NXP Semiconductors
Reference Manual

Document identifier: IMX93RM
Rev. 7, 2026-02-10

i.MX 93 Applications Processor Reference Manual

Contents

Preface About This Manual.....17

Chapter 1 Introduction..... 28
1.1 Overview.....28
1.2 Target Applications.....33
1.3 Endianness Support..... 33

Chapter 2 Memory Maps..... 34
2.1 Memory system overview.....34
2.2 System memory map used by all initiators other than the Cortex-M33 core.....34
2.3 System memory map (Cortex-M33)..... 45
2.4 AIPS Peripheral Bridge 1 Memory Map.....56
2.5 AIPS Peripheral Bridge 2 Memory Map.....68
2.6 AIPS Peripheral Bridge 3 Memory Map.....80
2.7 AIPS Peripheral Bridge 4 Memory Map.....85

Chapter 3 Interrupt and DMA Assignments..... 91
3.1 Overview.....91
3.2 System Interrupts..... 91
3.3 DMA MUX.....105

Chapter 4 Enhanced Direct Memory Access (eDMA3).....111
4.1 Chip-specific eDMA3 information..... 111
4.2 Overview.....111
4.3 Functional description.....114
4.4 External signals..... 119
4.5 Initialization.....119
4.6 Memory map and register definition..... 131
4.7 Glossary..... 166

Chapter 5 Enhanced Direct Memory Access (eDMA4).....167
5.1 Chip-specific eDMA4 information..... 167
5.2 Overview.....167
5.3 eDMA register descriptions..... 170
5.4 Functional description.....213
5.5 External signals..... 218
5.6 Initialization.....218
5.7 Application information..... 221
5.8 Glossary..... 228

Chapter 6 Security Overview..... 229
6.1 Security features.....229
6.2 Edgelock secure enclave.....231

Chapter 7 Fusemap 232
7.1 Overview 232
7.2 Boot Fusemap 232
7.3 Fusemap descriptions table 236
7.4 Programming Fuses 236

Chapter 8 Fuse Status Block 237
8.1 Introduction 237
8.2 Memory map and register definition 237
8.3 Functional description 247
8.4 Acronyms and abbreviations 248

Chapter 9 System Boot 249
9.1 Overview 249
9.2 Boot Types 249
9.3 Boot Modes 251
9.4 Boot Stages 253
9.5 Image Container Set 253
9.6 Boot Flow 264
9.7 Platform Initialization In ROM 275
9.8 Boot Devices 278
9.9 Serial Downloader Boot 293
9.10 Secure boot 295
9.11 ROM API 295

Chapter 10 IOMUXC GPR 298
10.1 Overview 298
10.2 Memory map and register definition 298

Chapter 11 Arm Modules Overview 300
11.1 Overview 300

Chapter 12 ARM Cortex A55 303
12.1 Overview 303
12.2 Functional Description 304
12.3 External Signals 305

Chapter 13 ARM Cortex CM33 TCM MCM and ECC MCM (CM33_CACHE_TCM_ECC_MCM) 306
13.1 Chip-specific CM33 information 306
13.2 Overview 307
13.3 Functional description 308
13.4 External signals 311
13.5 Initialization 312
13.6 Memory map and register definition 312

Chapter 14 ARM Cortex CM33 MCM (CM33_MCM) 362
14.1 Overview 362

14.2 Functional description....362
14.3 External signals....362
14.4 Application information....362
14.5 Memory map and register descriptions....363

Chapter 15 CM33 Cache Memory Controller (XCACHE)....374
15.1 Overview....374
15.2 Functional description....374
15.3 External signals....381
15.4 Initialization....382
15.5 Application information....382
15.6 XCACHE register descriptions....382

Chapter 16 CM33 Performance Monitor (CMX-PERFMON)....390
16.1 Chip-specific CMX-PERFMON information....390
16.2 Overview....390
16.3 Functional description....391
16.4 External signals....392
16.5 Initialization....392
16.6 Memory map and register definition....392

Chapter 17 Neural Processing Unit (NPU)....399
17.1 Overview....399
17.2 Functional descriptions....399
17.3 External signals....403
17.4 Initialization....403
17.5 Application information....403
17.6 Memory map and registers....403

Chapter 18 On-chip RAM (OCRAM)....404
18.1 Overview....404
18.2 Functional description....405
18.3 External signals....407
18.4 Initialization....407
18.5 Application information....407
18.6 Programmable registers....407

Chapter 19 Memory ECC Controller (MECC256)....408
19.1 Overview....408
19.2 Functional description....409
19.3 Signals....411
19.4 Initialization....411
19.5 Application information....411
19.6 Memory Map and register definition....411

Chapter 20 Messaging Unit (MU)....585
20.1 Overview....585
20.2 Functional description....586
20.3 External signals....589

20.4 Initialization....589
20.5 Application information....589
20.6 Register definition....595
20.7 Glossary....639

Chapter 21 Semaphores2 (SEMA42)....640
21.1 Overview....640
21.2 Functional description....641
21.3 External signals....644
21.4 Initialization....644
21.5 Memory map/register definition....644
21.6 Glossary....648

Chapter 22 Trusted Resource Domain Controller (TRDC)....649
22.1 Chip-specific TRDC information....649
22.2 Overview....656
22.3 Functional description....658
22.4 External signals....663
22.5 Initialization....664
22.6 Application information....664
22.7 Register descriptions....665

Chapter 23 System Counter (SYS_CTR)....1253
23.1 Chip-specific SYS_CTR information....1253
23.2 Overview....1253
23.3 Functional description....1254
23.4 External signals....1254
23.5 Initialization....1255
23.6 Application information....1255
23.7 Programming model....1255
23.8 Memory map and register definition....1255

Chapter 24 System Debug....1272
24.1 Overview....1272
24.2 Functional description....1274
24.3 External Signals....1281

Chapter 25 JTAG Controller (JTAGC)....1282
25.1 Chip-specific JTAGC information....1282
25.2 Overview....1282
25.3 Functional description....1283
25.4 External signals....1287
25.5 Initialization....1288
25.6 Application information....1288
25.7 Register description....1288
25.8 Glossary....1290

Chapter 26 External Signals and Pin Multiplexing....1291
26.1 Overview....1291

Chapter 27 IOMUX Controller (IOMUXC)..... 1292
27.1 Overview.....1292
27.2 Functional description.....1294
27.3 Memory Map and register definition..... 1296

Chapter 28 General-Purpose Input/Output (GPIO).....1713
28.1 Chip-specific GPIO information..... 1713
28.2 Overview.....1713
28.3 Functional description.....1714
28.4 External signals 1716
28.5 Initialization.....1717
28.6 Application information..... 1717
28.7 Memory map and register definition..... 1717

Chapter 29 Clock and Power Overview..... 1738
29.1 Overview.....1738
29.2 Components of Clock and Power Management..... 1738
29.3 Power Management..... 1739
29.4 Clocks 1743

Chapter 30 Clock Controller Module (CCM)..... 1747
30.1 Overview.....1747
30.2 System Clocks.....1748
30.3 Functional description.....1760
30.4 Reset..... 1780
30.5 CCM Memory Map and Register definition.....1780

Chapter 31 General Power Controller (GPC)..... 1917
31.1 Overview.....1917
31.2 Functional description.....1918
31.3 External Signals.....1925
31.4 Initialization.....1925
31.5 Memory Map and register definition..... 1926

Chapter 32 System Reset Controller (SRC)..... 1984
32.1 Overview.....1984
32.2 Functional description.....1985
32.3 External Signals.....1993
32.4 Register access permissions.....1993
32.5 Memory Map and register definition..... 1993

Chapter 33 Battery-Backed Non-Secure Module (BBNSM)..... 2073
33.1 Chip-specific BBNSM information..... 2073
33.2 Overview.....2073
33.3 Functional description.....2075
33.4 External Signals.....2078
33.5 Initialization.....2079
33.6 Memory Map and register definition..... 2079

Chapter 34 External Memories..... 2091
34.1 Overview.....2091
34.2 High Performance DRAM Interface (DDR).....2091
34.3 eMMC/SD/SDIO..... 2091
34.4 Flexible Serial Peripheral Interface.....2091

Chapter 35 DDR Memory Controller (DDRC).....2092
35.1 Chip-specific DDR information..... 2092
35.2 Overview.....2093
35.3 Functional description.....2094
35.4 External signals..... 2114
35.5 Initialization.....2116
35.6 DDRC register descriptions..... 2119
35.7 Glossary..... 2246

Chapter 36 Flexible Serial Peripheral Interface (FlexSPI)..... 2247
36.1 Chip-specific FLEXSPI information..... 2247
36.2 Overview.....2248
36.3 Glossary for FlexSPI module.....2250
36.4 Functional description.....2251
36.5 External signals..... 2296
36.6 Initialization.....2297
36.7 Application information..... 2298
36.8 Memory map and register definition..... 2315
36.9 AHB memory map definition.....2364

Chapter 37 FlexSPI Address Remap (AHB_ADDR_REMAP)..... 2366
37.1 Overview.....2366
37.2 Functional description.....2366
37.3 Application Information..... 2366
37.4 Memory map and register definition..... 2367

Chapter 38 Ultra Secured Digital Host Controller (uSDHC).....2369
38.1 Chip-specific USDHC information..... 2369
38.2 Overview.....2369
38.3 Functional description.....2372
38.4 External signals..... 2403
38.5 Application information..... 2404
38.6 uSDHC memory map and register definition.....2423
38.7 Glossary..... 2519

Chapter 39 CAN (FlexCAN).....2521
39.1 Chip-specific FLEXCAN information.....2521
39.2 Overview.....2521
39.3 Functional description.....2524
39.4 External signal descriptions 2578
39.5 Initialization and application information.....2578
39.6 Memory map and register definition..... 2579
39.7 Glossary..... 2688

Chapter 40 Flexible I/O (FlexIO)..... 2690
40.1 Chip-specific FLEXIO information..... 2690
40.2 Overview..... 2690
40.3 Functional description..... 2692
40.4 External signals..... 2702
40.5 Initialization..... 2702
40.6 Application information..... 2702
40.7 Memory map and registers..... 2722
40.8 Glossary..... 2767

Chapter 41 Universal Serial Bus Controller (USB)..... 2768
41.1 Chip-specific USB information..... 2768
41.2 Overview..... 2768
41.3 Functional description..... 2770
41.4 External signals..... 2802
41.5 Initialization..... 2803
41.6 Register interface..... 2803
41.7 Register descriptions..... 2806

Chapter 42 Universal Serial Bus 2.0 PHY (USB2_PHY)..... 2915
42.1 Overview..... 2915
42.2 Clock Generation..... 2917
42.3 Device Initialization..... 2917
42.4 Tuning..... 2917

Chapter 43 Ethernet Controller with TSN (ENET_QoS)..... 2918
43.1 Chip-specific ENET information..... 2918
43.3 Overview..... 2918
43.4 Functional description..... 2920
43.5 External signals..... 3033
43.6 Initialization..... 3034
43.7 Application Information..... 3036
43.8 Memory Map and register definition..... 3049

Chapter 44 Ethernet MAC (ENET)..... 3882
44.1 Overview..... 3882
44.2 Functional description..... 3885
44.3 External signals..... 3936
44.4 Initialization..... 3937
44.5 Memory map/register definition..... 3937

Chapter 45 Audio Overview..... 4051
45.1 Audio Overview..... 4051

Chapter 46 PDM Microphone Interface (PDM)..... 4053
46.1 Chip-specific PDM information..... 4053
46.2 Overview..... 4053
46.3 Functional description..... 4055
46.4 External signals..... 4068

46.5 Initialization....4068
46.6 Application information....4071
46.7 MICFIL register descriptions....4071

Chapter 47 Synchronous Audio Interface (SAI)....4102
47.1 Chip-specific SAI information....4102
47.2 Overview....4103
47.3 Functional description....4105
47.4 External signals....4112
47.5 Initialization....4113
47.6 Memory map and register definition....4113
47.7 Glossary....4164

Chapter 48 Audio Transceiver (SPDIF XCVR)....4165
48.1 Overview....4165
48.2 Functional description....4165
48.3 External signals....4169
48.4 Initialization....4169
48.5 Application information....4170
48.6 Memory Map and register definition....4171
48.7 Glossary....4208

Chapter 49 Medium Quality Sound (MQS)....4209
49.1 Chip-specific MQS information....4209
49.2 Overview....4209
49.3 Functional Description....4210
49.4 External Signals....4211
49.5 Initialization....4212
49.6 Application Information....4212

Chapter 50 Multimedia Overview....4214
50.1 Multimedia Overview....4214

Chapter 51 LCD Interface (LCDIF)....4217
51.1 Chip-specific LCDIF information....4217
51.2 Overview....4217
51.3 Functional description....4218
51.4 Memory Map and register definition....4220

Chapter 52 MIPI-DSI....4246
52.1 Overview....4246
52.2 Functional Description....4247
52.3 External signals....4283
52.4 Memory Map and register definition....4283

Chapter 53 MIPI-DSI DPHY....4371
53.1 Overview....4371
53.2 Functional description....4372

53.3 Initialization....4377

Chapter 54 MIPI-CSI2....4379
54.1 Overview....4379
54.2 Functional Description....4381
54.3 External signals....4414
54.4 Initialization....4414
54.5 Memory Map and register definition....4414

Chapter 55 MIPI-CSI2 DPHY....4480
55.1 Overview....4480
55.2 Functional description....4481
55.3 Initialization....4489

Chapter 56 Pixel Pipeline (PXP)....4491
56.1 Chip-specific PXP information....4491
56.2 Overview....4491
56.3 Functional Description....4492
56.4 External Signals....4525
56.5 Initialization....4525
56.6 PXP Memory Map/Register Definition....4526

Chapter 57 Image Sensing Interface (ISI)....4675
57.1 Chip-specific ISI information....4675
57.2 Overview....4675
57.3 Functional description....4677
57.4 External signals....4683
57.5 Initialization and channel configuration....4683
57.6 Application information....4685
57.7 ISI register descriptions....4687

Chapter 58 Parallel Camera Interface....4744
58.1 Overview....4744
58.2 Functional Description....4744
58.3 External Signals....4747

Chapter 59 LVDS Display Bridge and Pixel Mapper....4749
59.1 Chip-specific LVDS information....4749
59.2 Overview....4749
59.3 Functional Descriptions....4750

Chapter 60 Low Power Inter-Integrated Circuit (LPI2C)....4753
60.1 Chip-specific LPI2C information....4753
60.2 Overview....4753
60.3 Functional description....4755
60.4 External signals....4766
60.5 Initialization....4767
60.6 Application information....4768

60.7 Memory map and registers.....4768
60.8 Glossary..... 4822

Chapter 61 Low Power Serial Peripheral Interface (LPSPI)..... 4823
61.1 Chip-specific LPSPI information.....4823
61.2 Overview.....4823
61.3 Functional description.....4825
61.4 External signals..... 4836
61.5 Initialization.....4836
61.6 Memory map and registers.....4836
61.7 Glossary..... 4870

Chapter 62 Low Power Universal Asynchronous Receiver/Transmitter (LPUART).....4871
62.1 Chip-specific LPUART information.....4871
62.2 Overview.....4871
62.3 Functional description.....4874
62.4 External signals..... 4890
62.5 Initialization.....4891
62.6 LPUART register descriptions..... 4891
62.7 Glossary..... 4940

Chapter 63 Improved Inter-Integrated Circuit (I3C)..... 4941
63.1 Chip-specific I3C information.....4941
63.2 Overview.....4941
63.3 Functional description.....4943
63.4 External signals..... 4957
63.5 Initialization.....4957
63.6 Application information..... 4959
63.7 I3C register descriptions.....4961
63.8 Glossary..... 5069

Chapter 64 Timers Overview..... 5071
64.1 Overview.....5071

Chapter 65 Low-Power Timer (LPTMR)..... 5073
65.1 Chip-specific LPTMR information..... 5073
65.2 Overview.....5074
65.3 Functional description.....5074
65.4 External signals..... 5077
65.5 Initialization.....5077
65.6 Application information..... 5077
65.7 Memory map and register definition..... 5078

Chapter 66 Low Power Periodic Interrupt Timer (LPIT).....5085
66.1 Chip-specific LPIT information.....5085
66.2 Overview.....5085
66.3 Functional description.....5086
66.4 Initialization.....5106

66.5 Memory map and registers.....5107

Chapter 67 Timer/PWM Module (TPM).....5124
67.1 Chip-specific TPM information.....5124
67.2 Overview.....5124
67.3 Functional description.....5126
67.4 External signals.....5146
67.5 Initialization.....5146
67.6 Application information.....5147
67.7 Memory map and register definition.....5148

Chapter 68 Timer (TSTMR).....5173
68.1 Chip-specific TSTMR information.....5173
68.2 Overview.....5173
68.3 Functional description.....5174
68.4 External signals.....5174
68.5 Initialization.....5174
68.6 TSTMR_A memory map and register definition.....5174

Chapter 69 Watchdog timer (WDOG).....5177
69.1 Chip-specific WDOG information.....5177
69.2 Overview.....5179
69.3 Functional description.....5180
69.4 External signals.....5184
69.5 Initialization.....5184
69.6 Application information.....5184
69.7 Memory map and register definition.....5185

Chapter 70 Analog Overview.....5192
70.1 Overview.....5192
70.2 Analog-to-Digital Converter (ADC).....5192
70.3 Temperature Sensor.....5192
70.4 XTAL.....5192
70.5 PLL.....5192
70.6 DIGPROG Device ID.....5193

Chapter 71 Trigger Mux (TRGMUX).....5194
71.1 Chip-specific TRGMUX information.....5194
71.2 Overview.....5194
71.3 Functional description.....5195
71.4 External signals.....5195
71.5 Initialization.....5195
71.6 TRGMUX register descriptions.....5195

Chapter 72 Analog-to-Digital Converter (SAR_ADC).....5198
72.1 Chip-specific ADC information.....5198
72.2 Introduction.....5200
72.3 Functional description.....5201
72.4 External signals.....5225

72.5 Initialization....5225
72.6 SAR_ADC register descriptions....5226

Chapter 73 Thermal Monitoring Unit (TMU)....5337
73.1 Chip-specific TMU information....5337
73.2 Overview....5339
73.3 Functional description....5340
73.4 Initialization....5343
73.5 External signals....5343
73.6 TMU register descriptions....5343
73.7 Glossary....5374

Chapter 74 PLLs....5375
74.1 Overview....5375
74.2 Functional description....5376
74.3 External signals....5379
74.4 Initialization....5379
74.5 Memory map and register definitions....5383

Chapter 75 OSC....5425
75.1 Overview....5425
75.2 Memory map and register definition....5425

Chapter 76 Always-on Domain, Non-secure Block Control (blk_ctrl_ns_aonmix)....5427
76.1 Introduction....5427
76.2 Functional description....5428
76.3 Initialization....5429
76.4 Memory Map and register definition....5429

Chapter 77 Always-on Domain, Secure Block Control (blk_ctrl_s_aonmix)....5453
77.1 Introduction....5453
77.2 Functional description....5454
77.3 Initialization....5455
77.4 Memory Map and register definition....5455

Chapter 78 Wake-up Domain Block Control (blk_ctrl_wakeupmix)....5486
78.1 Introduction....5486
78.2 Functional description....5487
78.3 Initialization....5488
78.4 Memory Map and register definition....5488

Chapter 79 NIC Mix Block Control (blk_ctrl_nicmmix)....5510
79.1 Introduction....5510
79.2 Functional description....5510
79.3 Initialization....5511
79.4 Memory Map and register definition....5511

Chapter 80 ML Mix Domain Block Control (blk_ctrl_mlmix)..... 5523
80.1 Overview.....5523
80.2 Functional description.....5523
80.3 External signals..... 5523
80.4 Memory map and register definition..... 5524

Chapter 81 DDR Mix Domain Block Control (blk_ctrl_ddrmix).....5526
81.1 Overview.....5526
81.2 Functional Description..... 5526
81.3 HWFCC and SWFFC Procedure.....5527
81.4 DDR IO Retention Enter/Exit Procedure..... 5532
81.5 Memory Map and register definition..... 5538

Chapter 82 Media Mix Domain Block Control (blk_ctrl_mediamix)..... 5544
82.1 Overview.....5544
82.2 Functional description.....5545
82.3 External signals..... 5546
82.4 Memory map and register definition..... 5546

Appendix A Release notes.....5585
A.1 About_this_manual changes..... 5585
A.2 Introduction changes.....5585
A.3 Memory Map changes.....5585
A.4 IRQ and DMA assignment changes.....5585
A.5 Enhanced Direct Memory Access (eDMA3).....5585
A.6 Enhanced Direct Memory Access (eDMA4).....5585
A.7 Security Overview changes.....5585
A.8 Fusemap changes.....5586
A.9 FSB module changes 5586
A.10 SystemSecure Boot changes.....5586
A.11 IOMUXC GPR module changes.....5586
A.12 Arm Modules Overview changes..... 5586
A.13 A55 Platform Overview changes.....5586
A.14 ARM Cortex CM33 TCM MCM and ECC MCM (CM33_CACHE_TCM_ECC_MCM).....5586
A.15 MCM module changes..... 5586
A.16 XCACHE module changes.....5586
A.17 CM33 Performance Monitor (CMX-PERFMON)..... 5587
A.18 NPU module changes..... 5587
A.19 OCRAM module changes..... 5587
A.20 MECC256 changes.....5587
A.21 MU module changes..... 5587
A.22 SEMA42 module changes.....5587
A.23 Trusted Resource Domain Controller (TRDC)..... 5587
A.24 System Counter (SYS_CTR).....5588
A.25 System Debug changes.....5588
A.26 JTAG Controller (JTAGC)..... 5588
A.27 External Signals changes.....5588
A.28 IOMUXC module changes..... 5588
A.29 General-Purpose Input/Output (GPIO).....5588
A.30 Clocks and Power Overview changes.....5588

A.31 CCM module changes....5589
A.32 GPC changes....5589
A.33 SRC module changes....5589
A.34 Battery-Backed Non-Secure Module (BBNSM)....5589
A.35 External_memory controllers....5589
A.36 DDR Memory Controller (DDRC)....5589
A.37 Flexible Serial Peripheral Interface (FlexSPI)....5590
A.38 AHB Address Remap module changes....5590
A.39 Ultra Secured Digital Host Controller (uSDHC)....5590
A.40 CAN (FlexCAN)....5590
A.41 Flexible I/O (FlexIO)....5591
A.42 Universal Serial Bus Controller (USB)....5591
A.43 fusemap changes....5591
A.44 Ethernet Controller with TSN (ENET_QoS)....5591
A.45 ENET module changes....5591
A.46 Audio Overview changes....5591
A.47 PDM Microphone Interface (PDM)....5592
A.48 Synchronous Audio Interface (SAI)....5592
A.49 AUDIO_XCVR module changes....5592
A.50 Medium Quality Sound (MQS)....5592
A.51 Audio Overview changes....5592
A.52 LCD Interface (LCDIF)....5592
A.53 MIPI DSI changes....5593
A.54 MIPI Tx PHY changes....5593
A.55 MIPI_CSI2 changes....5593
A.56 MIPI Rx PHY changes....5593
A.57 Pixel Pipeline (PXP)....5593
A.58 Image Sensing Interface (ISI)....5593
A.59 Parallel Camera Interface changes....5594
A.60 LVDS Display Bridge and Pixel Mapper....5594
A.61 Low Power Inter-Integrated Circuit (LPI2C)....5594
A.62 Low Power Serial Peripheral Interface (LPSPI)....5594
A.63 Low Power Universal Asynchronous Receiver/Transmitter (LPUART)....5594
A.64 Improved Inter-Integrated Circuit (I3C)....5595
A.65 Timers Overview changes....5595
A.66 Low-Power Timer (LPTMR)....5595
A.67 Low Power Periodic Interrupt Timer (LPIT)....5595
A.68 Timer/PWM Module (TPM)....5596
A.69 Timer (TSTMR)....5596
A.70 Watchdog timer (WDOG)....5596
A.71 Analog Overview changes....5596
A.72 Trigger Mux (TRGMUX)....5596
A.73 Analog-to-Digital Converter (SAR_ADC)....5597
A.74 Thermal Monitoring Unit (TMU)....5597
A.75 PLL module changes....5597
A.76 OSC changes....5597
A.77 BLK_CTRL_NS_AONMIX module changes....5597
A.78 BLK_CTRL_S_AONMIX module changes....5597
A.79 BLK_CTRL_WAKEUPMIX changes....5597
A.80 BLK_CTRL_NICMIX module changes....5597
A.81 BLK_CTRL_MLMIX module changes....5598
A.82 DDRMIX module changes....5598
A.83 BLK_CTRL_MEDIAMIX module changes....5598

Legal information..... 5599

Preface
About This Manual

Audience

The reference manual is intended for the board-level product designers and product software developers. This manual assumes that the reader has a background in computer engineering and/or software engineering and understands the concepts of the digital system design, microprocessor architecture, input/output (I/O) devices, industry standard communication, and device interface protocols.

Organization

The reference manual describes the chip at a system level and provides an architectural overview. It also describes the system memory map, system-level interrupt events, external pins and pin multiplexing, external memory, system debug, system boot, multimedia subsystem, power management, and system security.

Attachments

This manual includes key information in the files attached to it. For example, memory map and I/O details. Use the content in these attachments in conjunction with this manual's content.

NOTE

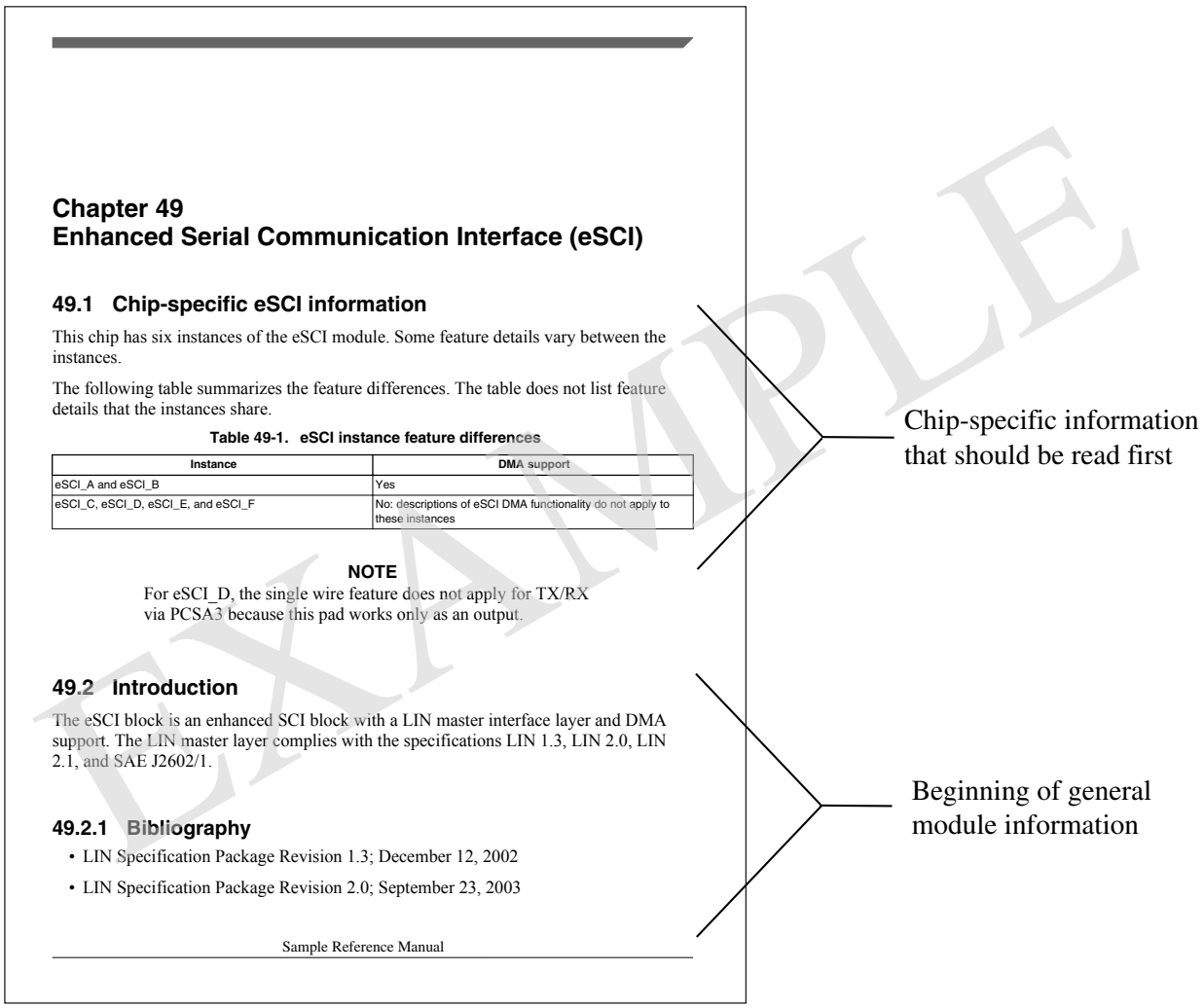
Select the paperclip icon on the left side of the PDF window to see the list of attachments.

Module descriptions

Each module chapter has two main parts:

- The first section, *chip-specific [module name]* information, provides details such as the number of module instances on the chip and connections between that module and the other ones. Read this section *first* because its content is crucial for understanding the information in the other sections of the chapter.
- The subsequent sections provide general information about the module, including its signals, registers, and functional description.

The following figure shows you an example of this demarcation.



Chapter 49
Enhanced Serial Communication Interface (eSCI)

49.1 Chip-specific eSCI information

This chip has six instances of the eSCI module. Some feature details vary between the instances.

The following table summarizes the feature differences. The table does not list feature details that the instances share.

Table 49-1. eSCI instance feature differences

Instance	DMA support
eSCI_A and eSCI_B	Yes
eSCI_C, eSCI_D, eSCI_E, and eSCI_F	No: descriptions of eSCI DMA functionality do not apply to these instances

NOTE
For eSCI_D, the single wire feature does not apply for TX/RX via PCSA3 because this pad works only as an output.

49.2 Introduction

The eSCI block is an enhanced SCI block with a LIN master interface layer and DMA support. The LIN master layer complies with the specifications LIN 1.3, LIN 2.0, LIN 2.1, and SAE J2602/1.

49.2.1 Bibliography

- LIN Specification Package Revision 1.3; December 12, 2002
- LIN Specification Package Revision 2.0; September 23, 2003

Sample Reference Manual

EXAMPLE

Figure 0. Example of chapter chip-specific information and general module information

Register Access

Register Diagram Field Access Type Legend

This figure provides the interpretation of the notation used in the register diagrams for a number of common field access types:

Access type	Access description	DITA output	Effect of Write on Value	Readback Value
RW	Read/write	R / W: Mnemonic	Changes to Written Value	Current Value
RO	Read-only	R: Mnemonic / W:	No Effect	Current Value
RU	Reserved, unimplemented	R / W: Reserved	No Effect	Value Undefined
ROZ	Reserved, Read-only zero	R: 0 / W:	No Effect	Returns All Zero
ROO	Reserved, Read-only one	R: 1 / W:	No Effect	Returns All Ones
WO	Write-only	R: / W: Mnemonic	Changed to Written Value	Value Undefined
WOZ	Reserved, Write-only zero	R: / W: 0	Write Value Must Be All 0s.	Value Undefined
WOO	Reserved, Write-only one	R: / W: 1	Write Value Must Be All 1s.	Value Undefined
W0C	Write zero to clear	R: Mnemonic / W: w0c	If a Bit in the Written Value is a 0, the Corresponding Bit in the Field is Set to 0. Otherwise, the Field Bit is Not Affected.	Current Value
W1C	Write one to clear	R: Mnemonic / W: w1c	If a Bit in the Written Value is a 1, the Corresponding Bit in the Field is set to 0. Otherwise, the Field Bit is Not Affected.	Current Value
R2C	Read to clear		Value is cleared following the read operation	Current Value
ROWZ	Read-only, writes zero	R: Mnemonic / W: 0	Write Value Must Be All 0s	Current Value
ROWO	Read-only writes one	R: Mnemonic / W: 1	Write Value Must Be All 1s	Current Value
ROWU	Ready-only writes undefined	R: Mnemonic / W: —	Writes Operation Undefine	Current Value
WORZ	Write-only reads zero	R: 0 / W: Mnemonic	Changes to Written Value	Returns All Zero
WORO	Write-only reads one	R: 1 / W: Mnemonic	Changes to Written Value	Returns All Ones
WORU	Write-only reads undefined	R: — / W: Mnemonic	Changes to Written Value	Value Undefined

Figure 0. Register Field Conventions

NOTE

For reserved register fields, the software should mask off the data in the field after a read (the software can't rely on the contents of data read from a reserved field) and always write all zeros.

Register Macro Usage

A common operation is to update one field without disturbing the contents of the remaining fields in the register. Normally, this requires a read-modify-write (RMW) operation, where the CPU reads the register, modifies the target field, then writes the results back to the register. This is an expensive operation in terms of CPU cycles, because of the initial register read.

To address this issue, some hardware registers are implemented as a group, including registers that can be used to either set, clear, or toggle (SCT) individual bits of the primary register. When writing to an SCT register, all the bits set to 1 perform the associated operation on the primary register, while the bits set to 0 are not affected. The SCT registers always read back 0, and should be considered write-only. The SCT registers are not implemented if the primary register is read-only.

With this architecture, it is possible to update one or more fields using only register writes. First, all bits of the target fields are cleared by a write to the associated clear register, then the desired value of the target fields is written to the set register. This sequence of two writes is referred to as a clear-set (CS) operation.

A CS operation does have one potential drawback. Whenever a field is modified, the hardware sees a value of 0 before the final value is written. For most fields, passing through the 0 state is not a problem. Nonetheless, this behavior is something to consider when using a CS operation.

Also, a CS operation is not required for fields that are one-bit wide. While the CS operation works in this case, it is more efficient to simply set or clear the target bit (that is, one write instead of two). A simple set or clear operation is also atomic, while a CS operation is not.

Note that not all macros for set, clear, or toggle (SCT) are atomic. For registers that do not provide hardware support for this functionality, these macros are implemented as a sequence of read-modify-write operations. When an atomic operation is required, the developer should pay attention to this detail, because unexpected behavior might result if an interrupt occurs in the middle of the critical section comprising the update sequence.

A set of SCT registers is offered for registers in many modules on this device, as described in this manual.
In a module memory map table, the suffix _SET, _CLR, or _TOG is added to the base name of the register.
For example, the CCM_ANALOG_PLL_ARM register has three other registers called CCM_ANALOG_PLL_ARM_SET, CCM_ANALOG_PLL_ARM_CLR, and CCM_ANALOG_PLL_ARM_TOG.

In the sub-section that describes one of these sets of registers, a short-hand convention is used to denote that a register has the SCT register set. There is an italicized *n* appended to the end of the short register name. Using the above example, the name used for this register is CCM_ANALOG_PLL_ARM*n*. When you see this designation, there is a SCT register set associated with the register, and you can verify this by checking it in the memory map table. The address offset for each of these registers is given in the form of the following example:

Address: 20C_8000h base + 0h offset + (4d × i), where i=0d to 3d

In this example, the address for each of the base registers and their three SCT registers can be calculated as:

Register	Address
CCM_ANALOG_PLL_ARM	20C_8000h
CCM_ANALOG_PLL_ARM_SET	20C_8004h
CCM_ANALOG_PLL_ARM_CLR	20C_8008h
CCM_ANALOG_PLL_ARM_TOG	20C_800Ch

Acronyms and Abbreviations

The table below contains acronyms and abbreviations used in this document.

Acronyms and Abbreviated Terms

Term	Meaning
ACMP	Analog Comparator
ADC	Analog-to-Digital Converter
AHB	Advanced High-performance Bus
AIPS	Arm IP Bus
ALU	Arithmetic Logic Unit
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
ASRC	Asynchronous Sample Rate Converter
AXI	Advanced eXtensible Interface
CA/CM	Arm Cortex-A/Cortex-M
CAAM	Cryptographic Acceleration and Assurance Module
CAN	Controller Area Network
CCM	Clock Controller Module
CM7	Arm Cortex M7 Core
CPU	Central Processing Unit
CSI	CMOS Sensor Interface
CSU	Central Security Unit
CTI	Cross Trigger Interface
D-cache	Data cache
DAP	Debug Access Port
DDR	Double data rate
DMA	Direct memory access
DPLL	Digital phase-locked loop
DRAM	Dynamic random access memory
ECC	Error correcting codes
LPSPI	Low-power SPI
EDMA	Enhanced Direct Memory Access
EIM	External Interface Module
ENET	Ethernet

Table continues on the next page...

Table continued from the previous page...

Term	Meaning
EPIT	Enhanced Periodic Interrupt Timer
EPROM	Erasable Programmable Read-Only Memory
ETF	Embedded Trace FIFO
ETM	Embedded Trace Macrocell
FIFO	First-In-First-Out
GIC	General Interrupt Controller
GPC	General Power Controller
GPIO	General-Purpose I/O
GPR	General-Purpose Register
GPS	Global Positioning System
GPU	Graphics Processing Unit
GPV	Global Programmers View
HAB	High-Assurance Boot
I-cache	Instruction cache
I2C or I^2C	Inter-Integrated Circuit
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IOMUX	Input-Output Multiplexer
IP	Intellectual Property
IrDA	Infrared Data Association
JTAG	Joint Test Action Group (a serial bus protocol usually used for test purposes)
LCDIF	Liquid Crystal Display Interface
LDO	Low-Dropout
LIFO	Last-In-First-Out
LRU	Least-Recently Used
LPSR	Low-Power State Retention
LSB	Least-Significant Byte

Table continues on the next page...

Table continued from the previous page...

Term	Meaning
LUT	Look-Up Table
LVDS	Low Voltage Differential Signaling
MAC	Medium Access Control
MCM	Miscellaneous Control Module
MMC	Multimedia Card
MSB	Most-Significant Byte
MT/s	Mega Transfers per second
OCRAM	On-Chip Random-Access Memory
OCOTP	On-Chip One-Time Programmable Controller
PCI	Peripheral Component Interconnect
PCIe	PCI express
PCMCIA	Personal Computer Memory Card International Association
BPC	Basic Power Controller
PIC	Programmable Interrupt Controller
PMU	Power Management Unit
POR	Power-On Reset
PSRAM	Pseudo-Static Random Access Memory
PWM	Pulse Width Modulation
PXP	Pixel Pipeline
QoS	Quality of Service
R2D	Radians to Degrees
RISC	Reduced Instruction Set Computing
ROM	Read-Only Memory
RTOS	Real-Time Operating System
Rx	Receive
SAI	Synchronous Audio Interface

Table continues on the next page...

Table continued from the previous page...

Term	Meaning
SCU	Snoop Control Unit
SD	Secure Digital
SDIO	Secure Digital Input/Output
SDLC	Synchronous Data Link Control
SDMA	Smart DMA
SIM	Subscriber Identification Module
SNVS	Secure Non-Volatile Storage
SoC	System-on-Chip
SPBA	Shared Peripheral Bus Arbiter
SPDIF	Sony Philips Digital Interface
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SRC	System Reset Controller
TFT	Thin-Film Transistor
TPIU	Trace Port Interface Unit
TSGEN	Time Stamp Generator
Tx	Transmit
TZASC	TrustZone Address Space Controller
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
USDHC	Ultra Secured Digital Host Controller
WDOG	Watchdog
WLAN	Wireless Local Area Network
WXGA	Wide Extended Graphics Array

Conventions

The reference manual uses the following notational conventions:

cleared / set	When a bit has a value of zero, it is said to be cleared; when it has a value of one, it is said to be set.
mnemonics	Instruction mnemonics are shown in lowercase bold.

italics	Italics indicate variable command parameters, for example, **bcctr***x*. The book titles in the text are set in italics.
15	An integer in decimal.
0x	the prefix to denote a hexadecimal number.
0b	The prefix to denote a binary number. Binary values of 0 and 1 are written without a prefix.
n'H4000CA00	The n-bit hexadecimal number.
BLK_REG_NAME	The register names are all uppercase. The block mnemonic is prepended with an underscore delimiter (_).
BLK_REG[FIELD]	The fields within registers appear in brackets. For example, ESR[RLS] refers to the Receive Last Slot field of the ESAI Status Register.
BLK_REG[*n*]	The bit number *n* within the BLK.REG register.
BLK_REG[*l:r*]	The register bit ranges. The ranges are indicated by the left-most bit number *l* and the right-most bit number *r*, separated by a colon (:). For example, ESR[15:0] refers to the lower half word in the ESAI Status Register.
x, U	In some contexts, such as signal encodings, an unitalicized x indicates a "don't care" or "uninitialized". The binary value can be 1 or 0.
x	An italicized *x* indicates an alphanumeric variable.
n, m	Italicized *n or m* represent integer variables.
!	Binary logic operator NOT.
&&	Binary logic operator AND.
||	Binary logic operator OR.
^ or <O+>	Binary logic operator XOR. For example, A <O+> B.
|	Bit-wise OR. For example, 0b0001 | 0b1000 yields the value of 0b1001.
&	Bit-wise AND. For example, 0b0001 & 0b1000 yields the value of 0b0000.
{A,B}	Concatenation, where the *n*-bit value A is prepended to the *m*-bit value B to form an (*n*+*m*)-bit value. For example, {0, REG*m* [14:0]} yeilds a 16-bit value with 0 in the most significant bit.
- or grey fill	Indicates a reserved bit field in a register. Although these bits can be written to with ones or zeros, they always read zeros.
>>	Shift right logical one position.
<<	Shift left logical one position.
<=	Assignment.
==	Compare equal.
!=	Compare not equal.
>	Greater than.
<	Less than.

Notes and cautions

Specific information is provided as part of notes and cautions throughout this manual.

NOTE

Emphasizes information that deserves extra attention.

CAUTION

Informs you of situations that could lead to highly undesirable outcomes—such as damage to the chip or irreversible malfunction.

Numbering systems

The following suffixes identify different numbering systems:

Table 1. Numbering systems

This suffix	Identifies a
b	Binary number. For example, the binary equivalent of the number 5 is mentioned as 101b. In some cases, *0b* is prefixed to binary numbers.
d	Decimal number. Decimal numbers are followed by this suffix only when there is a possibility of confusion. In general, decimal numbers are used without a suffix.
h	Hexadecimal number. For example, the hexadecimal equivalent of the number 60 is mentioned as 3Ch. In some cases, *0x* is prefixed to hexadecimal numbers.

Typographic notation

The following typographic notations are used throughout this document:

Table 2. Typographic notation

Example	Description
x and other italicized text	The italicized, lowercase *x* is used as a placeholder for replaceable numbers. In general, italicized text is used for titles of publications and for emphasis. Additionally, italics could be used for metasymbols in syntax descriptions. Plain lowercase letters are used as placeholders for single letters and numbers.
`code font`	Fixed-width font (such as Courier) used for code. It is used for a letter, word, or phrase that you want the user to type. For example, "Type `Read` and press Enter." This type of font is also used for instruction mnemonics, directives, symbols, subcommands, parameters, operators, computer-language elements, code listings, commands that appear in running text, and for sample code. Instruction mnemonics and directives in text and tables are mentioned in all caps; for example, BSR.
SR[SCM]	A mnemonic in square brackets represents the name of a register field. This example refers to the Scaling Mode (SCM) field in the Status Register (SR).
REVNO[6:4], XAD[7:0]	Numbers in brackets that are separated by a colon represent either: • A subset of a register's named field For example, REVNO[6:4] refers to bits 6-4 that are part of the COREREV field occupying bits 6-0 of the REVNO register. • A continuous range of individual signals of a bus For example, XAD[7:0] refers to signals 7-0 of the XAD bus.
MOD.REG	A period separates the elements of a hierarchy: subsystem.module.register. For example:

Table continues on the next page...

Table 2. Typographic notation (continued)

Example	Description
	• SWT.TO means that the TO register is located in the SWT module. • SMU.XRDC.CR means that the CR register is located in the XRDC module within the SMU subsystem.

Special terms

The following terms have special meanings.

Table 3. Special terms

Term	Meaning
Asserted	Refers to the state of a signal as follows: • An active-high signal is asserted when high (1). • An active-low signal is asserted when low (0).
Deasserted	Refers to the state of a signal as follows: • An active-high signal is deasserted when low (0). • An active-low signal is deasserted when high (1). In some cases, deasserted signals are described as *negated*.
Reserved	Refers to memory space, register, field, or programming setting. Writes to a reserved location can result in unpredictable functionality or behavior. You must: • Not modify the default value of a reserved programming setting, such as the reset value of a reserved register field. • Consider undefined locations in memory to be reserved. • Not use the reset values of reserved locations documented in this manual for diagnostic purposes. They are subject to change. **NOTE** BootROM could modify the reserved bit values after reset. Please refer to the BootROM settings attachment.
Write 1 to clear (w1c)	Refers to the access type of a register field that is used to clear the field by writing the value 1 to it.
Undefined (u)	Refers to undefined reset values

Chapter 1
Introduction

1.1 Overview

i.MX 93 applications processors deliver efficient machine learning (ML) acceleration and advanced security with integrated EdgeLock® secure enclave to support energy-efficient edge computing. The i.MX 93 applications processors are the first in the i.MX portfolio to integrate the scalable Arm Cortex®-A55 core, bringing performance and energy efficiency to Linux®-based edge applications and the Arm Ethos®-U65 microNPU, enabling developers to create more capable, cost-effective and energy-efficient ML applications. Optimizing performance and power efficiency for Industrial, IoT and automotive devices, i.MX 93 processors are built with NXP's innovative Energy Flex architecture. The SoCs offer a rich set of peripherals targeting automotive, industrial and consumer IoT market segments. Part of the EdgeVerse™ portfolio of intelligent edge solutions, the i.MX 93 family will be offered in commercial, industrial, extended industrial and automotive level qualification and backed by NXP's product longevity program.

1.1.1 Block Diagram

The functional block diagram is shown in the figure below. This diagram provides a view of the chip's major functional components and core complexes.

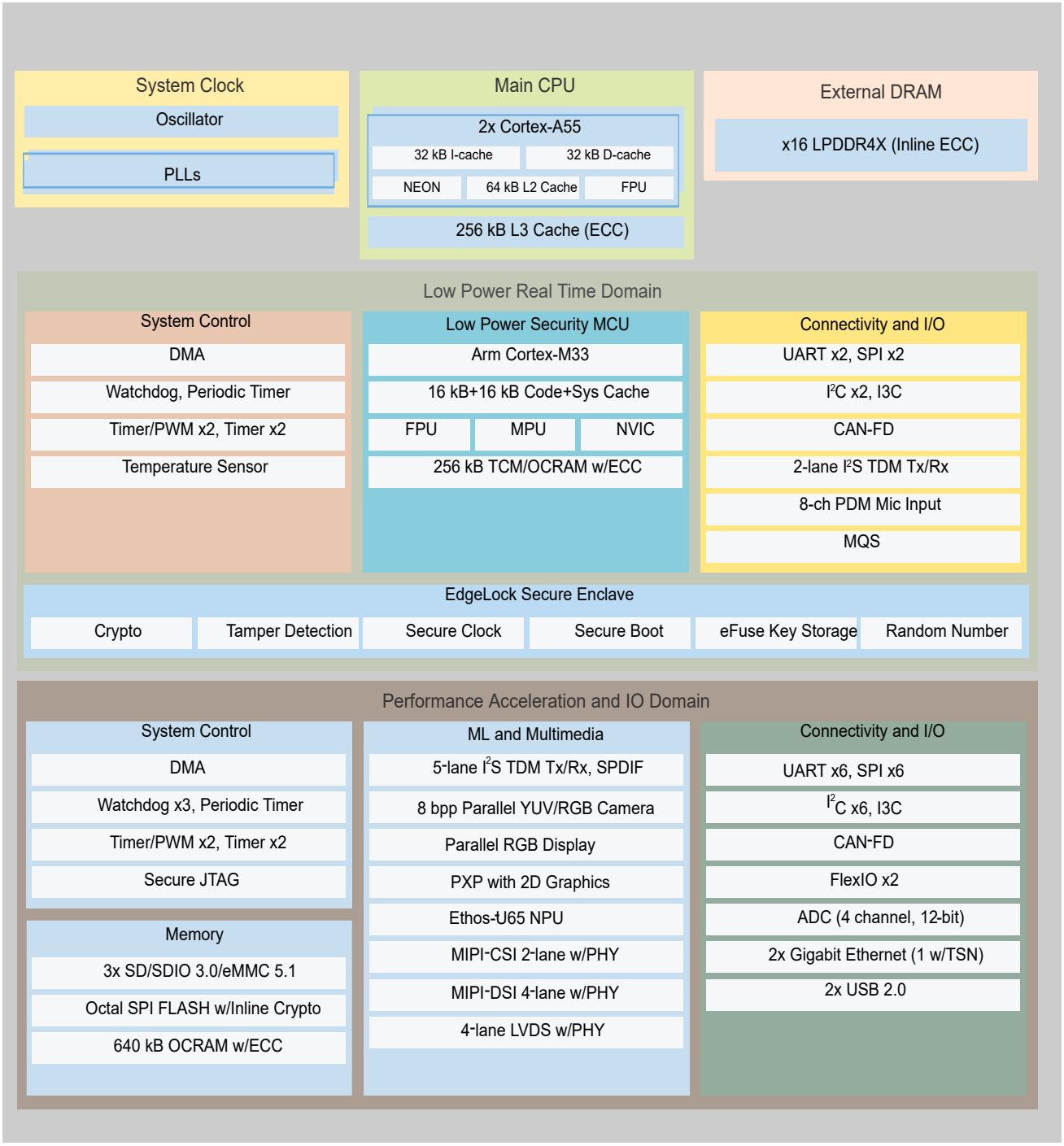


Figure 1. Simplified Block Diagram

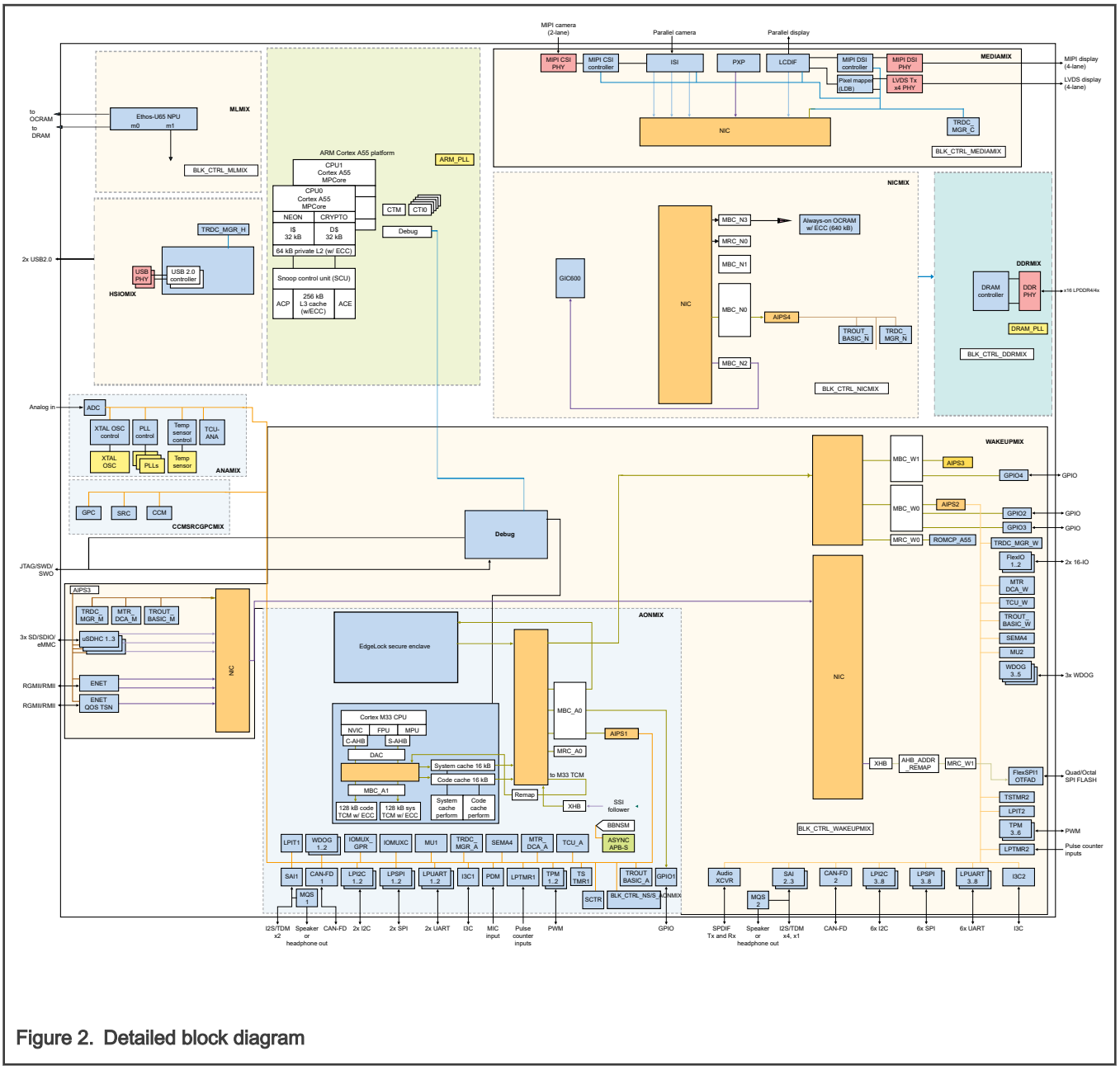


Figure 2. Detailed block diagram

1.1.2 Features

The i.MX93 processor is based on Arm Cortex®-A55 MPCore cluster and Arm Cortex®-M33 which have the following features:

Arm Cortex®-A55 MPCore cluster:

- One cluster of 2x Cortex-A55 cores
- Each core includes 32kB L1-I, 32kB L1-D and 64kB L2 cache per core
- 256kB shared cluster L3 cache
- Core cache protection (parity/ECC) supported

Arm Cortex®-M33 Platform:

- Microcontroller available both for boot and for customer application

- Arm Cortex®-M33 Processor:
  - 16KB L1 Instruction Cache
  - 16KB L1 Data Cache
  - 256 KByte TCM, also accessible as SRAM by the rest of the system
- ECC support for both cache and TCM

On Chip Memory:

- Boot ROM (256KB)
- 256KB Tightly Coupled RAM (TCM) for CM33, with ECC
- 640KB of SoC-specific OCRAM

External Memory Interfaces:

- One 16-bit DRAM controller:
  - Maximum supported capacity is 2GByte
  - LPDDR4X supported.
- FlexSPI FLASH with support for XIP and support for either one Octal SPI, two Quad SPI FLASH devices

Graphics:

- PXP 2D accelerator
- LCDIF display
  - LDB and 4-lane LVDS display (up to 1366x768 or 1280x800)
  - Parallel display (up to 1366x768 or 1280x800)
  - One 4-lane MIPI-DPHY DSI Tx PHY and MIPI-DSI Controller
- ISI camera interface
  - MIPI-DPHY CSI Rx PHY and MIPI-CSI Controller compliant to MIPI-DSI specification v1.2 and MIPI-DPHY specification v1.2
  - Parallel camera
  - Image processing for
    - One processed camera streams at 1080p30, or
  - Image down scaling via decimation and bi-phase filtering
  - Color space conversion
  - Interlaced to progressive conversions

Machine Learning

- High-efficiency Neural-network Processing Unit (NPU)

Audio:

- SPDIF supports raw capture mode that can save all the incoming bits into audio buffer.
- 3x Synchronous Audio Interface (SAI) modules supporting I2S, AC97, TDM, and Codec/DSP interfaces
- Digital microphone input, 8-channel PDM

Connectivity:

- 2x USB 2.0 OTG controller with integrated PHY interfaces
- 3x Ultra Secure Digital Host Controller (uSDHC) interfaces

  - uSDHC1 with boot support for eMMC 5.1 compliance with HS400 DDR signaling to support up to 400 MB/sec
  - uSDHC2 with boot support for SD/SDIO 3.0 compliance with 200 MHz SDR signaling to support up to 100 MB/sec
  - Support for SDXC (extended capacity)
  - USDHC3 to support use cases that need simultaneous support for all three of 1x eMMC, 1x SD Card and 1x SDIO (for WIFI connectivity)
- 8x Universal asynchronous receiver/transmitter (LPUARTs) modules
- 8x LPSPI modules
- 8x I2C modules
- 2x I3C modules
- 2x CAN-FD modules
- 2x 32-pin FlexIO modules
- 1x 1G-bit Ethernet with AVB support (ENET)
- 1x 1G-bit Ethernet QoS with TSN support

Timers and PWMs:

- 2x Low Power Periodical Interrupt Timer (LPIT)
  - 4 channel
  - 4 external trigger source
  - Generic 32-bit resolution timer
  - Periodical interrupt generation
- 6x Timer/PWM module (TPM)
  - Prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128
  - 16-bit counter, support free-running counter or modulo counter mode, counting up or down
  - Includes 6 channels that can be configured for input capture, output compare, edge-aligned PWM mode, or center-aligned PWM mode
- 2x Low-Power Timer (LPTMR)
- 5x WatchDog modules (WDOG)

GPIO and Pin Multiplexing:

- General-purpose input/output (GPIO) modules with interrupt capability
- Input/output multiplexing controller (IOMUXC) to provide centralized pad control

Analog:

- 1x 12-bit Analog Digital Converter (ADC)

Security:

- TRDC – Resource Domain Controller
  - Supports up to 16 resource domains
- Arm TrustZone® (TZ) architecture
- Secure and trusted access control
- Battery Backed Security Module (BBSM)
  - Monotonic counter - Secure real-time clock (RTC) - Zeroizable Master Key

System Debug:

- Arm CoreSight® debug and trace architecture
- Trace Port Interface Unit (TPIU) to support off-chip real-time trace
- Support for 5-pin (JTAG) and SWD debug interfaces

Power Management:

- One PMIC to supply all power rails.
- Temperature sensor with programmable trim points
- GPC hardware power management controller

1.2 Target Applications

The i.MX93 can be used but not limited to applications such as:

- Industrial
- IoT
- Automotive

1.3 Endianness Support

The chip supports only the Little Endian mode.

Chapter 2
Memory Maps

2.1 Memory system overview

This section introduces the memory architecture of the chip.

NOTE

Accessing the reserved memory regions can result in unpredictable behavior. Write/Read to the Reserved locations may not generate a transfer error.

2.2 System memory map used by all initiators other than the Cortex-M33 core

The following table provides the TrustZone "Non-Secure (NS)" and "Secure (S)" world view of the system (Cortex-A) memory map. Access through the "Secure" memory region can only be performed by code with the appropriate security settings.

NOTE

The System Memory Map (used by Cortex-A cores and all non-Cortex-M initiators in the chip) has TrustZone Non-Secure (NS) and Secure (S) world aliases of the address space below address 8000_0000 (hex). This is done for compatibility with TrustZone-M Cortex-M33 cores, which need these aliases for TrustZone-M Secure World access. Having these aliases in the system memory map allows address pointers to be shared between TrustZone-M Cortex-M33 cores and other initiators using the system memory map.

Table 4. System memory map

Start address (hex)	End address (hex)	Size (KB)	Description
C000_0000 (NS)	FFFF_FFFF (NS)	1048576	DRAM (offset 1GB)
B010_0000 (NS)	BFFF_FFFF (NS)	261120	DRAM (offset continued)
B004_4000 (NS)	B00F_FFFF (NS)	752	DRAM (offset continued)
B000_0000 (NS)	B004_3FFF (NS)	272	DRAM (offset 768 MB)
A000_0000 (NS)	AFFF_FFFF (NS)	262144	DRAM (offset 512 MB)
9000_0000 (NS)	9FFF_FFFF (NS)	262144	DRAM (offset 256 MB)
8000_0000 (NS)	8FFF_FFFF (NS)	262144	DRAM (offset 0 MB)
7000_0000 (NS) 7000_0000 (S)	7FFF_FFFF (NS) 7FFF_FFFF (S)	262144	Reserved
6000_0000 (NS) 7000_0000 (S)	6FFF_FFFF (NS) 7FFF_FFFF (S)	262144	Reserved
4FC0_0000 (NS) 5FC0_0000 (S)	4FFF_FFFF (NS) 5FFF_FFFF (S)	4096	Reserved
4E31_0000 (NS) 5E31_0000 (S)	4FBF_FFFF (NS) 5FBF_FFFF (S)	25536	Reserved
4E30_0000 (NS)	4E30_FFFF (NS)	64	DDR Controller

Table continues on the next page...

Table 4. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
5E30_0000 (S)	5E30_FFFF (S)		
4E10_0000 (NS) 5E10_0000 (S)	4E2F_FFFF (NS) 5E2F_FFFF (S)	2048	DDR PHY
4E05_0000 (NS) 5E05_0000 (S)	4E0F_FFFF (NS) 5E0F_FFFF (S)	704	Reserved
4E04_0000 (NS) 5E04_0000 (S)	4E04_FFFF (NS) 5E04_FFFF (S)	64	Reserved
4E03_0000 (NS) 5E03_0000 (S)	4E03_FFFF (NS) 5E03_FFFF (S)	64	Reserved
4E02_0000 (NS) 5E02_0000 (S)	4E02_FFFF (NS) 5E02_FFFF (S)	64	Reserved
4E01_0000 (NS) 5E01_0000 (S)	4E01_FFFF (NS) 5E01_FFFF (S)	64	DDRMIX Block Control
4E00_0000 (NS) 5E00_0000 (S)	4E00_FFFF (NS) 5E00_FFFF (S)	64	Reserved
4D00_0000 (NS) 5D00_0000 (S)	4DFF_FFFF (NS) 5DFF_FFFF (S)	16384	Reserved
4C30_0000 (NS) 5C30_0000 (S)	4CFF_FFFF (NS) 5CFF_FFFF (S)	13312	Reserved
4C20_0000 (NS) 5C20_0000 (S)	4C2F_FFFF (NS) 5C2F_FFFF (S)	1024	USB2 Controller
4C10_0000 (NS) 5C10_0000 (S)	4C1F_FFFF (NS) 5C1F_FFFF (S)	1024	USB1 Controller
4C05_0000 (NS) 5C05_0000 (S)	4C0F_FFFF (NS) 5C0F_FFFF (S)	704	Reserved
4C04_0000 (NS) 5C04_0000 (S)	4C04_FFFF (NS) 5C04_FFFF (S)	64	Reserved
4C03_0000 (NS) 5C03_0000 (S)	4C03_FFFF (NS) 5C03_FFFF (S)	64	HSIOMIX TRDC Manager
4C02_0000 (NS) 5C02_0000 (S)	4C02_FFFF (NS) 5C02_FFFF (S)	64	Reserved

Table continues on the next page...

Table 4. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
4C02_0000 (NS) 5C02_0000 (S)	4C02_FFFF (NS) 5C02_FFFF (S)	64	Reserved
4C01_0000 (NS) 5C01_0000 (S)	4C01_FFFF (NS) 5C01_FFFF (S)	64	Reserved
4C00_0000 (NS) 5C00_0000 (S)	4C00_FFFF (NS) 5C00_FFFF (S)	64	Reserved
4BC0_0000 (NS) 5BC0_0000 (S)	4BFF_FFFF (NS) 5BFF_FFFF (S)	4096	Reserved
4B80_0000 (NS) 5B80_0000 (S)	4BBF_FFFF (NS) 5BBF_FFFF (S)	4096	Reserved
4B40_0000 (NS) 5B40_0000 (S)	4B7F_FFFF (NS) 5B7F_FFFF (S)	4096	Reserved
4B00_0000 (NS) 5B00_0000 (S)	4B3F_FFFF (NS) 5B3F_FFFF (S)	4096	Reserved
4AF0_0000 (NS) 5AF0_0000 (S)	4AFF_FFFF (NS) 5AFF_FFFF (S)	1024	Reserved
4AE5_0000 (NS) 5AE5_0000 (S)	4AEF_FFFF (NS) 5AEF_FFFF (S)	704	Reserved
4AE4_0000 (NS) 5AE4_0000 (S)	4AE4_FFFF (NS) 5AE4_FFFF (S)	64	ISI
4AE3_0000 (NS) 5AE3_0000 (S)	4AE3_FFFF (NS) 5AE3_FFFF (S)	64	LCDIF
4AE2_0000 (NS) 5AE2_0000 (S)	4AE2_FFFF (NS) 5AE2_FFFF (S)	64	PXP
4AE1_0000 (NS) 5AE1_0000 (S)	4AE1_FFFF (NS) 5AE1_FFFF (S)	64	MIPI DSI
4AE0_0000 (NS) 5AE0_0000 (S)	4AE0_FFFF (NS) 5AE0_FFFF (S)	64	MIPI CSI
4AD0_0000 (NS) 5AD0_0000 (S)	4ADF_FFFF (NS) 5ADF_FFFF (S)	1024	Reserved
4AC5_0000 (NS)	4ACF_FFFF (NS)	704	Reserved

Table continues on the next page...

Table 4. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
5AC5_0000 (S)	5ACF_FFFF (S)		
4AC4_0000 (NS) 5AC4_0000 (S)	4AC4_FFFF (NS) 5AC4_FFFF (S)	64	Reserved
4AC3_0000 (NS) 5AC3_0000 (S)	4AC3_FFFF (NS) 5AC3_FFFF (S)	64	MEDIAMIX TRDC Manager
4AC2_0000 (NS) 5AC2_0000 (S)	4AC2_FFFF (NS) 5AC2_FFFF (S)	64	Reserved
4AC1_0000 (NS) 5AC1_0000 (S)	4AC1_FFFF (NS) 5AC1_FFFF (S)	64	MEDIAMIX Block Control
4AC0_0000 (NS) 5AC0_0000 (S)	4AC0_FFFF (NS) 5AC0_FFFF (S)	64	Reserved
4A90_1000 (NS) 5A90_1000 (S)	4ABF_FFFF (NS) 5ABF_FFFF (S)	3068	Reserved
4A90_0000 (NS) 5A90_0000 (S)	4A90_0FFF (NS) 5A90_0FFF (S)	4	NPU Controller
4A85_0000 (NS) 5A85_0000 (S)	4A8F_FFFF (NS) 5A8F_FFFF (S)	704	Reserved
4A84_0000 (NS) 5A84_0000 (S)	4A84_FFFF (NS) 5A84_FFFF (S)	64	Reserved
4A83_0000 (NS) 5A83_0000 (S)	4A83_FFFF (NS) 5A83_FFFF (S)	64	Reserved
4A82_0000 (NS) 5A82_0000 (S)	4A82_FFFF (NS) 5A82_FFFF (S)	64	Reserved
4A81_0000 (NS) 5A81_0000 (S)	4A81_FFFF (NS) 5A81_FFFF (S)	64	NPUMIX Block Control
4A80_0000 (NS) 5A80_0000 (S)	4A80_FFFF (NS) 5A80_FFFF (S)	64	Reserved
4A50_0000 (NS) 5A50_0000 (S)	4A7F_FFFF (NS) 5A7F_FFFF (S)	3072	Reserved
4A45_0000 (NS) 5A45_0000 (S)	4A4F_FFFF (NS) 5A4F_FFFF (S)	704	Reserved

Table continues on the next page...

Table 4. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
4A44_0000 (NS) 5A44_0000 (S)	4A44_FFFF (NS) 5A44_FFFF (S)	64	Reserved
4A43_0000 (NS) 5A43_0000 (S)	4A43_FFFF (NS) 5A43_FFFF (S)	64	Reserved
4A42_0000 (NS) 5A42_0000 (S)	4A42_FFFF (NS) 5A42_FFFF (S)	64	Reserved
4A41_0000 (NS) 5A41_0000 (S)	4A41_FFFF (NS) 5A41_FFFF (S)	64	Reserved
4A40_0000 (NS) 5A40_0000 (S)	4A40_FFFF (NS) 5A40_FFFF (S)	64	Reserved
4A10_0000 (NS) 5A10_0000 (S)	4A3F_FFFF (NS) 5A3F_FFFF (S)	3072	Reserved
4A05_0000 (NS) 5A05_0000 (S)	4A0F_FFFF (NS) 5A0F_FFFF (S)	704	Reserved
4A04_0000 (NS) 5A04_0000 (S)	4A04_FFFF (NS) 5A04_FFFF (S)	64	Reserved
4A03_0000 (NS) 5A03_0000 (S)	4A03_FFFF (NS) 5A03_FFFF (S)	64	Reserved
4A02_0000 (NS) 5A02_0000 (S)	4A02_FFFF (NS) 5A02_FFFF (S)	64	Reserved
4A01_0000 (NS) 5A01_0000 (S)	4A01_FFFF (NS) 5A01_FFFF (S)	64	Reserved
4A00_0000 (NS) 5A00_0000 (S)	4A00_FFFF (NS) 5A00_FFFF (S)	64	Reserved
4980_0000 (NS) 5980_0000 (S)	49FF_FFFF (NS) 59FF_FFFF (S)	8192	Reserved
4900_0000 (NS) 5900_0000 (S)	497F_FFFF (NS) 597F_FFFF (S)	8192	AIPS Peripheral Bridge 4
4800_0000 (NS) 5800_0000 (S)	48FF_FFFF (NS) 58FF_FFFF (S)	16384	GIC Programming Registers
4758_0000 (NS)	47FF_FFFF (NS)	10752	Reserved

Table continues on the next page...

Table 4. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
5758_0000 (S)	57FF_FFFF (S)		
4750_0000 (NS) 5750_0000 (S)	4757_FFFF (NS) 5757_FFFF (S)	512	EdgeLock Enclave AHB Slave
474F_0000 (NS) 574F_0000 (S)	474F_FFFF (NS) 574F_FFFF (S)	64	Reserved
474E_0000 (NS) 574E_0000 (S)	474E_FFFF (NS) 574E_FFFF (S)	64	Reserved
474D_0000 (NS) 574D_0000 (S)	474D_FFFF (NS) 574D_FFFF (S)	64	Reserved
474C_0000 (NS) 574C_0000 (S)	474C_FFFF (NS) 574C_FFFF (S)	64	Reserved
474B_0000 (NS) 574B_0000 (S)	474B_FFFF (NS) 574B_FFFF (S)	64	Reserved
474A_0000 (NS) 574A_0000 (S)	474A_FFFF (NS) 574A_FFFF (S)	64	Reserved
4749_0000 (NS) 5749_0000 (S)	4749_FFFF (NS) 5749_FFFF (S)	64	Reserved
4748_0000 (NS) 5748_0000 (S)	4748_FFFF (NS) 5748_FFFF (S)	64	Reserved
4747_0000 (NS) 5747_0000 (S)	4747_FFFF (NS) 5747_FFFF (S)	64	Reserved
4746_0000 (NS) 5746_0000 (S)	4746_FFFF (NS) 5746_FFFF (S)	64	Reserved
4745_0000 (NS) 5745_0000 (S)	4745_FFFF (NS) 5745_FFFF (S)	64	Reserved
4744_0000 (NS) 5744_0000 (S)	4744_FFFF (NS) 5744_FFFF (S)	64	Reserved
4743_0400 (NS) 5743_0400 (S)	4743_FFFF (NS) 5743_FFFF (S)	63	Reserved
4743_0000 (NS) 5743_0000 (S)	4743_03FF (NS) 5743_03FF (S)	1	FlexSPI TX FIFO

Table continues on the next page...

Table 4. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
4742_0200 (NS) 5742_0200 (S)	4742_FFFF (NS) 5742_FFFF (S)	63.5	Reserved
4742_0000 (NS) 5742_0000 (S)	4742_01FF (NS) 5742_01FF (S)	0.5	FlexSPI RX FIFO
4741_0000 (NS) 5741_0000 (S)	4741_FFFF (NS) 5741_FFFF (S)	64	Reserved
4740_0000 (NS) 5740_0000 (S)	4740_FFFF (NS) 5740_FFFF (S)	64	GPIO1
4730_0000 (NS) 5730_0000 (S)	473F_FFFF (NS) 573F_FFFF (S)	1024	Reserved
4720_0000 (NS) 5720_0000 (S)	472F_FFFF (NS) 572F_FFFF (S)	1024	Reserved
4710_0000 (NS) 5710_0000 (S)	471F_FFFF (NS) 571F_FFFF (S)	1024	Reserved
4700_0000 (NS) 5700_0000 (S)	470F_FFFF (NS) 570F_FFFF (S)	1024	Reserved
4680_0000 (NS) 5680_0000 (S)	46FF_FFFF (NS) 56FF_FFFF (S)	8192	Reserved
4600_0000 (NS) 5600_0000 (S)	467F_FFFF (NS) 567F_FFFF (S)	8192	Reserved
4580_0000 (NS) 5580_0000 (S)	45FF_FFFF (NS) 55FF_FFFF (S)	8192	Reserved
4500_0000 (NS) 5500_0000 (S)	457F_FFFF (NS) 557F_FFFF (S)	8192	Reserved
4480_0000 (NS) 5480_0000 (S)	44FF_FFFF (NS) 54FF_FFFF (S)	8192	Reserved
4400_0000 (NS) 5400_0000 (S)	447F_FFFF (NS) 547F_FFFF (S)	8192	AIPS Peripheral Bridge 1
43A0_0000 (NS) 53A0_0000 (S)	43FF_FFFF (NS) 53FF_FFFF (S)	6144	Reserved
4390_0000 (NS)	439F_FFFF (NS)	1024	Reserved

Table continues on the next page...

Table 4. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
5390_0000 (S)	539F_FFFF (S)		
438F_0000 (NS) 538F_0000 (S)	438F_FFFF (NS) 538F_FFFF (S)	64	Reserved
438E_0000 (NS) 538E_0000 (S)	438E_FFFF (NS) 538E_FFFF (S)	64	Reserved
438D_0000 (NS) 538D_0000 (S)	438D_FFFF (NS) 538D_FFFF (S)	64	Reserved
438C_0000 (NS) 538C_0000 (S)	438C_FFFF (NS) 538C_FFFF (S)	64	Reserved
438B_0000 (NS) 538B_0000 (S)	438B_FFFF (NS) 538B_FFFF (S)	64	Reserved
438A_0000 (NS) 538A_0000 (S)	438A_FFFF (NS) 538A_FFFF (S)	64	Reserved
4389_0000 (NS) 5389_0000 (S)	4389_FFFF (NS) 5389_FFFF (S)	64	Reserved
4388_0000 (NS) 5388_0000 (S)	4388_FFFF (NS) 5388_FFFF (S)	64	Reserved
4387_0000 (NS) 5387_0000 (S)	4387_FFFF (NS) 5387_FFFF (S)	64	Reserved
4386_0000 (NS) 5386_0000 (S)	4386_FFFF (NS) 5386_FFFF (S)	64	Reserved
4385_0000 (NS) 5385_0000 (S)	4385_FFFF (NS) 5385_FFFF (S)	64	Reserved
4384_0000 (NS) 5384_0000 (S)	4384_FFFF (NS) 5384_FFFF (S)	64	Reserved
4383_0000 (NS) 5383_0000 (S)	4383_FFFF (NS) 5383_FFFF (S)	64	GPIO4
4382_0000 (NS) 5382_0000 (S)	4382_FFFF (NS) 5382_FFFF (S)	64	GPIO3
4381_0000 (NS) 5381_0000 (S)	4381_FFFF (NS) 5381_FFFF (S)	64	GPIO2

Table continues on the next page...

Table 4. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
4380_0000 (NS) 5380_0000 (S)	4380_FFFF (NS) 5380_FFFF (S)	64	EdgeLock Secure Enclave ISP_AP
4300_0000 (NS) 5300_0000 (S)	437F_FFFF (NS) 537F_FFFF (S)	8192	Reserved
4280_0000 (NS) 5280_0000 (S)	42FF_FFFF (NS) 52FF_FFFF (S)	8192	AIPS Peripheral Bridge 3
4200_0000 (NS) 5200_0000 (S)	427F_FFFF (NS) 527F_FFFF (S)	8192	AIPS Peripheral Bridge 2
4110_0000 (NS) 5110_0000 (S)	41FF_FFFF (NS) 51FF_FFFF (S)	15360	Reserved
4000_0000 (NS) 5000_0000 (S)	410F_FFFF (NS) 510F_FFFF (S)	17408	Debug - DAP
2800_0000 (NS) 3800_0000 (S)	2FFF_FFFF (NS) 3FFF_FFFF (S)	131072	Flexible Serial Peripheral Interface
2400_0000 (NS) 3400_0000 (S)	27FF_FFFF (NS) 37FF_FFFF (S)	65536	Reserved
2200_0000 (NS) 3200_0000 (S)	23FF_FFFF (NS) 33FF_FFFF (S)	32768	Reserved
20D0_0000 (NS) 30D0_0000 (S)	21FF_FFFF (NS) 31FF_FFFF (S)	19456	Reserved
20C0_0000 (NS) 30C0_0000 (S)	20CF_FFFF (NS) 30CF_FFFF (S)	1024	Reserved
20B0_0000 (NS) 30B0_0000 (S)	20BF_FFFF (NS) 30BF_FFFF (S)	1024	Reserved
20A0_0000 (NS) 30A0_0000 (S)	20AF_FFFF (NS) 30AF_FFFF (S)	1024	Reserved
2090_0000 (NS) 3090_0000 (S)	209F_FFFF (NS) 309F_FFFF (S)	1024	Reserved
2080_0000 (NS) 3080_0000 (S)	208F_FFFF (NS) 308F_FFFF (S)	1024	Reserved
2070_8000 (NS)	207F_FFFF (NS)	992	Reserved

Table continues on the next page...

Table 4. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
3070_8000 (S)	307F_FFFF (S)		
2070_0000 (NS) 3070_0000 (S)	2070_7FFF (NS) 3070_7FFF (S)	32	Reserved
2068_0000 (NS) 3068_0000 (S)	206F_FFFF (NS) 306F_FFFF (S)	512	Reserved
2060_0000 (NS) 3060_0000 (S)	2067_FFFF (NS) 3067_FFFF (S)	512	Reserved
2058_0000 (NS) 3058_0000 (S)	205F_FFFF (NS) 305F_FFFF (S)	512	Reserved
2052_0000 (NS) 3052_0000 (S)	2057_FFFF (NS) 3057_FFFF (S)	384	Reserved
2048_0000 (NS) 3048_0000 (S)	2051_FFFF (NS) 3051_FFFF (S)	640	OCRAM (640 KB)
2044_0000 (NS) 3044_0000 (S)	2047_FFFF (NS) 3047_FFFF (S)	256	Reserved
2040_0000 (NS) 3040_0000 (S)	2043_FFFF (NS) 3043_FFFF (S)	256	Reserved
203C_0000 (NS) 303C_0000 (S)	203F_FFFF (NS) 303F_FFFF (S)	256	Reserved
2038_0000 (NS) 3038_0000 (S)	203B_FFFF (NS) 303B_FFFF (S)	256	Reserved
2030_0000 (NS) 3030_0000 (S)	2037_FFFF (NS) 3037_FFFF (S)	512	Reserved
2024_0000 (NS) 3024_0000 (S)	202F_FFFF (NS) 302F_FFFF (S)	768	Reserved
2022_0000 (NS) 3022_0000 (S)	2023_FFFF (NS) 3023_FFFF (S)	128	M33 System TCM (continued FlexRAM aliased if M33 SW config option Code TCM=0KB)
2020_0000 (NS) 3020_0000 (S)	2021_FFFF (NS) 3021_FFFF (S)	128	M33 System TCM (FlexRAM aliased)
201E_0000 (NS) 301E_0000 (S)	201F_FFFF (NS) 301F_FFFF (S)	128	M33 Code TCM (FlexRAM aliased)

Table continues on the next page...

Table 4. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
201C_0000 (NS) 301C_0000 (S)	201D_FFFF (NS) 301D_FFFF (S)	128	M33 Code TCM (continued FlexRAM aliased if M33 SW config option System TCM=0KB)
2010_0000 (NS) 3010_0000 (S)	201B_FFFF (NS) 301B_FFFF (S)	768	Reserved
2008_0000 (NS) 3008_0000 (S)	200F_FFFF (NS) 300F_FFFF (S)	512	Reserved
2004_0000 (NS) 3004_0000 (S)	2007_FFFF (NS) 3007_FFFF (S)	256	Reserved
2002_0000 (NS) 3002_0000 (S)	2003_FFFF (NS) 3003_FFFF (S)	128	Reserved
2000_0000 (NS) 3000_0000 (S)	2001_FFFF (NS) 3001_FFFF (S)	128	Reserved
0FFE_0000 (NS) 1FFE_0000 (S)	0FFF_FFFF (NS) 1FFF_FFFF (S)	128	Reserved
0FFC_0000 (NS) 1FFC_0000 (S)	0FFD_FFFF (NS) 1FFD_FFFF (S)	128	Reserved
0FF0_0000 (NS) 1FF0_0000 (S)	0FFB_FFFF (NS) 1FFB_FFFF (S)	768	Reserved
0E00_0000 (NS) 1E00_0000 (S)	0FEF_FFFF (NS) 1FEF_FFFF (S)	31744	Reserved
0C00_0000 (NS) 1C00_0000 (S)	0DFF_FFFF (NS) 1DFF_FFFF (S)	32768	Reserved
0400_0000 (NS) 1400_0000 (S)	0BFF_FFFF (NS) 1BFF_FFFF (S)	131072	Reserved
0200_0000 (NS) 1200_0000 (S)	03FF_FFFF (NS) 13FF_FFFF (S)	32768	Reserved
0030_0000 (NS) 1030_0000 (S)	01FF_FFFF (NS) 11FF_FFFF (S)	29696	Reserved
0024_0000 (NS) 1024_0000 (S)	002F_FFFF (NS) 102F_FFFF (S)	768	Reserved
0020_0000 (NS)	0023_FFFF (NS)	256	Reserved

Table continues on the next page...

Table 4. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
1020_0000 (S)	1023_FFFF (S)		
0014_0000 (NS) 1014_0000 (S)	001F_FFFF (NS) 101F_FFFF (S)	768	Reserved
0010_0000 (NS) 1010_0000 (S)	0013_FFFF (NS) 1013_FFFF (S)	256	M33 Boot ROM (aliased)
0008_0000 (NS) 1008_0000 (S)	000F_FFFF (NS) 100F_FFFF (S)	512	Reserved
0004_0000 (NS) 1004_0000 (S)	0007_FFFF (NS) 1007_FFFF (S)	256	Reserved
0002_0000 (NS) 1002_0000 (S)	0003_FFFF (NS) 1003_FFFF (S)	128	A55 Boot ROM (continued)
0000_0000 (NS) 1000_0000 (S)	0001_FFFF (NS) 1001_FFFF (S)	128	A55 Boot ROM (256 KB)

2.3 System memory map (Cortex-M33)

The following table provides the TrustZone-M "Non-Secure (NS)" and "Secure (S)" world view of the Cortex-M33 memory map. Access through the "Secure" memory region can only be performed by code with the appropriate security settings.

Table 5. System memory map (Cortex-M33)

Start address (hex)	End address (hex)	Size (KB)	Description
E010_0000 (NS) F010_0000 (S)	EFFF_FFFF (NS) FFFF_FFFF (S)	261120	Reserved
E004_4000 (NS) F004_4000 (S)	E00F_FFFF (NS) F00F_FFFF (S)	752	M33 Private Peripheral Bus - External
E000_0000 (NS) F000_0000 (S)	E004_3FFF (NS) F004_3FFF (S)	272	M33 Private Peripheral Bus - Internal
C000_0000 (NS) D000_0000 (S)	CFFF_FFFF (NS) DFFF_FFFF (S)	262144	DRAM (offset 1 GB)
A000_0000 (NS) B000_0000 (S)	AFFF_FFFF (NS) BFFF_FFFF (S)	262144	DRAM (offset 512 MB)
8000_0000 (NS) 9000_0000 (S)	8FFF_FFFF (NS) 9FFF_FFFF (S)	262144	DRAM (offset 0 MB)
6000_0000 (NS)	6FFF_FFFF (NS)	262144	Reserved

Table continues on the next page...

Table 5. System memory map (Cortex-M33) (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
7000_0000 (S)	7FFF_FFFF (S)		
4FC0_0000 (NS) 5FC0_0000 (S)	4FFF_FFFF (NS) 5FFF_FFFF (S)	4096	Reserved
4E31_0000 (NS) 5E31_0000 (S)	4FBF_FFFF (NS) 5FBF_FFFF (S)	25536	Reserved
4E30_0000 (NS) 5E30_0000 (S)	4E30_FFFF (NS) 5E30_FFFF (S)	64	DDR Controller
4E10_0000 (NS) 5E10_0000 (S)	4E2F_FFFF (NS) 5E2F_FFFF (S)	2048	DDR PHY
4E05_0000 (NS) 5E05_0000 (S)	4E0F_FFFF (NS) 5E0F_FFFF (S)	704	Reserved
4E04_0000 (NS) 5E04_0000 (S)	4E04_FFFF (NS) 5E04_FFFF (S)	64	Reserved
4E03_0000 (NS) 5E03_0000 (S)	4E03_FFFF (NS) 5E03_FFFF (S)	64	Reserved
4E02_0000 (NS) 5E02_0000 (S)	4E02_FFFF (NS) 5E02_FFFF (S)	64	Reserved
4E01_0000 (NS) 5E01_0000 (S)	4E01_FFFF (NS) 5E01_FFFF (S)	64	DDRMIX Block Control
4E00_0000 (NS) 5E00_0000 (S)	4E00_FFFF (NS) 5E00_FFFF (S)	64	Reserved
4D00_0000 (NS) 5D00_0000 (S)	4DFF_FFFF (NS) 5DFF_FFFF (S)	16384	Reserved
4C30_0000 (NS) 5C30_0000 (S)	4CFF_FFFF (NS) 5CFF_FFFF (S)	13312	Reserved
4C20_0000 (NS) 5C20_0000 (S)	4C2F_FFFF (NS) 5C2F_FFFF (S)	1024	USB2 Controller
4C10_0000 (NS) 5C10_0000 (S)	4C1F_FFFF (NS) 5C1F_FFFF (S)	1024	USB1 Controller
4C05_0000 (NS) 5C05_0000 (S)	4C0F_FFFF (NS) 5C0F_FFFF (S)	704	Reserved

Table continues on the next page...

Table 5. System memory map (Cortex-M33) (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
4C04_0000 (NS) 5C04_0000 (S)	4C04_FFFF (NS) 5C04_FFFF (S)	64	Reserved
4C03_0000 (NS) 5C03_0000 (S)	4C03_FFFF (NS) 5C03_FFFF (S)	64	HSIOMIX TRDC Manager
4C02_0000 (NS) 5C02_0000 (S)	4C02_FFFF (NS) 5C02_FFFF (S)	64	Reserved
4C01_0000 (NS) 5C01_0000 (S)	4C01_FFFF (NS) 5C01_FFFF (S)	64	Reserved
4C00_0000 (NS) 5C00_0000 (S)	4C00_FFFF (NS) 5C00_FFFF (S)	64	Reserved
4BC0_0000 (NS) 5BC0_0000 (S)	4BFF_FFFF (NS) 5BFF_FFFF (S)	4096	Reserved
4B80_0000 (NS) 5B80_0000 (S)	4BBF_FFFF (NS) 5BBF_FFFF (S)	4096	Reserved
4B40_0000 (NS) 5B40_0000 (S)	4B7F_FFFF (NS) 5B7F_FFFF (S)	4096	Reserved
4B00_0000 (NS) 5B00_0000 (S)	4B3F_FFFF (NS) 5B3F_FFFF (S)	4096	Reserved
4AF0_0000 (NS) 5AF0_0000 (S)	4AFF_FFFF (NS) 5AFF_FFFF (S)	1024	Reserved
4AE5_0000 (NS) 5AE5_0000 (S)	4AEF_FFFF (NS) 5AEF_FFFF (S)	704	Reserved
4AE4_0000 (NS) 5AE4_0000 (S)	4AE4_FFFF (NS) 5AE4_FFFF (S)	64	ISI
4AE3_0000 (NS) 5AE3_0000 (S)	4AE3_FFFF (NS) 5AE3_FFFF (S)	64	LCDIF
4AE2_0000 (NS) 5AE2_0000 (S)	4AE2_FFFF (NS) 5AE2_FFFF (S)	64	PXP
4AE1_0000 (NS) 5AE1_0000 (S)	4AE1_FFFF (NS) 5AE1_FFFF (S)	64	MIPI DSI
4AE0_0000 (NS)	4AE0_FFFF (NS)	64	MIPI CSI

Table continues on the next page...

Table 5. System memory map (Cortex-M33) (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
5AE0_0000 (S)	5AE0_FFFF (S)		
4AD0_0000 (NS) 5AD0_0000 (S)	4ADF_FFFF (NS) 5ADF_FFFF (S)	1024	NIC_MEDIA GPV Control Registers
4AC5_0000 (NS) 5AC5_0000 (S)	4ACF_FFFF (NS) 5ACF_FFFF (S)	704	Reserved
4AC4_0000 (NS) 5AC4_0000 (S)	4AC4_FFFF (NS) 5AC4_FFFF (S)	64	Reserved
4AC3_0000 (NS) 5AC3_0000 (S)	4AC3_FFFF (NS) 5AC3_FFFF (S)	64	MEDIAMIX TRDC Manager
4AC2_0000 (NS) 5AC2_0000 (S)	4AC2_FFFF (NS) 5AC2_FFFF (S)	64	Reserved
4AC1_0000 (NS) 5AC1_0000 (S)	4AC1_FFFF (NS) 5AC1_FFFF (S)	64	MEDIAMIX Block Control
4AC0_0000 (NS) 5AC0_0000 (S)	4AC0_FFFF (NS) 5AC0_FFFF (S)	64	Reserved
4A90_1000 (NS) 5A90_1000 (S)	4ABF_FFFF (NS) 5ABF_FFFF (S)	3068	Reserved
4A90_0000 (NS) 5A90_0000 (S)	4A90_0FFF (NS) 5A90_0FFF (S)	4	NPU Controller
4A85_0000 (NS) 5A85_0000 (S)	4A8F_FFFF (NS) 5A8F_FFFF (S)	704	Reserved
4A84_0000 (NS) 5A84_0000 (S)	4A84_FFFF (NS) 5A84_FFFF (S)	64	Reserved
4A83_0000 (NS) 5A83_0000 (S)	4A83_FFFF (NS) 5A83_FFFF (S)	64	Reserved
4A82_0000 (NS) 5A82_0000 (S)	4A82_FFFF (NS) 5A82_FFFF (S)	64	Reserved
4A81_0000 (NS) 5A81_0000 (S)	4A81_FFFF (NS) 5A81_FFFF (S)	64	NPUMIX Block Control
4A80_0000 (NS) 5A80_0000 (S)	4A80_FFFF (NS) 5A80_FFFF (S)	64	Reserved

Table continues on the next page...

Table 5. System memory map (Cortex-M33) (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
4A50_0000 (NS) 5A50_0000 (S)	4A7F_FFFF (NS) 5A7F_FFFF (S)	3072	Reserved
4A45_0000 (NS) 5A45_0000 (S)	4A4F_FFFF (NS) 5A4F_FFFF (S)	704	Reserved
4A44_0000 (NS) 5A44_0000 (S)	4A44_FFFF (NS) 5A44_FFFF (S)	64	Reserved
4A43_0000 (NS) 5A43_0000 (S)	4A43_FFFF (NS) 5A43_FFFF (S)	64	Reserved
4A42_0000 (NS) 5A42_0000 (S)	4A42_FFFF (NS) 5A42_FFFF (S)	64	Reserved
4A41_0000 (NS) 5A41_0000 (S)	4A41_FFFF (NS) 5A41_FFFF (S)	64	Reserved
4A40_0000 (NS) 5A40_0000 (S)	4A40_FFFF (NS) 5A40_FFFF (S)	64	Reserved
4A10_0000 (NS) 5A10_0000 (S)	4A3F_FFFF (NS) 5A3F_FFFF (S)	3072	Reserved
4A05_0000 (NS) 5A05_0000 (S)	4A0F_FFFF (NS) 5A0F_FFFF (S)	704	Reserved
4A04_0000 (NS) 5A04_0000 (S)	4A04_FFFF (NS) 5A04_FFFF (S)	64	Reserved
4A03_0000 (NS) 5A03_0000 (S)	4A03_FFFF (NS) 5A03_FFFF (S)	64	Reserved
4A02_0000 (NS) 5A02_0000 (S)	4A02_FFFF (NS) 5A02_FFFF (S)	64	Reserved
4A01_0000 (NS) 5A01_0000 (S)	4A01_FFFF (NS) 5A01_FFFF (S)	64	Reserved
4A00_0000 (NS) 5A00_0000 (S)	4A00_FFFF (NS) 5A00_FFFF (S)	64	Reserved
4980_0000 (NS) 5980_0000 (S)	49FF_FFFF (NS) 59FF_FFFF (S)	8192	Reserved
4900_0000 (NS)	497F_FFFF (NS)	8192	AIPS Peripheral Bridge 4

Table continues on the next page...

Table 5. System memory map (Cortex-M33) (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
5900_0000 (S)	597F_FFFF (S)		
4800_0000 (NS) 5800_0000 (S)	48FF_FFFF (NS) 58FF_FFFF (S)	16384	GIC Programming Registers
4758_0000 (NS) 5758_0000 (S)	47FF_FFFF (NS) 57FF_FFFF (S)	10752	Reserved
4750_0000 (NS) 5750_0000 (S)	4757_FFFF (NS) 5757_FFFF (S)	512	EdgeLock Secure Enclave AHB Slave
474F_0000 (NS) 574F_0000 (S)	474F_FFFF (NS) 574F_FFFF (S)	64	Reserved
474E_0000 (NS) 574E_0000 (S)	474E_FFFF (NS) 574E_FFFF (S)	64	Reserved
474D_0000 (NS) 574D_0000 (S)	474D_FFFF (NS) 574D_FFFF (S)	64	Reserved
474C_0000 (NS) 574C_0000 (S)	474C_FFFF (NS) 574C_FFFF (S)	64	Reserved
474B_0000 (NS) 574B_0000 (S)	474B_FFFF (NS) 574B_FFFF (S)	64	Reserved
474A_0000 (NS) 574A_0000 (S)	474A_FFFF (NS) 574A_FFFF (S)	64	Reserved
4749_0000 (NS) 5749_0000 (S)	4749_FFFF (NS) 5749_FFFF (S)	64	Reserved
4748_0000 (NS) 5748_0000 (S)	4748_FFFF (NS) 5748_FFFF (S)	64	Reserved
4747_0000 (NS) 5747_0000 (S)	4747_FFFF (NS) 5747_FFFF (S)	64	Reserved
4746_0000 (NS) 5746_0000 (S)	4746_FFFF (NS) 5746_FFFF (S)	64	Reserved
4745_0000 (NS) 5745_0000 (S)	4745_FFFF (NS) 5745_FFFF (S)	64	Reserved
4744_0000 (NS) 5744_0000 (S)	4744_FFFF (NS) 5744_FFFF (S)	64	Reserved

Table continues on the next page...

Table 5. System memory map (Cortex-M33) (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
4743_0400 (NS) 5743_0400 (S)	4743_FFFF (NS) 5743_FFFF (S)	63	Reserved
4743_0000 (NS) 5743_0000 (S)	4743_03FF (NS) 5743_03FF (S)	1	FlexSPI1 TX FIFO
4742_0200 (NS) 5742_0200 (S)	4742_FFFF (NS) 5742_FFFF (S)	63.5	Reserved
4742_0000 (NS) 5742_0000 (S)	4742_01FF (NS) 5742_01FF (S)	0.5	FlexSPI1 RX FIFO
4741_0000 (NS) 5741_0000 (S)	4741_FFFF (NS) 5741_FFFF (S)	64	Reserved
4740_0000 (NS) 5740_0000 (S)	4740_FFFF (NS) 5740_FFFF (S)	64	GPIO1
4730_0000 (NS) 5730_0000 (S)	473F_FFFF (NS) 573F_FFFF (S)	1024	Reserved
4720_0000 (NS) 5720_0000 (S)	472F_FFFF (NS) 572F_FFFF (S)	1024	Reserved
4710_0000 (NS) 5710_0000 (S)	471F_FFFF (NS) 571F_FFFF (S)	1024	Reserved
4700_0000 (NS) 5700_0000 (S)	470F_FFFF (NS) 570F_FFFF (S)	1024	Reserved
4680_0000 (NS) 5680_0000 (S)	46FF_FFFF (NS) 56FF_FFFF (S)	8192	Reserved
4600_0000 (NS) 5600_0000 (S)	467F_FFFF (NS) 567F_FFFF (S)	8192	Reserved
4580_0000 (NS) 5580_0000 (S)	45FF_FFFF (NS) 55FF_FFFF (S)	8192	Reserved
4500_0000 (NS) 5500_0000 (S)	457F_FFFF (NS) 557F_FFFF (S)	8192	Reserved
4480_0000 (NS) 5480_0000 (S)	44FF_FFFF (NS) 54FF_FFFF (S)	8192	Reserved
4400_0000 (NS)	447F_FFFF (NS)	8192	AIPS Peripheral Bridge 1

Table continues on the next page...

Table 5. System memory map (Cortex-M33) (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
5400_0000 (S)	547F_FFFF (S)		
43A0_0000 (NS) 53A0_0000 (S)	43FF_FFFF (NS) 53FF_FFFF (S)	6144	Reserved
4390_0000 (NS) 5390_0000 (S)	439F_FFFF (NS) 539F_FFFF (S)	1024	NIC_MAIN GPV Registers
438F_0000 (NS) 538F_0000 (S)	438F_FFFF (NS) 538F_FFFF (S)	64	Reserved
438E_0000 (NS) 538E_0000 (S)	438E_FFFF (NS) 538E_FFFF (S)	64	Reserved
438D_0000 (NS) 538D_0000 (S)	438D_FFFF (NS) 538D_FFFF (S)	64	Reserved
438C_0000 (NS) 538C_0000 (S)	438C_FFFF (NS) 538C_FFFF (S)	64	Reserved
438B_0000 (NS) 538B_0000 (S)	438B_FFFF (NS) 538B_FFFF (S)	64	Reserved
438A_0000 (NS) 538A_0000 (S)	438A_FFFF (NS) 538A_FFFF (S)	64	Reserved
4389_0000 (NS) 5389_0000 (S)	4389_FFFF (NS) 5389_FFFF (S)	64	Reserved
4388_0000 (NS) 5388_0000 (S)	4388_FFFF (NS) 5388_FFFF (S)	64	Reserved
4387_0000 (NS) 5387_0000 (S)	4387_FFFF (NS) 5387_FFFF (S)	64	Reserved
4386_0000 (NS) 5386_0000 (S)	4386_FFFF (NS) 5386_FFFF (S)	64	Reserved
4385_0000 (NS) 5385_0000 (S)	4385_FFFF (NS) 5385_FFFF (S)	64	Reserved
4384_0000 (NS) 5384_0000 (S)	4384_FFFF (NS) 5384_FFFF (S)	64	Reserved
4383_0000 (NS) 5383_0000 (S)	4383_FFFF (NS) 5383_FFFF (S)	64	GPIO4

Table continues on the next page...

Table 5. System memory map (Cortex-M33) (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
4382_0000 (NS) 5382_0000 (S)	4382_FFFF (NS) 5382_FFFF (S)	64	GPIO3
4381_0000 (NS) 5381_0000 (S)	4381_FFFF (NS) 5381_FFFF (S)	64	GPIO2
4380_0000 (NS) 5380_0000 (S)	4380_FFFF (NS) 5380_FFFF (S)	64	EdgeLock Secure Enclave SP_AP
4300_0000 (NS) 5300_0000 (S)	437F_FFFF (NS) 537F_FFFF (S)	8192	Reserved
4280_0000 (NS) 5280_0000 (S)	42FF_FFFF (NS) 52FF_FFFF (S)	8192	AIPS Peripheral Bridge 3
4200_0000 (NS) 5200_0000 (S)	427F_FFFF (NS) 527F_FFFF (S)	8192	AIPS Peripheral Bridge 2
4110_0000 (NS) 5110_0000 (S)	41FF_FFFF (NS) 51FF_FFFF (S)	15360	Reserved
4000_0000 (NS) 5000_0000 (S)	410F_FFFF (NS) 510F_FFFF (S)	17408	Debug - DAP
2800_0000 (NS) 3800_0000 (S)	2FFF_FFFF (NS) 3FFF_FFFF (S)	131072	Flexible Serial Peripheral Interface
2400_0000 (NS) 3400_0000 (S)	27FF_FFFF (NS) 37FF_FFFF (S)	65536	Reserved
2200_0000 (NS) 3200_0000 (S)	23FF_FFFF (NS) 33FF_FFFF (S)	32768	Reserved
20D0_0000 (NS) 30D0_0000 (S)	21FF_FFFF (NS) 31FF_FFFF (S)	19456	Reserved
20C0_0000 (NS) 30C0_0000 (S)	20CF_FFFF (NS) 30CF_FFFF (S)	1024	Reserved
20B0_0000 (NS) 30B0_0000 (S)	20BF_FFFF (NS) 30BF_FFFF (S)	1024	Reserved
20A0_0000 (NS) 30A0_0000 (S)	20AF_FFFF (NS) 30AF_FFFF (S)	1024	Reserved
2090_0000 (NS)	209F_FFFF (NS)	1024	Reserved

Table continues on the next page...

Table 5. System memory map (Cortex-M33) (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
3090_0000 (S)	309F_FFFF (S)		
2080_0000 (NS) 3080_0000 (S)	208F_FFFF (NS) 308F_FFFF (S)	1024	Reserved
2070_8000 (NS) 3070_8000 (S)	207F_FFFF (NS) 307F_FFFF (S)	992	Reserved
2070_0000 (NS) 3070_0000 (S)	2070_7FFF (NS) 3070_7FFF (S)	32	Reserved
2068_0000 (NS) 3068_0000 (S)	206F_FFFF (NS) 306F_FFFF (S)	512	Reserved
2060_0000 (NS) 3060_0000 (S)	2067_FFFF (NS) 3067_FFFF (S)	512	Reserved
2058_0000 (NS) 3058_0000 (S)	205F_FFFF (NS) 305F_FFFF (S)	512	Reserved
2052_0000 (NS) 3052_0000 (S)	2057_FFFF (NS) 3057_FFFF (S)	384	Reserved
2048_0000 (NS) 3048_0000 (S)	2051_FFFF (NS) 3051_FFFF (S)	640	OCRAM (640 KB)
2044_0000 (NS) 3044_0000 (S)	2047_FFFF (NS) 3047_FFFF (S)	256	Reserved
2040_0000 (NS) 3040_0000 (S)	2043_FFFF (NS) 3043_FFFF (S)	256	Reserved
203C_0000 (NS) 303C_0000 (S)	203F_FFFF (NS) 303F_FFFF (S)	256	Reserved
2038_0000 (NS) 3038_0000 (S)	203B_FFFF (NS) 303B_FFFF (S)	256	Reserved
2030_0000 (NS) 3030_0000 (S)	2037_FFFF (NS) 3037_FFFF (S)	512	Reserved
2024_0000 (NS) 3024_0000 (S)	202F_FFFF (NS) 302F_FFFF (S)	768	Reserved
2022_0000 (NS) 3022_0000 (S)	2023_FFFF (NS) 3023_FFFF (S)	128	Reserved

Table continues on the next page...

Table 5. System memory map (Cortex-M33) (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
2020_0000 (NS) 3020_0000 (S)	2021_FFFF (NS) 3021_FFFF (S)	128	Reserved
201E_0000 (NS) 301E_0000 (S)	201F_FFFF (NS) 301F_FFFF (S)	128	Reserved
201C_0000 (NS) 301C_0000 (S)	201D_FFFF (NS) 301D_FFFF (S)	128	Reserved
2010_0000 (NS) 3010_0000 (S)	201B_FFFF (NS) 301B_FFFF (S)	768	Reserved
2008_0000 (NS) 3008_0000 (S)	200F_FFFF (NS) 300F_FFFF (S)	512	Reserved
2004_0000 (NS) 3004_0000 (S)	2007_FFFF (NS) 3007_FFFF (S)	256	Reserved
2002_0000 (NS) 3002_0000 (S)	2003_FFFF (NS) 3003_FFFF (S)	128	M33 System TCM (128KB, if M33 SW config option Code TCM=0KB)
2000_0000 (NS) 3000_0000 (S)	2001_FFFF (NS) 3001_FFFF (S)	128	M33 System TCM (128KB)
0FFE_0000 (NS) 1FFE_0000 (S)	0FFF_FFFF (NS) 1FFF_FFFF (S)	128	M33 Code TCM (128KB, grows down from all F's)
0FFC_0000 (NS) 1FFC_0000 (S)	0FFD_FFFF (NS) 1FFD_FFFF (S)	128	M33 Code TCM (128KB, if M33 SW config option System TCM=0KB)
0FF0_0000 (NS) 1FF0_0000 (S)	0FFB_FFFF (NS) 1FFB_FFFF (S)	768	Reserved
0E00_0000 (NS) 1E00_0000 (S)	0FEF_FFFF (NS) 1FEF_FFFF (S)	31744	Reserved
0C00_0000 (NS) 1C00_0000 (S)	0DFF_FFFF (NS) 1DFF_FFFF (S)	32768	DRAM Code Space (aliased First 32MB of DRAM at 0x8000_0000)
0400_0000 (NS) 1400_0000 (S)	0BFF_FFFF (NS) 1BFF_FFFF (S)	131072	Reserved
0200_0000 (NS) 1200_0000 (S)	03FF_FFFF (NS) 13FF_FFFF (S)	32768	Flexible Serial Peripheral Interface (aliased first 32MB of FLEXSPI1)
0030_0000 (NS)	01FF_FFFF (NS)	29696	Reserved

Table continues on the next page...

Table 5. System memory map (Cortex-M33) (continued)

Start address (hex)	End address (hex)	Size (KB)	Description
1030_0000 (S)	11FF_FFFF (S)		
0024_0000 (NS) 1024_0000 (S)	002F_FFFF (NS) 102F_FFFF (S)	768	Reserved
0020_0000 (NS) 1020_0000 (S)	0023_FFFF (NS) 1023_FFFF (S)	256	A55 Boot ROM (aliased)
0014_0000 (NS) 1014_0000 (S)	001F_FFFF (NS) 101F_FFFF (S)	768	Reserved
0010_0000 (NS) 1010_0000 (S)	0013_FFFF (NS) 1013_FFFF (S)	256	Reserved
0008_0000 (NS) 1008_0000 (S)	000F_FFFF (NS) 100F_FFFF (S)	512	Reserved
0004_0000 (NS) 1004_0000 (S)	0007_FFFF (NS) 1007_FFFF (S)	256	Reserved
0002_0000 (NS) 1002_0000 (S)	0003_FFFF (NS) 1003_FFFF (S)	128	M33 Boot ROM (continued)
0000_0000 (NS) 1000_0000 (S)	0001_FFFF (NS) 1001_FFFF (S)	128	M33 Boot ROM (256 KB)

2.4 AIPS Peripheral Bridge 1 Memory Map

The following table provides the AIPS-1 memory map. The peripherals have both Secure (S) and Non-Secure (NS) access possibilities. Both, Secure and Non-Secure addresses for each peripheral is provided in the table below.

The start address of AIPS-1 peripherals for the Non-Secure access is 0x4400_0000.

The start address of AIPS-1 peripherals for the Secure access is 0x5400_0000.

NOTE

AIPS1 Peripherals are in the AONMIX domain.

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Enhanced Direct Memory Access 3 - Management Page	64	44000000	4400FFFF	54000000	5400FFFF	0	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Enhanced Direct Memory Access 3 - Channel 0 Control and Configuration	64	44010000	4401FFFF	54010000	5401FFFF	1	
Enhanced Direct Memory Access 3 - Channel 1 Control and Configuration	64	44020000	4402FFFF	54020000	5402FFFF	2	
Enhanced Direct Memory Access 3 - Channel 2 Control and Configuration	64	44030000	4403FFFF	54030000	5403FFFF	3	
Enhanced Direct Memory Access 3 - Channel 3 Control and Configuration	64	44040000	4404FFFF	54040000	5404FFFF	4	
Enhanced Direct Memory Access 3 - Channel 4 Control and Configuration	64	44050000	4405FFFF	54050000	5405FFFF	5	
Enhanced Direct Memory Access 3 - Channel 5 Control and Configuration	64	44060000	4406FFFF	54060000	5406FFFF	6	
Enhanced Direct Memory	64	44070000	4407FFFF	54070000	5407FFFF	7	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Access 3 - Channel 6 Control and Configuration							
Enhanced Direct Memory Access 3 - Channel 7 Control and Configuration	64	44080000	4408FFFF	54080000	5408FFFF	8	
Enhanced Direct Memory Access 3 - Channel 8 Control and Configuration	64	44090000	4409FFFF	54090000	5409FFFF	9	
Enhanced Direct Memory Access 3 - Channel 9 Control and Configuration	64	440A0000	440AFFFF	540A0000	540AFFFF	10	
Enhanced Direct Memory Access 3 - Channel 10 Control and Configuration	64	440B0000	440BFFFF	540B0000	540BFFFF	11	
Enhanced Direct Memory Access 3 - Channel 11 Control and Configuration	64	440C0000	440CFFFF	540C0000	540CFFFF	12	
Enhanced Direct Memory Access 3 - Channel 12	64	440D0000	440DFFFF	540D0000	540DFFFF	13	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Control and Configuration							
Enhanced Direct Memory Access 3 - Channel 13 Control and Configuration	64	440E0000	440EFFFF	540E0000	540EFFFF	14	
Enhanced Direct Memory Access 3 - Channel 14 Control and Configuration	64	440F0000	440FFFFF	540F0000	540FFFFF	15	
Enhanced Direct Memory Access 3 - Channel 15 Control and Configuration	64	44100000	4410FFFF	54100000	5410FFFF	16	
Enhanced Direct Memory Access 3 - Channel 16 Control and Configuration	64	44110000	4411FFFF	54110000	5411FFFF	17	
Enhanced Direct Memory Access 3 - Channel 17 Control and Configuration	64	44120000	4412FFFF	54120000	5412FFFF	18	
Enhanced Direct Memory Access 3 - Channel 18 Control and Configuration	64	44130000	4413FFFF	54130000	5413FFFF	19	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Enhanced Direct Memory Access 3 - Channel 19 Control and Configuration	64	44140000	4414FFFF	54140000	5414FFFF	20	
Enhanced Direct Memory Access 3 - Channel 20 Control and Configuration	64	44150000	4415FFFF	54150000	5415FFFF	21	
Enhanced Direct Memory Access 3 - Channel 21 Control and Configuration	64	44160000	4416FFFF	54160000	5416FFFF	22	
Enhanced Direct Memory Access 3 - Channel 22 Control and Configuration	64	44170000	4417FFFF	54170000	5417FFFF	23	
Enhanced Direct Memory Access 3 - Channel 23 Control and Configuration	64	44180000	4418FFFF	54180000	5418FFFF	24	
Enhanced Direct Memory Access 3 - Channel 24 Control and Configuration	64	44190000	4419FFFF	54190000	5419FFFF	25	
Enhanced Direct Memory	64	441A0000	441AFFFF	541A0000	541AFFFF	26	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Access 3 - Channel 25 Control and Configuration							
Enhanced Direct Memory Access 3 - Channel 26 Control and Configuration	64	441B0000	441BFFFF	541B0000	541BFFFF	27	
Enhanced Direct Memory Access 3 - Channel 27 Control and Configuration	64	441C0000	441CFFFF	541C0000	541CFFFF	28	
Enhanced Direct Memory Access 3 - Channel 28 Control and Configuration	64	441D0000	441DFFFF	541D0000	541DFFFF	29	
Enhanced Direct Memory Access 3 - Channel 29 Control and Configuration	64	441E0000	441EFFFF	541E0000	541EFFFF	30	
Enhanced Direct Memory Access 3 - Channel 30 Control and Configuration	64	441F0000	441FFFFF	541F0000	541FFFFF	31	
Reserved	64	44200000	4420FFFF	54200000	5420FFFF	32	
BLK_CTRL_NS_AONMIX - AONMIX	64	44210000	4421FFFF	54210000	5421FFFF	33	AONMIX

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Block Control Non-Secure							
MU1 - Messaging Unit - Side A (M33,A55 - to communicate with A55 core)	64	44220000	4422FFFF	54220000	5422FFFF	34	AONMIX
MU1 - Messaging Unit - Side B (M33,A55 - to communicate with M33 core)	64	44230000	4423FFFF	54230000	5423FFFF	35	AONMIX
SEMA42_ - Hardware Semaphore Module	64	44260000	4426FFFF	54260000	5426FFFF	38	
TRDC-MGR - AONMIX - Trusted Resource Domain Controller - Manager	64	44270000	4427FFFF	54270000	5427FFFF	39	AONMIX
TRDC-MC - AONMIX - Trusted Resource Domain Controller - Master Checker (MBC, MRC)	64	44280000	4428FFFF	54280000	5428FFFF	40	AONMIX
SYS_CTR_CTLBASE - System Counter - CTRL Interface	64	44290000	4429FFFF	54290000	5429FFFF	41	AONMIX
SYS_CTR_CMPBASE -	64	442A0000	442AFFFF	542A0000	542AFFFF	42	AONMIX

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
System Counter - CMP Interface							
SYS_CTR_RDBASE - System Counter - RD Interface	64	442B0000	442BFFFF	542B0000	542BFFFF	43	AONMIX
TSTMR1 - Timestamp Timer Module	64	442C0000	442CFFFF	542C0000	542CFFFF	44	AONMIX
LPWDOG1 - Low-Power Watch Dog Timer	64	442D0000	442DFFFF	542D0000	542DFFFF	45	AONMIX
LPWDOG 2- Low-Power Watch Dog Timer	64	442E0000	442EFFFF	542E0000	542EFFFF	46	AONMIX
LPIT1 - Low Power Interrupt Timer	64	442F0000	442FFFFF	542F0000	542FFFFF	47	AONMIX
LPTMR1 - Low-Power Timer	64	44300000	4430FFFF	54300000	5430FFFF	48	AONMIX
LPTPM1 - Low-Power Flexible Timer	64	44310000	4431FFFF	54310000	5431FFFF	49	AONMIX
LPTPM2- Low-Power Flexible Timer	64	44320000	4432FFFF	54320000	5432FFFF	50	AONMIX
I3C1 - Improved Inter Integrated Circuit	64	44330000	4433FFFF	54330000	5433FFFF	51	AONMIX
LPI2C1 - Low-Power	64	44340000	4434FFFF	54340000	5434FFFF	52	AONMIX

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Inter Integrated Circuit							
LPI2C2 - Low-Power Inter Integrated Circuit	64	44350000	4435FFFF	54350000	5435FFFF	53	AONMIX
LPSPI1 - Low-Power Serial Peripheral Interface	64	44360000	4436FFFF	54360000	5436FFFF	54	AONMIX
LPSPI2 - Low-Power Serial Peripheral Interface	64	44370000	4437FFFF	54370000	5437FFFF	55	AONMIX
LPUART1 - Low-Power Universal Asynchronous Receive/ Transmit	64	44380000	4438FFFF	54380000	5438FFFF	56	AONMIX
LPUART2 - Low-Power Universal Asynchronous Receive/ Transmit	64	44390000	4439FFFF	54390000	5439FFFF	57	AONMIX
CAN-FD1 - Controller Area Network	64	443A0000	443AFFFF	543A0000	543AFFFF	58	AONMIX
HMI - SAI1 - Serial Audio Interface	64	443B0000	443BFFFF	543B0000	543BFFFF	59	AONMIX
IOMUXC - Input/Output Multiplexing Controller	64	443C0000	443CFFFF	543C0000	543CFFFF	60	AONMIX
IPC - Integrated	64	443D0000	443DFFFF	543D0000	543DFFFF	61	AONMIX

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Peripheral Control							
M33_PCF - Cortex-M33 PCF Bus	64	443E0000	443EFFFF	543E0000	543EFFFF	62	AONMIX
M33_PSF - Cortex-M33 PSF Bus	64	443F0000	443FFFFF	543F0000	543FFFFF	63	AONMIX
M33_CACHE_CTRL - Cortex-M33 Cache APB Bus	64	44400000	4440FFFF	54400000	5440FFFF	64	AONMIX
BLK_CTRL_BBSMMIX - BBSMMIX Block Control	64	44410000	4441FFFF	54410000	5441FFFF	65	
M33 TCM ECC Control Peripheral Bus Interface	64	44420000	4442FFFF	54420000	5442FFFF	66	
M33 ROM Controller with Patch	64	44430000	4443FFFF	54430000	5443FFFF	67	
BBNSM - Battery Back Non-Secure Module	64	44440000	4444FFFF	54440000	5444FFFF	68	
CLK - CCM_CTRL - Clock Control Module Control	64	44450000	4445FFFF	54450000	5445FFFF	69	
SRC - System Reset Controller	64	44460000	4446FFFF	54460000	5446FFFF	70	
GPC - Global Power Controller	64	44470000	4447FFFF	54470000	5447FFFF	71	
Analog Top	64	44480000	4448FFFF	54480000	5448FFFF	72	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
CCMMIX Test Control Unit	64	444D0000	444DFFFF	544D0000	544DFFFF	77	
BLK_CTRL_S_AONMIX - AONMIX Block Control Secure	64	444F0000	444FFFFF	544F0000	544FFFFF	79	AONMIX
BASIC_TROUT - CM33_CPUWAIT - Basic Trusted Outside (TROUT)	64	44500000	4450FFFF	54500000	5450FFFF	80	AONMIX
Reserved	64	44510000	4451FFFF	54510000	5451FFFF	81	AONMIX
HMI - MICFIL - PDM Microphone Interface	64	44520000	4452FFFF	54520000	5452FFFF	82	AONMIX
ANA - ADC - Analog-to-Digital Converter	64	44530000	4453FFFF	54530000	5453FFFF	83	
Reserved	64	44540000	4454FFFF	54540000	5454FFFF	84	
Reserved	64	44550000	4455FFFF	54550000	5455FFFF	85	
Reserved	64	44560000	4456FFFF	54560000	5456FFFF	86	
Reserved	64	44570000	4457FFFF	54570000	5457FFFF	87	
Reserved	64	44580000	4458FFFF	54580000	5458FFFF	88	
Reserved	64	44590000	4459FFFF	54590000	5459FFFF	89	
Reserved	64	445A0000	445AFFFF	545A0000	545AFFFF	90	
Reserved	64	445B0000	445BFFFF	545B0000	545BFFFF	91	
Reserved	64	445C0000	445CFFFF	545C0000	545CFFFF	92	
Reserved	64	445D0000	445DFFFF	545D0000	545DFFFF	93	
Reserved	64	445E0000	445EFFFF	545E0000	545EFFFF	94	
Reserved	64	445F0000	445FFFFF	545F0000	545FFFFF	95	
Reserved	64	44600000	4460FFFF	54600000	5460FFFF	96	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Reserved	64	44610000	4461FFFF	54610000	5461FFFF	97	
Reserved	64	44620000	4462FFFF	54620000	5462FFFF	98	
Reserved	64	44630000	4463FFFF	54630000	5463FFFF	99	
Reserved	64	44640000	4464FFFF	54640000	5464FFFF	100	
Reserved	64	44650000	4465FFFF	54650000	5465FFFF	101	
Reserved	64	44660000	4466FFFF	54660000	5466FFFF	102	
Reserved	64	44670000	4467FFFF	54670000	5467FFFF	103	
Reserved	64	44680000	4468FFFF	54680000	5468FFFF	104	
Reserved	64	44690000	4469FFFF	54690000	5469FFFF	105	
Reserved	64	446A0000	446AFFFF	546A0000	546AFFFF	106	
Reserved	64	446B0000	446BFFFF	546B0000	546BFFFF	107	
Reserved	64	446C0000	446CFFFF	546C0000	546CFFFF	108	
Reserved	64	446D0000	446DFFFF	546D0000	546DFFFF	109	
Reserved	64	446E0000	446EFFFF	546E0000	546EFFFF	110	
Reserved	64	446F0000	446FFFFF	546F0000	546FFFFF	111	
Reserved	64	44700000	4470FFFF	54700000	5470FFFF	112	
Reserved	64	44710000	4471FFFF	54710000	5471FFFF	113	
Reserved	64	44720000	4472FFFF	54720000	5472FFFF	114	
Reserved	64	44730000	4473FFFF	54730000	5473FFFF	115	
Reserved	64	44740000	4474FFFF	54740000	5474FFFF	116	
Reserved	64	44750000	4475FFFF	54750000	5475FFFF	117	
Reserved	64	44760000	4476FFFF	54760000	5476FFFF	118	
Reserved	64	44770000	4477FFFF	54770000	5477FFFF	119	
Reserved	64	44780000	4478FFFF	54780000	5478FFFF	120	
Reserved	64	44790000	4479FFFF	54790000	5479FFFF	121	
Reserved	64	447A0000	447AFFFF	547A0000	547AFFFF	122	
Reserved	64	447B0000	447BFFFF	547B0000	547BFFFF	123	
Reserved	64	447C0000	447CFFFF	547C0000	547CFFFF	124	
Reserved	64	447D0000	447DFFFF	547D0000	547DFFFF	125	
Reserved	64	447E0000	447EFFFF	547E0000	547EFFFF	126	
Reserved	64	447F0000	447FFFFF	547F0000	547FFFFF	127	

2.5 AIPS Peripheral Bridge 2 Memory Map

The following table provides the AIPS-2 memory map. The peripherals have both Secure (S) and Non-Secure (NS) access possibilities. Both, Secure and Non-Secure addresses for each peripheral is provided in the table below.

The start address of AIPS-2 peripherals for the Non-Secure access is 0x4200_0000.

The start address of AIPS-2 peripherals for the Secure access is 0x5200_0000.

NOTE

AIPS2 Peripherals are in WAKEUPMIX domain.

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Enhanced Direct Memory Access 4 - Management Page	64	42000000	4200FFFF	52000000	5200FFFF	0	
Enhanced Direct Memory Access 4 - Channels 0 & 1 Control and Configuration	64	42010000	4201FFFF	52010000	5201FFFF	1	
Enhanced Direct Memory Access 4 - Channels 2 & 3 Control and Configuration	64	42020000	4202FFFF	52020000	5202FFFF	2	
Enhanced Direct Memory Access 4 - Channels 4 & 5 Control and Configuration	64	42030000	4203FFFF	52030000	5203FFFF	3	
Enhanced Direct Memory Access 4 - Channels 6 & 7 Control and Configuration	64	42040000	4204FFFF	52040000	5204FFFF	4	
Enhanced Direct	64	42050000	4205FFFF	52050000	5205FFFF	5	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Memory Access 4 - Channels 8 & 9 Control and Configuration							
Enhanced Direct Memory Access 4 - Channels 10 & 11 Control and Configuration	64	42060000	4206FFFF	52060000	5206FFFF	6	
Enhanced Direct Memory Access 4 - Channels 12 & 13 Control and Configuration	64	42070000	4207FFFF	52070000	5207FFFF	7	
Enhanced Direct Memory Access 4 - Channels 14 & 15 Control and Configuration	64	42080000	4208FFFF	52080000	5208FFFF	8	
Enhanced Direct Memory Access 4 - Channels 16 & 17 Control and Configuration	64	42090000	4209FFFF	52090000	5209FFFF	9	
Enhanced Direct Memory Access 4 - Channels 18 & 19 Control and Configuration	64	420A0000	420AFFFF	520A0000	520AFFFF	10	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Enhanced Direct Memory Access 4 - Channels 20 & 21 Control and Configuration	64	420B0000	420BFFFF	520B0000	520BFFFF	11	
Enhanced Direct Memory Access 4 - Channels 22 & 23 Control and Configuration	64	420C0000	420CFFFF	520C0000	520CFFFF	12	
Enhanced Direct Memory Access 4 - Channels 24 & 25 Control and Configuration	64	420D0000	420DFFFF	520D0000	520DFFFF	13	
Enhanced Direct Memory Access 4 - Channels 26 & 27 Control and Configuration	64	420E0000	420EFFFF	520E0000	520EFFFF	14	
Enhanced Direct Memory Access 4 - Channels 28 & 29 Control and Configuration	64	420F0000	420FFFFF	520F0000	520FFFFF	15	
Enhanced Direct Memory Access 4 - Channels 30	64	42100000	4210FFFF	52100000	5210FFFF	16	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
& 31 Control and Configuration							
Enhanced Direct Memory Access 4 - Channels 32 & 33 Control and Configuration	64	42110000	4211FFFF	52110000	5211FFFF	17	
Enhanced Direct Memory Access 4 - Channels 34 & 35 Control and Configuration	64	42120000	4212FFFF	52120000	5212FFFF	18	
Enhanced Direct Memory Access 4 - Channels 36 & 37 Control and Configuration	64	42130000	4213FFFF	52130000	5213FFFF	19	
Enhanced Direct Memory Access 4 - Channels 38 & 39 Control and Configuration	64	42140000	4214FFFF	52140000	5214FFFF	20	
Enhanced Direct Memory Access 4 - Channels 40 & 41 Control and Configuration	64	42150000	4215FFFF	52150000	5215FFFF	21	
Enhanced Direct	64	42160000	4216FFFF	52160000	5216FFFF	22	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Memory Access 4 - Channels 42 & 43 Control and Configuration							
Enhanced Direct Memory Access 4 - Channels 44 & 45 Control and Configuration	64	42170000	4217FFFF	52170000	5217FFFF	23	
Enhanced Direct Memory Access 4 - Channels 46 & 47 Control and Configuration	64	42180000	4218FFFF	52180000	5218FFFF	24	
Enhanced Direct Memory Access 4 - Channels 48 & 49 Control and Configuration	64	42190000	4219FFFF	52190000	5219FFFF	25	
Enhanced Direct Memory Access 4 - Channels 50 & 51 Control and Configuration	64	421A0000	421AFFFF	521A0000	521AFFFF	26	
Enhanced Direct Memory Access 4 - Channels 52 & 53 Control	64	421B0000	421BFFFF	521B0000	521BFFFF	27	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
and Configuration							
Enhanced Direct Memory Access 4 - Channels 54 & 55 Control and Configuration	64	421C0000	421CFFFF	521C0000	521CFFFF	28	
Enhanced Direct Memory Access 4 - Channels 56 & 57 Control and Configuration	64	421D0000	421DFFFF	521D0000	521DFFFF	29	
Enhanced Direct Memory Access 4 - Channels 58 & 59 Control and Configuration	64	421E0000	421EFFFF	521E0000	521EFFFF	30	
Enhanced Direct Memory Access 4 - Channels 60 & 61 Control and Configuration	64	421F0000	421FFFFF	521F0000	521FFFFF	31	
Enhanced Direct Memory Access 4 - Channels 62 & 63 Control and Configuration	64	42200000	4220FFFF	52200000	5220FFFF	32	
Reserved	64	42210000	4221FFFF	52210000	5221FFFF	33	
Reserved	64	42220000	4222FFFF	52220000	5222FFFF	34	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Reserved	64	42230000	4223FFFF	52230000	5223FFFF	35	
Reserved	64	42240000	4224FFFF	52240000	5224FFFF	36	
Reserved	64	42250000	4225FFFF	52250000	5225FFFF	37	
Reserved	64	42260000	4226FFFF	52260000	5226FFFF	38	
Reserved	64	42270000	4227FFFF	52270000	5227FFFF	39	
Reserved	64	42280000	4228FFFF	52280000	5228FFFF	40	
Reserved	64	42290000	4229FFFF	52290000	5229FFFF	41	
Reserved	64	422A0000	422AFFFF	522A0000	522AFFFF	42	
Reserved	64	422B0000	422BFFFF	522B0000	522BFFFF	43	
Reserved	64	422C0000	422CFFFF	522C0000	522CFFFF	44	
Reserved	64	422D0000	422DFFFF	522D0000	522DFFFF	45	
Reserved	64	422E0000	422EFFFF	522E0000	522EFFFF	46	
Reserved	64	422F0000	422FFFFF	522F0000	522FFFFF	47	
Reserved	64	42300000	4230FFFF	52300000	5230FFFF	48	
Reserved	64	42310000	4231FFFF	52310000	5231FFFF	49	
Reserved	64	42320000	4232FFFF	52320000	5232FFFF	50	
Reserved	64	42330000	4233FFFF	52330000	5233FFFF	51	
Reserved	64	42340000	4234FFFF	52340000	5234FFFF	52	
Reserved	64	42350000	4235FFFF	52350000	5235FFFF	53	
Reserved	64	42360000	4236FFFF	52360000	5236FFFF	54	
Reserved	64	42370000	4237FFFF	52370000	5237FFFF	55	
Reserved	64	42380000	4238FFFF	52380000	5238FFFF	56	
Reserved	64	42390000	4239FFFF	52390000	5239FFFF	57	
Reserved	64	423A0000	423AFFFF	523A0000	523AFFFF	58	
Reserved	64	423B0000	423BFFFF	523B0000	523BFFFF	59	
Reserved	64	423C0000	423CFFFF	523C0000	523CFFFF	60	
Reserved	64	423D0000	423DFFFF	523D0000	523DFFFF	61	
Reserved	64	423E0000	423EFFFF	523E0000	523EFFFF	62	
Reserved	64	423F0000	423FFFFF	523F0000	523FFFFF	63	
Reserved	64	42400000	4240FFFF	52400000	5240FFFF	64	
Reserved	64	42410000	4241FFFF	52410000	5241FFFF	65	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
BLK_CTRL_WAKEUPMIX - WAKEUPMIX Block Control	64	42420000	4242FFFF	52420000	5242FFFF	66	WAKEUPMIX
MU2 - Messaging Unit - Side A (M33,A55 - to communicate with A55 core)	64	42430000	4243FFFF	52430000	5243FFFF	67	WAKEUPMIX
MU2 - Messaging Unit - Side B (M33,A55 - to communicate with M33 core)	64	42440000	4244FFFF	52440000	5244FFFF	68	WAKEUPMIX
SEMA42_ - Hardware Semaphore Module	64	42450000	4245FFFF	52450000	5245FFFF	69	WAKEUPMIX
TRDC-MGR - WAKEUPMIX - Trusted Resource Domain Controller - Manager	64	42460000	4246FFFF	52460000	5246FFFF	70	WAKEUPMIX
TRDC-MC - WAKEUPMIX - Trusted Resource Domain Controller - Master Checker (MBC, MRC)	64	42470000	4247FFFF	52470000	5247FFFF	71	WAKEUPMIX
TSTMR2 - Timestamp Timer Module	64	42480000	4248FFFF	52480000	5248FFFF	72	WAKEUPMIX
LPWDOG3 - Low-Power	64	42490000	4249FFFF	52490000	5249FFFF	73	WAKEUPMIX

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Watch Dog Timer							
LPWDOG4 - Low-Power Watch Dog Timer	64	424A0000	424AFFFF	524A0000	524AFFFF	74	WAKEUPMIX
LPWDOG5 - Low-Power Watch Dog Timer	64	424B0000	424BFFFF	524B0000	524BFFFF	75	WAKEUPMIX
LPIT2 - Low Power Interrupt Timer	64	424C0000	424CFFFF	524C0000	524CFFFF	76	WAKEUPMIX
LPTMR2 - Low-Power Timer	64	424D0000	424DFFFF	524D0000	524DFFFF	77	WAKEUPMIX
LPTPM3 - Low-Power Flexible Timer	64	424E0000	424EFFFF	524E0000	524EFFFF	78	WAKEUPMIX
LPTPM4 - Low-Power Flexible Timer	64	424F0000	424FFFFF	524F0000	524FFFFF	79	WAKEUPMIX
LPTPM5 - Low-Power Flexible Timer	64	42500000	4250FFFF	52500000	5250FFFF	80	WAKEUPMIX
LPTPM6 - Low-Power Flexible Timer	64	42510000	4251FFFF	52510000	5251FFFF	81	WAKEUPMIX
I3C2 - Improved Inter Integrated Circuit	64	42520000	4252FFFF	52520000	5252FFFF	82	WAKEUPMIX
LPI2C3 - Low-Power Inter	64	42530000	4253FFFF	52530000	5253FFFF	83	WAKEUPMIX

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Integrated Circuit							
LPI2C4 - Low-Power Inter Integrated Circuit	64	42540000	4254FFFF	52540000	5254FFFF	84	WAKEUPMIX
LPSPI 3- Low-Power Serial Peripheral Interface	64	42550000	4255FFFF	52550000	5255FFFF	85	WAKEUPMIX
LPSPI 4- Low-Power Serial Peripheral Interface	64	42560000	4256FFFF	52560000	5256FFFF	86	WAKEUPMIX
LPUART3 - Low-Power Universal Asynchronous Receive/ Transmit	64	42570000	4257FFFF	52570000	5257FFFF	87	WAKEUPMIX
LPUART4 - Low-Power Universal Asynchronous Receive/ Transmit	64	42580000	4258FFFF	52580000	5258FFFF	88	WAKEUPMIX
LPUART5 - Low-Power Universal Asynchronous Receive/ Transmit	64	42590000	4259FFFF	52590000	5259FFFF	89	WAKEUPMIX
LPUART6 - Low-Power Universal Asynchronous Receive/ Transmit	64	425A0000	425AFFFF	525A0000	525AFFFF	90	WAKEUPMIX

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
CAN-FD2-Controller Area Network	64	425B0000	425BFFFF	525B0000	525BFFFF	91	WAKEUPMIX
Flexible Input/Output FLEXIO1	64	425C0000	425CFFFF	525C0000	525CFFFF	92	WAKEUPMIX
Flexible Input/Output FLEXIO2	64	425D0000	425DFFFF	525D0000	525DFFFF	93	WAKEUPMIX
FlexSPI_OTFAD - Flexible Serial Peripheral Interface with OTFAD	64	425E0000	425EFFFF	525E0000	525EFFFF	94	WAKEUPMIX
Reserved	64	425F0000	425FFFFF	525F0000	525FFFFF	95	
Reserved	64	42600000	4260FFFF	52600000	5260FFFF	96	
BASIC_TROUT - WAKEUPMIX - Basic Trusted Outside (TROUT)	64	42630000	4263FFFF	52630000	5263FFFF	99	WAKEUPMIX
MEM - ROMCP - A55 ROM Controller with Patch	64	42640000	4264FFFF	52640000	5264FFFF	100	WAKEUPMIX
HMI - SAI 2-Serial Audio Interface	64	42650000	4265FFFF	52650000	5265FFFF	101	WAKEUPMIX
HMI - SAI 3-Serial Audio Interface	64	42660000	4266FFFF	52660000	5266FFFF	102	WAKEUPMIX
Reserved	64	42670000	4267FFFF	52670000	5267FFFF	103	
HMI - SPDIF - Sony/Philips Digital Interface	64	42680000	4268FFFF	52680000	5268FFFF	104	WAKEUPMIX

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
LPUART7 - Low-Power Universal Asynchronous Receive/ Transmit	64	42690000	4269FFFF	52690000	5269FFFF	105	WAKEUPMIX
LPUART8 - Low-Power Universal Asynchronous Receive/ Transmit	64	426A0000	426AFFFF	526A0000	526AFFFF	106	WAKEUPMIX
LPI2C5 - Low-Power Inter Integrated Circuit	64	426B0000	426BFFFF	526B0000	526BFFFF	107	WAKEUPMIX
LPI2C6 - Low-Power Inter Integrated Circuit	64	426C0000	426CFFFF	526C0000	526CFFFF	108	WAKEUPMIX
LPI2C7 - Low-Power Inter Integrated Circuit	64	426D0000	426DFFFF	526D0000	526DFFFF	109	WAKEUPMIX
LPI2C8 - Low-Power Inter Integrated Circuit	64	426E0000	426EFFFF	526E0000	526EFFFF	110	WAKEUPMIX
LPSPI 5- Low-Power Serial Peripheral Interface	64	426F0000	426FFFFF	526F0000	526FFFFF	111	WAKEUPMIX
LPSPI 6- Low-Power Serial Peripheral Interface	64	42700000	4270FFFF	52700000	5270FFFF	112	WAKEUPMIX

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
LPSPI 7-Low-Power Serial Peripheral Interface	64	42710000	4271FFFF	52710000	5271FFFF	113	WAKEUPMIX
LPSPI8 - Low-Power Serial Peripheral Interface	64	42720000	4272FFFF	52720000	5272FFFF	114	WAKEUPMIX
Reserved	64	42730000	4273FFFF	52730000	5273FFFF	115	
Reserved	64	42740000	4274FFFF	52740000	5274FFFF	116	
Reserved	64	42750000	4275FFFF	52750000	5275FFFF	117	
Reserved	64	42760000	4276FFFF	52760000	5276FFFF	118	
Reserved	64	42770000	4277FFFF	52770000	5277FFFF	119	
Reserved	64	42780000	4278FFFF	52780000	5278FFFF	120	
Reserved	64	42790000	4279FFFF	52790000	5279FFFF	121	
Reserved	64	427A0000	427AFFFF	527A0000	527AFFFF	122	
Reserved	64	427B0000	427BFFFF	527B0000	527BFFFF	123	
Reserved	64	427C0000	427CFFFF	527C0000	527CFFFF	124	
Reserved	64	427D0000	427DFFFF	527D0000	527DFFFF	125	
Reserved	64	427E0000	427EFFFF	527E0000	527EFFFF	126	
Reserved	64	427F0000	427FFFFF	527F0000	527FFFFF	127	

2.6 AIPS Peripheral Bridge 3 Memory Map

The following table provides the AIPS-3 memory map. The peripherals have both Secure (S) and Non-Secure (NS) access possibilities. Both, Secure and Non-Secure addresses for each peripheral is provided in the table below.

The start address of AIPS-3 peripherals for the Non-Secure access is 0x4280_0000.

The start address of AIPS-3 peripherals for the Secure access is 0x5280_0000.

NOTE

AIPS3 Peripherals are in the MEGAMIX domain.

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Reserved	64	42800000	4280FFFF	52800000	5280FFFF	0	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
TRDC-MGR - MEGAMIX - Trusted Resource Domain Controller - Manager	64	42810000	4281FFFF	52810000	5281FFFF	1	MEGAMIX
Reserved	64	42820000	4282FFFF	52820000	5282FFFF	2	MEGAMIX
MEGAMIX NIC GPV Control	64	42830000	4283FFFF	52830000	5283FFFF	3	MEGAMIX
MEGAMIX NIC GPV Control	64	42840000	4284FFFF	52840000	5284FFFF	4	MEGAMIX
uSDHC1 - Ultra Secure Digital Host Controller	64	42850000	4285FFFF	52850000	5285FFFF	5	MEGAMIX
uSDHC2 - Ultra Secure Digital Host Controller	64	42860000	4286FFFF	52860000	5286FFFF	6	MEGAMIX
BASIC_TROUT - MEGAMIX - Basic Trusted Outside (TROUT)	64	42880000	4288FFFF	52880000	5288FFFF	8	MEGAMIX
ENET - Ethernet Controller	64	42890000	4289FFFF	52890000	5289FFFF	9	MEGAMIX
ENET_QoS - TSN-capable Ethernet Controller	64	428A0000	428AFFFF	528A0000	528AFFFF	10	MEGAMIX
uSDHC3 - Ultra Secure Digital Host Controller	64	428B0000	428BFFFF	528B0000	528BFFFF	11	MEGAMIX
Reserved	64	428C0000	428CFFFF	528C0000	528CFFFF	12	
Reserved	64	428D0000	428DFFFF	528D0000	528DFFFF	13	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Reserved	64	428E0000	428EFFFF	528E0000	528EFFFF	14	
Reserved	64	428F0000	428FFFFF	528F0000	528FFFFF	15	
Reserved	64	42900000	4290FFFF	52900000	5290FFFF	16	
Reserved	64	42910000	4291FFFF	52910000	5291FFFF	17	
Reserved	64	42920000	4292FFFF	52920000	5292FFFF	18	
Reserved	64	42930000	4293FFFF	52930000	5293FFFF	19	
Reserved	64	42940000	4294FFFF	52940000	5294FFFF	20	
Reserved	64	42950000	4295FFFF	52950000	5295FFFF	21	
Reserved	64	42960000	4296FFFF	52960000	5296FFFF	22	
Reserved	64	42970000	4297FFFF	52970000	5297FFFF	23	
Reserved	64	42980000	4298FFFF	52980000	5298FFFF	24	
Reserved	64	42990000	4299FFFF	52990000	5299FFFF	25	
Reserved	64	429A0000	429AFFFF	529A0000	529AFFFF	26	
Reserved	64	429B0000	429BFFFF	529B0000	529BFFFF	27	
Reserved	64	429C0000	429CFFFF	529C0000	529CFFFF	28	
Reserved	64	429D0000	429DFFFF	529D0000	529DFFFF	29	
Reserved	64	429E0000	429EFFFF	529E0000	529EFFFF	30	
Reserved	64	429F0000	429FFFFF	529F0000	529FFFFF	31	
Reserved	64	42A00000	42A0FFFF	52A00000	52A0FFFF	32	
Reserved	64	42A10000	42A1FFFF	52A10000	52A1FFFF	33	
Reserved	64	42A20000	42A2FFFF	52A20000	52A2FFFF	34	
Reserved	64	42A30000	42A3FFFF	52A30000	52A3FFFF	35	
Reserved	64	42A40000	42A4FFFF	52A40000	52A4FFFF	36	
Reserved	64	42A50000	42A5FFFF	52A50000	52A5FFFF	37	
Reserved	64	42A60000	42A6FFFF	52A60000	52A6FFFF	38	
Reserved	64	42A70000	42A7FFFF	52A70000	52A7FFFF	39	
Reserved	64	42A80000	42A8FFFF	52A80000	52A8FFFF	40	
Reserved	64	42A90000	42A9FFFF	52A90000	52A9FFFF	41	
Reserved	64	42AA0000	42AAFFFF	52AA0000	52AAFFFF	42	
Reserved	64	42AB0000	42ABFFFF	52AB0000	52ABFFFF	43	
Reserved	64	42AC0000	42ACFFFF	52AC0000	52ACFFFF	44	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Reserved	64	42AD0000	42ADFFFF	52AD0000	52ADFFFF	45	
Reserved	64	42AE0000	42AEFFFF	52AE0000	52AEFFFF	46	
Reserved	64	42AF0000	42AFFFFF	52AF0000	52AFFFFF	47	
Reserved	64	42B00000	42B0FFFF	52B00000	52B0FFFF	48	
Reserved	64	42B10000	42B1FFFF	52B10000	52B1FFFF	49	
Reserved	64	42B20000	42B2FFFF	52B20000	52B2FFFF	50	
Reserved	64	42B30000	42B3FFFF	52B30000	52B3FFFF	51	
Reserved	64	42B40000	42B4FFFF	52B40000	52B4FFFF	52	
Reserved	64	42B50000	42B5FFFF	52B50000	52B5FFFF	53	
Reserved	64	42B60000	42B6FFFF	52B60000	52B6FFFF	54	
Reserved	64	42B70000	42B7FFFF	52B70000	52B7FFFF	55	
Reserved	64	42B80000	42B8FFFF	52B80000	52B8FFFF	56	
Reserved	64	42B90000	42B9FFFF	52B90000	52B9FFFF	57	
Reserved	64	42BA0000	42BAFFFF	52BA0000	52BAFFFF	58	
Reserved	64	42BB0000	42BBFFFF	52BB0000	52BBFFFF	59	
Reserved	64	42BC0000	42BCFFFF	52BC0000	52BCFFFF	60	
Reserved	64	42BD0000	42BDFFFF	52BD0000	52BDFFFF	61	
Reserved	64	42BE0000	42BEFFFF	52BE0000	52BEFFFF	62	
Reserved	64	42BF0000	42BFFFFF	52BF0000	52BFFFFF	63	
Reserved	64	42C00000	42C0FFFF	52C00000	52C0FFFF	64	
Reserved	64	42C10000	42C1FFFF	52C10000	52C1FFFF	65	
Reserved	64	42C20000	42C2FFFF	52C20000	52C2FFFF	66	
Reserved	64	42C30000	42C3FFFF	52C30000	52C3FFFF	67	
Reserved	64	42C40000	42C4FFFF	52C40000	52C4FFFF	68	
Reserved	64	42C50000	42C5FFFF	52C50000	52C5FFFF	69	
Reserved	64	42C60000	42C6FFFF	52C60000	52C6FFFF	70	
Reserved	64	42C70000	42C7FFFF	52C70000	52C7FFFF	71	
Reserved	64	42C80000	42C8FFFF	52C80000	52C8FFFF	72	
Reserved	64	42C90000	42C9FFFF	52C90000	52C9FFFF	73	
Reserved	64	42CA0000	42CAFFFF	52CA0000	52CAFFFF	74	
Reserved	64	42CB0000	42CBFFFF	52CB0000	52CBFFFF	75	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Reserved	64	42CC0000	42CCFFFF	52CC0000	52CCFFFF	76	
Reserved	64	42CD0000	42CDFFFF	52CD0000	52CDFFFF	77	
Reserved	64	42CE0000	42CEFFFF	52CE0000	52CEFFFF	78	
Reserved	64	42CF0000	42CFFFFF	52CF0000	52CFFFFF	79	
Reserved	64	42D00000	42D0FFFF	52D00000	52D0FFFF	80	
Reserved	64	42D10000	42D1FFFF	52D10000	52D1FFFF	81	
Reserved	64	42D20000	42D2FFFF	52D20000	52D2FFFF	82	
Reserved	64	42D30000	42D3FFFF	52D30000	52D3FFFF	83	
Reserved	64	42D40000	42D4FFFF	52D40000	52D4FFFF	84	
Reserved	64	42D50000	42D5FFFF	52D50000	52D5FFFF	85	
Reserved	64	42D60000	42D6FFFF	52D60000	52D6FFFF	86	
Reserved	64	42D70000	42D7FFFF	52D70000	52D7FFFF	87	
Reserved	64	42D80000	42D8FFFF	52D80000	52D8FFFF	88	
Reserved	64	42D90000	42D9FFFF	52D90000	52D9FFFF	89	
Reserved	64	42DA0000	42DAFFFF	52DA0000	52DAFFFF	90	
Reserved	64	42DB0000	42DBFFFF	52DB0000	52DBFFFF	91	
Reserved	64	42DC0000	42DCFFFF	52DC0000	52DCFFFF	92	
Reserved	64	42DD0000	42DDFFFF	52DD0000	52DDFFFF	93	
Reserved	64	42DE0000	42DEFFFF	52DE0000	52DEFFFF	94	
Reserved	64	42DF0000	42DFFFFF	52DF0000	52DFFFFF	95	
Reserved	64	42E00000	42E0FFFF	52E00000	52E0FFFF	96	
Reserved	64	42E10000	42E1FFFF	52E10000	52E1FFFF	97	
Reserved	64	42E20000	42E2FFFF	52E20000	52E2FFFF	98	
Reserved	64	42E30000	42E3FFFF	52E30000	52E3FFFF	99	
Reserved	64	42E40000	42E4FFFF	52E40000	52E4FFFF	100	
Reserved	64	42E50000	42E5FFFF	52E50000	52E5FFFF	101	
Reserved	64	42E60000	42E6FFFF	52E60000	52E6FFFF	102	
Reserved	64	42E70000	42E7FFFF	52E70000	52E7FFFF	103	
Reserved	64	42E80000	42E8FFFF	52E80000	52E8FFFF	104	
Reserved	64	42E90000	42E9FFFF	52E90000	52E9FFFF	105	
Reserved	64	42EA0000	42EAFFFF	52EA0000	52EAFFFF	106	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Reserved	64	42EB0000	42EBFFFF	52EB0000	52EBFFFF	107	
Reserved	64	42EC0000	42ECFFFF	52EC0000	52ECFFFF	108	
Reserved	64	42ED0000	42EDFFFF	52ED0000	52EDFFFF	109	
Reserved	64	42EE0000	42EEFFFF	52EE0000	52EEFFFF	110	
Reserved	64	42EF0000	42EFFFFF	52EF0000	52EFFFFF	111	
Reserved	64	42F00000	42F0FFFF	52F00000	52F0FFFF	112	
Reserved	64	42F10000	42F1FFFF	52F10000	52F1FFFF	113	
Reserved	64	42F20000	42F2FFFF	52F20000	52F2FFFF	114	
Reserved	64	42F30000	42F3FFFF	52F30000	52F3FFFF	115	
Reserved	64	42F40000	42F4FFFF	52F40000	52F4FFFF	116	
Reserved	64	42F50000	42F5FFFF	52F50000	52F5FFFF	117	
Reserved	64	42F60000	42F6FFFF	52F60000	52F6FFFF	118	
Reserved	64	42F70000	42F7FFFF	52F70000	52F7FFFF	119	
Reserved	64	42F80000	42F8FFFF	52F80000	52F8FFFF	120	
Reserved	64	42F90000	42F9FFFF	52F90000	52F9FFFF	121	
Reserved	64	42FA0000	42FAFFFF	52FA0000	52FAFFFF	122	
Reserved	64	42FB0000	42FBFFFF	52FB0000	52FBFFFF	123	
Reserved	64	42FC0000	42FCFFFF	52FC0000	52FCFFFF	124	
Reserved	64	42FD0000	42FDFFFF	52FD0000	52FDFFFF	125	
Reserved	64	42FE0000	42FEFFFF	52FE0000	52FEFFFF	126	
Reserved	64	42FF0000	42FFFFFF	52FF0000	52FFFFFF	127	

2.7 AIPS Peripheral Bridge 4 Memory Map

The following table provides the AIPS-4 memory map. The peripherals have both Secure (S) and Non-Secure (NS) access possibilities. Both, Secure and Non-Secure addresses for each peripheral is provided in the table below.

The start address of AIPS-4 peripherals for the Non-Secure access is 0x4900_0000.

The start address of AIPS-4 peripherals for the Secure access is 0x5900_0000.

NOTE

AIPS4 Peripherals are in the NICMIX domain.

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
BLK_CTRL_NIC_WRAPPER - NIC_WRAPPER Block Control	64	49000000	4900FFFF	59000000	5900FFFF	0	NICMIX
TRDC-MGR - NIC_WRAPPER - Trusted Resource Domain Controller - Manager	64	49010000	4901FFFF	59010000	5901FFFF	1	NICMIX
TRDC-MC - NIC_WRAPPER - Trusted Resource Domain Controller - Master Checker (MBC, MRC)	64	49020000	4902FFFF	59020000	5902FFFF	2	NICMIX
BASIC_TROUT - NIC_WRAPPER -Basic Trusted Outside (TROUT)	64	49050000	4905FFFF	59050000	5905FFFF	5	NICMIX
Reserved	64	49060000	4906FFFF	59060000	5906FFFF	6	NICMIX
Reserved	64	49070000	4907FFFF	59070000	5907FFFF	7	NICMIX
Reserved	64	49080000	4908FFFF	59080000	5908FFFF	8	
Reserved	64	49090000	4909FFFF	59090000	5909FFFF	9	
OCRAM_MECC - On Chip SRAM ECC Control Peripheral Bus Interface	64	490A0000	490AFFFF	590A0000	590AFFFF	10	NICMIX
Reserved	64	490B0000	490BFFFF	590B0000	590BFFFF	11	
Reserved	64	490C0000	490CFFFF	590C0000	590CFFFF	12	
Reserved	64	490D0000	490DFFFF	590D0000	590DFFFF	13	
Reserved	64	490E0000	490EFFFF	590E0000	590EFFFF	14	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Reserved	64	490F0000	490FFFFF	590F0000	590FFFFF	15	
Reserved	64	49100000	4910FFFF	59100000	5910FFFF	16	
Reserved	64	49110000	4911FFFF	59110000	5911FFFF	17	
Reserved	64	49120000	4912FFFF	59120000	5912FFFF	18	
Reserved	64	49130000	4913FFFF	59130000	5913FFFF	19	
Reserved	64	49140000	4914FFFF	59140000	5914FFFF	20	
Reserved	64	49150000	4915FFFF	59150000	5915FFFF	21	
Reserved	64	49160000	4916FFFF	59160000	5916FFFF	22	
Reserved	64	49170000	4917FFFF	59170000	5917FFFF	23	
Reserved	64	49180000	4918FFFF	59180000	5918FFFF	24	
Reserved	64	49190000	4919FFFF	59190000	5919FFFF	25	
Reserved	64	491A0000	491AFFFF	591A0000	591AFFFF	26	
Reserved	64	491B0000	491BFFFF	591B0000	591BFFFF	27	
Reserved	64	491C0000	491CFFFF	591C0000	591CFFFF	28	
Reserved	64	491D0000	491DFFFF	591D0000	591DFFFF	29	
Reserved	64	491E0000	491EFFFF	591E0000	591EFFFF	30	
Reserved	64	491F0000	491FFFFF	591F0000	591FFFFF	31	
Reserved	64	49200000	4920FFFF	59200000	5920FFFF	32	
Reserved	64	49210000	4921FFFF	59210000	5921FFFF	33	
Reserved	64	49220000	4922FFFF	59220000	5922FFFF	34	
Reserved	64	49230000	4923FFFF	59230000	5923FFFF	35	
Reserved	64	49240000	4924FFFF	59240000	5924FFFF	36	
Reserved	64	49250000	4925FFFF	59250000	5925FFFF	37	
Reserved	64	49260000	4926FFFF	59260000	5926FFFF	38	
Reserved	64	49270000	4927FFFF	59270000	5927FFFF	39	
Reserved	64	49280000	4928FFFF	59280000	5928FFFF	40	
Reserved	64	49290000	4929FFFF	59290000	5929FFFF	41	
Reserved	64	492A0000	492AFFFF	592A0000	592AFFFF	42	
Reserved	64	492B0000	492BFFFF	592B0000	592BFFFF	43	
Reserved	64	492C0000	492CFFFF	592C0000	592CFFFF	44	
Reserved	64	492D0000	492DFFFF	592D0000	592DFFFF	45	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Reserved	64	492E0000	492EFFFF	592E0000	592EFFFF	46	
Reserved	64	492F0000	492FFFFF	592F0000	592FFFFF	47	
Reserved	64	49300000	4930FFFF	59300000	5930FFFF	48	
Reserved	64	49310000	4931FFFF	59310000	5931FFFF	49	
Reserved	64	49320000	4932FFFF	59320000	5932FFFF	50	
Reserved	64	49330000	4933FFFF	59330000	5933FFFF	51	
Reserved	64	49340000	4934FFFF	59340000	5934FFFF	52	
Reserved	64	49350000	4935FFFF	59350000	5935FFFF	53	
Reserved	64	49360000	4936FFFF	59360000	5936FFFF	54	
Reserved	64	49370000	4937FFFF	59370000	5937FFFF	55	
Reserved	64	49380000	4938FFFF	59380000	5938FFFF	56	
Reserved	64	49390000	4939FFFF	59390000	5939FFFF	57	
Reserved	64	493A0000	493AFFFF	593A0000	593AFFFF	58	
Reserved	64	493B0000	493BFFFF	593B0000	593BFFFF	59	
Reserved	64	493C0000	493CFFFF	593C0000	593CFFFF	60	
Reserved	64	493D0000	493DFFFF	593D0000	593DFFFF	61	
Reserved	64	493E0000	493EFFFF	593E0000	593EFFFF	62	
Reserved	64	493F0000	493FFFFF	593F0000	593FFFFF	63	
Reserved	64	49400000	4940FFFF	59400000	5940FFFF	64	
Reserved	64	49410000	4941FFFF	59410000	5941FFFF	65	
Reserved	64	49420000	4942FFFF	59420000	5942FFFF	66	
Reserved	64	49430000	4943FFFF	59430000	5943FFFF	67	
Reserved	64	49440000	4944FFFF	59440000	5944FFFF	68	
Reserved	64	49450000	4945FFFF	59450000	5945FFFF	69	
Reserved	64	49460000	4946FFFF	59460000	5946FFFF	70	
Reserved	64	49470000	4947FFFF	59470000	5947FFFF	71	
Reserved	64	49480000	4948FFFF	59480000	5948FFFF	72	
Reserved	64	49490000	4949FFFF	59490000	5949FFFF	73	
Reserved	64	494A0000	494AFFFF	594A0000	594AFFFF	74	
Reserved	64	494B0000	494BFFFF	594B0000	594BFFFF	75	
Reserved	64	494C0000	494CFFFF	594C0000	594CFFFF	76	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Reserved	64	494D0000	494DFFFF	594D0000	594DFFFF	77	
Reserved	64	494E0000	494EFFFF	594E0000	594EFFFF	78	
Reserved	64	494F0000	494FFFFF	594F0000	594FFFFF	79	
Reserved	64	49500000	4950FFFF	59500000	5950FFFF	80	
Reserved	64	49510000	4951FFFF	59510000	5951FFFF	81	
Reserved	64	49520000	4952FFFF	59520000	5952FFFF	82	
Reserved	64	49530000	4953FFFF	59530000	5953FFFF	83	
Reserved	64	49540000	4954FFFF	59540000	5954FFFF	84	
Reserved	64	49550000	4955FFFF	59550000	5955FFFF	85	
Reserved	64	49560000	4956FFFF	59560000	5956FFFF	86	
Reserved	64	49570000	4957FFFF	59570000	5957FFFF	87	
Reserved	64	49580000	4958FFFF	59580000	5958FFFF	88	
Reserved	64	49590000	4959FFFF	59590000	5959FFFF	89	
Reserved	64	495A0000	495AFFFF	595A0000	595AFFFF	90	
Reserved	64	495B0000	495BFFFF	595B0000	595BFFFF	91	
Reserved	64	495C0000	495CFFFF	595C0000	595CFFFF	92	
Reserved	64	495D0000	495DFFFF	595D0000	595DFFFF	93	
Reserved	64	495E0000	495EFFFF	595E0000	595EFFFF	94	
Reserved	64	495F0000	495FFFFF	595F0000	595FFFFF	95	
Reserved	64	49600000	4960FFFF	59600000	5960FFFF	96	
Reserved	64	49610000	4961FFFF	59610000	5961FFFF	97	
Reserved	64	49620000	4962FFFF	59620000	5962FFFF	98	
Reserved	64	49630000	4963FFFF	59630000	5963FFFF	99	
Reserved	64	49640000	4964FFFF	59640000	5964FFFF	100	
Reserved	64	49650000	4965FFFF	59650000	5965FFFF	101	
Reserved	64	49660000	4966FFFF	59660000	5966FFFF	102	
Reserved	64	49670000	4967FFFF	59670000	5967FFFF	103	
Reserved	64	49680000	4968FFFF	59680000	5968FFFF	104	
Reserved	64	49690000	4969FFFF	59690000	5969FFFF	105	
Reserved	64	496A0000	496AFFFF	596A0000	596AFFFF	106	
Reserved	64	496B0000	496BFFFF	596B0000	596BFFFF	107	

Table continues on the next page...

Table continued from the previous page...

NIC Port	Size (KB)	Start Address (NS)	End Address (NS)	Start Address (S)	End Address (S)	IPS slot	Domain (MIX)
Reserved	64	496C0000	496CFFFF	596C0000	596CFFFF	108	
Reserved	64	496D0000	496DFFFF	596D0000	596DFFFF	109	
Reserved	64	496E0000	496EFFFF	596E0000	596EFFFF	110	
Reserved	64	496F0000	496FFFFF	596F0000	596FFFFF	111	
Reserved	64	49700000	4970FFFF	59700000	5970FFFF	112	
Reserved	64	49710000	4971FFFF	59710000	5971FFFF	113	
Reserved	64	49720000	4972FFFF	59720000	5972FFFF	114	
Reserved	64	49730000	4973FFFF	59730000	5973FFFF	115	
Reserved	64	49740000	4974FFFF	59740000	5974FFFF	116	
Reserved	64	49750000	4975FFFF	59750000	5975FFFF	117	
Reserved	64	49760000	4976FFFF	59760000	5976FFFF	118	
Reserved	64	49770000	4977FFFF	59770000	5977FFFF	119	
Reserved	64	49780000	4978FFFF	59780000	5978FFFF	120	
Reserved	64	49790000	4979FFFF	59790000	5979FFFF	121	
Reserved	64	497A0000	497AFFFF	597A0000	597AFFFF	122	
Reserved	64	497B0000	497BFFFF	597B0000	597BFFFF	123	
Reserved	64	497C0000	497CFFFF	597C0000	597CFFFF	124	
Reserved	64	497D0000	497DFFFF	597D0000	597DFFFF	125	
Reserved	64	497E0000	497EFFFF	597E0000	597EFFFF	126	
Reserved	64	497F0000	497FFFFF	597F0000	597FFFFF	127	

Chapter 3
Interrupt and DMA Assignments

3.1 Overview

This section provides information on the assignments of the interrupts and DMA.

3.2 System Interrupts

The system interrupts are routed to both the Cortex-A55 Generic Interrupt Controller (GIC) and the Cortex-M33 Nested Vectored Interrupt Controller (NVIC). The following table describes the interrupt sources.

Table 6. System Interrupt Summary

Module	IRQ	Logic	Description
-	0	-	Reserved
DAP	1	-	DAP Interrupt
-	2	OR	Reserved
-	2	OR	Reserved
CM33	3	OR	CTI trigger outputs from CM33 platform (internal: CTIIRQ[0])
	3	OR	CTI trigger outputs from CM33 platform (internal: CTIIRQ[1])
CA55	4	OR	CTI trigger outputs from CA55 platform (internal: nCTIIRQ[0])
	4	OR	CTI trigger outputs from CA55 platform (internal: nCTIIRQ[1])
	5	OR	Performance Unit Interrupts from CA55 platform (internally: PMUIRQ[0])
	5	OR	Performance Unit Interrupts from CA55 platform (internally: PMUIRQ[1])
	5	OR	Performance Unit Interrupts from CA55 platform (internally: PMUIRQ[2])
	6	OR	ECC error with potential data corruption or loss of coherency from CA55 platform L3 Cache
	6	OR	ECC error with potential data corruption or loss of coherency from CA55 platform Core-0 L1 or L2 Cache
	6	OR	ECC error with potential data corruption or loss of coherency from CA55 platform Core-1 L1 or L2 Cache
	7	OR	1-bit or 2-bit ECC or Parity error from CA55 platform L3 Cache
	7	OR	1-bit or 2-bit ECC or Parity error from CA55 platform Core-0 L1 or L2 Cache

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
	7	OR	1-bit or 2-bit ECC or Parity error from CA55 platform Core-1 L1 or L2 Cache
CAN_FD1	8	OR	AONMIX CAN-FD Interrupt from busoff
	8	OR	AONMIX CAN-FD Interrupt from CAN line error
	8	OR	AONMIX CAN-FD Ored interrupts from ipi_int_MB
	8	OR	AONMIX CAN-FD Rx warning Interrupt
	8	OR	AONMIX CAN-FD Tx warning Interrupt
	8	OR	AONMIX CAN-FD Interrupt from wake up
	8	OR	Reserved
	8	OR	Reserved
	8	OR	AONMIX CAN-FD Busoff done interrupt
	8	OR	AONMIX CAN-FD FD error interrupt
	8	OR	AONMIX CAN-FD FIFO underflow interrupt
	8	OR	AONMIX CAN-FD FIFO overflow interrupt
	8	OR	AONMIX CAN-FD FIFO watermarker interrupt
	8	OR	AONMIX CAN-FD FIFO data available interrupt
	9	OR	AONMIX CAN-FD Correctable error interrupt
	9	OR	AONMIX CAN-FD Non correctable error int host
	9	OR	AONMIX CAN-FD Non correctable error int internal
GPIO1	10	OR	AONMIX GPIO interrupt request 0
	10	OR	AONMIX GPIO async wakeup interrupt request 0
	11	OR	AONMIX GPIO interrupt request 1
	11	OR	AONMIX GPIO async wakeup interrupt request 1
I3C1	12	OR	AONMIX I3C interrupt request
	12	OR	AONMIX I3C interrupt request
LPI2C1	13	OR4[1]	AONMIX LPI2C Master interrupt
	13	OR4[1]	AONMIX LPI2C Master async interrupt
	13	OR4[1]	AONMIX LPI2C Slave interrupt
	13	OR4[1]	AONMIX LPI2C Slave async interrupt
LPI2C2	14	OR4[1]	AONMIX LPI2C Master interrupt
	14	OR4[1]	AONMIX LPI2C Master async interrupt
	14	OR4[1]	AONMIX LPI2C Slave interrupt
	14	OR4[1]	AONMIX LPI2C Slave async interrupt

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
LPIT1	15	OR	AONMIX LPIT global interrupt
LPSPI1	16	OR	AONMIX LPSPI interrupt
	16	OR	AONMIX LPSPI async interrupt
LPSPI2	17	OR	AONMIX LPSPI interrupt
	17	OR	AONMIX LPSPI async interrupt
LPTMR1	18	OR	AONMIX LPTMR async interrupt
LPUART1	19	OR4[1]	AONMIX LPUART TX interrupt
	19	OR4[1]	AONMIX LPUART TX async interrupt
	19	OR4[1]	AONMIX LPUART RX interrupt
	19	OR4[1]	AONMIX LPUART RX async interrupt
LPUART2	20	OR4[1]	AONMIX LPUART TX interrupt
	20	OR4[1]	AONMIX LPUART TX async interrupt
	20	OR4[1]	AONMIX LPUART RX interrupt
	20	OR4[1]	AONMIX LPUART RX async interrupt
MU1	21		AONMIX MU Ored of all (tx,rx,gp,core,murip) interrupt to MUA
	22		AONMIX MU Ored of all (tx,rx,gp,core,murip) interrupt to MUB
MU2	23		WAKEUPMIX MU Ored of all (tx,rx,gp,core,murip) interrupt to MUA
	24		WAKEUPMIX MU Ored of all (tx,rx,gp,core,murip) interrupt to MUB
-	25	-	Reserved
-	26	-	Reserved
-	27	-	Reserved
Edgelock	28		AONMIX Edgelock Trust MUA RX full interrupt
	29		AONMIX Edgelock Trust MUA TX empty interrupt
	30		AONMIX Edgelock Apps Core MUA RX full interrupt
	31		AONMIX Edgelock Apps Core MUA TX empty interrupt
	32		AONMIX Edgelock Realtime Core MUA RX full interrupt
	33		AONMIX Edgelock Realtime Core MUA TX empty interrupt
	34		AONMIX Edgelock secure interrupt

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
	35		AONMIX Edgelock non-secure interrupt
TPM1	36	OR	AONMIX LPTPM global interrupt
	36	OR	AONMIX LPTPM async interrupt
TPM2	37	OR	AONMIX LPTPM global interrupt
	37	OR	AONMIX LPTPM async interrupt
WDOG1	38	OR	AONMIX Watchdog interrupt request. SRC reset source.
	38	OR	AONMIX Watchdog async interrupt request. SRC reset source.
WDOG2	39	OR	AONMIX Watchdog interrupt request. SRC reset source.
	39	OR	AONMIX Watchdog async interrupt request. SRC reset source.
TRDC_MGR_A	40		AONMIX TRDC transfer error interrupt
-	41	-	Reserved
-	42	-	Reserved
-	43	OR	Reserved
-	43	OR	Reserved
-	44	OR	Reserved
-	44	OR	Reserved
SAI1	45	OR4[1]	AONMIX SAI TX interrupt
	45	OR4[1]	AONMIX SAI TX async interrupt
	45	OR4[1]	AONMIX SAI RX interrupt
	45	OR4[1]	AONMIX SAI RX async interrupt
CM33	46	OR	AONMIX M33 PS Tag Parity Error
	46	OR	AONMIX M33 PS Data Parity Error
CM33_ECC_MCM	47		AONMIX M33 TCM ECC interrupt FIXME - temporary holder for TCM ECC.
	48		AONMIX M33 TCM Error interrupt FIXME - temporary holder for TCM ECC.
-	49	-	Reserved
-	50	-	Reserved
CAN_FD2	51	OR	WAKEUPMIX CAN-FD Interrupt from busoff
	51	OR	WAKEUPMIX CAN-FD Interrupt from CAN line error

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
	51	OR	WAKEUPMIX CAN-FD Ored interrupts from ipi_int_MB
	51	OR	WAKEUPMIX CAN-FD Rx warning Interrupt
	51	OR	WAKEUPMIX CAN-FD Tx warning Interrupt
	51	OR	WAKEUPMIX CAN-FD Interrupt from wake up
	51	OR	Reserved
	51	OR	Reserved
	51	OR	WAKEUPMIX CAN-FD Busoff done interrupt
	51	OR	WAKEUPMIX CAN-FD FD error interrupt
	51	OR	WAKEUPMIX CAN-FD FIFO underflow interrupt
	51	OR	WAKEUPMIX CAN-FD FIFO overflow interrupt
	51	OR	WAKEUPMIX CAN-FD FIFO watermarker interrupt
	51	OR	WAKEUPMIX CAN-FD FIFO data available interrupt
	52	OR	WAKEUPMIX CAN-FD Correctable error interrupt
	52	OR	WAKEUPMIX CAN-FD Non correctable error int host
	52	OR	WAKEUPMIX CAN-FD Non correctable error int internal
FLEXIO1	53	OR	WAKEUPMIX FLEXIO interrupt
	53	OR	WAKEUPMIX FLEXIO async interrupt
FLEXIO2	54	OR	WAKEUPMIX FLEXIO interrupt
	54	OR	WAKEUPMIX FLEXIO async interrupt
FLEXSPI1	55	-	WAKEUPMIX FLEXSPI interrupt
-	56	-	Reserved
GPIO2	57	OR	WAKEUPMIX GPIO interrupt request 0
	57	OR	WAKEUPMIX GPIO async wakeup interrupt request 0
	58	OR	WAKEUPMIX GPIO interrupt request 1
	58	OR	WAKEUPMIX GPIO async wakeup interrupt request 1
GPIO3	59	OR	WAKEUPMIX GPIO interrupt request 0
	59	OR	WAKEUPMIX GPIO async wakeup interrupt request 0
	60	OR	WAKEUPMIX GPIO interrupt request 1
	60	OR	WAKEUPMIX GPIO async wakeup interrupt request 1
I3C2	61	OR	WAKEUPMIX I3C interrupt request
	61	OR	WAKEUPMIX I3C interrupt request
LPI2C3	62	OR4[1]	WAKEUPMIX LPI2C Master interrupt

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
	62	OR4[1]	WAKEUPMIX LPI2C Master async interrupt
	62	OR4[1]	WAKEUPMIX LPI2C Slave interrupt
	62	OR4[1]	WAKEUPMIX LPI2C Slave async interrupt
LPI2C4	63	OR4[1]	WAKEUPMIX LPI2C Master interrupt
	63	OR4[1]	WAKEUPMIX LPI2C Master async interrupt
	63	OR4[1]	WAKEUPMIX LPI2C Slave interrupt
	63	OR4[1]	WAKEUPMIX LPI2C Slave async interrupt
LPIT2	64		WAKEUPMIX LPIT global interrupt
LPSPI3	65	OR	WAKEUPMIX LPSPI interrupt
	65	OR	WAKEUPMIX LPSPI async interrupt
LPSPI4	66	OR	WAKEUPMIX LPSPI interrupt
	66	OR	WAKEUPMIX LPSPI async interrupt
LPTMR2	67		WAKEUPMIX LPTMR async interrupt
LPUART3	68	OR4[1]	WAKEUPMIX LPUART TX interrupt
	68	OR4[1]	WAKEUPMIX LPUART TX async interrupt
	68	OR4[1]	WAKEUPMIX LPUART RX interrupt
	68	OR4[1]	WAKEUPMIX LPUART RX async interrupt
LPUART4	69	OR4[1]	WAKEUPMIX LPUART TX interrupt
	69	OR4[1]	WAKEUPMIX LPUART TX async interrupt
	69	OR4[1]	WAKEUPMIX LPUART RX interrupt
	69	OR4[1]	WAKEUPMIX LPUART RX async interrupt
LPUART5	70	OR4[1]	WAKEUPMIX LPUART TX interrupt
	70	OR4[1]	WAKEUPMIX LPUART TX async interrupt
	70	OR4[1]	WAKEUPMIX LPUART RX interrupt
	70	OR4[1]	WAKEUPMIX LPUART RX async interrupt
LPUART6	71	OR4[1]	WAKEUPMIX LPUART TX interrupt
	71	OR4[1]	WAKEUPMIX LPUART TX async interrupt
	71	OR4[1]	WAKEUPMIX LPUART RX interrupt
	71	OR4[1]	WAKEUPMIX LPUART RX async interrupt
MTR_MSTR	72		AONMIX MTR Master error interrupt
BBNSM	73	OR	BBNSM Non-Secure external wakeup alarm
	73	OR	BBNSM Non-Secure functional interrupt

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
	73	OR	BBNSM Non-Secure functional interrupt
	73	OR	BBNSM Non-Secure request to power on interrupt
	73	OR	BBNSM Non-Secure request to power on interrupt
SYS_CTR1	74	OR	WAKEUPMIX System Counter compare interrupt 0
	74	OR	WAKEUPMIX System counter compare interrupt 1
TPM3	75	OR	WAKEUPMIX LPTPM global interrupt
	75	OR	WAKEUPMIX LPTPM async interrupt
TPM4	76	OR	WAKEUPMIX LPTPM global interrupt
	76	OR	WAKEUPMIX LPTPM async interrupt
TPM5	77	OR	WAKEUPMIX LPTPM global interrupt
	77	OR	WAKEUPMIX LPTPM async interrupt
TPM6	78	OR	WAKEUPMIX LPTPM global interrupt
	78	OR	WAKEUPMIX LPTPM async interrupt
WDOG3	79	OR	WAKEUPMIX Watchdog interrupt request. SRC reset source.
	79	OR	WAKEUPMIX Watchdog async interrupt request. SRC reset source.
WDOG4	80	OR	WAKEUPMIX Watchdog interrupt request. SRC reset source.
	80	OR	WAKEUPMIX Watchdog async interrupt request. SRC reset source.
WDOG5	81	OR	WAKEUPMIX Watchdog interrupt request. SRC reset source.
	81	OR	WAKEUPMIX Watchdog async interrupt request. SRC reset source.
TRDC_MGR_W	82		WAKEUPMIX TRDC transfer error interrupt
TEMPSENSOR	83	OR	ANAMIX TempSensor (Temperature alarm).
	83	OR	ANAMIX TempSensor (Temperature critical alarm). SRC Reset source.
-	84	-	Reserved
-	85	-	Reserved
USDHC1	86		MEGAMIX uSDHC Enhanced SDHC Interrupt Request
USDHC2	87		MEGAMIX uSDHC Enhanced SDHC Interrupt Request
TRDC_MGR_M	88		MEGAMIX TRDC transfer error interrupt
TRDC_MGR_N	89		NIC_WRAPPER TRDC transfer error interrupt

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
DDR_TOP	90		DRAM controller Performance Monitor Interrupt
	91		DRAM controller Critical Interrupt
	92		DRAM PHY Critical Interrupt
-	93	-	Reserved
eDMA3_1	94		AONMIX EDMA error interrupt
	95		AONMIX EDMA channel 0 interrupt
	96		AONMIX EDMA channel 1 interrupt
	97		AONMIX EDMA channel 2 interrupt
	98		AONMIX EDMA channel 3 interrupt
	99		AONMIX EDMA channel 4 interrupt
	100		AONMIX EDMA channel 5 interrupt
	101		AONMIX EDMA channel 6 interrupt
	102		AONMIX EDMA channel 7 interrupt
	103		AONMIX EDMA channel 8 interrupt
	104		AONMIX EDMA channel 9 interrupt
	105		AONMIX EDMA channel 10 interrupt
	106		AONMIX EDMA channel 11 interrupt
	107		AONMIX EDMA channel 12 interrupt
	108		AONMIX EDMA channel 13 interrupt
	109		AONMIX EDMA channel 14 interrupt
	110		AONMIX EDMA channel 15 interrupt
	111		AONMIX EDMA channel 16 interrupt
	112		AONMIX EDMA channel 17 interrupt
	113		AONMIX EDMA channel 18 interrupt
	114		AONMIX EDMA channel 19 interrupt
	115		AONMIX EDMA channel 20 interrupt
	116		AONMIX EDMA channel 21 interrupt
	117		AONMIX EDMA channel 22 interrupt
	118		AONMIX EDMA channel 23 interrupt
	119		AONMIX EDMA channel 24 interrupt
	120		AONMIX EDMA channel 25 interrupt
	121		AONMIX EDMA channel 26 interrupt

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
	122		AONMIX EDMA channel 27 interrupt
	123		AONMIX EDMA channel 28 interrupt
	124		AONMIX EDMA channel 29 interrupt
	125		AONMIX EDMA channel 30 interrupt
	126		Reserved
eDMA4_2	127		WAKEUPMIX EDMA error interrupt
	128	OR	WAKEUPMIX EDMA channel 0 interrupt
	128	OR	WAKEUPMIX EDMA channel 1 interrupt
	129	OR	WAKEUPMIX EDMA channel 2 interrupt
	129	OR	WAKEUPMIX EDMA channel 3 interrupt
	130	OR	WAKEUPMIX EDMA channel 4 interrupt
	130	OR	WAKEUPMIX EDMA channel 5 interrupt
	131	OR	WAKEUPMIX EDMA channel 6 interrupt
	131	OR	WAKEUPMIX EDMA channel 7 interrupt
	132	OR	WAKEUPMIX EDMA channel 8 interrupt
	132	OR	WAKEUPMIX EDMA channel 9 interrupt
	133	OR	WAKEUPMIX EDMA channel 10 interrupt
	133	OR	WAKEUPMIX EDMA channel 11 interrupt
	134	OR	WAKEUPMIX EDMA channel 12 interrupt
	134	OR	WAKEUPMIX EDMA channel 13 interrupt
	135	OR	WAKEUPMIX EDMA channel 14 interrupt
	135	OR	WAKEUPMIX EDMA channel 15 interrupt
	136	OR	WAKEUPMIX EDMA channel 16 interrupt
	136	OR	WAKEUPMIX EDMA channel 17 interrupt
	137	OR	WAKEUPMIX EDMA channel 18 interrupt
	137	OR	WAKEUPMIX EDMA channel 19 interrupt
	138	OR	WAKEUPMIX EDMA channel 20 interrupt
	138	OR	WAKEUPMIX EDMA channel 21 interrupt
	139	OR	WAKEUPMIX EDMA channel 22 interrupt
	139	OR	WAKEUPMIX EDMA channel 23 interrupt
	140	OR	WAKEUPMIX EDMA channel 24 interrupt
	140	OR	WAKEUPMIX EDMA channel 25 interrupt

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
	141	OR	WAKEUPMIX EDMA channel 26 interrupt
	141	OR	WAKEUPMIX EDMA channel 27 interrupt
	142	OR	WAKEUPMIX EDMA channel 28 interrupt
	142	OR	WAKEUPMIX EDMA channel 29 interrupt
	143	OR	WAKEUPMIX EDMA channel 30 interrupt
	143	OR	WAKEUPMIX EDMA channel 31 interrupt
	144	OR	WAKEUPMIX EDMA channel 32 interrupt
	144	OR	WAKEUPMIX EDMA channel 33 interrupt
	145	OR	WAKEUPMIX EDMA channel 34 interrupt
	145	OR	WAKEUPMIX EDMA channel 35 interrupt
	146	OR	WAKEUPMIX EDMA channel 36 interrupt
	146	OR	WAKEUPMIX EDMA channel 37 interrupt
	147	OR	WAKEUPMIX EDMA channel 38 interrupt
	147	OR	WAKEUPMIX EDMA channel 39 interrupt
	148	OR	WAKEUPMIX EDMA channel 40 interrupt
	148	OR	WAKEUPMIX EDMA channel 41 interrupt
	149	OR	WAKEUPMIX EDMA channel 42 interrupt
	149	OR	WAKEUPMIX EDMA channel 43 interrupt
	150	OR	WAKEUPMIX EDMA channel 44 interrupt
	150	OR	WAKEUPMIX EDMA channel 45 interrupt
	151	OR	WAKEUPMIX EDMA channel 46 interrupt
	151	OR	WAKEUPMIX EDMA channel 47 interrupt
	152	OR	WAKEUPMIX EDMA channel 48 interrupt
	152	OR	WAKEUPMIX EDMA channel 49 interrupt
	153	OR	WAKEUPMIX EDMA channel 50 interrupt
	153	OR	WAKEUPMIX EDMA channel 51 interrupt
	154	OR	WAKEUPMIX EDMA channel 52 interrupt
	154	OR	WAKEUPMIX EDMA channel 53 interrupt
	155	OR	WAKEUPMIX EDMA channel 54 interrupt
	155	OR	WAKEUPMIX EDMA channel 55 interrupt
	156	OR	WAKEUPMIX EDMA channel 56 interrupt
	156	OR	WAKEUPMIX EDMA channel 57 interrupt

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
	157	OR	WAKEUPMIX EDMA channel 58 interrupt
	157	OR	WAKEUPMIX EDMA channel 59 interrupt
	158	OR	WAKEUPMIX EDMA channel 60 interrupt
	158	OR	WAKEUPMIX EDMA channel 61 interrupt
	159	OR	WAKEUPMIX EDMA channel 62 interrupt
	159	OR	WAKEUPMIX EDMA channel 63 interrupt
BOOT ROM Debug Control	160		Debug Wakeup Interrupt
EdgeLock	161		Edgelock Group 1 reset source if no Edgelock enclave reference clock is detected. Output synchronized to 32KHz clk.
	162		Edgelock Group 2 reset source Edgelock enclave reference clock is not detected or too slow. Output synchronized to ref1_clk.
	163		Edgelock Group 2 reset source Edgelock enclave reference clock is not detected or too slow. Output synchronized to ref1_clk.
DBG_TRACE	164		JTAGSW DAP MDM-AP SRC reset source
JTAGC	165		JTAGC SRC reset source
CM33	166		CM33 SYSREQRST SRC reset source
	167		CM33 LOCKUP SRC reset source
-	168	-	Reserved
-	169	-	Reserved
SAI2	170	OR4[1]	WAKEUPMIX SAI TX interrupt
	170	OR4[1]	WAKEUPMIX SAI TX async interrupt
	170	OR4[1]	WAKEUPMIX SAI RX interrupt
	170	OR4[1]	WAKEUPMIX SAI RX async interrupt
SAI3	171	OR4[1]	WAKEUPMIX SAI TX interrupt
	171	OR4[1]	WAKEUPMIX SAI TX async interrupt
	171	OR4[1]	WAKEUPMIX SAI RX interrupt
	171	OR4[1]	WAKEUPMIX SAI RX async interrupt
ISI	172		WAKEUPMIX ISI interrupt
PXP	173		WAKEUPMIX PXP interrupt 0
	174		WAKEUPMIX PXP interrupt 1
CSI	175		WAKEUPMIX CSI interrupt

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
LCDIF1	176		WAKEUPMIX LCDIF Sync Interrupt
MIPI_DSI_CONTROLLER	177		WAKEUPMIX MIPI DSI Interrupt Request
ML	178		Machine Learning Processor Interrupt
ENET1	179	OR4[1]	MAC 0 Receive Buffer Done
	179	OR4[1]	MAC 0 Receive Frame Done
	179	OR4[1]	MAC 0 Transmit Buffer Done
	179	OR4[1]	MAC 0 Transmit Frame Done
	180	OR4[1]	MAC 0 Receive Buffer Done
	180	OR4[1]	MAC 0 Receive Frame Done
	180	OR4[1]	MAC 0 Transmit Buffer Done
	180	OR4[1]	MAC 0 Transmit Frame Done
	181	OR	MAC 0 Periodic Timer Overflow
	181	OR	MAC 0 Time Stamp Available
	181	OR	MAC 0 Payload Receive Error
	181	OR	MAC 0 Transmit FIFO Underrun
	181	OR	MAC 0 Collision Retry Limit
	181	OR	MAC 0 Late Collision
	181	OR	MAC 0 Ethernet Bus Error
	181	OR	MAC 0 MII Data Transfer Done
	181	OR	MAC 0 Receive Buffer Done
	181	OR	MAC 0 Receive Frame Done
	181	OR	MAC 0 Transmit Buffer Done
	181	OR	MAC 0 Transmit Frame Done
	181	OR	MAC 0 Graceful Stop
	181	OR	MAC 0 Babbling Transmit Error
	181	OR	MAC 0 Babbling Receive Error
	181	OR	MAC 0 Receive Flush Frame0
	181	OR	MAC 0 Receive Flush Frame1
	181	OR	MAC 0 Receive Flush Frame2
	181	OR	MAC 0 Wakeup Request (sync)
	181	OR	MAC 0 Babbling Receive Error
	181	OR	MAC 0 Wakeup Request (sync)

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
	182		MAC 0 1588 Timer Interrupt – synchronous
ENET_QOS	183		ENET QOS TSN Interrupt from PMT
	184	OR	ENET QOS TSN LPI RX exit Interrupt
	184	OR	ENET QOS TSN Host System Interrupt
	184	OR	ENET QOS TSN Host System RX Channel Interrupts, Logical OR of channels[4:0]
	184	OR	ENET QOS TSN Host System TX Channel Interrupts, Logical OR of channels[4:0]
-	185	-	Reserved
-	186	-	Reserved
USB1	187		USB-1 Wake-up Interrupt
USB2	188		USB-2 Wake-up Interrupt
GPIO4	189	OR	WAKEUPMIX GPIO interrupt request 0
	189	OR	WAKEUPMIX GPIO async wakeup interrupt request 0
	190	OR	WAKEUPMIX GPIO interrupt request 1
	190	OR	WAKEUPMIX GPIO async wakeup interrupt request 1
LPSPI5	191	OR	WAKEUPMIX LPSPI interrupt
	191	OR	WAKEUPMIX LPSPI async interrupt
LPSPI6	192	OR	WAKEUPMIX LPSPI interrupt
	192	OR	WAKEUPMIX LPSPI async interrupt
LPSPI7	193	OR	WAKEUPMIX LPSPI interrupt
	193	OR	WAKEUPMIX LPSPI async interrupt
LPSPI8	194	OR	WAKEUPMIX LPSPI interrupt
	194	OR	WAKEUPMIX LPSPI async interrupt
LPI2C5	195	OR4[1]	WAKEUPMIX LPI2C Master interrupt
	195	OR4[1]	WAKEUPMIX LPI2C Master async interrupt
	195	OR4[1]	WAKEUPMIX LPI2C Slave interrupt
	195	OR4[1]	WAKEUPMIX LPI2C Slave async interrupt
LPI2C6	196	OR4[1]	WAKEUPMIX LPI2C Master interrupt
	196	OR4[1]	WAKEUPMIX LPI2C Master async interrupt
	196	OR4[1]	WAKEUPMIX LPI2C Slave interrupt
	196	OR4[1]	WAKEUPMIX LPI2C Slave async interrupt
LPI2C7	197	OR4[1]	WAKEUPMIX LPI2C Master interrupt

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
	197	OR4[1]	WAKEUPMIX LPI2C Master async interrupt
	197	OR4[1]	WAKEUPMIX LPI2C Slave interrupt
	197	OR4[1]	WAKEUPMIX LPI2C Slave async interrupt
LPI2C8	198	OR4[1]	WAKEUPMIX LPI2C Master interrupt
	198	OR4[1]	WAKEUPMIX LPI2C Master async interrupt
	198	OR4[1]	WAKEUPMIX LPI2C Slave interrupt
	198	OR4[1]	WAKEUPMIX LPI2C Slave async interrupt
PDM	199		AONMIXMIX PDM interrupt
	200		AONMIXMIX PDM interrupt
	201		AONMIXMIX PDM interrupt
	202		AONMIXMIX PDM interrupt
AUDIOXCVR	203		WAKEUPMIX AUDIO XCVR interrupt
	204		WAKEUPMIX AUDIO XCVR interrupt
USDHC3	205		MEGAMIX uSDHC Enhanced SDHC Interrupt Request
MECC	206		OCRAM MECC interrupt
	207		OCRAM MECC interrupt
TRDC_MGR_H	208		HSIOMIX TRDC transfer error interrupt
TRDC_MGR_C	209		MEDIAMIX TRDC transfer error interrupt
LPUART7	210	OR4[1]	WAKEUPMIX LPUART TX interrupt
	210	OR4[1]	WAKEUPMIX LPUART TX async interrupt
	210	OR4[1]	WAKEUPMIX LPUART RX interrupt
	210	OR4[1]	WAKEUPMIX LPUART RX async interrupt
LPUART8	211	OR4[1]	WAKEUPMIX LPUART TX interrupt
	211	OR4[1]	WAKEUPMIX LPUART TX async interrupt
	211	OR4[1]	WAKEUPMIX LPUART RX interrupt
	211	OR4[1]	WAKEUPMIX LPUART RX async interrupt
CM33	212		CM33 MCM interrupt
ANAMIX	213		ANAMIX SFA interrupt
GIC600	214		GIC600 INTERRUPT
	215		GIC600 INTERRUPT
	216		GIC600 INTERRUPT
ADC1	217		ADC interrupt

Table continues on the next page...

Table 6. System Interrupt Summary (continued)

Module	IRQ	Logic	Description
	218		ADC interrupt
	219		ADC interrupt
-	220	-	Reserved
I3C1_RAW_IRQ_GLUE	221		I3C1 wakeup irq after double sync
I3C2_RAW_IRQ_GLUE	222		I3C2 wakeup irq after double sync
-	223 - 267	-	Reserved

1. This interrupt will be asserted if any one of the four interrupts with the same IRQ# are asserted.

3.3 DMA MUX

The table below shows the DMA request signals for the peripherals in the chip. Channel Mux assignments noted as FULL are all channels (0,1,2,3,..), EVEN are even channels only (0,2,4,6,..), ODD are odd channels (1,3,5,7,..), and NONE are 1:1 with the requestor and channel.

NOTE

eDMA3_1 AONMIX supports 31 channels.

Table 7. eDMA3_1 MUX Mapping - AONMIX

DMA	Module	Logic	Channel MUX	Description
0	-	-	NONE	Reserved
1	CAN1	-	NONE	CAN-FD 1 DMA request and response
2	-	-	NONE	Reserved
3	GPIO1	-	NONE	GPIO1 channel 0 DMA request and response
4	GPIO1	-	NONE	GPIO1 channel 1 DMA request and response
5	I3C1	-	NONE	I3C1 "To-bus" DMA request and response
6	I3C1	-	NONE	I3C1 "from-bus" DMA request and response
7	LPI2C1	OR	NONE	LPI2C1 Master TX DMA request
7	LPI2C1	OR	NONE	LPI2C1 Slave TX DMA request
8	LPI2C1	OR	NONE	LPI2C1 Master RX DMA request
8	LPI2C1	OR	NONE	LPI2C1 Slave RX DMA request
9	LPI2C2	OR	NONE	LPI2C2 Master TX DMA request
9	LPI2C2	OR	NONE	LPI2C2 Slave TX DMA request
10	LPI2C2	OR	NONE	LPI2C2 Master RX DMA request
10	LPI2C2	OR	NONE	LPI2C2 Slave RX DMA request
11	LPSPI1	-	NONE	LPSPI1 TX DMA request
12	LPSPI1	-	NONE	LPSPI1 RX DMA request

Table continues on the next page...

Table 7. eDMA3_1 MUX Mapping - AONMIX (continued)

DMA	Module	Logic	Channel MUX	Description
13	LPSPI2	-	NONE	LPSPI2 TX DMA request
14	LPSPI2	-	NONE	LPSPI2 RX DMA request
15	LPTMR1	-	NONE	LPTIMER1 DMA request and response
16	LPUART1	-	NONE	LPUART1 DMA TX request
17	LPUART1	-	NONE	LPUART1 DMA RX request and response
18	LPUART2	-	NONE	LPUART2 DMA TX request
19	LPUART2	-	NONE	LPUART2 DMA RX request and response
20	EdgeLock secure enclave	-	NONE	Edgelock enclave DMA request and response
21	SAI1	-	NONE	SAI1 TX DMA request and response
22	SAI1	-	NONE	SAI1 RX DMA request and response
23	TPM1	OR	NONE	LPTPM1 DMA Channel 0 request and response
23	TPM1	OR	NONE	LPTPM1 DMA Channel 2 request and response
24	TPM1	OR	NONE	LPTPM1 DMA Channel 1 request and response
24	TPM1	OR	NONE	LPTPM1 DMA Channel 3 request and response
25	TPM1	-	NONE	LPTPM1 DMA Overflow request and response
26	TPM2	OR	NONE	LPTPM2 DMA Channel 0 request and response
26	TPM2	OR	NONE	LPTPM2 DMA Channel 2 request and response
27	TPM2	OR	NONE	LPTPM2 DMA Channel 1 request and response
27	TPM2	OR	NONE	LPTPM2 DMA Channel 3 request and response
28	TPM2	-	NONE	LPTPM2 DMA Overflow request and response
29	PDM	-	NONE	PDM DMA request and response
30	ADC1	-	NONE	ADC DMA request

Table 8. eDMA4_2 Mux Mapping - WAKEUPMIX

DMA	Module	Logic	Channel Mux	Description
0	-	-	EVEN	Reserved
1	CAN2	-	ODD	CAN-FD 2 DMA request and response
2	GPIO2	-	EVEN	GPIO2 channel 0 DMA request and response
3	GPIO2	-	ODD	GPIO2 channel 1 DMA request and response
4	GPIO3	-	EVEN	GPIO3 channel 0 DMA request and response
5	GPIO3	-	ODD	GPIO3 channel 1 DMA request and response
6	I3C2	-	EVEN	I3C2 "To-bus" DMA request and response
7	I3C2	-	ODD	I3C2 "from-bus" DMA request and response

Table continues on the next page...

Table 8. eDMA4_2 Mux Mapping - WAKEUPMIX (continued)

DMA	Module	Logic	Channel Mux	Description
8	LPI2C3	OR	EVEN	LPI2C3 Master TX DMA request
8	LPI2C3	OR	EVEN	LPI2C3 Slave TX DMA request
9	LPI2C3	OR	ODD	LPI2C3 Master RX DMA request
9	LPI2C3	OR	ODD	LPI2C3 Slave RX DMA request
10	LPI2C4	OR	EVEN	LPI2C4 Master TX DMA request
10	LPI2C4	OR	EVEN	LPI2C4 Slave TX DMA request
11	LPI2C4	OR	ODD	LPI2C4 Master RX DMA request
11	LPI2C4	OR	ODD	LPI2C4 Slave RX DMA request
12	LPSPI3	-	EVEN	LPSPI3 TX DMA request
13	LPSPI3	-	ODD	LPSPI3 RX DMA request
14	LPSPI4	-	EVEN	LPSPI4 TX DMA request
15	LPSPI4	-	ODD	LPSPI4 RX DMA request
16	LPTMR2	-	EVEN	LPTIMER2 DMA request and response
17	LPUART3	-	ODD	LPUART3 DMA TX request
18	LPUART3	-	EVEN	LPUART3 DMA RX request and response
19	LPUART4	-	ODD	LPUART4 DMA TX request
20	LPUART4	-	EVEN	LPUART4 DMA RX request and response
21	LPUART5	-	ODD	LPUART5 DMA TX request
22	LPUART5	-	EVEN	LPUART5 DMA RX request and response
23	LPUART6	-	ODD	LPUART6 DMA TX request
24	LPUART6	-	EVEN	LPUART6 DMA RX request and response
25	TPM3	OR	ODD	LPTPM3 DMA Channel 0 request and response
25	TPM3	OR	ODD	LPTPM3 DMA Channel 2 request and response
26	TPM3	OR	EVEN	LPTPM3 DMA Channel 1 request and response
26	TPM3	OR	EVEN	LPTPM3 DMA Channel 3 request and response
27	TPM3	-	ODD	LPTPM3 DMA Overflow request and response
28	TPM4	OR	EVEN	LPTPM4 DMA Channel 0 request and response
28	TPM4	OR	EVEN	LPTPM4 DMA Channel 2 request and response
29	TPM4	OR	ODD	LPTPM4 DMA Channel 1 request and response
29	TPM4	OR	ODD	LPTPM4 DMA Channel 3 request and response
30	TPM4	-	EVEN	LPTPM4 DMA Overflow request and response
31	TPM5	OR	ODD	LPTPM5 DMA Channel 0 request and response

Table continues on the next page...

Table 8. eDMA4_2 Mux Mapping - WAKEUPMIX (continued)

DMA	Module	Logic	Channel Mux	Description
31	TPM5	OR	ODD	LPTPM5 DMA Channel 2 request and response
32	TPM5	OR	EVEN	LPTPM5 DMA Channel 1 request and response
32	TPM5	OR	EVEN	LPTPM5 DMA Channel 3 request and response
33	TPM5	-	ODD	LPTPM5 DMA Overflow request and response
34	TPM6	OR	EVEN	LPTPM6 DMA Channel 0 request and response
34	TPM6	OR	EVEN	LPTPM6 DMA Channel 2 request and response
35	TPM6	OR	ODD	LPTPM6 DMA Channel 1 request and response
35	TPM6	OR	ODD	LPTPM6 DMA Channel 3 request and response
36	TPM6	-	EVEN	LPTPM6 DMA Overflow request and response
37	FLEXIO1	OR	ODD	FLEXIO1 Shifter DMA Channel 0 request
37	FLEXIO1	OR	ODD	FLEXIO1 Timer DMA Channel 0 request
38	FLEXIO1	OR	EVEN	FLEXIO1 Shifter DMA Channel 1 request
38	FLEXIO1	OR	EVEN	FLEXIO1 Timer DMA Channel 1 request
39	FLEXIO1	OR	ODD	FLEXIO1 Shifter DMA Channel 2 request
39	FLEXIO1	OR	ODD	FLEXIO1 Timer DMA Channel 2 request
40	FLEXIO1	OR	EVEN	FLEXIO1 Shifter DMA Channel 3 request
40	FLEXIO1	OR	EVEN	FLEXIO1 Timer DMA Channel 3 request
41	FLEXIO1	OR	ODD	FLEXIO1 Shifter DMA Channel 4 request
41	FLEXIO1	OR	ODD	FLEXIO1 Timer DMA Channel 4 request
42	FLEXIO1	OR	EVEN	FLEXIO1 Shifter DMA Channel 5 request
42	FLEXIO1	OR	EVEN	FLEXIO1 Timer DMA Channel 5 request
43	FLEXIO1	OR	ODD	FLEXIO1 Shifter DMA Channel 6 request
43	FLEXIO1	OR	ODD	FLEXIO1 Timer DMA Channel 6 request
44	FLEXIO1	OR	EVEN	FLEXIO1 Shifter DMA Channel 7 request
44	FLEXIO1	OR	EVEN	FLEXIO1 Timer DMA Channel 7 request
45	FLEXIO2	OR	ODD	FLEXIO2 Shifter DMA Channel 0 request
45	FLEXIO2	OR	ODD	FLEXIO2 Timer DMA Channel 0 request
46	FLEXIO2	OR	EVEN	FLEXIO2 Shifter DMA Channel 1 request
46	FLEXIO2	OR	EVEN	FLEXIO2 Timer DMA Channel 1 request
47	FLEXIO2	OR	ODD	FLEXIO2 Shifter DMA Channel 2 request
47	FLEXIO2	OR	ODD	FLEXIO2 Timer DMA Channel 2 request
48	FLEXIO2	OR	EVEN	FLEXIO2 Shifter DMA Channel 3 request

Table continues on the next page...

Table 8. eDMA4_2 Mux Mapping - WAKEUPMIX (continued)

DMA	Module	Logic	Channel Mux	Description
48	FLEXIO2	OR	EVEN	FLEXIO2 Timer DMA Channel 3 request
49	FLEXIO2	OR	ODD	FLEXIO2 Shifter DMA Channel 4 request
49	FLEXIO2	OR	ODD	FLEXIO2 Timer DMA Channel 4 request
50	FLEXIO2	OR	EVEN	FLEXIO2 Shifter DMA Channel 5 request
50	FLEXIO2	OR	EVEN	FLEXIO2 Timer DMA Channel 5 request
51	FLEXIO2	OR	ODD	FLEXIO2 Shifter DMA Channel 6 request
51	FLEXIO2	OR	ODD	FLEXIO2 Timer DMA Channel 6 request
52	FLEXIO2	OR	EVEN	FLEXIO2 Shifter DMA Channel 7 request
52	FLEXIO2	OR	EVEN	FLEXIO2 Timer DMA Channel 7 request
53	FLEXSPI1	-	ODD	FLEXSPI1 DMA TX request
54	FLEXSPI1	-	EVEN	FLEXSPI1 DMA RX request
55	-	-	ODD	Reserved
56	-	-	EVEN	Reserved
57	-	-	ODD	Reserved
58	SAI2	-	EVEN	SAI2 TX DMA request and response
59	SAI2	-	ODD	SAI2 RX DMA request and response
60	SAI3	-	EVEN	SAI3 TX DMA request and response
61	SAI3	-	ODD	SAI3 RX DMA request and response
62	GPIO4	-	EVEN	GPIO4 channel 0 DMA request and response
63	GPIO4	-	ODD	GPIO4 channel 1 DMA request and response
64	-	-	EVEN	Reserved
65	SPDIF	-	ODD	SPDIF XCVR DMA request
66	SPDIF	-	EVEN	SPDIF XCVR DMA request
67	ENET1	-	ODD	ENET DMA request and response
68	ENET1	-	EVEN	ENET DMA request and response
69	ENET1	-	ODD	ENET DMA request and response
70	ENET1	-	EVEN	ENET DMA request and response
71	LPI2C5	OR	ODD	LPI2C5 Master TX DMA request
71	LPI2C5	OR	ODD	LPI2C5 Slave TX DMA request
72	LPI2C5	OR	EVEN	LPI2C5 Master RX DMA request
72	LPI2C5	OR	EVEN	LPI2C5 Slave RX DMA request
73	LPI2C6	OR	ODD	LPI2C6 Master TX DMA request

Table continues on the next page...

Table 8. eDMA4_2 Mux Mapping - WAKEUPMIX (continued)

DMA	Module	Logic	Channel Mux	Description
73	LPI2C6	OR	ODD	LPI2C6 Slave TX DMA request
74	LPI2C6	OR	EVEN	LPI2C6 Master RX DMA request
74	LPI2C6	OR	EVEN	LPI2C6 Slave RX DMA request
75	LPI2C7	OR	ODD	LPI2C7 Master TX DMA request
75	LPI2C7	OR	ODD	LPI2C7 Slave TX DMA request
76	LPI2C7	OR	EVEN	LPI2C7 Master RX DMA request
76	LPI2C7	OR	EVEN	LPI2C7 Slave RX DMA request
77	LPI2C8	OR	ODD	LPI2C8 Master TX DMA request
77	LPI2C8	OR	ODD	LPI2C8 Slave TX DMA request
78	LPI2C8	OR	EVEN	LPI2C8 Master RX DMA request
78	LPI2C8	OR	EVEN	LPI2C8 Slave RX DMA request
79	LPSPI5	-	ODD	LPSPI5 TX DMA request
80	LPSPI5	-	EVEN	LPSPI5 RX DMA request
81	LPSPI6	-	ODD	LPSPI6 TX DMA request
82	LPSPI6	-	EVEN	LPSPI6 RX DMA request
83	LPSPI7	-	ODD	LPSPI7 TX DMA request
84	LPSPI7	-	EVEN	LPSPI7 RX DMA request
85	LPSPI8	-	ODD	LPSPI8 TX DMA request
86	LPSPI8	-	EVEN	LPSPI8 RX DMA request
87	LPUART7	-	ODD	LPUART7 DMA TX request
88	LPUART7	-	EVEN	LPUART7 DMA RX request and response
89	LPUART8	-	ODD	LPUART8 DMA TX request
90	LPUART8	-	EVEN	LPUART8 DMA RX request and response
91	ENET_QOS	-	ODD	ENET DMA request and response
92	-	-		Reserved.
93	-	-		Reserved
94	-	-		Reserved

Chapter 4
Enhanced Direct Memory Access (eDMA3)

4.1 Chip-specific eDMA3 information

Table 9. Reference links to related information

Topic	Related module	Reference
Full description	Arm Cortex CM33 Platform (CM33)	CM33 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The Enhanced Direct Memory Access Type 3 (eDMA3) is an AHB-based 32-bit-addressing-capable eDMA module. For the rest of the chapter, eDMA refers to the eDMA3 type. There is one instance of eDMA3 in the AON domain. This is called the eDMA3_1 instance.

4.2 Overview

The enhanced direct memory access (eDMA) controller is capable of performing complex data transfers with minimal intervention from a host processor. The hardware microarchitecture includes:

- A DMA engine that performs:
  - Source address and destination address calculations
  - Data-movement operations
- Local memory containing transfer control descriptors for each of the 31 channels

NOTE

The terminology in this chapter has been updated to align with NXP's inclusive language standards, as shown in the table below.

Table 10. Updated terms

Updated term	Deprecated term
Initiator	Master
Target	Slave

4.2.1 Block diagram

Figure 3 illustrates the components of the eDMA system, including the eDMA module (engine).

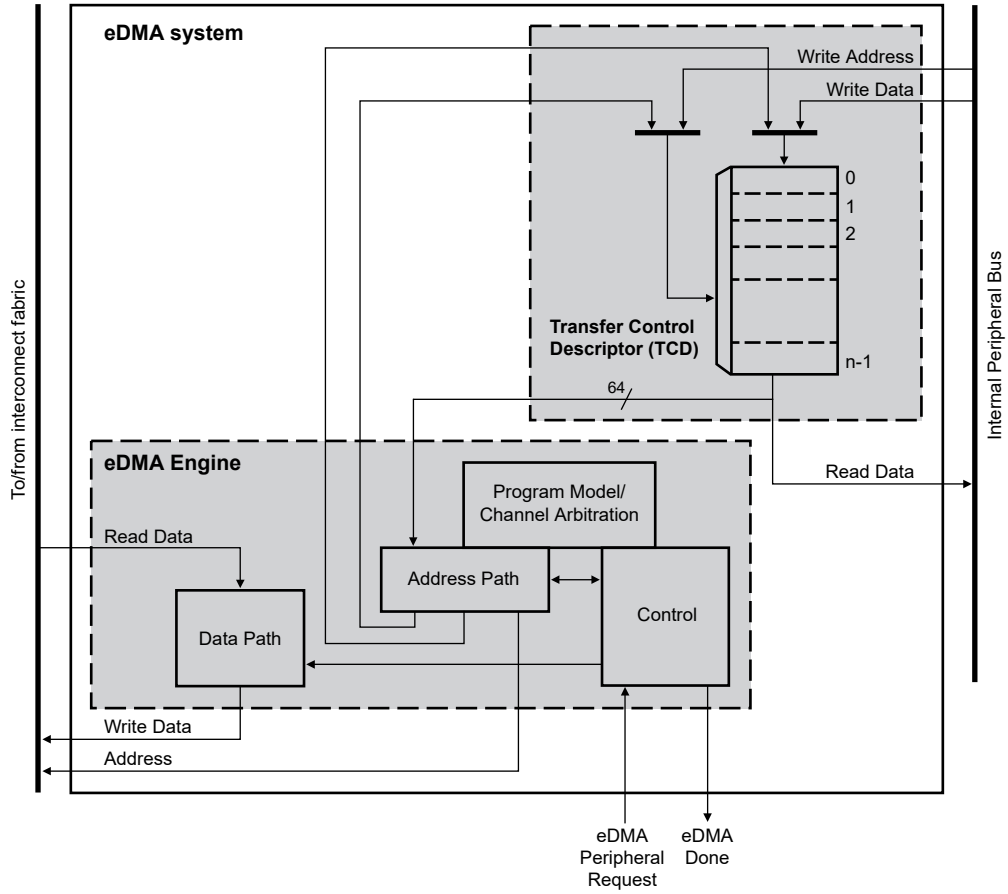


Figure 3. Block diagram

4.2.2 Block parts

The eDMA module is partitioned into two major modules: the eDMA engine and the transfer control descriptor local memory.

The eDMA engine is further partitioned into four submodules:

Table 11. eDMA engine submodules

Submodule	Function
Address path	This block: • Implements a primary channel and secondary (preempt) channel • Manages all initiator bus-address calculations All the channels provide the same functionality. This structure allows data transfers associated with one channel to be preempted after the completion of a read/write sequence if a higher priority channel activation is asserted while the primary channel is active. After a channel is activated, it runs until the minor loop is completed, unless preempted by a higher priority channel. This provides a mechanism (enabled by CH*n*_PRI[ECP]) where a large data transfer can be preempted to minimize the time another channel is blocked from execution.

Table continues on the next page...

Table 11. eDMA engine submodules (continued)

Submodule	Function
	When any channel is selected to execute, the contents of its TCD are read from local memory and loaded into the address path channel x registers for a normal start and into channel y registers for a preemption start. After the minor loop completes execution, the address path hardware writes the new values for the TCD*n*_{SADDR, DADDR, CITER} back to local memory. If the major iteration count is exhausted, additional processing is performed, including the final address pointer updates, reloading the TCD*n*_CITER field, and a possible fetch of a new TCD*n* from memory as part of a scatter/gather operation. See Dynamic scatter/gather for more details.
Data path	This block implements the bus initiator read/write data path. It includes a data buffer and the necessary multiplex logic to support any required data alignment. The internal read data bus is the primary input, and the internal write data bus is the primary output. The address and data path modules directly support the 2-stage pipelined internal bus. The address path module represents the first stage of the bus pipeline (address phase), and the data path module implements the second stage of the pipeline (data phase).
Program model/channel arbitration	This block implements the first section of the eDMA programming model as well as the channel arbitration logic. The programming model registers are connected to the internal peripheral bus. The eDMA peripheral request inputs and interrupt request outputs are also connected to this block (via control logic).
Control	This block provides all the control functions for the eDMA engine. For data transfers where the source and destination sizes are equal, the eDMA engine performs a series of source read/destination write operations until the number of bytes specified in the minor loop byte count has been moved from the source to the destination. For descriptors where the sizes are not equal, multiple accesses of the smaller size data are required for each reference of the larger size. As an example, if the source size references 16-bit data and the destination is 32-bit data, the eDMA performs two reads, then one 32-bit write.

The transfer control descriptor local memory is further partitioned into:

Table 12. Transfer control descriptor memory

Submodule	Description
Memory controller	This logic implements the required dual-ported controller, and manages accesses from the eDMA engine as well as references from the internal peripheral bus. In simultaneous accesses, the eDMA engine is given priority and the peripheral transaction is stalled.
Memory array	TCD storage for each channel's transfer profile.

4.2.3 Features

The eDMA is a highly programmable data-transfer engine optimized to minimize any required intervention from the host processor. It is intended for use in applications where the data size to be transferred is statically known and is not defined within the transferred data itself. The eDMA module features:

- All data movement via dual-address transfers: read from source, write to destination
  - Programmable source and destination addresses and transfer size
  - Support for complex address calculations
- 31-channel implementation that performs complex data transfers with minimal intervention from a host processor
  - Internal data buffer, used as temporary storage for all transfers

  - Connections to the crossbar switch for bus driving the data movement
- TCD organized to support two-deep, nested transfer operations
  - 32-byte TCD stored in local memory for each channel
  - An inner data transfer loop defined by a minor byte transfer count
  - An outer data transfer loop defined by a major iteration count
- Channel activation via one of three methods:
  - Explicit software initiation
  - Initiation via a channel-to-channel linking mechanism for continuous transfers
  - Peripheral-paced hardware requests, one per channel
- Fixed-priority and round-robin channel arbitration
- Channel completion reported via programmable interrupt requests
  - One interrupt per channel, which can be asserted at completion of major iteration count
  - Programmable error terminations per channel that are logically summed together to form one error interrupt to the interrupt controller
- Programmable support for scatter/gather DMA processing
- Support for complex data structures

In the discussion of this module, *n* is used to reference the channel number.

4.3 Functional description

The operation of eDMA is described in the following subsections.

4.3.1 Modes of operation

The eDMA operates in the following modes:

Table 13. Modes of operation

Mode	Description
Normal	In Normal mode, eDMA transfers data between a source and a destination. The source and destination can be a memory block or an I/O block capable of operation with eDMA. A service request initiates a transfer of a specific number of bytes (NBYTES) as specified in the TCD. The minor loop is the sequence of read-write operations that transfers these NBYTES per service request. Each service request executes one iteration of the major loop, which transfers NBYTES of data.
Debug	eDMA operation is configurable in Debug mode via the control register: • If CSR[EDBG] is cleared to 0, eDMA continues to operate. • If CSR[EDBG] is set to 1, eDMA stops transferring data. If Debug mode is entered when a channel is active, eDMA continues operation until the channel retires.

4.3.2 eDMA basic data flow

The basic flow of a data transfer can be partitioned into three segments.

As shown in the following diagram, the first segment involves the channel activation:

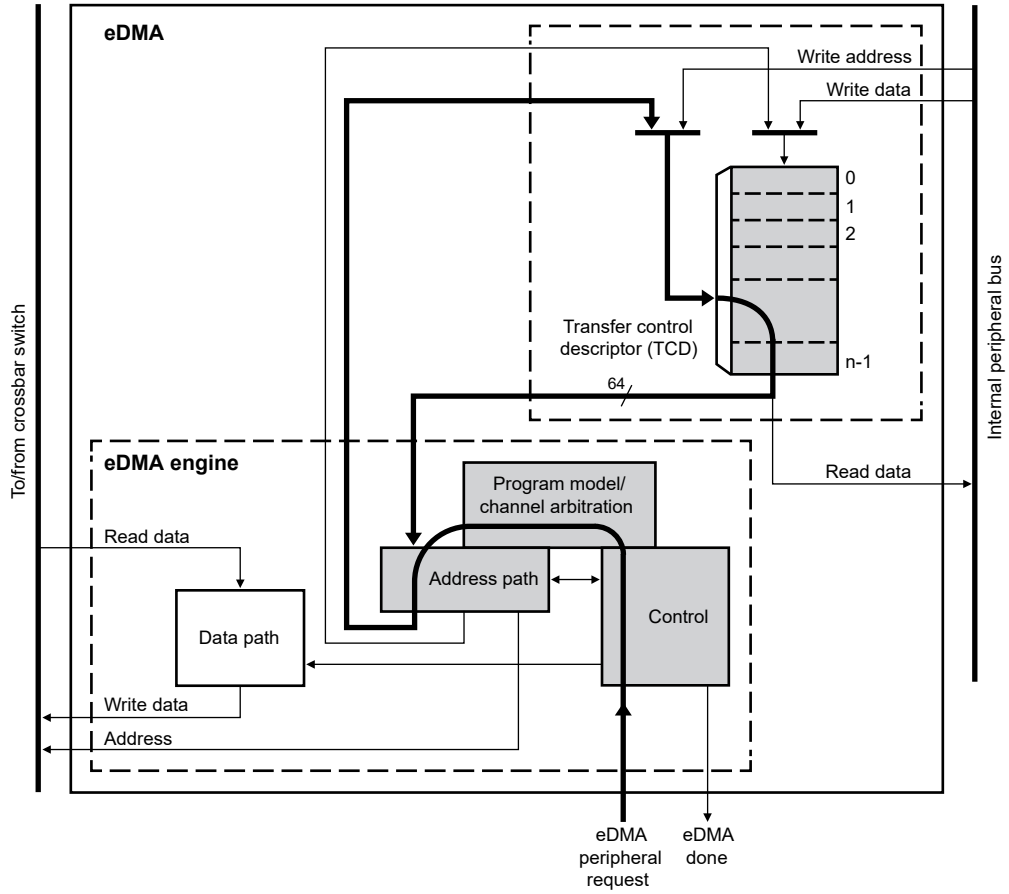


Figure 4. eDMA operation, part 1

This example uses the assertion of the eDMA peripheral request signal to request service for channel *n*. Channel activation via software and the TCD*n*_CSR[START] field follows the same basic flow as peripheral requests. The eDMA request input signal is registered internally and then routed through the eDMA engine: first through the control module, then into the program model and channel arbitration.

In the next cycle, the channel arbitration begins using fixed-priority plus the optional round-robin algorithm. After arbitration is complete, the activated channel number is sent through the address path and converted into the required address to access the local memory for TCD*n*. Next, the TCD memory is accessed and the required descriptor is read from the local memory and then loaded into the eDMA engine address path's primary or secondary channel execution registers. The TCD memory is 64 bits wide to minimize the time needed to fetch the activated channel descriptor and load it into the address path registers.

The following diagram illustrates the second part of the basic data flow:

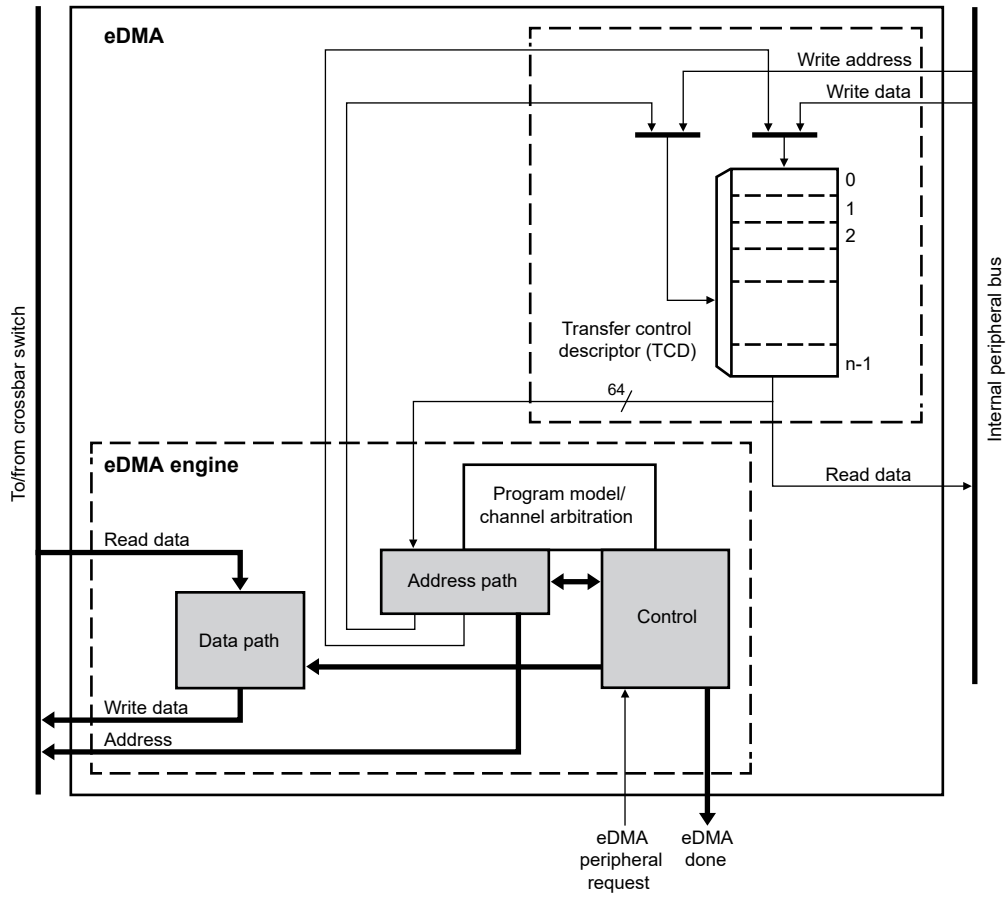


Figure 5. eDMA operation, part 2

The modules associated with the data transfer (address path, data path, and control) go through the required sequence of source reads and destination writes to perform the actual data movement. The source reads are initiated, and the fetched data is temporarily stored in the data path block until it is gated onto the internal bus during the destination write. This source read/destination write processing continues until the byte count, NBYTES, has been transferred.

After NBYTES of data has been moved, the final phase of the basic data flow is performed. In this segment, the address path logic performs the required updates to certain fields in the appropriate TCD (for example, SADDR, DADDR, CITER). If the major iteration count is exhausted, additional operations are performed. These include the final address adjustments and reloading of the BITER field into the CITER field. Assertion of an optional interrupt request also occurs at this time, as does a possible fetch of a new TCD from memory using the scatter/gather address pointer included in the descriptor (if scatter/gather is enabled). The updates to the TCD memory and the assertion of an interrupt request are shown in the following diagram.

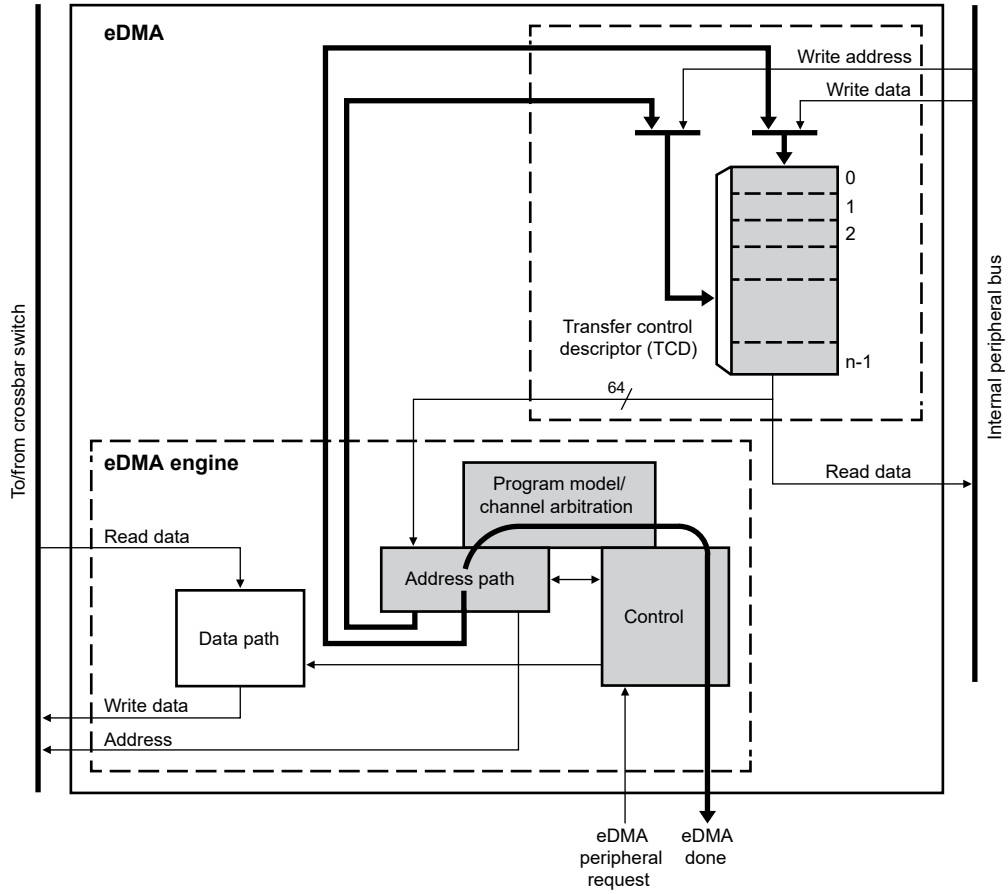


Figure 6. eDMA operation, part 3

4.3.3 Fault reporting and handling

Channel errors are reported in the Error Status register (CHn_ES) and can be caused by any of the following:

- A configuration error, which is an illegal setting in the transfer control descriptor
- An active channel canceled via a "cancel transfer with error" hardware or software request
- An error termination to a bus initiator read or write cycle

A configuration error is reported when an inconsistent state is represented by one of these factors:

- Starting source or destination address
- Source or destination offsets
- Minor loop byte count
- Transfer size

Each of these possible causes is detailed below:

- The addresses and offsets must be aligned on zero-modulo-transfer-sized boundaries.
- The minor loop byte count must be a multiple of the source and destination transfer sizes.
- All source reads and destination writes must be configured to the natural boundary of the programmed transfer size.

NOTE

To aid in debugging, set the Halt After Error field in the DMA's Control Status register, CSR[HAE]. Upon any error condition, the DMA is halted after the error is recorded. The DMA remains halted and does not process any channel service requests. After the error is fixed, the DMA may be enabled again by clearing the Halt field, CSR[HALT].

- If a scatter/gather operation is enabled upon channel completion, a configuration error is reported if the scatter/gather address (TCDn_DLAST_SGA) is not aligned on a 32-byte boundary.
- If minor loop channel linking is enabled upon channel completion, a configuration error is reported when the link is attempted if the TCDn_CITER[ELINK] field does not equal the TCDn_BITER[ELINK] field.

If enabled, all configuration error conditions, except the scatter/gather and minor-loop link errors, are reported as the channel activates and asserts an error interrupt request. A scatter/gather configuration error is reported when the scatter/gather operation begins at major loop completion if properly enabled. A minor loop channel link configuration error is reported when the link operation is serviced at minor loop completion.

If a system bus read or write is terminated with an error, the data transfer is stopped and the appropriate bus error flag set. In this case, the state of the channel's transfer control descriptor is updated by the eDMA engine with the current source address, destination address, and current iteration count at the point of the fault. When a system bus error occurs, the channel terminates after the next transfer. Due to pipeline effect, the next transfer is already in progress when the bus error is received by the eDMA. If a bus error occurs on the last read prior to beginning the write sequence, the write executes using the data captured during the bus error. If a bus error occurs on the last write prior to switching to the next read sequence, the read sequence executes before the channel terminates due to the destination bus error.

The occurrence of any error causes the eDMA engine to stop normal processing of the active channel immediately (it goes to its error processing states and the transaction to the system bus still has pipeline effect), and the appropriate channel field in the eDMA error register is set to 1. At the same time, the details of the error condition are loaded into the Error Status register (CHn_ES). The major loop complete indicators, setting the transfer control descriptor DONE flag, and the possible assertion of an interrupt request are not affected when an error is detected.

After the error status has been updated, the eDMA engine continues operating by servicing the next appropriate channel. A channel that experiences an error condition is not automatically disabled. If a channel is terminated by an error and then issues another service request before the error is fixed, that channel executes and terminates with the same error condition.

The error status fields are read-only. These error indicators are sticky and cannot be cleared. They show the last recorded error until the DMA is reset. CHn_ES[ERR] is used to determine if a new error condition exists. This field is the logical OR of each channel's error interrupt field (ERR).

After the software has resolved all errors and cleared all of the error interrupt fields, the MP_ES[VLD] is cleared to 0 but the cause of the last error is still indicated.

4.3.4 Channel preemption

The eDMA uses a priority vector value to determine the highest priority channel requesting service.

The priority vector is a combination of:

1. the channel's group priority, CHn_GRPRI
2. the channel's priority level, CHn_PRI[APL]

It can be considered a number composed of these concatenated priority levels: CHn_GRPRI : CHn_PRI[APL]

Priority vector = ((CHn_GRPRI << 8) + (CHn_PRI[APL] << 5) + CH_NUM)

A channel requesting service with the highest priority vector value will receive the next execution slot.

An execution slot is available:

1. immediately if the eDMA is idle
2. when an active channel retires
3. when valid preemption conditions exist

NOTE

Preemption is strictly priority based. Preemption is not bound by a specific group number as defined by CHn_GRPRI.

Channel preemption is enabled on a per-channel basis by setting the CHn_PRI[ECP] field. Channel preemption allows the executing channel's data transfers to temporarily suspend in favor of starting a higher-priority channel. After the preempting channel has completed all of its minor loop data transfers, the preempted channel is restored and resumes execution.

After the restored channel completes one read/write sequence, it is again eligible for preemption. If any higher priority channel is requesting service, the restored channel is suspended, and the higher-priority channel is serviced. Nested preemption, that is, attempting to preempt a preempting channel, is not supported. After a preempting channel begins execution, it cannot be preempted.

A channel's ability to preempt another channel can be disabled by setting CHn_PRI[DPA] to 1. When a channel's preempt ability is disabled, that channel cannot suspend a lower-priority channel's data transfer, regardless of the lower-priority channel's ECP setting. This allows for a pool of low-priority, large-data-moving channels to be defined.

You can configure these low-priority channels to not preempt each other, thus preventing a low-priority channel from consuming the preempt slot normally available to a true high-priority channel. When you enable round-robin channel arbitration mode (CSR[ERCA] is set to 1), any channel with a priority level equal to 0 (CHn_PRI[APL] = 0) has preemption disabled and cannot preempt another channel.

4.3.5 Clocking

The following table describes the clock sources:

Table 14. Clocks

Clock	Description
hclk	Clock used for the DMA engine
ipg_clk	Clock used for the DMA programming model
dma_ram_clk	Clock used for the TCD RAMs

4.3.6 Interrupts

Software can enable the interrupt for each channel for the following events:

1. The major loop is half complete (INTHALF)
2. The major loop is complete (INTMAJOR)
3. A configuration error occurs (EEI)

4.4 External signals

This module has no external signals.

4.5 Initialization

The following sections discuss initialization of the eDMA and programming considerations.

4.5.1 eDMA initialization

To initialize the eDMA:

1. Write to the MP_CSR if a configuration other than the default is wanted.
2. Write the channel priority levels to the CHn_PRI registers and group priority levels to the CHn_GRPRI registers if a configuration other than the default is wanted.

3. Enable error interrupts in the CHn_CSR[EEI] registers if they are wanted.
4. Write the 32-byte TCD for each channel that may request service.
5. Enable any hardware service requests via the CHn_CSR[ERQ] registers.
6. Request channel service via either:
   - Software: setting TCDn_CSR[START]
   - Hardware: target device asserting its eDMA peripheral request signal

After any channel requests service, a channel is selected for execution based on the arbitration and priority levels written into the programmer's model. The eDMA engine reads the entire TCD, including the TCD control and status fields, as shown in Table 15, for the selected channel into its internal address path module.

As the TCD is read, the first transfer is initiated on the internal bus, unless a configuration error is detected. Transfers from the source, defined by TCDn_SADDR, to the destination, defined by TCD_DADDR, continue until the number of bytes specified by TCDn_NBYTES are transferred.

When the transfer is complete, the eDMA engine's local TCDn_SADDR, TCDn_DADDR, and TCDn_CITER are written back to the main TCD memory and any minor loop channel linking is performed, if enabled. If the major loop is exhausted, then eDMA executes further post-processing, such as interrupts, major loop channel linking, and scatter/gather operations, if enabled.

Table 15. TCD control and status (TCDn_CSR) fields

TCDn_CSR field name	Description
START	Control field to start the channel explicitly when using a software-initiated DMA service (automatically cleared by hardware)
EEOP	Control field to enable end-of-packet processing
ESDA	Control field to enable storing of the destination address to system memory after the major loop completes
DREQ	Control field to disable hardware-initiated DMA service requests after major loop completion
BWC	Control field for throttling the bandwidth control of a channel
ESG	Control field to enable the scatter-gather feature
INTHALF	Control field to enable interrupt when major loop is half-complete
INTMAJOR	Control field to enable interrupt when major loop completes

Table 16. Channel control and status (CHn_CSR) fields

CHn_CSR field name	Description
ACTIVE	Status field indicating the channel is currently in execution
DONE	Status field indicating major loop completion (cleared by software when a channel begins execution)
EEI	Control field to enable error interrupts
EARQ	Control field to enable external, asynchronous wakeup event in conjunction with the ERQ field
ERQ	Control field to enable hardware service requests

The following figure shows how each DMA request initiates one minor-loop transfer, or iteration, without CPU intervention. DMA arbitration can occur after each minor loop, and one level of minor loop DMA preemption is allowed. The number of minor loops in a major loop is specified by the beginning iteration count (BITER).

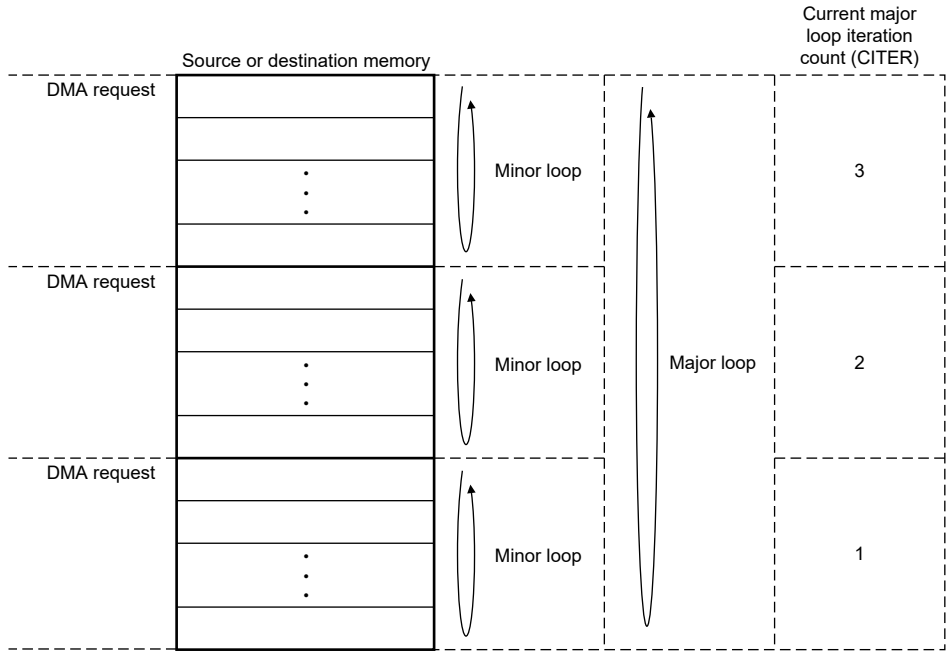


Figure 7. Example of multiple loop iterations

The following figure lists the memory array terms and how the TCD settings are related.

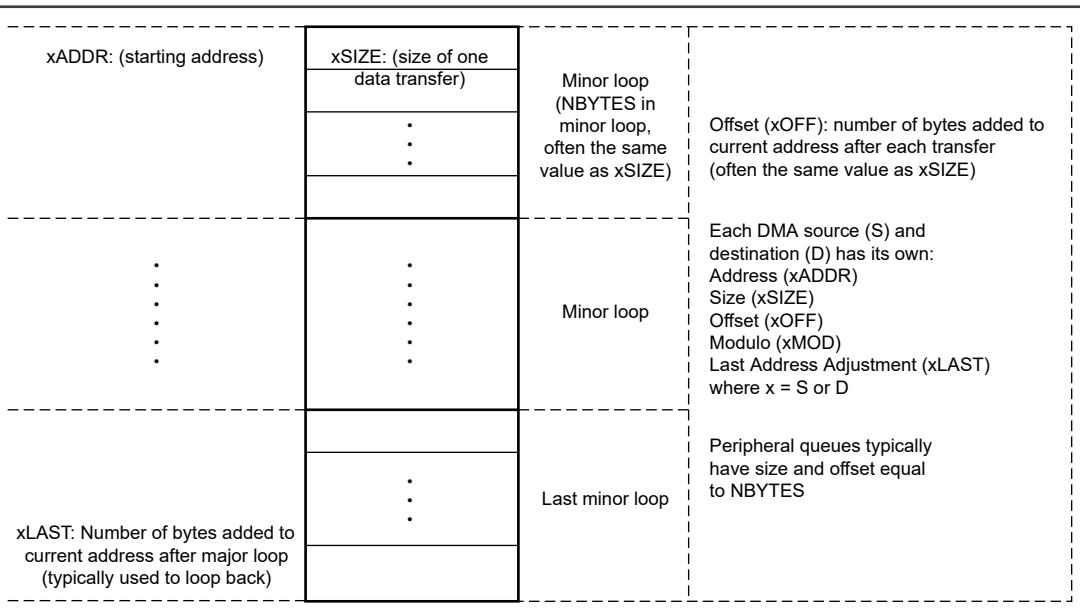


Figure 8. Memory array terms

4.5.2 eDMA arbitration

The eDMA uses a layered arbitration scheme composed of multiple priority levels. The eDMA uses a fixed-priority arbitration scheme with optional round-robin arbitration under specific conditions. The priorities are evaluated in the following order:

Table 17. eDMA arbitration priorities

Priority	Scheme	Description
1 (Highest)	Arbitration group priority	Each channel is assigned an arbitration group via the CHn_GRPRI registers. Priority is given to the highest value (31 being the highest possible value) down to the lowest value (zero, the default).
2	Channel priority	Each channel is assigned a channel priority level via the CHn_PRI registers. The channel priority is a relative priority level within an arbitration group. Priority is given to the highest value (seven being the highest possible value) down to the lowest value (zero, the default). Channel priorities within each arbitration group need not be unique. If multiple channels have the same channel priority level, the channel number will be used to determine priority as defined in row three.
3	Channel number	When two or more channels have the same arbitration group priority and channel priority, the channel number (CHn_NUM) is used to determine the highest priority. Priority is given to the highest channel number. Lowest priority is channel 0. The channel numbers are static and cannot be changed in the programmer's model.
4 (Lowest)	Round-robin	When round-robin is enabled, any channel configured for round-robin operation has lowest priority within an arbitration group. Round-robin is enabled by setting the MP_CSR[ERCA] field to 1. After being enabled, channels with a channel priority of zero (CHn_PRI=0) will use round-robin arbitration. Round-robin arbitration will rotate the channel selection among the channels requesting service with CHn_PRI=0 within the arbitration group. Any non-zero channel within the arbitration group will continue to use fixed-priority arbitration, and if requesting service will be selected over any round-robin channels.

For fixed arbitration, the overall priority can be considered a number composed of three concatenated priority levels: CHn_GRPRI:CHn_PRI:CH_NUM. The largest number has the highest priority and the lowest number has the lowest priority.

For round-robin arbitration, the priority number is CHn_GRPRI:0:X. The module rotates through the CHn_PRI=0 channels requesting service without regard to priority among these channels. Any channel within the arbitration group for which CHn_PRI is greater than 0 will be serviced before the round-robin channels.

4.5.3 Programming errors

The eDMA performs various tests on the transfer control descriptor to verify consistency in the descriptor data.

The channel number causing the error is recorded in the Error Status register (CHn_ES). If the error source is not removed before the next activation of the problematic channel, the error is detected and recorded again. Setting the halt after error field, CSR[HAE], will halt the DMA and prevent recurrence of the error.

4.5.4 Arbitration mode considerations

This section discusses arbitration considerations for eDMA.

4.5.4.1 Fixed group arbitration, fixed channel arbitration

In this mode, eDMA selects for execution the channel service request from the highest-priority channel in the highest-priority group. If eDMA is programmed so that the channels within a high-priority group have a high number of requests or large data transfers, that group may consume all the bandwidth of the eDMA controller. That is, no lower-priority groups are serviced if there is always at least one DMA request pending on a channel in the highest-priority group when the controller arbitrates the next DMA request. The advantage of this scenario is that latency can be small for channels that need to be serviced quickly.

4.5.4.2 Fixed group arbitration, round-robin channel arbitration

The highest-priority group with a request is serviced. Lower-priority groups are serviced if no pending requests exist in the higher-priority groups.

Within each group, channels are serviced starting with the highest non-zero channel priority. For all channels with a channel priority programmed to 0, selection begins with the highest channel number requesting service and then rotates through to the lowest channel number requesting service. The round-robin channel arbitration can provide a fairness mechanism to lower-priority channels.

This scenario could cause the same bandwidth consumption problem as indicated in Fixed group arbitration, fixed channel arbitration, but all the channels in the highest-priority group will be serviced. Service latency is short on the highest-priority group, but could potentially be very much longer as the group priority decreases.

4.5.5 Performing DMA transfers

This section presents examples on how to perform DMA transfers with the eDMA.

4.5.5.1 Single request

To perform a simple transfer of n bytes of data with one activation, set the major loop to one (TCD*n*_CITER = TCD*n*_BITER = 1). The data transfer begins after the channel service request is acknowledged and the channel is selected to execute. After the transfer is complete, the CH*n*_CSR[DONE] field is set to 1 and an interrupt is generated if properly enabled.

For example, the following TCD entry is configured to transfer 16 bytes of data. The eDMA is programmed for one iteration of the major loop transferring 16 bytes per iteration. The source memory has a byte-wide memory port located at 0x1000. The destination memory has a 32-bit port located at 0x2000. The address offsets are programmed in increments to match the transfer size: one byte for the source, and four bytes for the destination. The final source and destination addresses are adjusted to return to their beginning values.

```
TCDn_CITER = TCDn_BITER = 1
TCDn_NBYTES = 16
TCDn_SADDR = 0x1000
TCDn_SOFF = 1
TCDn_ATTR[SSIZE] = 0
```

```
TCDn_SLAST = -16
TCDn_DADDR = 0x2000
TCDn_DOFF = 4
TCDn_ATTR[DSIZE] = 2
TCDn_DLAST_SGA= -16
TCDn_CSR[INTMAJ] = 1
TCDn_CSR[START] = 1 (should be written last after all other fields have been initialized)
All other TCDn fields = 0
```

This generates the following event sequence:

1. User write to the TCD*n*_CSR[START] field requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes:
   - CH*n*_CSR[DONE] = 0
   - TCD*n*_CSR[START] = 0
   - CH*n*_CSR[ACTIVE] = 1
4. eDMA engine reads: channel TCD data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
   a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
   b. Write 32 bits to location 0x2000 → first iteration of the minor loop.
   c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
   d. Write 32 bits to location 0x2004 → second iteration of the minor loop.
   e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
   f. Write 32 bits to location 0x2008 → third iteration of the minor loop.
   g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
   h. Write 32 bits to location 0x200C → last iteration of the minor loop → major loop complete.
6. The eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 1 (TCD*n*_BITER).
7. The eDMA engine writes: CH*n*_CSR[ACTIVE] = 0, CH*n*_CSR[DONE] = 1, CH*n*_INT[INT] = 1.
8. The channel retires and the eDMA goes idle or services the next channel.

4.5.5.2 Multiple requests

The following example transfers 32 bytes via two hardware requests, but is otherwise the same as the previous example. The only fields that change are the major loop iteration count and the final address offsets. The eDMA is programmed for two iterations of the major loop, transferring 16 bytes per iteration. After the channel's hardware requests are enabled via the CHn_CSR[ERQ] register field, the target device initiates channel service requests.

```
TCDn_CITER = TCDn_BITER = 2
TCDn_SLAST = -32
TCDn_DLAST_SGA = -32
```

This would generate the following sequence of events:

1. First hardware (eDMA peripheral) requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: CH*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, CH*n*_CSR[ACTIVE] = 1.

4. eDMA engine reads: channel TCD*n* data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
    a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
    b. Write 32 bits to location 0x2000 → first iteration of the minor loop.
    c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
    d. Write 32 bits to location 0x2004 → second iteration of the minor loop.
    e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
    f. Write 32 bits to location 0x2008 → third iteration of the minor loop.
    g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
    h. Write 32 bits to location 0x200C → last iteration of the minor loop.
6. eDMA engine writes: TCD*n*_SADDR = 0x1010, TCD*n*_DADDR = 0x2010, TCD*n*_CITER = 1.
7. eDMA engine writes: CH*n*_CSR[ACTIVE] = 0.
8. The channel retires, which concludes one iteration of the major loop. The eDMA goes idle or services the next channel.
9. Second hardware (eDMA peripheral) requests channel service.
10. The channel is selected by arbitration for servicing.
11. eDMA engine writes: CH*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, CH*n*_CSR[ACTIVE] = 1.
12. eDMA engine reads: Channel TCD data from local memory to internal register file.
13. The source-to-destination transfers are executed as follows:
    a. Read byte from location 0x1010, read byte from location 0x1011, read byte from 0x1012, read byte from 0x1013.
    b. Write 32 bits to location 0x2010 → first iteration of the minor loop.
    c. Read byte from location 0x1014, read byte from location 0x1015, read byte from 0x1016, read byte from 0x1017.
    d. Write 32 bits to location 0x2014 → second iteration of the minor loop.
    e. Read byte from location 0x1018, read byte from location 0x1019, read byte from 0x101A, read byte from 0x101B.
    f. Write 32 bits to location 0x2018 → third iteration of the minor loop.
    g. Read byte from location 0x101C, read byte from location 0x101D, read byte from 0x101E, read byte from 0x101F.
    h. Write 32 bits to location 0x201C → last iteration of the minor loop → major loop complete.
14. eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 2 (TCD*n*_BITER).
15. eDMA engine writes: CH*n*_CSR[ACTIVE] = 0, CH*n*_CSR[DONE] = 1, CH*n*_INT[INT] = 1.
16. The channel retires, which concludes with the major loop complete. The eDMA goes idle or services the next channel.

4.5.5.3 Using the modulo feature

The modulo feature of the eDMA allows implementation of a circular data queue in which the size of the queue is a power of 2. xMOD is a 5-bit field for the source and destination in the TCD, and it specifies which lower address bits increment from their original value after the address+offset calculation. All upper address bits remain the same as in the original value. A setting of 0 for this field disables the modulo feature. Modulo addressing applies to cases where the minor loop offset is enabled; that is, the upper address bits remain the same after the minor loop offset is added to the source or destination address.

The following table shows how the transfer addresses are specified based on the setting of the MOD field. Here a circular buffer is created where the address wraps to the original value but the 28 upper address bits (0x1234567*x*) retain their original value. In this example, the source address is set to 0x12345670, the offset is set to four bytes, and the MOD field is set to four, which allows for a 2^4 byte (16 byte) queue size.

Table 18. Modulo example

Transfer number	Address
1	0x12345670
2	0x12345674
3	0x12345678
4	0x1234567C
5	0x12345670
6	0x12345674

4.5.6 Monitoring transfer descriptor status

This section discusses how to monitor eDMA status.

4.5.6.1 Testing for minor loop completion

There are two methods to test for minor loop completion when using software-initiated service requests.

1. The first method is to read the TCDn_CITER field and test for a change.
2. The second method, extracted from the sequence shown below, is to test the TCDn_CSR[START] field and the CHn_CSR[ACTIVE] field. The minor-loop-complete condition is indicated by both fields reading 0 after TCDn_CSR[START] is set to 1. Polling the CHn_CSR[ACTIVE] field only may be inconclusive because the active status may be missed if the channel execution is short in duration.

The CHn_CSR and TCDn_CSR status fields execute the following sequence for a software-activated channel:

Stage	TCDn_CSR field	CHn_CSR fields		State
	START	ACTIVE	DONE	
1	1	0	0	Initiate channel service request via software.
2	0	1	0	Channel is executing.
3a	0	0	0	Channel has completed the minor loop and is idle.
3b	0	0	1	Channel has completed the major loop and is idle.

The best method to test for minor-loop completion when using hardware-initiated (that is, peripheral-initiated) service requests is to read the TCDn_CITER field and test for a change. The hardware request and acknowledge handshake signals are not visible in the programmer's model.

The TCD status fields execute the following sequence for a hardware-activated channel:

Stage	TCDn_CSR field	CHn_CSR fields		State
	START	ACTIVE	DONE	
1	0	0	0	Initiate channel service request via hardware (peripheral request asserted).
2	0	1	0	Channel is executing.
3a	0	0	0	Channel has completed the minor loop and is idle.
3b	0	0	1	Channel has completed the major loop and is idle.

For both activation types, the major-loop-complete status is explicitly indicated via the CHn_CSR[DONE] field.

The TCDn_CSR[START] field is cleared to 0 automatically when the channel begins execution, regardless of how the channel activates.

4.5.6.2 Reading the transfer descriptors of active channels

The eDMA reads back the true TCDn_SADDR, TCDn_DADDR, and TCDn_NBYTES values if they are read when a channel executes. The true values of SADDR, DADDR, and NBYTES are the values the eDMA engine currently uses in its internal register file, and not the values in the TCD local memory for that channel. The addresses, SADDR and DADDR, and NBYTES (which decrements to zero as the transfer progresses), can give an indication of the progress of the transfer. All other values are read back from the TCD local memory.

4.5.6.3 Checking channel preemption status

A preemptive situation is one in which a preempt-enabled channel is executing and a higher-priority request becomes active. When round-robin channel arbitration mode is enabled, all channels with their channel priority set to 0 lose their preempt ability. Channel priorities of 0 are treated as equal, that is, they are constantly rotating, when round-robin arbitration mode is enabled.

The CHn_CSR[ACTIVE] field for the preempted channel remains asserted throughout the preemption. The preempted channel is temporarily suspended when the preempting channel executes one major loop iteration. If two CHn_CSR[ACTIVE] fields are set simultaneously in the global TCD map, a higher-priority channel is actively preempting a lower-priority channel.

4.5.7 Channel linking

Channel linking (or chaining) is a mechanism in which one channel sets the TCDn_CSR[START] field of another channel (or itself), thus initiating a service request for that channel. When properly enabled, the eDMA engine automatically performs this operation at the major or minor loop completion.

The minor loop channel linking occurs at the completion of the minor loop (or one iteration of the major loop). The TCDn_CITER[ELINK] field determines whether a minor loop link is requested. When enabled, the channel link is made after each iteration of the major loop except for the last. When the major loop is exhausted, only the major loop channel link fields are used to determine if a channel link should be made. For example, using an initial field setting of:

```
MP_CSR[GCLC] = 1
TCDn_CITER[ELINK] = 1
TCDn_CITER[LINKCH] = 0xC
TCDn_CITER[CITER] value = 0x4
TCDn_CSR[MAJORELINK] = 1
TCDn_CSR[MAJORLINKCH] = 0x7
```

executes as:

1. Minor loop done → set TCD12_CSR[START] field
2. Minor loop done → set TCD12_CSR[START] field
3. Minor loop done → set TCD12_CSR[START] field
4. Minor loop done, major loop done→ set TCD7_CSR[START] field

When minor loop linking is enabled (TCDn_CITER[ELINK] = 1), the TCDn_CITER[CITER] field uses a nine-bit vector to form the current iteration count. When minor loop linking is disabled (TCDn_CITER[ELINK] = 0), the TCDn_CITER[CITER] field uses a 15-bit vector to form the current iteration count. The bits associated with the TCDn_CITER[LINKCH] field are concatenated onto the CITER value to increase the range of the CITER.

NOTE

The TCDn_CITER[ELINK] field and the TCDn_BITER[ELINK] field must be equal — if they are not, a configuration error is reported. The CITER and BITER vector widths must be equal to calculate the major loop halfway done interrupt point.

The following table summarizes how a DMA channel can link to another DMA channel, that is, use another channel's TCD, at the end of a loop.

Table 19. Channel linking parameters

Wanted link behavior	TCD control field name	Description
Link at end of minor loop	TCDn_CITER[ELINK]	Enable channel-to-channel linking on minor loop completion (current iteration)
	TCDn_CITER[LINKCH]	Link channel number when linking at end of minor loop (current iteration)
Link at end of major loop	TCDn_CSR[MAJORELINK]	Enable channel-to-channel linking on major loop completion
	TCDn_CSR[MAJORLINKCH]	Link channel number when linking at end of major loop

4.5.8 Dynamic programming

This section provides recommended methods to change the programming model during channel execution.

Steps for channel linking:

1. Set GCLC = 1
2. For linking after major loop complete:
   a. Set MAJORLINKCH = target channel number
   b. Set MAJORELINK = 1, enable the major loop link
   c. Upon major loop completion (CITER decrements to 0), the link will be made.
3. For linking on minor loop completion:
   a. Set LINKCH = target channel number
   b. Set ELINK = 1, enable the minor loop link
   c. Set the beginning iteration count to the same values; LINKCH = target channel and ELINK = 1,
   d. Upon minor loop completion (that is, after each service request except when the major loop is complete) the link will be made.

4.5.8.1 Dynamically changing the channel priority

To change group or channel priority levels:

1. Halt the DMA by writing 1 to the CSR[HALT] field.
2. Change the group or channel priorities as wanted.
3. Enable normal DMA operations by writing 0 to the CSR[HALT] field.

4.5.8.2 Dynamic channel linking

Dynamic channel linking is the process of setting the TCDn_CSR[MAJORELINK] field during channel execution (see the diagram in TCD structure). This field is read from the TCD local memory at the end of channel execution, thus allowing you to enable the feature during channel execution.

Because you are allowed to change the configuration during execution, you need a coherency model. Consider the scenario where you attempt to execute a dynamic channel link by enabling the TCDn_CSR[MAJORELINK] field at the same time the eDMA engine is retiring the channel. TCDn_CSR[MAJORELINK] would be set in the programmer's model, but it would be unclear whether the actual link was made before the channel retired.

We recommend that you use the following coherency model when executing a dynamic channel link request.

1. Write 1 to the TCDn_CSR[MAJORELINK] field.
2. Read back the TCDn_CSR[MAJORELINK] field.
3. Test the TCDn_CSR[MAJORELINK] request status:
   - If TCDn_CSR[MAJORELINK] = 1, the dynamic link attempt was successful.
   - If TCDn_CSR[MAJORELINK] = 0, the attempted dynamic link did not succeed (the channel was already retiring).

For this request, the TCD local memory controller forces the TCDn_CSR[MAJORELINK] field to 0 on any writes to a channel's TCDn_CSR[7:0] after that channel's CHn_CSR[DONE] field is set to 1, indicating the major loop is complete.

NOTE

You must clear the CHn_CSR[DONE] field to 0 before writing to the TCDn_CSR[MAJORELINK] field. The CHn_CSR[DONE] field is cleared to 0 automatically by the eDMA engine after a channel begins execution.

4.5.8.3 Dynamic scatter/gather

Scatter/gather is the process of automatically loading a new TCD into a channel. It allows a DMA channel to use multiple TCDs; this enables a DMA channel to scatter the DMA data to multiple destinations or gather it from multiple sources. When scatter/gather is enabled and the channel has finished its major loop, a new TCD is fetched from system memory and loaded into that channel's descriptor location in the eDMA programmer's model, thus replacing the current descriptor.

Because you are allowed to change the configuration during execution, you need a coherency model. Consider the scenario where you attempt to execute a dynamic scatter/gather operation by enabling the TCDn_CSR[ESG] field at the same time the eDMA engine is retiring the channel. The TCDn_CSR[ESG] field would be set in the programmer's model, but it would be unclear whether the actual scatter/gather request was honored before the channel retired.

Two methods are recommended for executing a dynamic scatter/gather request. Whenever the TCDn_CSR is written, the TCD local memory controller forces the TCDn_CSR[ESG] field to 0 on any writes to a channel's TCDn_CSR[7:0] after that channel's CHn_CSR[DONE] field has been set to 1, indicating the major loop is complete. If attempting to set the ESG, ensure the DONE field is cleared to 0.

NOTE

You must clear the CHn_CSR[DONE] field to 0 before writing the TCDn_CSR[MAJORELINK] or TCDn_CSR[ESG] fields. The CHn_CSR[DONE] field is cleared to 0 automatically by the eDMA engine after a channel begins execution and is set to 1 upon major loop completion.

4.5.8.3.1 Method 1 (channel not using major loop channel linking)

For a channel not using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request.

When the TCDn_CSR[MAJORELINK] field is 0, the TCDn_CSR[MAJORLINKCH] field is not used by the eDMA. In this case, the TCDn_CSR[MAJORLINKCH] bits may be used for other purposes. This method uses the TCDn_CSR[MAJORLINKCH] field as a TCDn_CSR identification (ID).

When the descriptors are built, write a unique TCDn_CSR ID in the TCDn_CSR[MAJORLINKCH] field for each TCDn_CSR associated with a channel using dynamic scatter/gather.

1. Write a 1 to the TCDn_CSR[DREQ] field. Should a dynamic scatter/gather attempt fail, setting the TCDn_CSR[DREQ] field to 1 will prevent future hardware activation of this channel. This stops the channel from executing with a destination address (daddr) that was calculated using a scatter/gather address (written in the next step) instead of a DLAST final offset value.
2. Write the TCDn_DLAST_SGA field with the scatter/gather address.
3. Write a 1 to the TCDn_CSR[ESG] field.
4. Read back the 16-bit TCDn_CSR control/status field.
5. Test the TCDn_CSR[ESG] request status and TCDn_CSR[MAJORLINKCH] value:
   - If ESG = 1, the dynamic scatter/gather attempt was successful.
   - If ESG = 0 and the MAJORLINKCH (ID) did not change, the dynamic scatter/gather attempt was not successful (the channel was already retiring).
   - If ESG = 0 and the MAJORLINKCH (ID) changed, the dynamic scatter/gather attempt was successful (the new TCDn_CSR's ESG value cleared the ESG field to 0).

4.5.8.3.2 Method 2 (channel using major loop channel linking)

For a channel using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request. This method uses the TCDn_DLAST_SGA field as a TCD identification (ID).

1. Write a 1 to the TCDn_CSR[DREQ] field. Should a dynamic scatter/gather attempt fail, setting the DREQ field to 1 will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (DADDR) that was calculated using a scatter/gather address (written in the next step) instead of a DLAST final offset value.
2. Write the TCDn_DLAST_SGA field with the scatter/gather address.
3. Write a 1 to the TCDn_CSR[ESG] field.
4. Read back the TCDn_CSR[ESG] field.
5. Test the TCDn_CSR[ESG] request status:
   - If ESG = 1, the dynamic scatter/gather attempt was successful.
   - If ESG = 0, read the 32-bit TCDn_DLAST_SGA field.
   - If ESG = 0 and the TCDn_DLAST_SGA did not change, the dynamic scatter/gather attempt was not successful (the channel was already retiring).
   - If ESG = 0 and the TCDn_DLAST_SGA changed, the dynamic scatter/gather attempt was successful (the new TCDn_CSR's ESG value cleared the ESG field to 0).

4.5.9 Suspend/resume a DMA channel with active hardware service requests

The DMA allows you to move data from memory or peripheral registers to another location in memory or to peripheral registers without CPU interaction. After the DMA and peripherals are configured and active, it is rare but supported to suspend a peripheral's

service request dynamically. In this scenario, there are certain restrictions to disabling a DMA hardware service request. For coherency, you must follow a specific procedure.

4.5.9.1 Suspend an active DMA channel

To suspend an active DMA channel:

1. Stop the DMA service request at the peripheral first. Confirm it has been disabled by reading back the appropriate register in the peripheral.
2. Check the DMA's Hardware Request Status (MP_HRS) to ensure there is no service request to the DMA channel being suspended. Then disable the hardware service request by clearing the ERQ field to 0 on the appropriate DMA channel.

For example, assume the SPI is set as a initiator for transmitting data via a DMA service request when the TXFIFO has an empty slot. The DMA will transfer the next command and data to the TXFIFO upon the request. If you need to suspend the DMA/SPI transfer loop, perform the following steps:

1. Disable the DMA service request at the SPI.
2. Ensure there is no DMA service request from the SPI by verifying that MP_HRS[HRS] is 0 for the appropriate channel. If no service request is present, disable the DMA channel by clearing the channel's ERQ field to 0. If a service request is present, wait until the request has been processed and the HRS field reads 0.

4.5.9.2 Resume a DMA channel

To resume a DMA channel:

1. Enable the DMA service request on the appropriate channel by setting its ERQ field to 1.
2. Enable the DMA service request at the peripheral.

4.6 Memory map and register definition

The eDMA programming model is partitioned into three parts:

1. The first part defines a number of registers providing overall control functions and is known as the management page.
2. The second part corresponds to the channel (CH) control, status, and configuration.
3. The third part corresponds to the local TCD memory.

TCD memory

Each channel requires a 32-byte transfer control descriptor for defining the data movement operation. Each TCDn definition is presented as 11 registers of 16 or 32 bits. See EDMA_TCD register descriptions for details.

TCD initialization

Prior to activating a channel, you must initialize its TCD with the appropriate transfer profile.

TCD structure

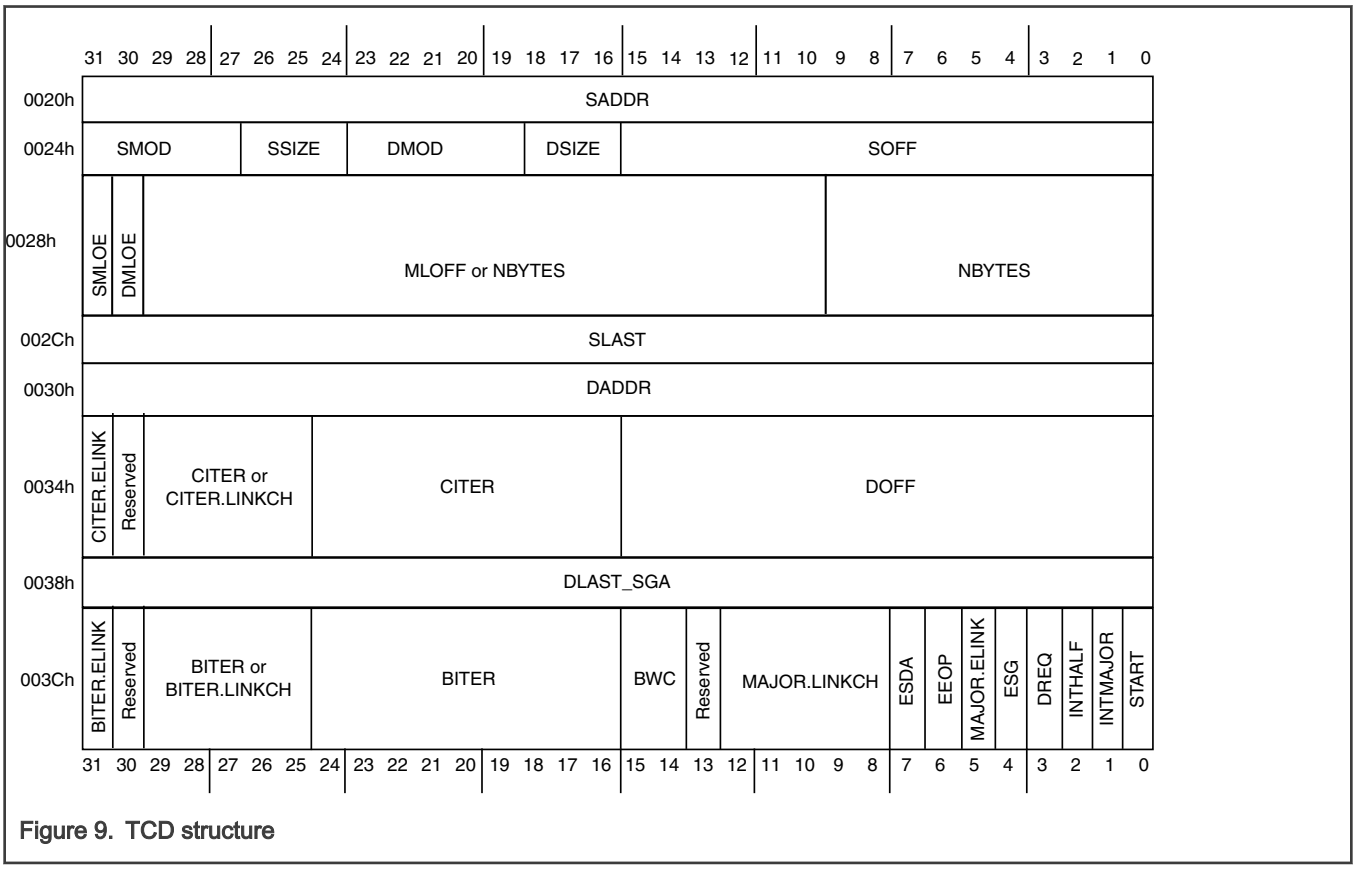


Figure 9. TCD structure

Accesses to reserved memory and fields

- Reading reserved fields in a register returns the value of zero.
- Writes to reserved fields in a register are ignored.
- Reading or writing a reserved memory location generates a bus error.

4.6.1 EDMA_MP register descriptions

4.6.1.1 EDMA_MP memory map

EDMA3_1.MP base address: 4400_0000h

NOTE

For registers in the following table with *Protection*, see the REG_PROT details for more information.

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)	Protection
0	Management Page Control (MP_CSR)	32	RW	0031_0000	Yes
4	Management Page Error Status (MP_ES)	32	R	0000_0000	No
8	Management Page Interrupt Request Status (MP_INT)	32	R	0000_0000	No

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)	Protection
C	Management Page Hardware Request Status (MP_HRS)	32	R	0000_0000	No
100 - 178	Channel Arbitration Group (CH0_GRPRI - CH30_GRPRI)	32	RW	0000_0000	Yes

4.6.1.2 Management Page Control (MP_CSR)

The Management Page Control register defines the basic operating configuration of the DMA.

Arbitration uses a two-tier priority system; group and channel priority. The eDMA assigns each channel to a priority group. Group arbitration is fixed-priority and cannot be changed. Channel arbitration uses fixed priority and may be configured to use a selective round-robin scheme for specified channels within each priority group. For fixed-priority arbitration, eDMA selects for execution the highest priority channel requesting service in the highest priority arbitration group.

The channel priority registers assign the relative priorities within each arbitration group; see CHn_PRI. All channels with a non-zero CHn_PRI value use fixed-priority arbitration.

When you enable round-robin arbitration, all channels with channel priority set to zero do not have a priority and, of those channels requesting service, are cycled through (from high to low channel number) without regard to priority relative to each other within the same priority group. Any channel with a non-zero CHn_PRI value automatically has a higher priority over the round-robin channels. A channel's priority group is assigned in Channel Arbitration Group (CH0_GRPRI - CH30_GRPRI).

NOTE

For correct operation, changes to the MP_CSR[ERCA, GCLC, GMRC] fields must be performed when the DMA channels are inactive; that is, when the MP_CSR[ACTIVE] field is 0.

Offset

Register	Offset
MP_CSR	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ACTIVE	Reserved			ACTIVE_ID				Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						CX	ECX	GMRC	GCLC	HALT	HAE	Reserved	ERCA	EDBG	Reserved
W	0												0			0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 ACTIVE	DMA Active Status 0 - eDMA is idle 1 - eDMA is executing a channel
30-29 —	Reserved
28-24 ACTIVE_ID	Active Channel ID This field identifies the channel number that is executing when the ACTIVE bit is 1.
23-16 —	Reserved
15-10 —	Reserved
9 CX	Cancel Transfer When set to 1, this field cancels the remaining data transfer, stops the executing channel, and forces the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. CX clears itself to 0 after the cancel has been honored. This cancel retires the channel normally as if the minor loop had been completed. 0 - Normal operation 1 - Cancel the remaining data transfer
8 ECX	Cancel Transfer With Error Cancellation of the remaining data transfer is similar to that of the CX field. Execution of the channel is stopped and the minor loop is forced to finish. The cancellation takes effect after the last write of the current read/write sequence. The ECX field clears itself to 0 after the cancel is honored. In addition to cancelling the transfer, ECX treats the cancel as an error condition, thus updating Management Page Error Status (MP_ES) and generating an optional error interrupt. 0 - Normal operation 1 - Cancel the remaining data transfer
7 GMRC	Global Initiator ID Replication Control **NOTE** If initiator ID replication is disabled, the nonsecure, privileged protection level for DMA transfers is used. 0 - Initiator ID replication disabled for all channels 1 - Initiator ID replication available and controlled by each channel's CHn_SBR[EMI] setting
6	Global Channel Linking Control 0 - Channel linking disabled for all channels

Table continues on the next page...

Table continued from the previous page...

Field	Description
GCLC	1 - Channel linking available and controlled by each channel's link settings
5 HALT	Halt DMA Operations This field stalls the start of any new channels. Executing channels are allowed to complete. Channel execution resumes when this field is cleared to 0. 0 - Normal operation 1 - Stall the start of any new channels
4 HAE	Halt After Error When this field is set to 1, any error causes the HALT field to be set to 1. Then all service requests are ignored until the HALT field is cleared to 0. 0 - Normal operation 1 - Any error causes the HALT field to be set to 1
3 —	Reserved
2 ERCA	Enable Round Robin Channel Arbitration 0 - Round-robin channel arbitration disabled. Fixed priority arbitration used for channel selection 1 - Round-robin channel arbitration enabled. Round-robin arbitration used for channel selection
1 EDBG	Enable Debug When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. DMA resumes channel execution when the system exits debug mode or clears the EDBG field to 0. 0 - Debug mode disabled. When in debug mode, the DMA continues to operate 1 - Debug mode is enabled. When in debug mode, the DMA stalls the start of a new channel
0 —	Reserved

4.6.1.3 Management Page Error Status (MP_ES)

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

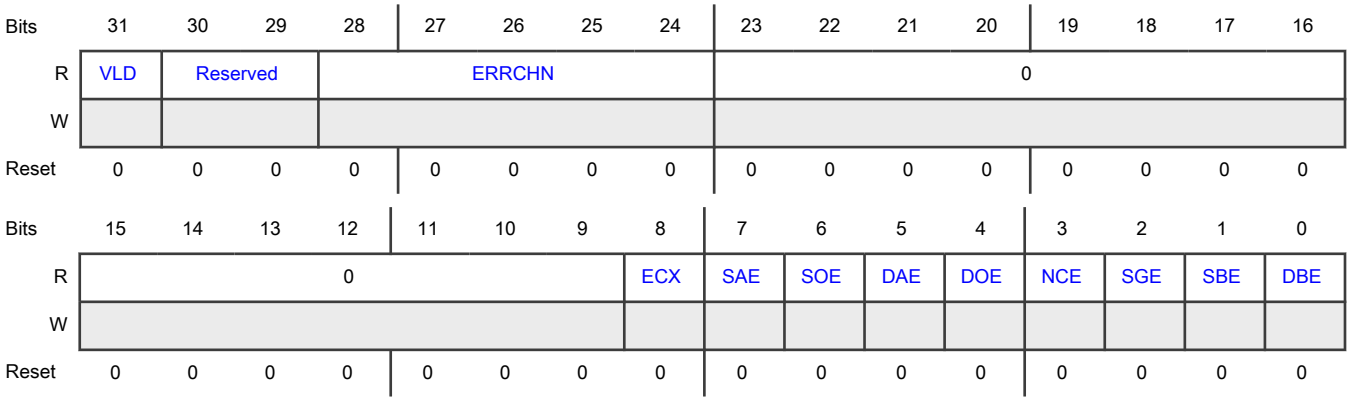
- An illegal setting in the transfer control descriptor
- An error termination to a bus initiator read or write cycle
- An uncorrectable error that occurred when the device was accessing the TCD SRAM
- A "cancel transfer with error" request was made via the corresponding cancel transfer field or input signal

Upon any error condition, the software must initialize the TCD of the channel that contains the error, as it is in an incomplete state after an error. See Fault reporting and handling for more details.

Offset

Register	Offset
MP_ES	4h

Diagram



Fields

Field	Description
31 VLD	Valid Logical OR of all CHn_ES[ERR] status fields. 0 - No CHn_ES[ERR] fields are set to 1 1 - At least one CHn_ES[ERR] field is set to 1, indicating a valid error exists that software has not cleared
30-29 —	Reserved
28-24 ERRCHN	Error Channel Number or Canceled Channel Number The channel number of the last recorded error or last recorded error-canceled transfer.
23-9 —	Reserved
8 ECX	Transfer Canceled The ECX operation is a management page function. When employed, the targeted channel's CHn_ES register reports an unspecified error; that is, only the CHn_ES[ERR] field is set to 1. The management page has full view of the error condition. 0 - No canceled transfers 1 - Last recorded entry was a canceled transfer by the error cancel transfer input
7	Source Address Error When this field is 1, it indicates that TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE].

Table continues on the next page...

Table continued from the previous page...

Field	Description
SAE	0 - No source address configuration error 1 - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error When this field is 1, it indicates that TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0 - No source offset configuration error 1 - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error When this field is 1, it indicates that TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE]. 0 - No destination address configuration error 1 - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error When this field is 1, it indicates that TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0 - No destination offset configuration error 1 - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0 - No NBYTES/CITER configuration error 1 - The last recorded error was NBYTES equal to zero or a CITER not equal to BITER error. Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. 0 - No scatter/gather configuration error 1 - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0 - No source bus error 1 - Last recorded error was a bus error on a source read
0	Destination Bus Error

Table continues on the next page...

Table continued from the previous page...

Field	Description
DBE	0 - No destination bus error 1 - Last recorded error was a bus error on a destination write

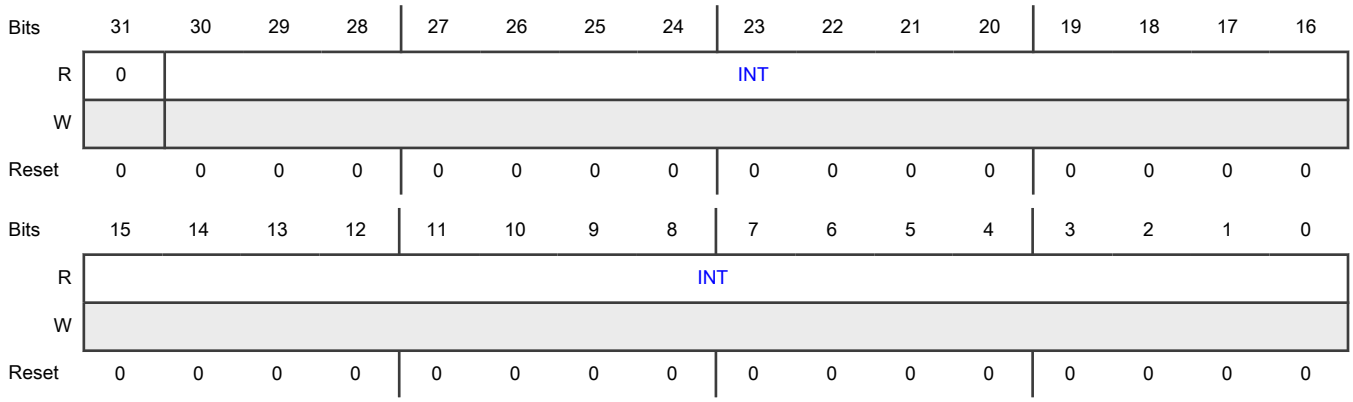
4.6.1.4 Management Page Interrupt Request Status (MP_INT)

This register shows the current state of the interrupt service requests for all eDMA channels.

Offset

Register	Offset
MP_INT	8h

Diagram



Fields

Field	Description
31 —	Reserved
30-0 INT	Interrupt Request Status The INT register presents the interrupt request status for each eDMA channel. Depending on the appropriate field setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion. The eDMA routes channel interrupt requests to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate field in the channel's interrupt request register, CHn_INT, thus negating the interrupt request. 0b - Interrupt request for corresponding channel not present 1b - Interrupt request for corresponding channel present

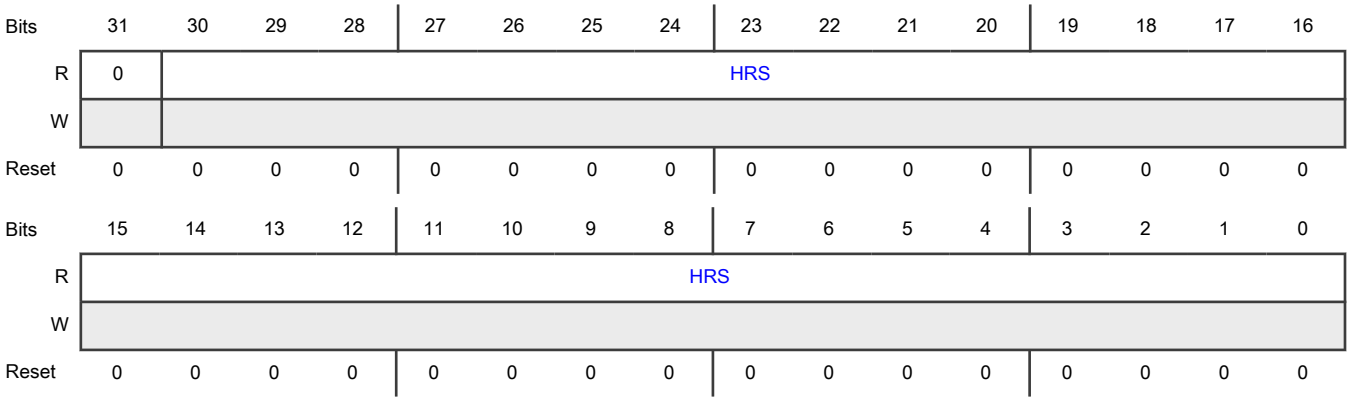
4.6.1.5 Management Page Hardware Request Status (MP_HRS)

The hardware request status register (HRS) shows the current state of the hardware service request signaling as seen by eDMA's arbitration logic.

Offset

Register	Offset
MP_HRS	Ch

Diagram



Fields

Field	Description
31 —	Reserved
30-0 HRS	Hardware Request Status The HRS bit for its respective channel remains asserted for the period when a hardware request is present on the channel. 0b - Hardware service request for corresponding channel is not present 1b - Hardware service request for corresponding channel is present

4.6.1.6 Channel Arbitration Group (CH0_GRPRI - CH30_GRPRI)

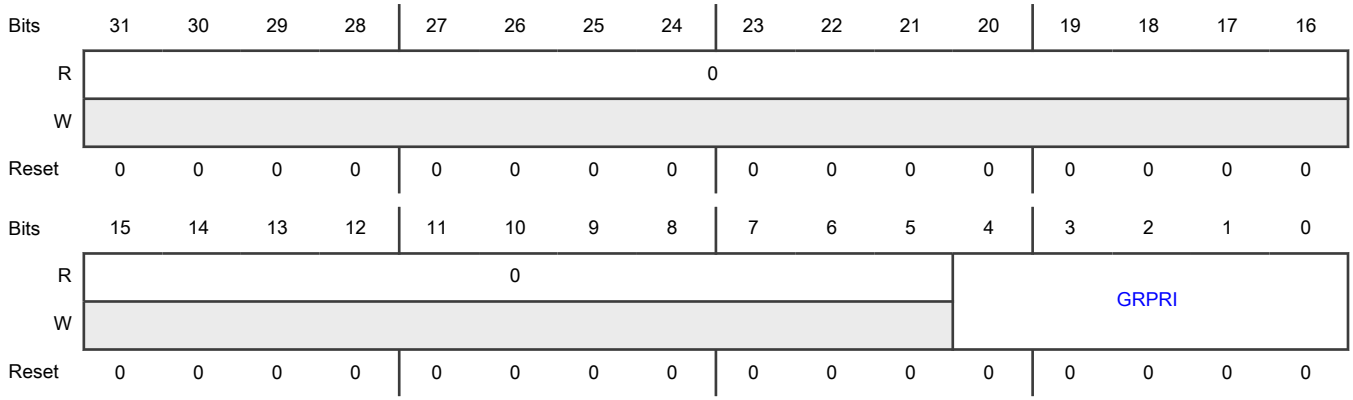
The contents of this register define the arbitration group associated with each channel. Using a fixed-priority group arbitration scheme, eDMA evaluates the arbitration group priorities by numeric value from highest group number to lowest; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. The range of the group priority values is limited to the values of 0 through 31. Within each arbitration group, the channel priority assignment CH*n*_PRI determines the highest-priority channel.

Offset

For n = 0 to 30:

Register	Offset
CHn_GRPRI	100h + (n × 4h)

Diagram



Fields

Field	Description
31-5 —	Reserved
4-0 GRPRI	Arbitration Group For Channel n Fixed-priority arbitration group number.

4.6.2 EDMA_TCD register descriptions

4.6.2.1 EDMA_TCD memory map

EDMA3_1.TCD base address: 4401_0000h

NOTE

For registers in the following table with *Protection*, see the REG_PROT details for more information.

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)	Protection
0 - 1E_0000	Channel Control and Status (CH0_CSR - CH30_CSR)	32	RW	0000_0000	Yes
4 - 1E_0004	Channel Error Status (CH0_ES - CH30_ES)	32	RW	0000_0000	No
8 - 1E_0008	Channel Interrupt Status (CH0_INT - CH30_INT)	32	RW	0000_0000	No

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)	Protection
C - 1E_000C	Channel System Bus (CH0_SBR - CH30_SBR)	32	RW	0000_8007	Yes
10 - 1E_0010	Channel Priority (CH0_PRI - CH30_PRI)	32	RW	0000_0000	Yes
20 - 1E_0020	TCD Source Address (TCD0_SADDR - TCD30_SADDR)	32	RW	See section	Yes
24 - 1E_0024	TCD Signed Source Address Offset (TCD0_SOFF - TCD30_SOFF)	16	RW	See section	Yes
26 - 1E_0026	TCD Transfer Attributes (TCD0_ATTR - TCD30_ATTR)	16	RW	See section	Yes
28 - 1E_0028	TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD30_NBYTES_MLOFFNO)	32	RW	See section	Yes
28 - 1E_0028	TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD30_NBYTES_MLOFFYES)	32	RW	See section	No
2C - 1E_002C	TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD30_SLAST_SDA)	32	RW	See section	Yes
30 - 1E_0030	TCD Destination Address (TCD0_DADDR - TCD30_DADDR)	32	RW	See section	Yes
34 - 1E_0034	TCD Signed Destination Address Offset (TCD0_DOFF - TCD30_DOFF)	16	RW	See section	Yes
36 - 1E_0036	TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD30_CITER_ELINKNO)	16	RW	See section	Yes
36 - 1E_0036	TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD30_CITER_ELINKYES)	16	RW	See section	No
38 - 1E_0038	TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD30_DLAST_SGA)	32	RW	See section	Yes
3C - 1E_003C	TCD Control and Status (TCD0_CSR - TCD30_CSR)	16	RW	See section	Yes
3E - 1E_003E	TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD30_BITER_ELINKNO)	16	RW	See section	Yes
3E - 1E_003E	TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD30_BITER_ELINKYES)	16	RW	See section	No

4.6.2.2 Channel Control and Status (CH0_CSR - CH30_CSR)

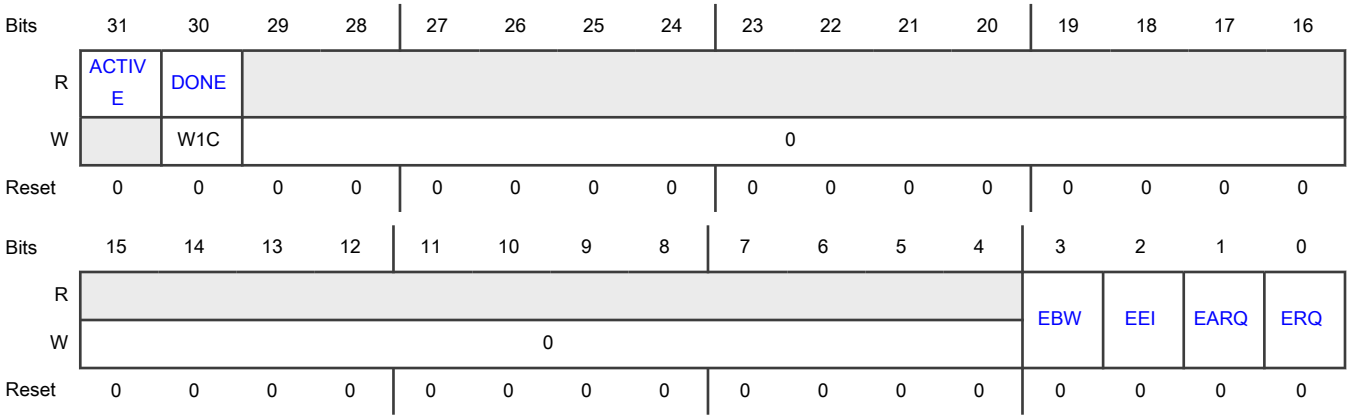
This register contains several fields related to hardware and interrupt requests, configuration, and status for the given channel.

Offset

For n = 0 to 30:

Register	Offset
CHn_CSR	0h + (n × 1_0000h)

Diagram



Fields

Field	Description
31 ACTIVE	Channel Active The ACTIVE field indicates the channel was selected by arbitration and is executing the prescribed transfers. The eDMA sets it to 1 when channel service begins, and clears it to 0 as the minor loop completes or when any error condition is detected. Except for dynamic scatter/gather or dynamic channel linking, you must not modify the transfer control descriptor when a channel is active.
30 DONE	Channel Done The DONE field indicates the eDMA has completed the major loop. The eDMA engine sets this field as the CITER count reaches zero. If enabled, the eDMA generates an interrupt request corresponding to this completed channel. The software clears it, or the hardware clears it when the channel is activated. **NOTE** This field must be cleared to 0 before writing the MAJORELINK or ESG fields.
29-4 —	Reserved
3 EBW	Enable Buffered Writes

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When buffered writes are enabled, all writes except for the last write sequence of the minor loop are signaled by the eDMA as bufferable. 0 - Buffered writes on system bus disabled. Buffered writes on system bus disabled 1 - Buffered writes on system bus enabled. Bufferable write signal asserted on all system bus writes except during last write sequence
2 EEI	Enable Error Interrupt The EEI field enables the error interrupt signal for the channel. The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller. 0 - Error signal for corresponding channel does not generate error interrupt 1 - Assertion of error signal for corresponding channel generates error interrupt request
1 EARQ	Enable Asynchronous DMA Request The enable asynchronous DMA request field (EARQ) does not affect DMA operations. When set to 1, this field allows the hardware service request enable field (ERQ) to propagate out of the DMA to the power controller. When cleared to 0, this field masks the hardware service request enable field to the power controller. 0 - Disable asynchronous DMA request for the channel 1 - Enable asynchronous DMA request for the channel
0 ERQ	Enable DMA Request Disable a channel's hardware service request at the source before clearing the channel's ERQ field. The DMA hardware request input signal and the enable request field (ERQ) must be asserted before a channel's hardware service request is accepted. The state of the eDMA enable request field does not affect a channel service request made explicitly through software or channel linking. The state of the ERQ field does not affect the channel's START field. 0 - DMA hardware request signal for corresponding channel disabled 1 - DMA hardware request signal for corresponding channel enabled

4.6.2.3 Channel Error Status (CH0_ES - CH30_ES)

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus initiator read or write cycle

The ERR field signals the presence of an error for the channel. The eDMA engine signals the occurrence of an error condition by setting the appropriate field in this register. The outputs of this register are enabled by the contents of the CHn_CSR[EEI] field, then logically summed across all channels to form an error interrupt request, which may be routed to the interrupt controller.

During the execution of the interrupt service routine associated with any DMA errors, it is software's responsibility to clear the appropriate bit, negating the error-interrupt request. The normal DMA channel completion indicators (setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request) are not affected when eDMA detects an error. The contents of this ERR register field can also be polled because a non-zero value indicates the presence of a channel error, regardless of the state of the EEI mask.

The state of any given channel's error indicators is affected by writes to this register. Writing a 1 to the ERR field clears the channel's error status, and writing a 0 has no effect.

An unspecified error, where only the ERR field is set to 1, indicates that either a transfer was cancelled with an error. The Management Page Error Status register has full view of the error condition.

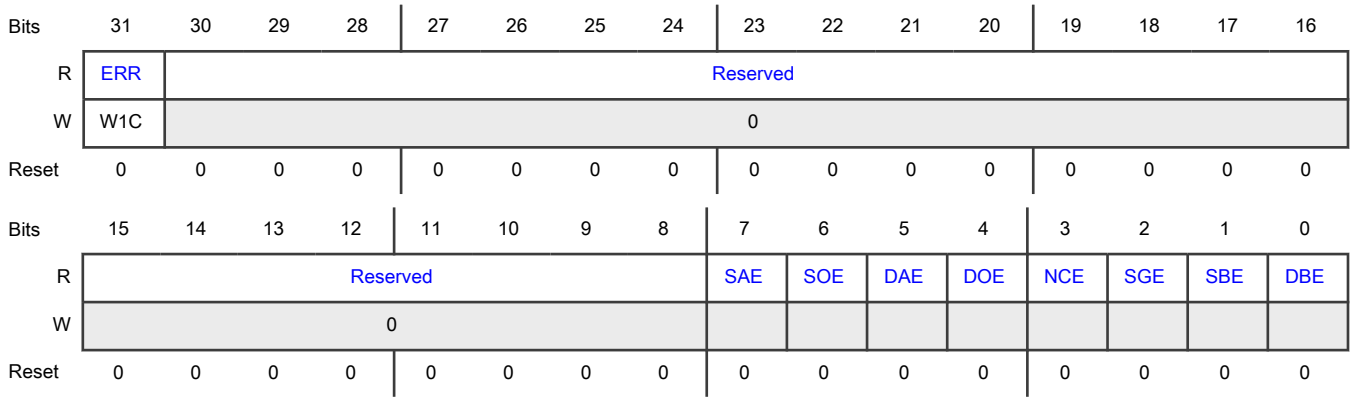
See Fault reporting and handling for more details.

Offset

For n = 0 to 30:

Register	Offset
CHn_ES	4h + (n × 1_0000h)

Diagram



Fields

Field	Description
31 ERR	Error In Channel 0 - An error in this channel has not occurred 1 - An error in this channel has occurred
30-8 —	Reserved
7 SAE	Source Address Error TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE]. 0 - No source address configuration error 1 - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0 - No source offset configuration error

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE]. 0 - No destination address configuration error 1 - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0 - No destination offset configuration error 1 - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0 - No NBYTES/CITER configuration error 1 - Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG]is enabled. 0 - No scatter/gather configuration error 1 - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0 - No source bus error 1 - Last recorded error was bus error on source read
0 DBE	Destination Bus Error 0 - No destination bus error 1 - Last recorded error was bus error on destination write

4.6.2.4 Channel Interrupt Status (CH0_INT - CH30_INT)

The INT field signals the presence of an interrupt request for the channel. Depending on the appropriate bit setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion.

The outputs of this register are directly routed to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit, negating the interrupt request. On writes to INT, a 1 clears the channel's interrupt request. A zero has no effect on the channel's current interrupt status.

Offset

For n = 0 to 30:

Register	Offset
CHn_INT	8h + (n × 1_0000h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															INT
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-1 —	Reserved
0 INT	Interrupt Request 0 - Interrupt request for corresponding channel cleared 1 - Interrupt request for corresponding channel active

4.6.2.5 Channel System Bus (CH0_SBR - CH30_SBR)

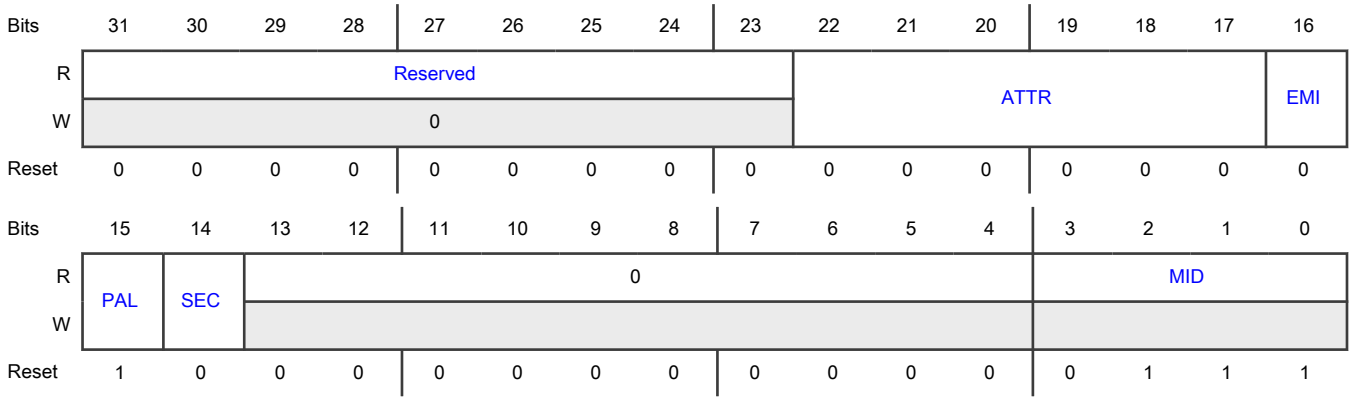
The Channel System Bus register places identification and attribute information on the system bus interface for the eDMA.

Offset

For n = 0 to 30:

Register	Offset
CHn_SBR	Ch + (n × 1_0000h)

Diagram



Fields

Field	Description
31-23 —	Reserved
22-17 ATTR	Attribute Output DMA's system bus attribute output value.
16 EMI	Enable Initiator ID Replication The eDMA initiator ID replication field allows the eDMA to use the same protection level and system bus ID of the initiator programming the eDMA's TCD. When enabled, the eDMA uses the initiator ID and protection level stored in the CHn_SBR registers, instead of the eDMA's default values. When a initiator (for example a core) programs a TCD, its initiator ID is captured when the TCD*n*_CSR control attributes are written. A scatter/gather operation does not affect the CHn_SBR registers. You can write the EMI only if MP_CSR[GMRC] = 1, which means Global Initiator ID Replication Control is enabled; otherwise, the EMI is forced to zero. **NOTE** If initiator ID replication is disabled, the nonsecure, privileged protection level for DMA transfers is used. 0 - Initiator ID replication is disabled 1 - Initiator ID replication is enabled
15 PAL	Privileged Access Level This field controls DMA's protection level on the system bus when the channel is active. **NOTE** The value written into this register cannot exceed the security and privilege level of the core or other initiator writing the channel's system bus register; CH*n*_SBR. The order of precedence is SecurePriv>SecureUser>NonsecurePriv>NonsecureUser 0 - User protection level for DMA transfers 1 - Privileged protection level for DMA transfers

Table continues on the next page...

Table continued from the previous page...

Field	Description
14 SEC	Security Level DMA's security level on the system bus when the channel is active. **NOTE** The value written into this register cannot exceed the security and privilege level of the core or other initiator writing the channel's system bus register; CH*n*_SBR. The order of precedence is SecurePriv>SecureUser>NonsecurePriv>NonsecureUser 0 - Nonsecure protection level for DMA transfers 1 - Secure protection level for DMA transfers
13-4 —	Reserved
3-0 MID	Initiator ID This field controls the DMA's initiator ID on the system bus when the channel is active. **NOTE** The ID captured in this register reflects the initiator ID of the core or other initiator writing the channel's security attributes, TCD*n*_SBR[SEC].

4.6.2.6 Channel Priority (CH0_PRI - CH30_PRI)

The contents of these registers define unique priorities associated with each channel within the same channel group. Channel grouping is programmed via Channel Arbitration Group (CH0_GRPRI - CH30_GRPRI).

The channel priorities within a group are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. Software must program the channel priorities with unique values; otherwise, channel numbers with the same, non-zero value, will be selected based on channel number with the higher channel number having higher priority.

If more than one channel in a group has an arbitration priority level value of zero, then the arbitration mode field MP_CSR[ERCA] is used to determine the arbitration scheme for all channels with APL=0 within a group.

When you enable round-robin channel arbitration (MP_CSR[ERCA] = 1), all channels with APL=0 within a group will use a round-robin arbitration scheme, which rotates among these channels requesting service without regard to priority. Round-robin provides a fairness mechanism within an arbitration group.

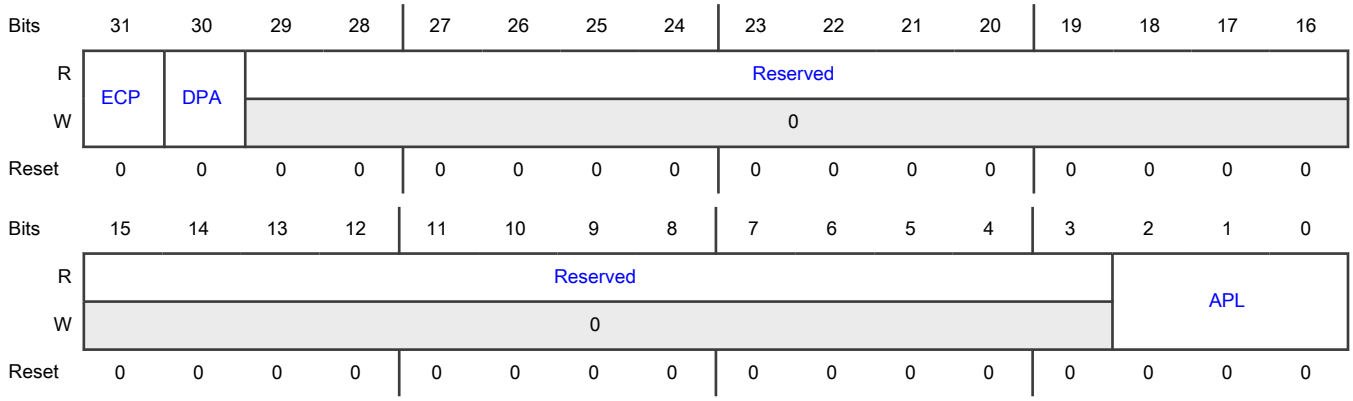
When you enable fixed-priority channel arbitration (MP_CSR[ERCA] = 0), eDMA selects channels with APL=0 based on channel number, with the higher channel number having higher priority.

Offset

For n = 0 to 30:

Register	Offset
CHn_PRI	10h + (n × 1_0000h)

Diagram



Fields

Field	Description
31 ECP	Enable Channel Preemption 0 - Channel cannot be suspended by a higher-priority channel's service request 1 - Channel can be temporarily suspended by a higher-priority channel's service request
30 DPA	Disable Preempt Ability 0 - Channel can suspend a lower-priority channel 1 - Channel cannot suspend any other channel, regardless of channel priority
29-3 —	Reserved
2-0 APL	Arbitration Priority Level Channel priority level for arbitration within the assigned arbitration group.

4.6.2.7 TCD Source Address (TCD0_SADDR - TCD30_SADDR)

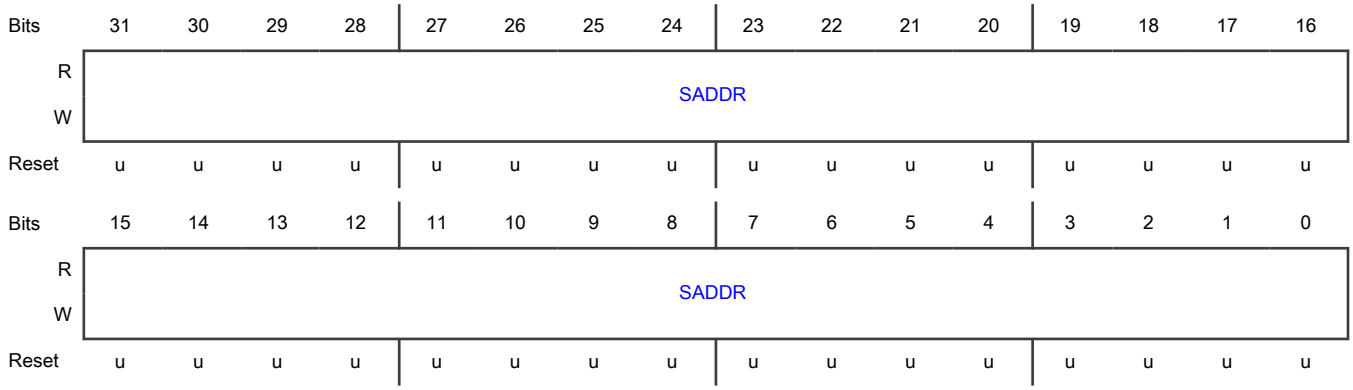
This register contains the address for the read transactions.

Offset

For n = 0 to 30:

Register	Offset
TCDn_SADDR	20h + (n × 1_0000h)

Diagram



Fields

Field	Description
31-0 SADDR	Source Address Memory address pointing to the source data.

4.6.2.8 TCD Signed Source Address Offset (TCD0_SOFF - TCD30_SOFF)

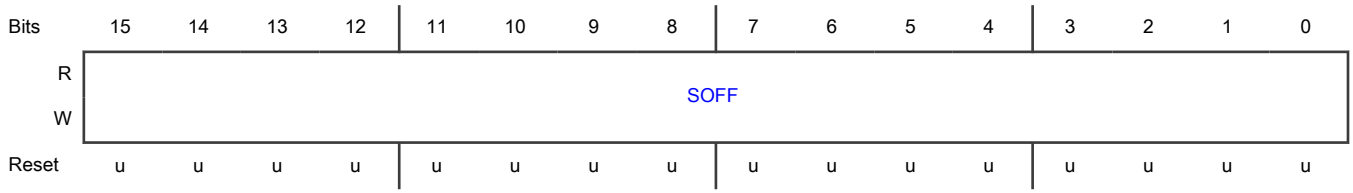
This register contains the sign-extended value added to Source Address register after each read transaction.

Offset

For n = 0 to 30:

Register	Offset
TCDn_SOFF	24h + (n × 1_0000h)

Diagram



Fields

Field	Description
15-0 SOFF	Source Address Signed Offset Sign-extended offset applied to the current source address to form the next-state value as each source read is completed.

4.6.2.9 TCD Transfer Attributes (TCD0_ATTR - TCD30_ATTR)

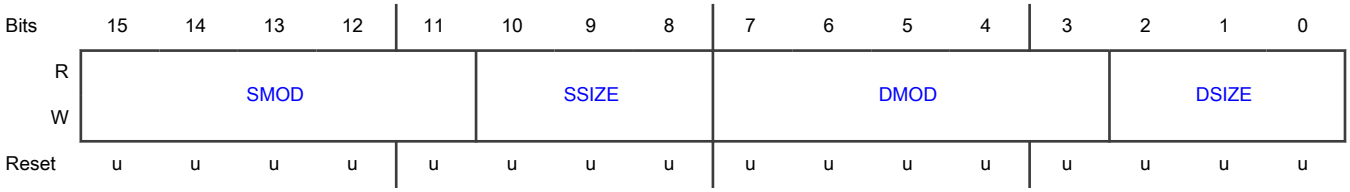
This register contains size and option modulo addressing information for source and destination addresses.

Offset

For n = 0 to 30:

Register	Offset
TCDn_ATTR	26h + (n × 1_0000h)

Diagram



Fields

Field	Description
15-11 SMOD	Source Address Modulo This field defines a specific address range, which is the value after the SADDR + SOFF calculation is performed on the original register value. Setting this field makes it easy to implement a circular data queue. For data queues requiring power-of-2-sized bytes, the queue must start at a 0-modulo-size address and the SMOD field must be set to the appropriate value for the queue, freezing the required number of upper address bits. The value programmed into this field specifies the number of lower address bits that are allowed to change. For a circular queue application, you typically set TCDn_SOFF[SOFF] to the transfer size to implement post-increment addressing, with the SMOD function constraining the addresses to a 0-modulo-size range. 0_0000 - Source address modulo feature disabled
10-8 SSIZE	Source Data Transfer Size 000 - 8-bit 001 - 16-bit 010 - 32-bit 011 - 64-bit 100 - 16-byte 101 - 32-byte 110 - 64-byte 111 - Reserved
7-3	Destination Address Modulo

Table continues on the next page...

Table continued from the previous page...

Field	Description
DMOD	See the SMOD definition.
2-0 DSIZE	Destination Data Transfer Size See the SSIZE definition.

4.6.2.10 TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD30_NBYTES_MLOFFNO)

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offsets are address offset values added to the final source address (TCDn_SADDR), or destination address (TCDn_DADDR), upon minor loop completion. Minor loop completion is when the channel has finished the service request and has transferred NBYTES. When minor loop offsets are enabled, the minor loop offset value (TCDn_NBYTES_MLOFFYES[MLOFF]) is added to the final source address (TCDn_SADDR), to the final destination address (TCDn_DADDR), or to both, prior to the addresses being written back to the TCD. If the major loop is complete, the minor loop offset is ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

When minor loop mapping is enabled (SMLOE or DMLOE is 1), TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is redefined. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is used to specify multiple fields:

- A source enable bit (SMLOE) to specify the minor loop offset must be applied to the source address (TCDn_SADDR) upon minor loop completion
- A destination enable bit (DMLOE) to specify the minor loop offset must be applied to the destination address (TCDn_DADDR) upon minor loop completion
- The sign extended minor loop offset value (MLOFF)

The same offset value (MLOFF) is used for both source and destination minor loop offsets. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is defined as follows:

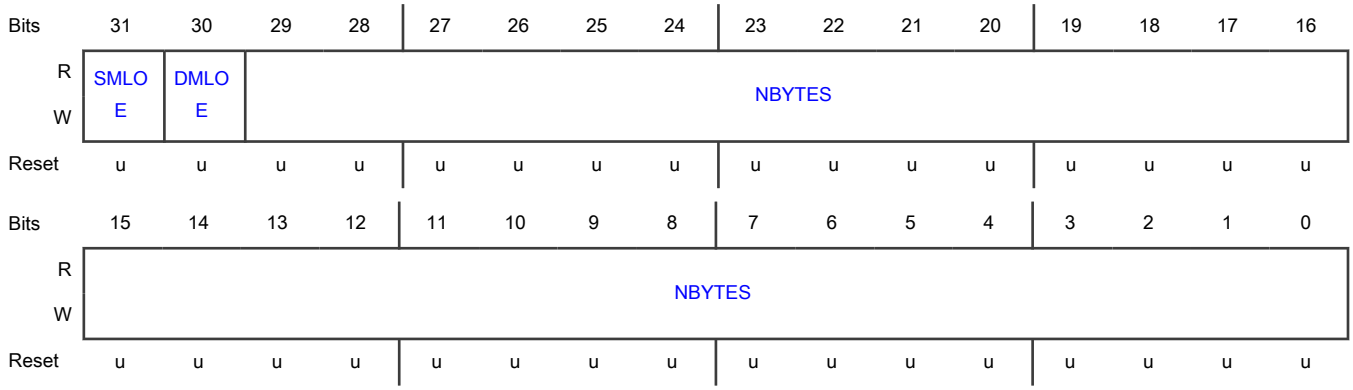
- If SMLOE = 0 and DMLOE = 0, then see the TCDn_NBYTES_MLOFFNO register description.
- If either SMLOE or DMLOE is 1, then see the TCDn_NBYTES_MLOFFYES register description.

Offset

For n = 0 to 30:

Register	Offset
TCDn_NBYTES_MLOFFNO	28h + (n × 1_0000h)

Diagram



Fields

Field	Description
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0 - Minor loop offset not applied to SADDR 1 - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0 - Minor loop offset not applied to DADDR 1 - Minor loop offset applied to DADDR
29-0 NBYTES	Number of Bytes To Transfer Per Service Request Number of bytes to be transferred for each service request of the channel. When a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the byte transfer count has been reached. This process is normally an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the byte count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major loop iteration count (CITER) is decremented by one and written back to the TCD memory. If the major iteration count is complete, additional processing is performed.

4.6.2.11 TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD30_NBYTES_MLOFFYES)

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offset is an address offset value added to the final source address (TCDn_SADDR) or destination address (TCDn_DADDR) upon minor loop completion. Minor loop completion occurs when the channel has finished the service request and has transferred NBYTES. Minor loop offsets are enabled by setting either the source enable bit (SMLOE) or the destination enable bit (DMLOE).

The source enable bit (SMLOE) specifies the minor loop offset value (MLOFF) that is to be applied to the source address (TCDn_SADDR) upon minor loop completion. The destination enable bit (DMLOE) specifies the minor loop offset (MLOFF) that is to be applied to the destination address (TCDn_DADDR) upon minor loop completion.

If the major loop is complete, the minor loop offsets are ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

When you enable the minor loop offset overlay (either SMLOE or DMLOE is 1), eDMA redefines TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES specifies the sign-extended minor loop offset value (MLOFF). The same offset value (MLOFF) applies to both source and destination minor loop offsets. When the minor loop offset is enabled, you must align it to the transfer size of the source or destination it is associated with. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFNO) defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFYES is defined as follows:

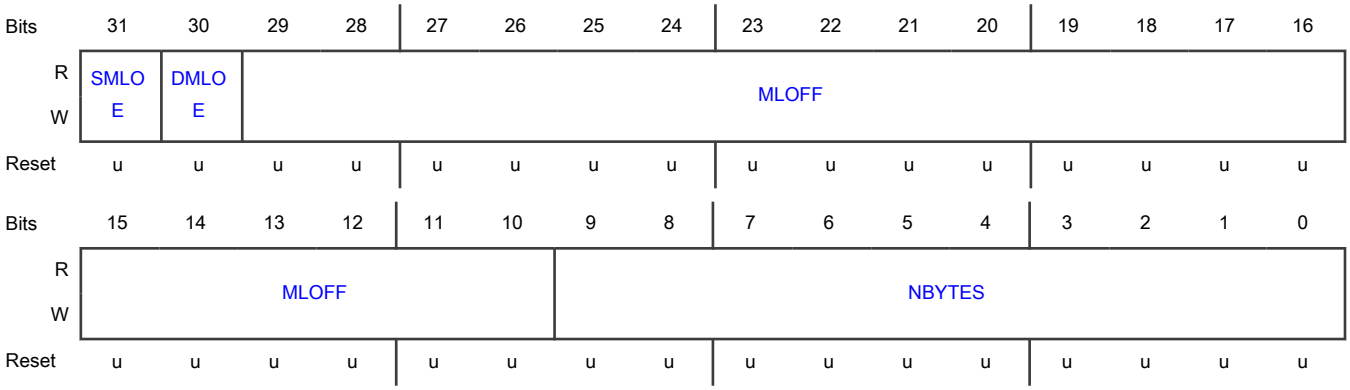
- If either minor loop offset is enabled (SMLOE or DMLOE = 1), then see the TCDn_NBYTES_MLOFFYES register description.
- If SMLOE and DMLOE are both 0, then see the TCDn_NBYTES_MLOFFNO register description.

Offset

For n = 0 to 30:

Register	Offset
TCDn_NBYTES_MLOFFYES	28h + (n × 1_0000h)

Diagram



Fields

Field	Description
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0 - Minor loop offset not applied to SADDR

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0 - Minor loop offset not applied to DADDR 1 - Minor loop offset applied to DADDR
29-10 MLOFF	Minor Loop Offset If SMLOE or DMLOE is 1, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.
9-0 NBYTES	Number of Bytes To Transfer Per Service Request The number of bytes to be transferred in each service request of the channel. As a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the minor byte transfer count has been reached. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major iteration count is decremented and restored to the TCD memory. If the major iteration count is complete, additional processing is performed.

4.6.2.12 TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD30_SLAST_SDA)

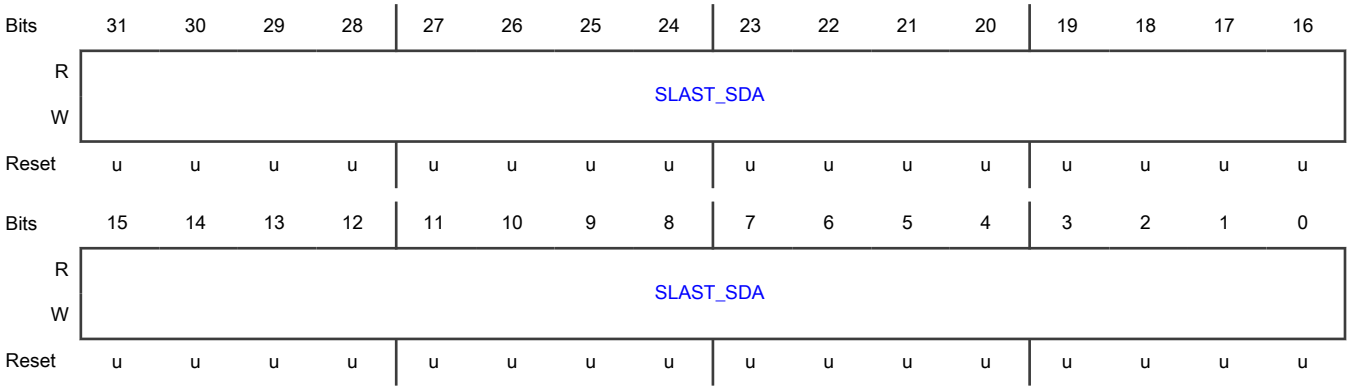
This register contains the value added to the source address when the major loop is complete. When the store destination address option is enabled, this field provides a pointer to memory for storing the final destination address.

Offset

For n = 0 to 30:

Register	Offset
TCDn_SLAST_SDA	2Ch + (n × 1_0000h)

Diagram



Fields

Field	Description
31-0 SLAST_SDA	Last Source Address Adjustment / Store DADDR Address Source last address adjustment or the system memory address for destination address (DADDR) storage. If (TCDn_CSR[ESDA] = 0), then: • Adjustment value is added to the source address at the completion of the major iteration count. This value can be used to restore the source address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final source address adjustment. Otherwise: • This address points to the 32-bit-aligned memory location where the destination address (DADDR) is to be stored in system memory. By saving the final destination address in system memory via the ESDA feature, you are able to compute the size of a variable destination data buffer by simply subtracting the beginning DADDR from the final, saved DADDR. This feature is used together with the scatter/gather operation to prevent the loss of the final DADDR, which is overwritten during the scatter/gather operation. The "Store Destination Address" (SDA) value must be a 32-bit-aligned location because the eDMA forces the lower two address bits of the SLAST_SDA field to zero when ESDA is enabled. The module performs this write operation when the major loop is done; that is, when the major iteration count (CITER) decrements to zero.

4.6.2.13 TCD Destination Address (TCD0_DADDR - TCD30_DADDR)

This register contains the address for the write transactions.

Offset

For n = 0 to 30:

Register	Offset
TCDn_DADDR	30h + (n × 1_0000h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DADDR															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DADDR															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Description
31-0 DADDR	Destination Address Memory address pointing to the destination data.

4.6.2.14 TCD Signed Destination Address Offset (TCD0_DOFF - TCD30_DOFF)

This register contains the sign-extended value added to Destination Address register after each write transaction.

Offset

For n = 0 to 30:

Register	Offset
TCDn_DOFF	34h + (n × 1_0000h)

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DOFF															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Description
15-0 DOFF	Destination Address Signed Offset Sign-extended offset that is applied to the current destination address to form the next-state value as each destination write is completed.

4.6.2.15 TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD30_CITER_ELINKNO)

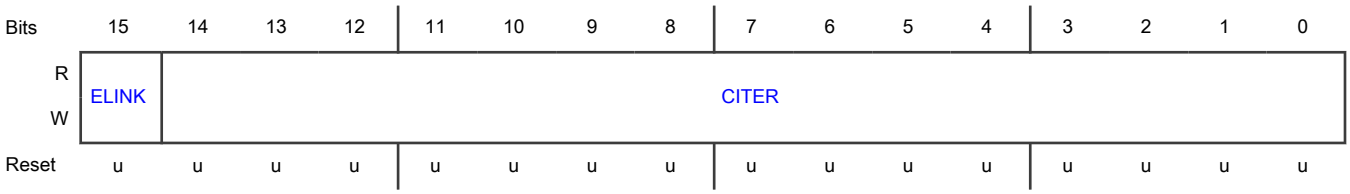
If TCDn_CITER[ELINK] is 0, the TCDn_CITER register is defined as follows.

Offset

For n = 0 to 30:

Register	Offset
TCDn_CITER_ELINKNO	36h + (n × 1_0000h)

Diagram



Fields

Field	Description
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel to 1. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking. **NOTE** This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0 - Channel-to-channel linking disabled 1 - Channel-to-channel linking enabled
14-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

4.6.2.16 TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD30_CITER_ELINKYES)

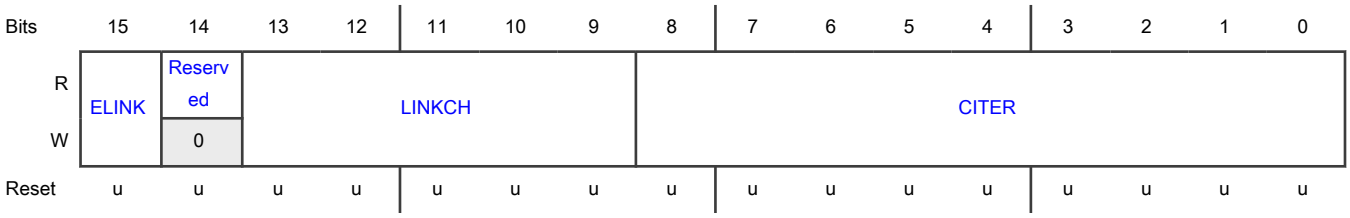
If TCDn_CITER[ELINK] is 1, the TCDn_CITER register is defined as follows.

Offset

For n = 0 to 30:

Register	Offset
TCDn_CITER_ELINKYES	36h + (n × 1_0000h)

Diagram



Fields

Field	Description
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. When enabled, an internal mechanism sets the TCDn_CSR[START] field of the specified channel (LINKCH) upon minor loop completion. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking. **NOTE** This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0 - Channel-to-channel linking disabled 1 - Channel-to-channel linking enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
14 —	Reserved
13-9 LINKCH	Minor Loop Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted the eDMA engine initiates a channel service request to the channel defined by this field by writing that channel's TCDn_CSR[START] field to 1.
8-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to the TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

4.6.2.17 TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD30_DLAST_SGA)

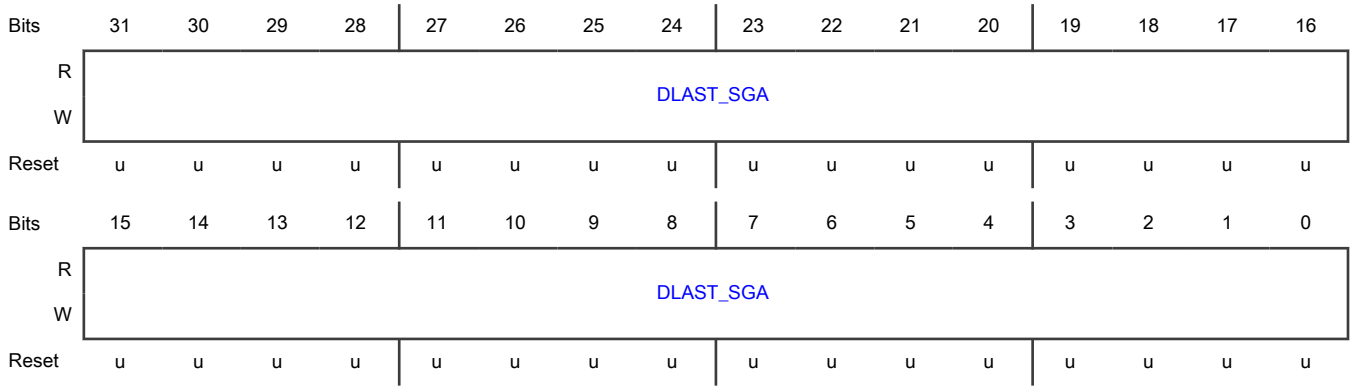
This register contains the value added to the destination address when the major loop is complete. When the Scatter/Gather option is enabled, this field provides a pointer to memory for fetching a transfer control descriptor to reprogram the channel.

Offset

For n = 0 to 30:

Register	Offset
TCDn_DLAST_SGA	38h + (n × 1_0000h)

Diagram



Fields

Field	Description
31-0 DLAST_SGA	Last Destination Address Adjustment / Scatter Gather Address Adjustment of the last destination address or the memory address for the next transfer control descriptor to be loaded into this channel (scatter/gather). If (TCDn_CSR[ESG] = 0) then: • Adjustment value is added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final destination address adjustment. Otherwise: • This address points to the beginning of a 0-modulo 32-byte region containing the next transfer control descriptor to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo 32-byte, or else a configuration error is reported.

4.6.2.18 TCD Control and Status (TCD0_CSR - TCD30_CSR)

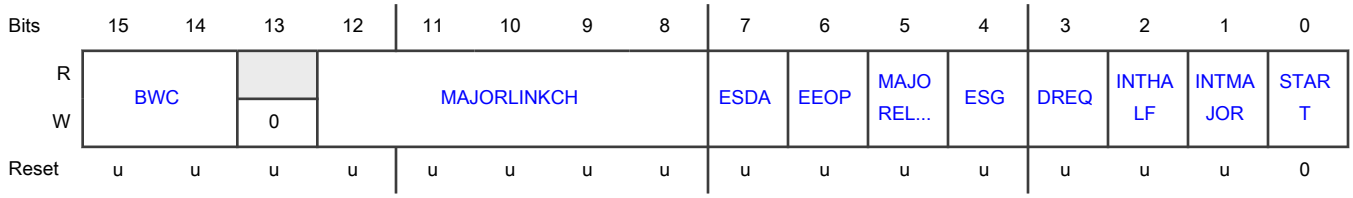
This register is used to enable optional features.

Offset

For n = 0 to 30:

Register	Offset
TCDn_CSR	3Ch + (n × 1_0000h)

Diagram



Fields

Field	Description
15-14 BWC	Bandwidth Control Throttles the amount of bus bandwidth consumed by the eDMA. Generally, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces eDMA to stall after the completion of each read/write access, to control the bus request bandwidth seen by the system bus interconnect. **NOTE** If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency. 00 - No eDMA engine stalls 01 - Reserved 10 - eDMA engine stalls for 4 cycles after each R/W 11 - eDMA engine stalls for 8 cycles after each R/W
13 —	Reserved
12-8 MAJORLINKCH	Major Loop Link Channel Number If (MAJORELINK = 0) then: • No channel-to-channel linking, or chaining, is performed after the major loop counter is exhausted. Otherwise: • After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field to 1.
7 ESDA	Enable Store Destination Address As the channel completes the major loop by either the current iteration counter (CITER) decrementing to 0, or by receiving an enabled end-of-packet signal, this field enables writing the destination address (DADDR) to the address stored in the SLAST_SDA field. The value written to system memory is the next DADDR value prior to the DLAST_SGA offset being applied, or overwritten by an enabled scatter/gather operation. When the ESDA bit is 1, SLAST_SDA contains the write pointer instead of the final source address offset. Because this is a pointer and not a final offset, a last source address offset of zero is applied to SADDR instead of the SLAST_SGA value. 0 - Ability to store destination address to system memory disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Ability to store destination address to system memory enabled
6 EEOP	Enable End-Of-Packet Processing When enabled by the EEOP field, an end-of-packet hardware input signal directs eDMA to discontinue executing the active channel, and to treat the shutdown as the major-loop-completed event. If the EEOP field is 1, the end-of-packet signal from supported peripherals is accepted. If the EEOP field is 0, the end-of-packet input is ignored. With an end-of-packet retirement, the current TCD destination address (or ESDA-saved destination address), minus the software-saved initial address (DADDR), reflects the total amount of data transferred. 0 - End-of-packet operation disabled 1 - End-of-packet hardware input signal enabled
5 MAJORELINK	Enable Link When Major Loop Complete As the channel completes the major loop, this flag enables linking to another channel defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. **NOTE** To support the dynamic linking coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0 - Channel-to-channel linking disabled 1 - Channel-to-channel linking enabled
4 ESG	Enable Scatter/Gather Processing As the channel completes the major loop, this flag enables scatter/gather processing in the current channel. If enabled, the eDMA engine uses TCDn_DLAST_SGA as a memory pointer to a 0-modulo 32-bit address containing a 32-byte data structure, which is loaded as the transfer control descriptor into local memory. **NOTE** To support the dynamic scatter/gather coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0 - Current channel's TCD is normal format 1 - Current channel's TCD specifies scatter/gather format.
3 DREQ	Disable Request If this flag is 1, the eDMA hardware automatically clears the corresponding ERQ bit when the current major iteration count reaches 0. 0 - No operation. Channel's ERQ field not affected 1 - Clear the ERQ field to 0 upon major loop completion, thus disabling hardware service requests. Channel's ERQ field cleared to 0 when major loop complete
2	Enable Interrupt If Major Counter Half-complete

Table continues on the next page...

Table continued from the previous page...

Field	Description
INTHALF	If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER = (BITER/2)). This halfway point interrupt request is provided to support double-buffered, also known as ping-pong, schemes, or other types of data movement where the processor needs an early indication of the transfer's progress. **NOTE** If BITER = 1, do not use INTHALF; use INTMAJOR instead. 0 - Halfway point interrupt disabled 1 - Halfway point interrupt enabled
1 INTMAJOR	Enable Interrupt If Major count complete If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count (CITER) reaches 0. 0 - End-of-major loop interrupt disabled 1 - End-of-major loop interrupt enabled
0 START	Channel Start If this flag is 1, the channel is requesting service. The eDMA hardware automatically clears this flag to 0 after the channel begins execution. 0 - Channel not explicitly started 1 - Channel explicitly started via a software-initiated service request

4.6.2.19 TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD30_BITER_ELINKNO)

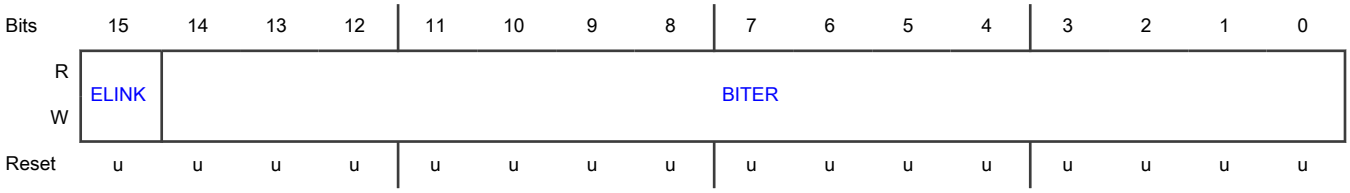
If the TCDn_BITER[ELINK] field is 0, the TCDn_BITER register is defined as follows.

Offset

For n = 0 to 30:

Register	Offset
TCDn_BITER_ELINKNO	3Eh + (n × 1_0000h)

Diagram



Fields

Field	Description
15 ELINK	Enables Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0 - Channel-to-channel linking disabled 1 - Channel-to-channel linking enabled
14-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

4.6.2.20 TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD30_BITER_ELINKYES)

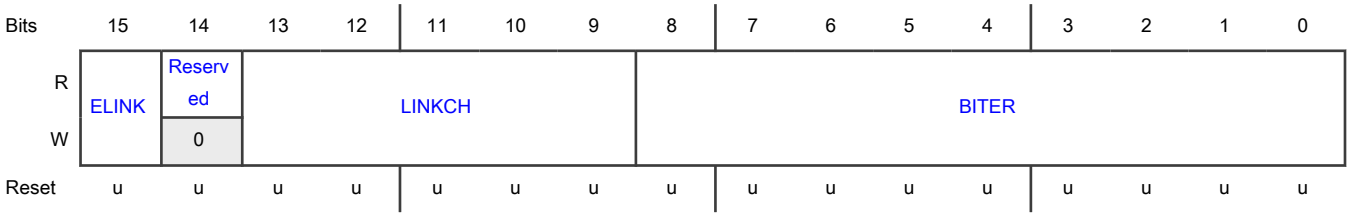
If the TCDn_BITER[ELINK] field is set, the TCDn_BITER register is defined as follows.

Offset

For n = 0 to 30:

Register	Offset
TCDn_BITER_ELINKYES	3Eh + (n × 1_0000h)

Diagram



Fields

Field	Description
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0 - Channel-to-channel linking disabled 1 - Channel-to-channel linking enabled
14 —	Reserved
13-9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field.
8-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

4.7 Glossary

TCD Transfer control descriptor

Chapter 5
Enhanced Direct Memory Access (eDMA4)

5.1 Chip-specific eDMA4 information

Table 20. Reference links to related information

Topic	Related module	Reference
Full description	Arm Cortex CM33 Platform (CM33)	CM33 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The Enhanced Direct Memory Access Type 4 (eDMA4) is an AXI-based 32-bit-addressing-capable eDMA module. For the rest of the chapter, eDMA refers to the eDMA4 type. There is one instance of eDMA4 in the Wakeup domain. This is called the eDMA4_2 instance.

5.2 Overview

eDMA is a highly programmable data-transfer engine that performs complex data transfers with minimal intervention from a host processor. You can use it in applications where the data size to be transferred is statically known and not defined within the transferred data.

The eDMA microarchitecture includes:

- A DMA engine that performs:
  - Source address and destination address calculations
  - Data-movement operations
- Local memory containing TCDs for each of the 64 channels

5.2.1 Block diagram

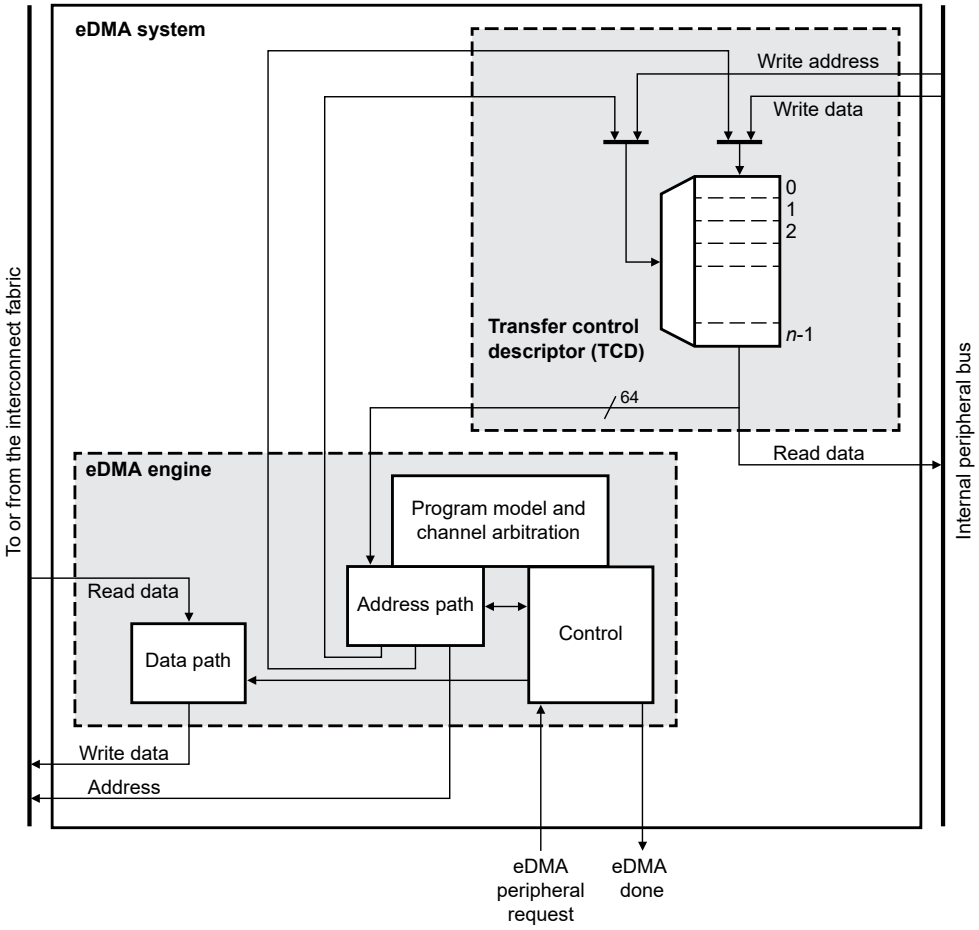


Figure 10. Block diagram

5.2.2 Block parts

eDMA is partitioned into two major modules: the eDMA engine and the TCD local memory.

Table 21 shows the submodules that the eDMA engine is partitioned into.

Table 21. eDMA engine submodules

Submodule	Function
Address path	Implements registered versions of the following channel TCDs: • Primary channel • Secondary (preempt) channel Additionally, this submodule manages all controller bus-address calculations. All channels provide the same functionality. This structure allows data transfers associated with one channel to be preempted after the completion of a read/write sequence if a higher-priority channel activation is asserted when the first channel is active. After a channel is activated, it runs until the minor loop is completed, unless a higher-priority channel preempts it. This provides a mechanism (enabled by CH0_PRI[ECP]) where a large data move operation can be preempted to minimize the time another channel is blocked from execution.

Table continues on the next page...

Table 21. eDMA engine submodules (continued)

Submodule	Function
	When you select a channel to execute, the contents of its TCD are read from local memory and loaded into the address path channel *x* registers for a normal start and into channel *y* registers for a preemption start. After the minor loop completes execution, the address path hardware writes the new values for TCD*n*[SADDR], TCD*n*[DADDR], and TCD*n*[CITER] back to local memory. If the major iteration count is exhausted, additional processing is performed, including the final address pointer updates, reloading TCD*n*_CITER, and a possible fetch of the next TCD*n* from memory as part of a scatter/gather operation.
Data path	Implements the bus controller read/write datapath. It includes a data buffer and the necessary multiplex logic to support any required data alignment. The internal read data bus is the primary input, and the internal write data bus is the primary output. The address and data path modules directly support the two-stage pipelined internal bus. The address path submodule represents the first stage of the bus pipeline (address phase), while the data path module implements the second stage of the pipeline (data phase).
Program model and channel arbitration	Implements the first section of the eDMA programming model and channel arbitration logic. The programming model registers are connected to the internal peripheral bus. The eDMA peripheral request inputs and interrupt request outputs are also connected to this block (via the control logic).
Control	Describes the control functions for the eDMA engine. For data transfers where the source and destination sizes are equal, the eDMA engine performs a series of source read/destination write operations until the number of bytes specified in the minor loop byte count has moved. For descriptors where the sizes are not equal, multiple accesses of the small-size data are required for each reference of the larger size. For example, if the source size references 16-bit data and the destination is 32-bit data, two reads are performed followed by a one 32-bit write.

Table 22 shows further partitioning of the TCD local memory.

Table 22. Transfer control descriptor memory

Submodule	Description
Memory controller	Implements the required dual-ported controller, managing accesses from the eDMA engine and references from the internal peripheral bus. If there are simultaneous accesses, the eDMA engine is given priority and the peripheral transaction is stalled.
Memory array	TCD storage for each channel's transfer profile.

5.2.3 Features

- Facilitates data movement through dual-address transfers read from source, write to destination plus a write-only initialization mode:
  - Programmable source and destination addresses and transfer size
  - Support for complex address calculations
- Supports 64-channel implementation that performs complex data transfers with minimal intervention from a host processor:
  - Internal data buffer, used as temporary storage to support large multibyte transfers
  - Connections to the interconnect fabric for bus mastering the data movement
- Supports a TCD that is organized to support two-deep, nested transfer operations:
  - 32-byte TCD stored in local memory for each channel

  - An inner data transfer loop defined by a minor byte transfer count
  - An outer data transfer loop defined by a major iteration count
- Initializes channel activation through one of the following methods:
  - Explicit software initiation
  - Initiation through channel-to-channel linking mechanism for continuous transfers
  - Peripheral-paced hardware requests, one per channel
- Supports fixed-priority and round-robin channel arbitration
- Reports channel completion reported through programmable interrupt requests:
  - One interrupt per channel, which can be asserted after the completion of a major iteration count
  - Programmable error terminations per channel and logically summed together to form one error interrupt to the interrupt controller
- Provides programmable support for the scatter/gather DMA processing
- Provides support for complex data structures

eDMA uses *n* for reference to the channel number.

5.3 eDMA register descriptions

The eDMA's programming model is partitioned into three parts:

- The first part defines the number of registers providing the overall control functions: the management page.
- The second part corresponds to the channel (CH) control, status, and configuration.
- The third part corresponds to the local TCD memory.

5.3.1 TCD memory

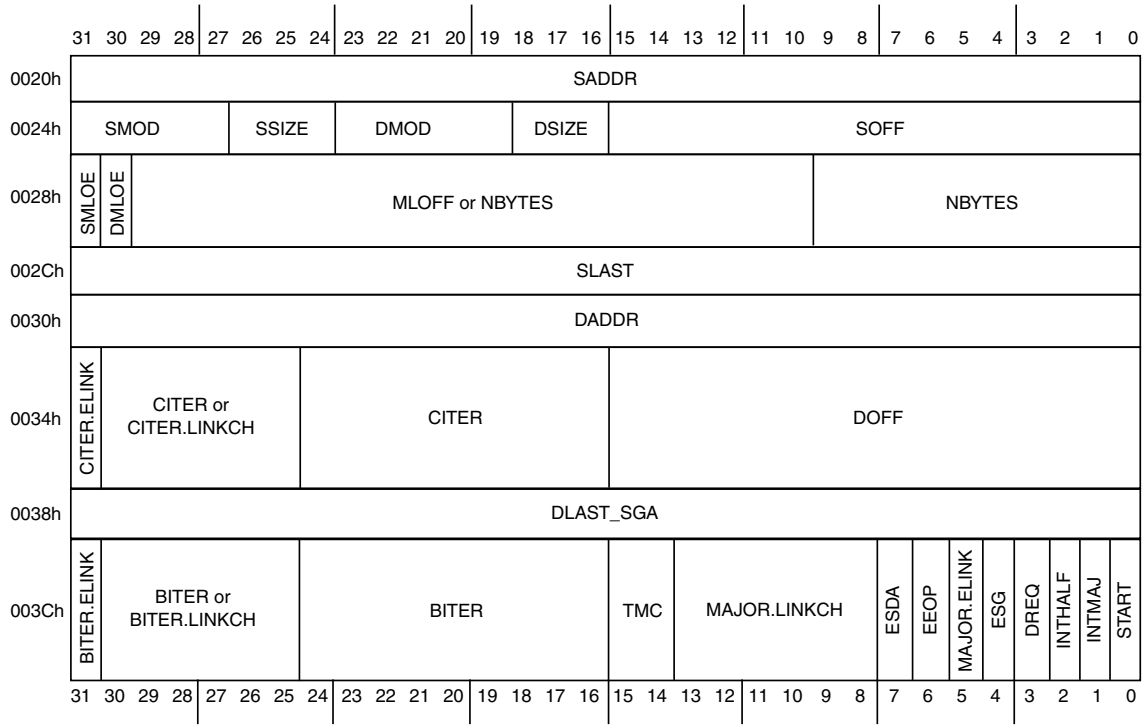
Each channel requires a 32-byte TCD for defining the desired data movement operation. The local memory stores the channel descriptors in a sequential order: channel 0, channel 1, ... channel 63.

This table provides a 32-bit view of the eDMA memory map.

5.3.2 TCD initialization

The TCD memory is in an unknown state after reset. Only the TCD START bit is initialized to 0. Before activating a channel, you must initialize its TCD with the appropriate transfer profile.

5.3.3 TCD structure



5.3.3.1 Reserved memory and bit fields

- Reading reserved bits in a register returns the value of zero.
- Writes to reserved bits in a register are ignored.
- Reading or writing to a reserved memory location generates a bus error.

5.3.4 EDMA4_MP register descriptions

5.3.4.1 EDMA4_MP memory map

EDMA4_2.MP base address: 4200_0000h

NOTE

For registers in the following table with *Protection*, see the REG_PROT details for more information.

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)	Protection
0	Management Page Control (CSR)	32	RW	0040_0000	No
4	Management Page Error Status (ES)	32	R	0000_0000	No
8	Management Page Interrupt Request Status Register - Low (INT_LOW)	32	R	0000_0000	No

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)	Protection
C	Management Page Interrupt Request Status Register-High (INT_HIGH)	32	R	0000_0000	No
10	Management Page Hardware Request Status Register - Low (HRS_LOW)	32	R	0000_0000	No
14	Management Page Hardware Request Status Register - High (HRS_HIGH)	32	R	0000_0000	No
100 - 1FC	Channel Arbitration Group (CH0_GRPRI - CH63_GRPRI)	32	RW	0000_0000	No

5.3.4.2 Management Page Control (CSR)

Defines the basic operating configuration of DMA.

Arbitration uses a two-tier priority system: group and channel priority. Each channel is assigned a priority group. Group arbitration is fixed-priority and cannot be changed. Channel arbitration uses fixed-priority, and you can configure it to use a selective round-robin scheme for specified channels within each priority group.

For fixed-priority arbitration, select the highest-priority channel requesting service in the highest-priority arbitration group to execute. The channel priority registers assign the relative priorities within each arbitration group (see Channel Priority (CH0_PRI - CH63_PRI)). All channels with a nonzero value in Channel Priority (CH0_PRI - CH63_PRI) use fixed-priority arbitration.

When you enable round-robin arbitration, any channel with its channel priority set to 0 does not have a priority. The channels requesting service are cycled through (from high to low channel number) without considering the relative priority within the same priority group. Any channel with a nonzero value in Channel Priority (CH0_PRI - CH63_PRI) automatically has a higher priority over the round-robin channels. A channel's priority group is assigned in Channel Arbitration Group (CH0_GRPRI - CH63_GRPRI).

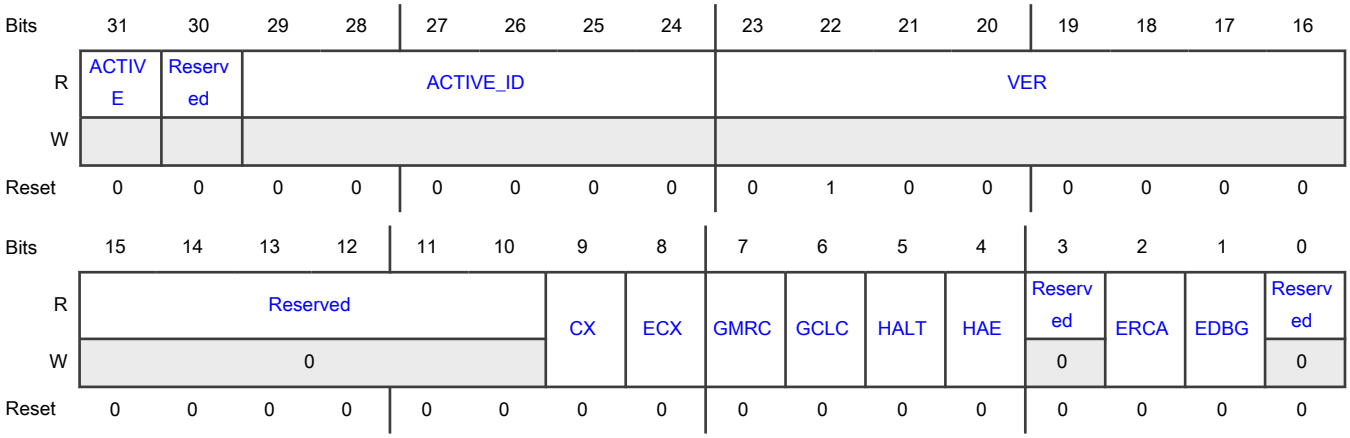
NOTE

For correct operation, perform changes to ERCA, GCLC, and GMRC when the DMA channels are inactive; that is, when ACTIVE is 0.

Offset

Register	Offset
CSR	0h

Diagram



Fields

Field	Description
31 ACTIVE	DMA Active Status Indicates whether eDMA is executing a channel. 0 - Idle 1 - Executing
30 —	Reserved
29-24 ACTIVE_ID	Active Channel ID Indicates the active channel ID.
23-16 VER	eDMA Version Indicates the eDMA version.
15-10 —	Reserved
9 CX	Cancel Transfer Cancels the remaining data transfer. If this field is 1, it stops the executing channel and forces the minor loop to finish. The transfer cancelation takes effect after the last write of the current read/write sequence. The field becomes 0 after the cancel is honored. This cancel retires the channel normally as if the minor loop is completed. 0 - Normal operation 1 - Cancel transfer
8 ECX	Cancel Transfer with Error

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Cancels the remaining data transfer (similar to CX) and treats the transfer cancelation as an error condition, updates the Error Status register (DMA*n*_ES), and generates an optional error interrupt. If this field is 1, it stops the executing channel and forces the minor loop to finish. The transfer cancelation takes effect after the last write of the current read/write sequence. This field becomes 0 after the cancelation is honored. 0 - Normal operation 1 - Cancel transfer
7 GMRC	Global Master ID Replication Control Specifies whether the master ID replication is available. If this field is 1, the master ID replication is available and is controlled by the configuration of CH0_SBR[EMI] for each channel. **NOTE** If master ID replication is disabled, the nonsecure, privileged protection level for DMA transfers is used. 0 - Not available 1 - Available
6 GCLC	Global Channel Linking Control Specifies whether channel linking is available and controlled by each channel's link settings. 0 - Disable 1 - Available
5 HALT	Halt DMA Operations Specifies whether DMA operations are halted. If this field is 1, it stalls the start of any new channels. The channels in execution are allowed to complete. Channel execution resumes when you write 0 to this field. 0 - Normal operation 1 - Halt
4 HAE	Halt after Error Specifies whether the operation halts after an error. If this field is 1, an error causes HALT to become 1. Subsequently, all service requests are ignored until HALT becomes 0. 0 - Normal operation 1 - Halt
3 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 ERCA	Enable Round-Robin Channel Arbitration Enables round-robin channel arbitration. If this field is 0, round-robin channel arbitration is disabled and fixed-priority arbitration is used for channel selection within each group. If this field is 1, round-robin channel arbitration is enabled and is used for channel selection within each group. 0 - Disable 1 - Enable
1 EDBG	Enable Debug Enables Debug mode. If this field is 0, Debug mode is disabled. When in Debug mode, DMA continues to operate. If this field is 1, Debug mode is enabled. When in Debug mode, DMA stalls the start of a new channel. Executing channels are allowed to complete. Channel execution resumes when the system exits Debug mode or when this field becomes 0. 0 - Disable 1 - Enable
0 —	Reserved

5.3.4.3 Management Page Error Status (ES)

Provides information concerning the last recorded channel error, which can be caused by:

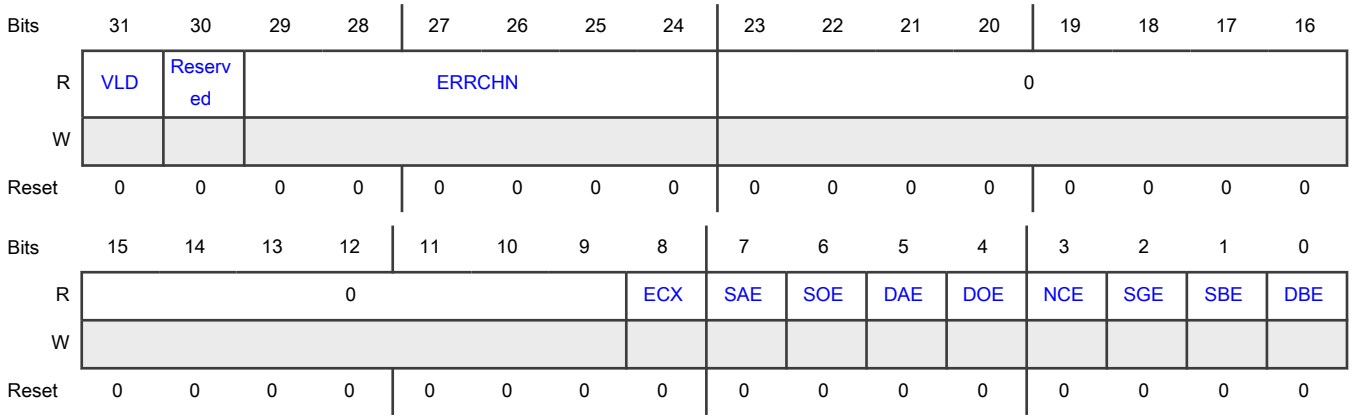
- An illegal setting in TCD.
- An error termination to a bus controller read or write cycle.
- An uncorrectable error occurring when the chip is accessing the TCD SRAM.
- A cancel transfer with error request made through the corresponding cancel transfer bit or input signal.

After an error condition occurs, you must initialize the errant channel's TCD as it is in an incomplete state after an error. See Fault reporting and handling for details.

Offset

Register	Offset
ES	4h

Diagram



Fields

Field	Description
31 VLD	Valid Indicates the logical OR of all ERR status bits. If this field is 1, it indicates that at least one ERR bit is set, indicating an existing valid error that has not been cleared. 0 - No bits are set 1 - At least one bit is set
30 —	Reserved
29-24 ERRCHN	Error Channel Number or Canceled Channel Number Indicates the channel number of the last recorded error or last recorded error canceled transfer.
23-9 —	Reserved
8 ECX	Transfer Canceled Indicates whether the last recorded entry is a canceled transfer by the error cancel transfer input. The ECX operation is a management page function. When this operation is used, the targeted channel's Channel Error Status (CH0_ES - CH63_ES) register reports an unspecified error (only ERR becomes 1). The management page has a complete view of the error condition. 0 - No canceled transfers 1 - Canceled transfers
7 SAE	Source Address Error Indicates whether the last recorded error is a configuration error detected in TCD Source Address (TCD0_SADDR - TCD63_SADDR), which is inconsistent with TCD0_ATTR[SSIZE]. 0 - No error detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Error detected
6 SOE	Source Offset Error Indicates whether the last recorded error is a configuration error detected in TCD Signed Source Address Offset (TCD0_SOFF - TCD63_SOFF), which is inconsistent with TCD0_ATTR[SSIZE]. 0 - No error detected 1 - Error detected
5 DAE	Destination Address Error Indicates whether the last recorded error is a configuration error detected in TCD Destination Address (TCD0_DADDR - TCD63_DADDR), which is inconsistent with TCD0_ATTR[DSIZE]. 0 - No error detected 1 - Error detected
4 DOE	Destination Offset Error Indicates whether the last recorded error is a configuration error detected in TCD Signed Destination Address Offset (TCD0_DOFF - TCD63_DOFF), which is inconsistent with TCD0_ATTR[DSIZE]. 0 - No error detected 1 - Error detected
3 NCE	NBYTES or CITER Configuration Error Indicates whether NBYTES or CITER configuration error occurred. If this field is 1, the last recorded error is NBYTES equal to zero or a CITER not equal to BITER. Also, the last recorded error is a configuration error detected in TCD*n*_NBYTES or TCD*n*_CITER fields. The configuration error on TCD*n*_NBYTES occurs when TCD*n*_NBYTES is not configured as an integer multiple of TCD0_ATTR[SSIZE] and TCD0_ATTR[DSIZE]. The configuration error on TCD*n*_CITER occurs when TCD*n*_CITER[CITER] is equal to zero or when TCD*n*_CITER[CITER] is not equal to TCD*n*_BITER[ELINK]. 0 - No error detected 1 - Error detected
2 SGE	Scatter/Gather Configuration Error Indicates whether a scatter/gather configuration error is detected. If this field is 1, the last recorded error is a configuration error detected in TCD0_DLAST_SGA[DLAST_SGA]. This field is checked at the beginning of a scatter/gather operation after major loop completion if you enable TCD0_CSR[ESG]. TCD*n*_DLAST_SGA is not on a 32-byte boundary. 0 - No error detected 1 - Error detected
1	Source Bus Error

Table continues on the next page...

Table continued from the previous page...

Field	Description
SBE	Indicates whether the last recorded error is a bus error on a source read. 0 - No error 1 - Error
0 DBE	Destination Bus Error Indicates whether the last recorded error is a bus error on a destination write. 0 - No error 1 - Error

5.3.4.4 Management Page Interrupt Request Status Register - Low (INT_LOW)

This register shows the current state of the interrupt service request for all eDMA channels.

Offset

Register	Offset
INT_LOW	8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 INT	Interrupt Request Status for channels 31 - 0 The INT register presents the interrupt request status for each eDMA channel. Depending on the appropriate bit setting in the transfer-control descriptors, the eDMA engine generates an interrupt on data transfer completion . The eDMA channel interrupt requests are routed to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit in the channel’s interrupt request register, CHn_INT, thus negating the interrupt request.

Table continues on the next page...

Field	Description
	• 0 - An interrupt request for the corresponding channel is not present. • 1 - An interrupt request for the corresponding channel is present.

5.3.4.5 Management Page Interrupt Request Status Register- High (INT_HIGH)

This register shows the current state of the interrupt service request for all eDMA channels.

Offset

Register	Offset
INT_HIGH	Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 INT	Interrupt Request Status for channels 63-32 The INT register presents the interrupt request status for each eDMA channel. Depending on the appropriate bit setting in the transfer-control descriptors, the eDMA engine generates an interrupt on data transfer completion . The eDMA channel interrupt requests are routed to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit in the channel's interrupt request register, CHn_INT, thus negating the interrupt request. • 0 - An interrupt request for the corresponding channel is not present. • 1 - An interrupt request for the corresponding channel is present.

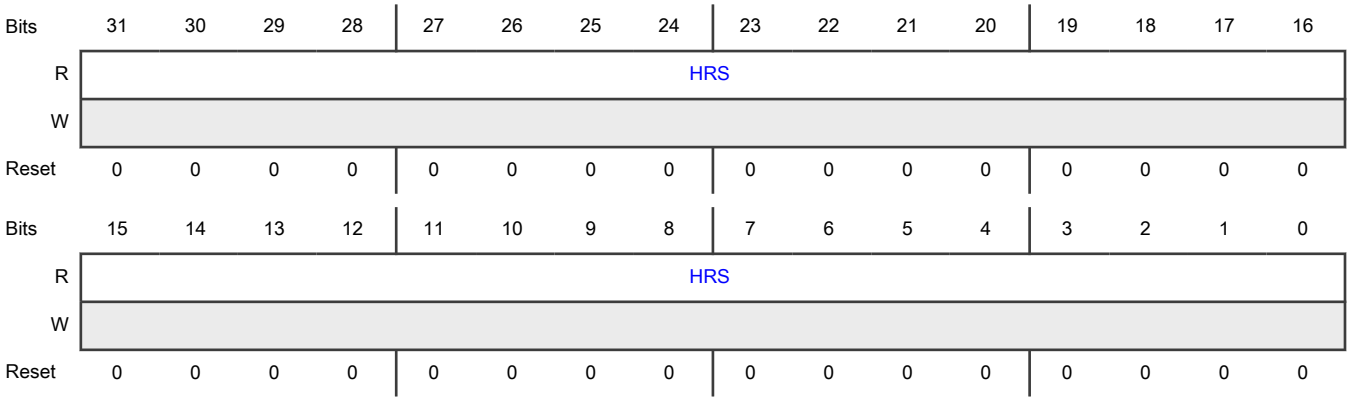
5.3.4.6 Management Page Hardware Request Status Register - Low (HRS_LOW)

The hardware request status register (HRS) shows the current state of the hardware service request signaling as seen by the eDMA's arbitration logic.

Offset

Register	Offset
HRS_LOW	10h

Diagram



Fields

Field	Description
31-0 HRS	Hardware Request Status for channels 31 - 0 The HRS bit for its respective channel remains asserted for the period when a hardware request is present on the channel. After the request is completed and channel is free, the HRS bit is automatically cleared by hardware. 0000_0000_0000_0000_0000_0000_0000_0000 - A hardware service request for the channel is not present 0000_0000_0000_0000_0000_0000_0000_0001 - A hardware service request for channel 0 is present

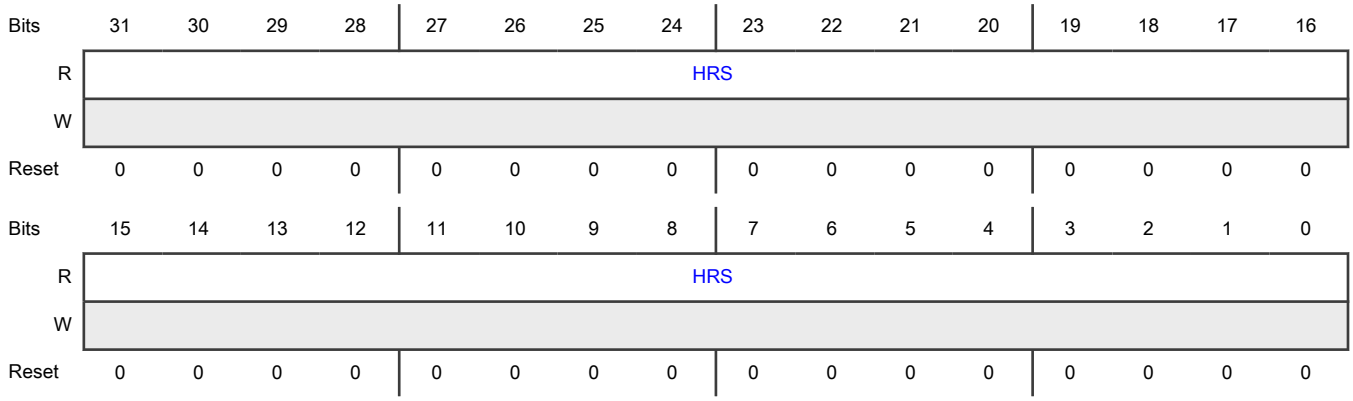
5.3.4.7 Management Page Hardware Request Status Register - High (HRS_HIGH)

The hardware request status register (HRS) shows the current state of the hardware service request signaling as seen by the eDMA's arbitration logic.

Offset

Register	Offset
HRS_HIGH	14h

Diagram



Fields

Field	Description
31-0 HRS	Hardware Request Status for channels 63-32 The HRS bit for its respective channel remains asserted for the period when a hardware request is present on the channel. After the request is completed and channel is free, the HRS bit is automatically cleared by hardware. 0000_0000_0000_0000_0000_0000_0000_0000 - A hardware service request for the channel is not present 0000_0000_0000_0000_0000_0000_0000_0001 - A hardware service request for channel 0 is present

5.3.4.8 Channel Arbitration Group (CH0_GRPRI - CH63_GRPRI)

Defines the arbitration group associated with each channel.

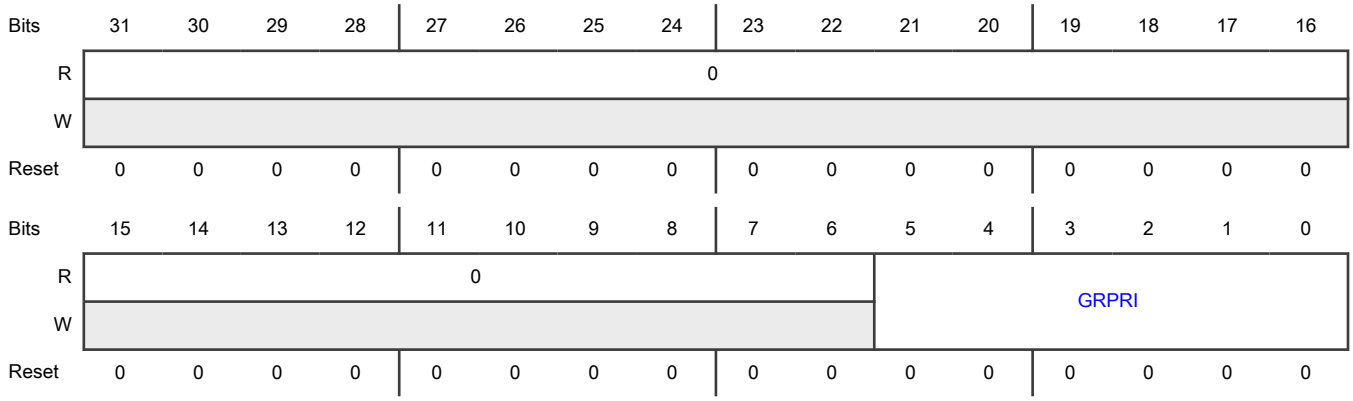
The arbitration group priorities are evaluated by a numeric value from the highest group number to the lowest using the fixed-priority group arbitration scheme; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. The range of the group priority values is limited to the values of 0 through 31. Within each arbitration group, Channel Priority (CH0_PRI - CH63_PRI) determines the highest priority channel.

Offset

For n = 0 to 63:

Register	Offset
CHn_GRPRI	100h + (n × 4h)

Diagram



Fields

Field	Description
31-6 —	Reserved
5-0 GRPRI	Arbitration Group Per Channel Specifies the fixed-priority arbitration group number.

5.3.5 EDMA4_TCD register descriptions

5.3.5.1 EDMA4_TCD memory map

EDMA4_2.TCD base address: 4201_0000h

NOTE

For registers in the following table with *Protection*, see the REG_PROT details for more information.

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)	Protection
0 - 3_F000	Channel Control and Status (CH0_CSR - CH63_CSR)	32	RW	0000_0000	No
4 - 3_F004	Channel Error Status (CH0_ES - CH63_ES)	32	RW	0000_0000	No
8 - 3_F008	Channel Interrupt Status (CH0_INT - CH63_INT)	32	RW	0000_0000	No
C - 3_F00C	Channel System Bus (CH0_SBR - CH63_SBR)	32	RW	0000_8007	No
10 - 3_F010	Channel Priority (CH0_PRI - CH63_PRI)	32	RW	0000_0000	No
14 - 3_F014	Channel Multiplexor Configuration (CH0_MUX - CH63_MUX)	32	RW	0000_0000	No

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)	Protection
18 - 3_F018	Memory Attributes (CH0_MATTR - CH63_MATTR)	16	RW	0000	No
20 - 3_F020	TCD Source Address (TCD0_SADDR - TCD63_SADDR)	32	RW	See section	No
24 - 3_F024	TCD Signed Source Address Offset (TCD0_SOFF - TCD63_SOFF)	16	RW	See section	No
26 - 3_F026	TCD Transfer Attributes (TCD0_ATTR - TCD63_ATTR)	16	RW	See section	No
28 - 3_F028	TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD63_NBYTES_MLOFFNO)	32	RW	See section	No
28 - 3_F028	TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD63_NBYTES_MLOFFYES)	32	RW	See section	No
2C - 3_F02C	TCD Last Source Address Adjustment and Store DADDR Address (TCD0_SLAST_SDA - TCD63_SLAST_SDA)	32	RW	See section	No
30 - 3_F030	TCD Destination Address (TCD0_DADDR - TCD63_DADDR)	32	RW	See section	No
34 - 3_F034	TCD Signed Destination Address Offset (TCD0_DOFF - TCD63_DOFF)	16	RW	See section	No
36 - 3_F036	TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD63_CITER_ELINKNO)	16	RW	See section	No
36 - 3_F036	TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD63_CITER_ELINKYES)	16	RW	See section	No
38 - 3_F038	TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD63_DLAST_SGA)	32	RW	See section	No
3C - 3_F03C	TCD Control and Status (TCD0_CSR - TCD63_CSR)	16	RW	See section	No
3E - 3_F03E	TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD63_BITER_ELINKNO)	16	RW	See section	No
3E - 3_F03E	TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD63_BITER_ELINKYES)	16	RW	See section	No

5.3.5.2 Channel Control and Status (CH0_CSR - CH63_CSR)

Provides channel control and status.

The DMA hardware request input signal must be asserted and ERQ must be 1 before a channel's hardware service request is accepted. The state of ERQ does not affect a channel's service request made explicitly through software or channel linking. The state of ERQ does not affect the channel's START field.

EARQ does not affect DMA operations. If this field is 1, this field allows ERQ to propagate out of DMA to the power controller. When 0, this field masks ERQ to the power controller.

EEI enables the error interrupt signal for the channel. The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller.

DONE indicates whether the channel has completed the major loop. If enabled, eDMA generates an interrupt request corresponding to this completed channel.

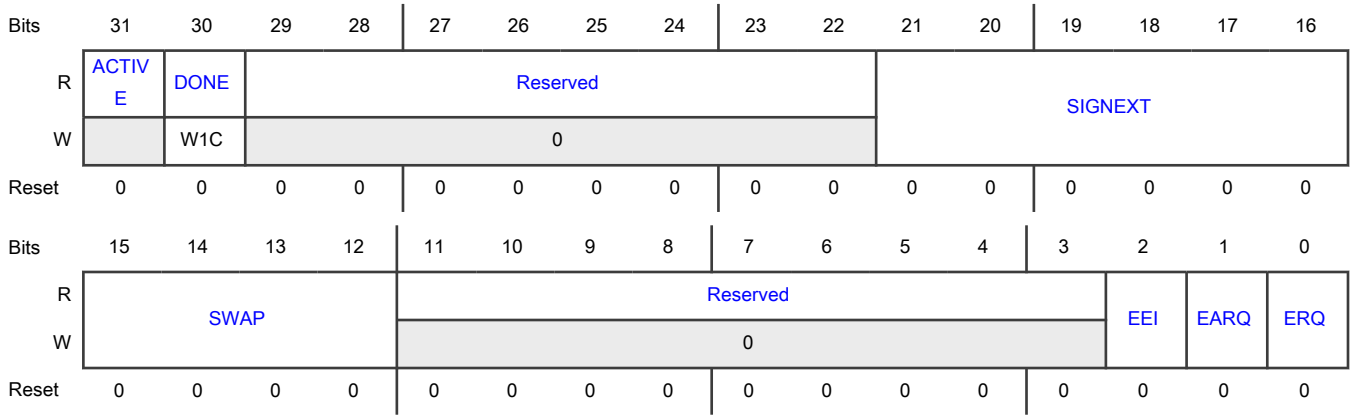
ACTIVE indicates whether the channel is selected by arbitration and is executing the prescribed transfers. Except for dynamic channel linking, you must not modify TCD when a channel is active.

Offset

For n = 0 to 63:

Register	Offset
CHn_CSR	0h + (n × 1000h)

Diagram



Fields

Field	Description
31 ACTIVE	Channel Active Indicates whether the channel is currently in execution. This field becomes 1 when channel service begins; eDMA writes 0 to it when the minor loop completes or when any error condition is detected.
30	Channel Done

Table continues on the next page...

Table continued from the previous page...

Field	Description
DONE	Indicates whether eDMA has completed the major loop. The eDMA engine sets this flag when the CITER count reaches 0. You or hardware can clear the flag when the channel is activated. **NOTE** Clear this flag to write 1 to MAJORELINK. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not done 1 - Done When writing 0 - No effect 1 - Clear the flag
29-22 —	Reserved
21-16 SIGNEXT	Sign Extension Specifies whether the data written on each transaction is sign extended at the bit specified in this field. 00_0000 - Sign extension disabled 00_0001 - Nonzero value specifying the sign extend bit position
15-12 SWAP	Swap Size Swaps data elements within the specified transfer. The swapping occurs with respect to the specified transfer size (TCD0_ATTR[SSIZE] or TCD0_ATTR[DSIZE]). 0000 - Disabled 0001 - Read with 8-bit swap 0010 - Read with 16-bit swap 0011 - Read with 32-bit swap 0100-1000 - Reserved 1001 - Write with 8-bit swap 1010 - Write with 16-bit swap 1011 - Write with 32-bit swap 1100-1111 - Reserved
11-3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
2 EEI	Enable Error Interrupt Specifies whether the assertion of the error signal for the corresponding channel generates an error interrupt request. 0 - Do not generate 1 - Generate
1 EARQ	Enable Asynchronous DMA Request Enables asynchronous DMA request for the channel. 0 - Disable 1 - Enable
0 ERQ	Enable DMA Request Enables DMA hardware request for the corresponding channel. Disable the channel's hardware service request at the source prior to writing 0 to this field for the corresponding channel. 0 - Disable 1 - Enable

5.3.5.3 Channel Error Status (CH0_ES - CH63_ES)

Provides information concerning the last recorded channel error. Channel errors can occur because of the following:

- An illegal setting in TCD
- An error termination to a bus master read or write cycle

ERR signals the presence of an error for the channel. The eDMA engine signals the occurrence of an error condition by configuring the appropriate fields in this register. The outputs of this register are enabled by the contents of CH0_CSR[EEI], and then logically summed across all channels to form an error interrupt request, which can be routed to the interrupt controller.

During execution of the interrupt-service routine associated with any DMA errors, it is your responsibility to clear the appropriate field, negating the error-interrupt request. The normal DMA channel completion indicators (setting the TCD DONE flag and the possible assertion of an interrupt request) are not affected when an error is detected. The contents of ERR can also be polled because a nonzero value indicates the presence of a channel error regardless of the state of the EEI mask.

The state of any given channel's error indicators is affected by writes to this register. Writing 1 to ERR clears the channel's error status. Writing 0 has no effect on the channel's current error status.

An unspecified error, where only ERR becomes 1 indicates any of the following:

- A transfer is canceled with an error

Management Page Error Status (ES) has a full view of the error condition.

See Fault reporting and handling for details.

Offset

For n = 0 to 63:

Register	Offset
CHn_ES	4h + (n × 1000h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR	Reserved														
W	W1C	0														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								SAE	SOE	DAE	DOE	NCE	SGE	SBE	DBE
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 ERR	Error in Channel Indicates whether an error occurred in this channel. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not occurred 1 - Occurred When writing 0 - No effect 1 - Clear the flag
30-8 —	Reserved
7 SAE	Source Address Error Indicates whether the last recorded error is a configuration error detected in TCD Source Address (TCD0_SADDR - TCD63_SADDR), which is inconsistent with TCD0_ATTR[SSIZE]. 0 - No error detected 1 - Error detected
6 SOE	Source Offset Error

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Indicates whether the last recorded error is a configuration error detected in TCD Signed Source Address Offset (TCD0_SOFF - TCD63_SOFF), which is inconsistent with TCD0_ATTR[SSIZE]. 0 - No error detected 1 - Error detected
5 DAE	Destination Address Error Indicates whether the last recorded error is a configuration error detected in TCD Destination Address (TCD0_DADDR - TCD63_DADDR), which is inconsistent with TCD0_ATTR[DSIZE]. 0 - No error detected 1 - Error detected
4 DOE	Destination Offset Error Indicates whether the last recorded error is a configuration error detected in TCD Signed Destination Address Offset (TCD0_DOFF - TCD63_DOFF), which is inconsistent with TCD0_ATTR[DSIZE]. 0 - No error detected 1 - Error detected
3 NCE	NBYTES/CITER Configuration Error Indicates whether NBYTES or CITER configuration error occurred. If this field is 1, the last recorded error is a configuration error detected in TCD*n*_NBYTES or TCD*n*_CITER fields. The configuration error on TCD*n*_NBYTES occurs when TCD*n*_NBYTES is not configured as an integer multiple of TCD0_ATTR[SSIZE] and TCD0_ATTR[DSIZE]. The configuration error on TCD*n*_CITER occurs when TCD*n*_CITER[CITER] is equal to zero or when TCD*n*_CITER[CITER] is not equal to TCD*n*_BITER[ELINK]. 0 - No error occurred 1 - Error occurred
2 SGE	Scatter/Gather Configuration Error Indicates whether a scatter/gather configuration error is detected. If this field is 1, the last recorded error is a configuration error detected in TCD0_DLAST_SGA[DLAST_SGA]. This field is checked at the beginning of a scatter/gather operation after major loop completion, if you enable TCD0_CSR[ESG]. TCD*n*_DLASTSGA is not on a 32-byte boundary. 0 - No error detected 1 - Error detected
1 SBE	Source Bus Error Indicates whether the last recorded error is a bus error on a source read. 0 - No error 1 - Error

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 DBE	Destination Bus Error Indicates whether the last recorded error is a bus error on a destination write. 0 - No error 1 - Error

5.3.5.4 Channel Interrupt Status (CH0_INT - CH63_INT)

Indicates the presence of an interrupt request for the channel.

Depending on the appropriate field setting in TCDs, the eDMA engine generates an interrupt on data transfer completion. The outputs of this register are directly routed to the interrupt controller. During the interrupt service routine associated with any given channel, you must clear the appropriate bit to negate the interrupt request.

Offset

For n = 0 to 63:

Register	Offset
CHn_INT	8h + (n × 1000h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															INT
W	0															W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-1 —	Reserved
0 INT	Interrupt Request Indicates whether the interrupt request for the corresponding channel is active.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - Cleared 1 - Active When writing 0 - No effect 1 - Clear the flag

5.3.5.5 Channel System Bus (CH0_SBR - CH63_SBR)

Places identification and protection information on the system bus interface for eDMA.

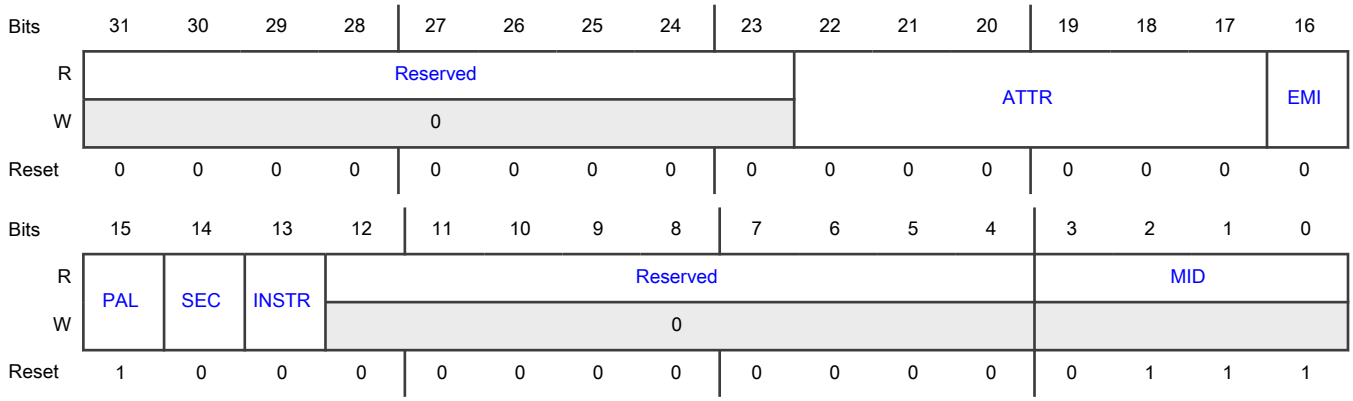
ATTR outputs the register values onto the system bus interface for further decoding by the security system.

Offset

For n = 0 to 63:

Register	Offset
CHn_SBR	Ch + (n × 1000h)

Diagram



Fields

Field	Description
31-23 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
22-17 ATTR	Attribute Output Specifies the DMA's system bus attribute output value.
16 EMI	Enable Master ID Replication Enables master ID replication. This field allows eDMA to use the same protection level and system bus ID of the master programming eDMA's TCD. When you enable this field, eDMA uses the master ID and protection level stored in SBR registers instead of eDMA's default values. When a master, for example a core, programs a TCD, its master ID is captured when TCD*n*_CSR control is written. The protection and security levels are captured when CH*n* _SBR security attributes are written. A scatter/gather operation does not affect Channel System Bus (CH0_SBR - CH63_SBR). You can write to this field only if CSR[GMRC] = 1; otherwise, this field is forced to 0. **NOTE** If you disable master ID replication, the nonsecure, privileged protection level for DMA transfers is used. 0 - Disable 1 - Enable
15 PAL	Privileged Access Level DMA's protection level on the system bus when the channel is active. **NOTE** The value written into this register cannot exceed the security and privilege level of the core or other master writing the channel's system bus register; CH *n* _SBR. The order of precedence is SecurePriv>SecureUser>NonsecurePriv>NonsecureUser 0 - User protection level for DMA transfers 1 - Privileged protection level for DMA transfers
14 SEC	Security Level DMA's security level on the system bus when the channel is active. **NOTE** The value written into this register cannot exceed the security and privilege level of the core or other master writing the channel's system bus register; CH *n* _SBR. The order of precedence is SecurePriv>SecureUser>NonsecurePriv>NonsecureUser 0 - Nonsecure protection level for DMA transfers 1 - Secure protection level for DMA transfers
13 INSTR	Instruction/Data Access DMA's transactions type on the system bus when the channel is active. 0 - Data access for DMA transfers

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Instruction access for DMA transfers
12-4 —	Reserved
3-0 MID	Master ID Indicates the DMA master ID on the system bus when the channel is active. **NOTE** When you enable EMI, the ID captured in this field reflects the master ID of the core or another master writing the channel's control attributes (TCD Control and Status (TCD0_CSR - TCD63_CSR)).

5.3.5.6 Channel Priority (CH0_PRI - CH63_PRI)

Defines the unique priorities associated with each channel within the same channel group.

Channel grouping is programmed through the group priority registers, CH*n*_GRPRI. The channel priorities within a group are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. Program the channel priorities with unique values; otherwise, channel numbers with the same nonzero value are selected based on the higher channel number having the higher priority. If more than one channel in a group has an arbitration priority level value of 0, then CSR[ERCA] is used to determine the arbitration scheme for all channels with APL = 0 within a group. When round-robin channel arbitration is enabled (CSR[ERCA] = 1), all channels with APL = 0 within a group use a round-robin arbitration scheme that rotates among these channels requesting service without regard to priority. Round-robin provides a fairness mechanism within an arbitration group. When fixed-priority channel arbitration is enabled (CSR[ERCA] = 0), channels with APL = 0 are selected based on the higher channel number having higher priority.

Offset

For n = 0 to 63:

Register	Offset
CHn_PRI	10h + (n × 1000h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ECP		DPA		Reserved											
W					0											
Reset	0		0		0		0		0		0		0		0	

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												APL			
W	0															
Reset	0		0		0		0		0		0		0	0[1]	0	

1. See bit field description.

Fields

Field	Description
31 ECP	Enable Channel Preemption Specifies whether the channel can be temporarily suspended by a service request from a higher priority channel. 0 - Do not suspend 1 - Suspend temporarily
30 DPA	Disable Preempt Ability Specifies whether the channel can suspend any other channel, regardless of channel priority. 0 - Can suspend a lower-priority channel 1 - Cannot suspend any other channel
29-3 —	Reserved
2-0 APL	Arbitration Priority Level Specifies the channel priority level for arbitration within the assigned arbitration group.

5.3.5.7 Channel Multiplexor Configuration (CH0_MUX - CH63_MUX)

Each of the DMA channels can be independently associated with various peripherals in the system. The Channel Multiplexor Configuration register selects the peripheral assigned to each channel. Service requests from the peripheral should be disabled when configuring a channel to a peripheral source.

Each channel must have a unique value when selecting a peripheral slot in the channel mux configuration. The only value that may overlap is source 0. If there is an attempt to write a mux configuration value that is already consumed by another channel, a mux configuration of 0 (SRC = 0) will be written.

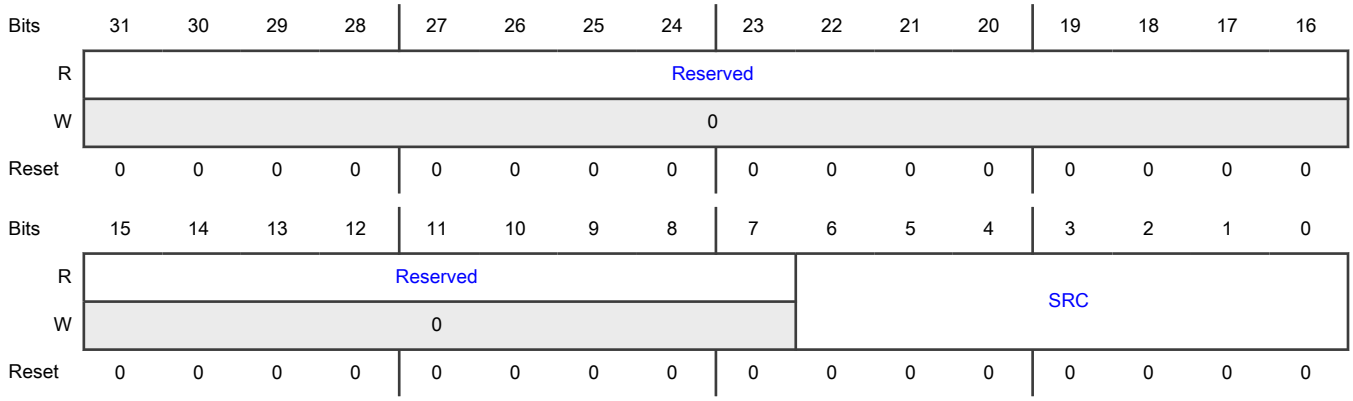
All channels will default to source 0. When a particular peripheral is needed, the channel's mux configuration is set to that source number. When the peripheral is no longer needed, the mux configuration for that channel should be written to 0, thus releasing the resource.

Offset

For n = 0 to 63:

Register	Offset
CHn_MUX	14h + (n × 1000h)

Diagram



Fields

Field	Description
31-7 —	Reserved
6-0 SRC	Service Request Source Hardware service request source for the channel. **NOTE** With the exception of 0, attempts to write a value already in use will be forced to 0.

5.3.5.8 Memory Attributes (CH0_MATTR - CH63_MATTR)

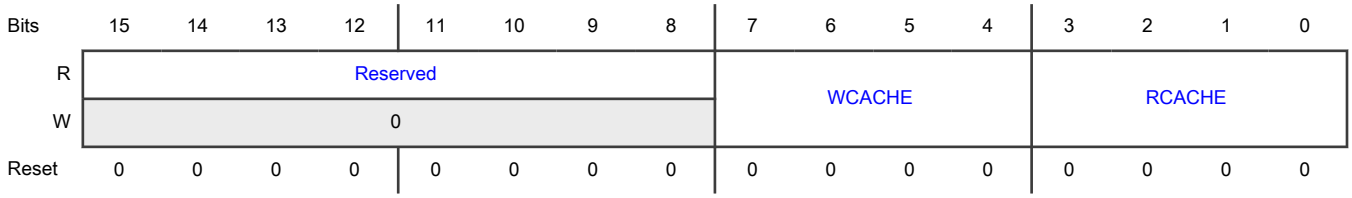
Defines a set of transaction attributes for memory and peripherals. The cache attributes specify how a transaction progresses through the system and how they are handled.

Offset

For n = 0 to 63:

Register	Offset
CHn_MATTR	18h + (n × 1000h)

Diagram



Fields

Field	Description
15-8 —	Reserved
7-4 WCACHE	Write Cache Attributes Defines the attributes associated with a write transaction. • [7] 0b: no write allocate; 1b: write-allocate • [6] 0b: no read allocate; 1b: read-allocate • [5] 0b: non-cacheable; 1b: cacheable • [4] 0b: non-bufferable; 1b: bufferable
3-0 RCACHE	Read Cache Attributes Defines the attributes associated with a read transaction. • [3] 0b: no write allocate; 1b: write-allocate • [2] 0b: no read allocate; 1b: read-allocate • [1] 0b: non-cacheable; 1b: cacheable • [0] 0b: non-bufferable; 1b: bufferable

5.3.5.9 TCD Source Address (TCD0_SADDR - TCD63_SADDR)

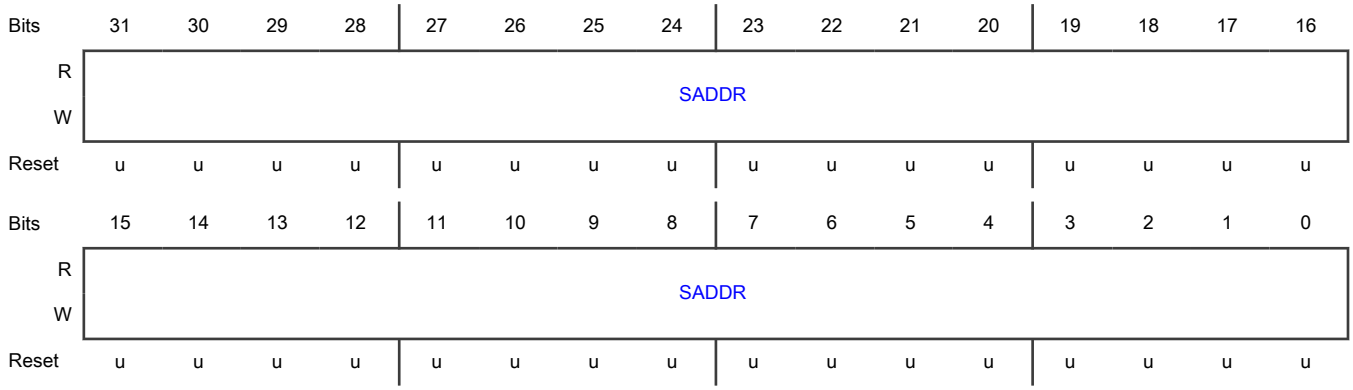
Contains the address for the read transactions.

Offset

For n = 0 to 63:

Register	Offset
TCDn_SADDR	20h + (n × 1000h)

Diagram



Fields

Field	Description
31-0 SADDR	Source Address Specifies the memory address pointing to the source data.

5.3.5.10 TCD Signed Source Address Offset (TCD0_SOFF - TCD63_SOFF)

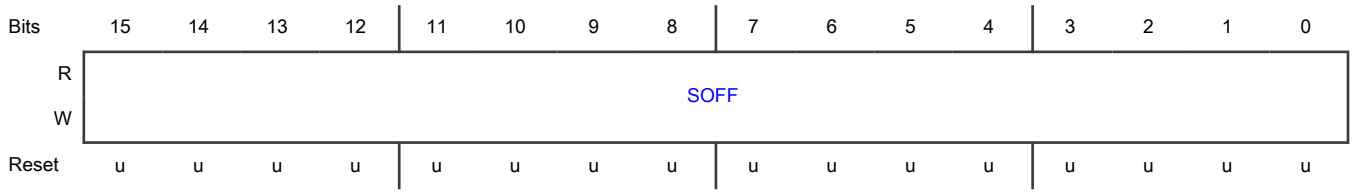
Specifies the sign-extended value added to TCD0_SADDR[SADDR] after each read transaction.

Offset

For n = 0 to 63:

Register	Offset
TCDn_SOFF	24h + (n × 1000h)

Diagram



Fields

Field	Description
15-0 SOFF	Source Address Signed Offset Specifies the sign-extended offset that applies to the current source address to form the next-state value as each source read completes.

5.3.5.11 TCD Transfer Attributes (TCD0_ATTR - TCD63_ATTR)

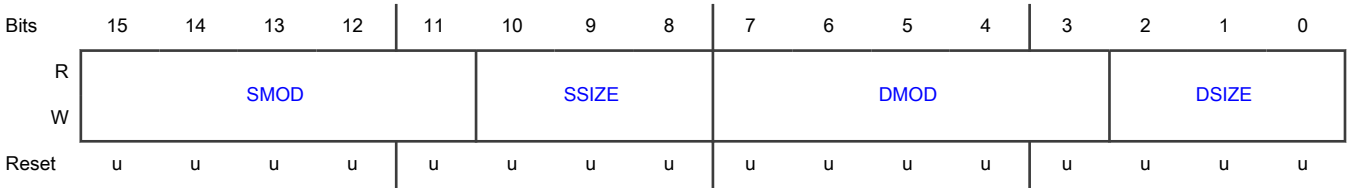
Specifies the size and option modulo addressing information for source and destination addresses.

Offset

For n = 0 to 63:

Register	Offset
TCDn_ATTR	26h + (n × 1000h)

Diagram



Fields

Field	Description
15-11 SMOD	Source Address Modulo Specifies whether the source address modulo feature is enabled for any nonzero value [1-31]. This field defines a specific address range specified to be the value after you perform the TCD0_SADDR[SADDR] + TCD0_SOFF[SOFF] calculation on the original register value. Writing 1 to this field provides the ability to implement a circular data queue easily. For data queues requiring power-of-2 size bytes, the queue must start at a 0-modulo-size address and you must set this field to an appropriate value for the queue, freezing the desired number of the upper address bits. The value that you program for this field specifies the number of lower address bits it allows to change. For a circular queue application, TCD0_SOFF[SOFF] is typically set to the transfer size to implement post-increment addressing with the SMOD function constraining the addresses to a 0-modulo-size range. 0_0000 - Disable 0_0001 - Enable
10-8 SSIZE	Source Data Transfer Size Specifies the size of each read transaction on the system bus. **NOTE** NBYTES must be an integer multiple of the source transfer size. 000 - 8-bit 001 - 16-bit 010 - 32-bit 011 - 64-bit 100 - 16-byte

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - 32-byte 110 - 64-byte 111 - 128-byte
7-3 DMOD	Destination Address Modulo Specifies the destination address modulo. See SMOD for more information.
2-0 DSIZE	Destination Data Transfer Size Specifies the size of each write transaction on the system bus. **NOTE** NBYTES must be an integer multiple of the destination transfer size. 000 - 8-bit 001 - 16-bit 010 - 32-bit 011 - 64-bit 100 - 16-byte 101 - 32-byte 110 - 64-byte 111 - 128-byte

5.3.5.12 TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD63_NBYTES_MLOFFNO)

Defines the number of bytes to transfer per service request.

Minor loop offsets are address offset values added to the final source address (TCD Source Address (TCD0_SADDR - TCD63_SADDR)) or destination address (TCD Destination Address (TCD0_DADDR - TCD63_DADDR)) upon minor loop completion, which happens when the channel has finished the service request and has transferred NBYTES. When you enable minor loop offsets, the minor loop offset value (MLOFF) is added to the final source address (TCD*n*_SADDR), to the final destination address (TCD*n*_DADDR), or to both prior to the address being written back to TCD. If the major loop is complete, the minor loop offset is ignored, and the major loop address offsets (TCD*n*_SLAST_SDA and TCD*n*_DLAST_SGA) are used to compute the next TCD*n*_SADDR and TCD*n*_DADDR values.

When you enable minor loop mapping (SMLOE or DMLOE = 1), TCD*n* word2 is redefined. A portion of TCD*n* word2 is used to specify following fields:

- A source enable field (SMLOE) must be applied to the source address (TCD*n*_SADDR) after minor loop completion, to specify the minor loop offset.
- A destination enable field (DMLOE) must be applied to the destination address (TCD*n*_DADDR) after minor loop completion, to specify the minor loop offset.
- A sign-extended minor loop offset value (MLOFF)

The same offset value (MLOFF) is used for both source and destination minor loop offsets. When either of the two minor loop offsets are enabled (SMLOE or DMLOE = 1), NBYTES is reduced to 10 bits. When both minor loop offsets are disabled (SMLOE and DMLOE = 0), NBYTES is a 30-bit vector.

One of the two register profiles (TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD63_NBYTES_MLOFFNO) or TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD63_NBYTES_MLOFFYES)) defines the number of bytes to transfer per request. You can select the register depending on whether the source or destination minor loop mapping is enabled.

TCD word 2 is defined when SMLOE and DMLOE are 0. If either SMLOE or DMLOE is 1, see the description of TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD63_NBYTES_MLOFFYES)).

NOTE

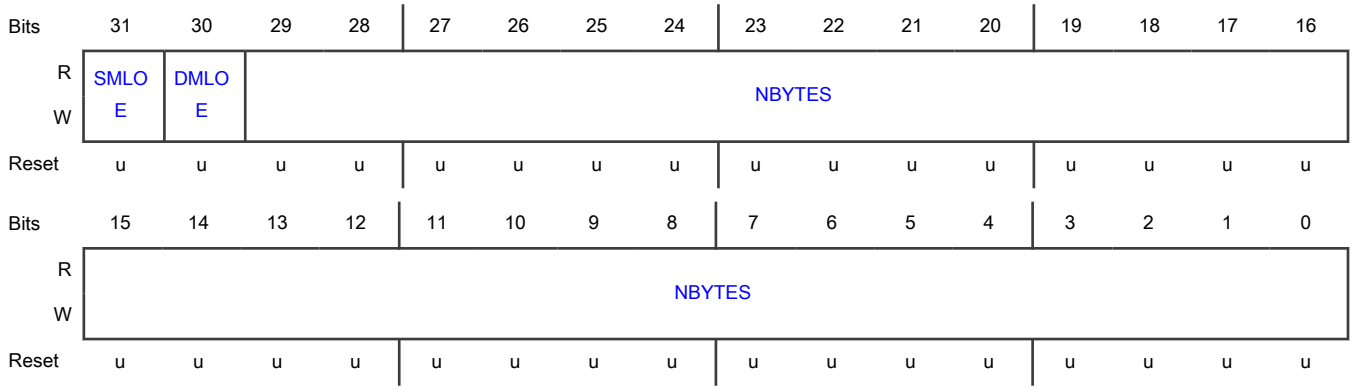
NBYTES must be an integer multiple of the source transfer size and the destination transfer size.

Offset

For n = 0 to 63:

Register	Offset
TCDn_NBYTES_MLOFFNO	28h + (n × 1000h)

Diagram



Fields

Field	Description
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address (SADDR) after minor loop completion. 0 - Do not apply 1 - Apply
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address (DADDR) after minor loop completion. 0 - Do not apply 1 - Apply

Table continues on the next page...

Table continued from the previous page...

Field	Description
29-0 NBYTES	Number of Bytes to Transfer Per Service Request Specifies the number of bytes to be transferred for each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes are performed until the byte transfer count is transferred. This is normally an indivisible operation and you cannot halt the operation. However, you can stall the operation using the bandwidth control field, or through preemption. After the byte count is exhausted, SADDR and DADDR values are written back into the TCD memory, and the major loop iteration count (CITER) is decremented by one and written back to the TCD memory. If the major iteration count is completed, additional processing is performed.

5.3.5.13 TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD63_NBYTES_MLOFFYES)

Defines the number of bytes to transfer per service request.

Minor loop offsets are address offset values added to the final source address (TCD Source Address (TCD0_SADDR - TCD63_SADDR)) or destination address (TCD Destination Address (TCD0_DADDR - TCD63_DADDR)) upon minor loop completion, which happens when the channel has finished the service request and has transferred NBYTES. You can enable minor loop offsets by either writing 1 to SMLOE or to DMLOE.

SMLOE specifies MLOFF that must be applied to the source address (TCD*n*_SADDR) upon minor loop completion. DMLOE specifies MLOFF that must be applied to the destination address (TCD*n*_DADDR) upon minor loop completion. If the major loop is complete, the minor loop offsets are ignored and the major loop address offsets (TCD*n*_SLAST_SDA and TCD*n*_DLAST_SGA) are used to compute the next TCD*n*_SADDR and TCD*n*_DADDR values.

When the minor loop offset overlay is enabled (either SMLOE or DMLOE = 1), TCD*n*_NBYTES_* is redefined. A portion of TCD*n*_NBYTES_* is used to specify the sign extended minor loop offset value (MLOFF). The same offset value (MLOFF) is used for both source and destination minor loop offsets. When enabled, the minor loop offset must be aligned to the transfer size of the source or destination it is associated with. When either of the two minor loop offsets are enabled (SMLOE or DMLOE is 1), NBYTES is reduced to 10 bits. When both minor loop offsets are disabled (SMLOE and DMLOE are 0), NBYTES is a 30-bit vector.

One of the two register profiles (TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD63_NBYTES_MLOFFYES) or TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD63_NBYTES_MLOFFNO)), defines the number of bytes to transfer per request. You can use the register depending on whether the source or destination minor loop mapping is enabled.

TCD*n*_NBYTES is defined when either of the two minor loop offsets are enabled (SMLOE or DMLOE = 1); see TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD63_NBYTES_MLOFFYES) for more information. If SMLOE and DMLOE = 0, see TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD63_NBYTES_MLOFFNO).

NOTE

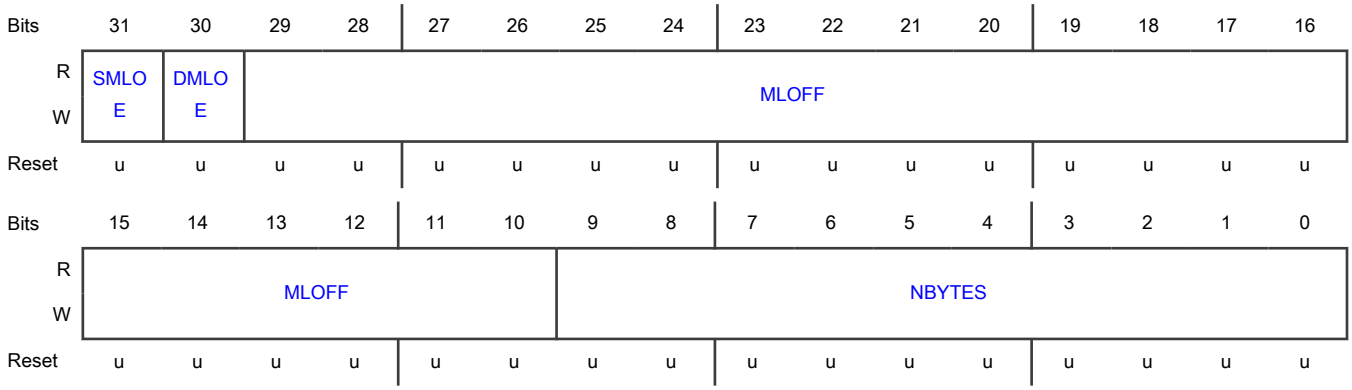
NBYTES must be an integer multiple of the source transfer size and the destination transfer size.

Offset

For n = 0 to 63:

Register	Offset
TCDn_NBYTES_MLOFFYES	28h + (n × 1000h)

Diagram



Fields

Field	Description
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address (SADDR) after minor loop completion. 0 - Do not apply 1 - Apply
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address (DADDR) after minor loop completion. 0 - Do not apply 1 - Apply
29-10 MLOFF	Minor Loop Offset Represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes, if SMLOE or DMLOE = 1.
9-0 NBYTES	Number of Bytes to Transfer Per Service Request Specifies the number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes are performed until the minor byte transfer count is transferred. This is an indivisible operation and you cannot halt the operation. However, you can stall the operation by using the bandwidth control field, or through preemption. After the minor count is exhausted, SADDR and DADDR values are written back into the TCD memory, and the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed.

5.3.5.14 TCD Last Source Address Adjustment and Store DADDR Address (TCD0_SLAST_SDA - TCD63_SLAST_SDA)

Specifies the value added to the source address after the major loop completes.

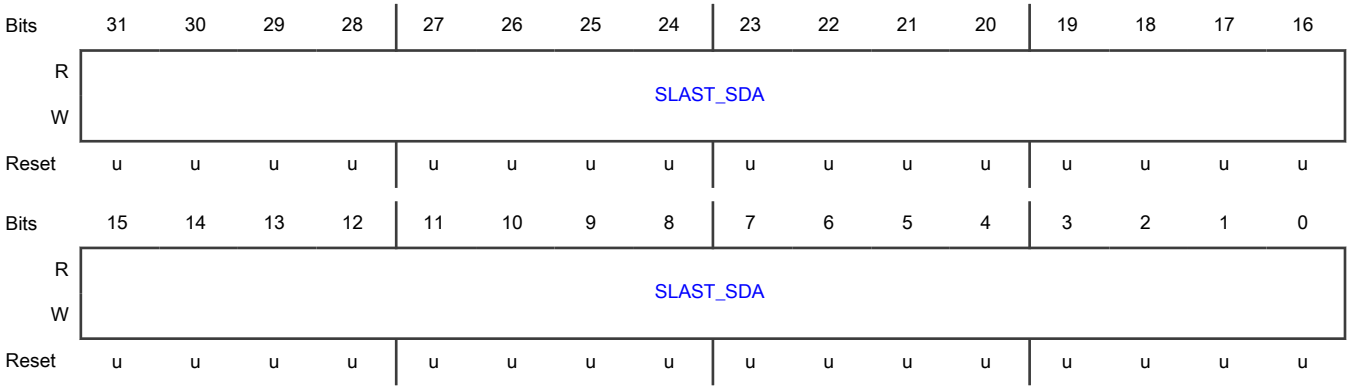
Provides a pointer to memory for storing the final destination address when the store destination address option is enabled.

Offset

For n = 0 to 63:

Register	Offset
TCDn_SLAST_SDA	2Ch + (n × 1000h)

Diagram



Fields

Field	Description
31-0 SLAST_SDA	Last Source Address Adjustment and Store DADDR Address Specifies the last source address adjustment or the system memory address for the destination address (DADDR) storage. When TCD0_CSR[ESDA] = 0, then: • Adjustment value is added to the source address after the completion of the major iteration count. You can use this value to restore the source address to the initial value or adjust the address to reference the next data structure. • This field uses the two's-complement notation for the final source address adjustment. Otherwise: • This address points to the 32-bit-aligned memory location where the destination address (DADDR) is stored in system memory. By saving the final destination address in system memory through the ESDA feature, you can compute the size of a variable destination data buffer by subtracting the beginning DADDR from the final saved DADDR. This feature is used together with the scatter/gather operation to prevent the loss of the final DADDR, which is overwritten during the scatter/gather operation. • The "Store Destination Address" (SDA) value must be a 32-bit-aligned location because eDMA forces the lower two address bits of SLAST_SDA to 0 when ESDA is enabled. eDMA performs this write operation when the major loop is complete; that is, when the major iteration count (CITER) decrements to 0.

5.3.5.15 TCD Destination Address (TCD0_DADDR - TCD63_DADDR)

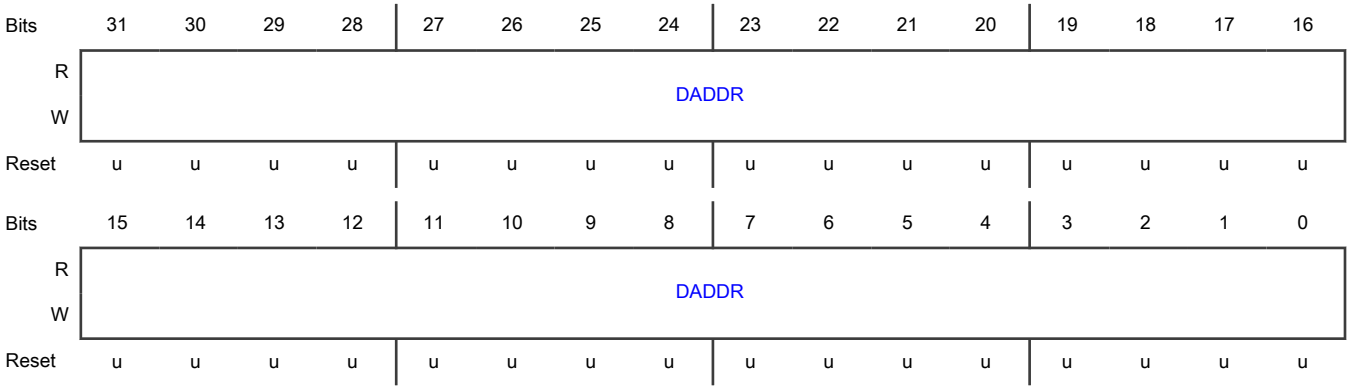
Specifies the address for write transactions.

Offset

For n = 0 to 63:

Register	Offset
TCDn_DADDR	30h + (n × 1000h)

Diagram



Fields

Field	Description
31-0 DADDR	Destination Address Specifies the memory address pointing to the destination data.

5.3.5.16 TCD Signed Destination Address Offset (TCD0_DOFF - TCD63_DOFF)

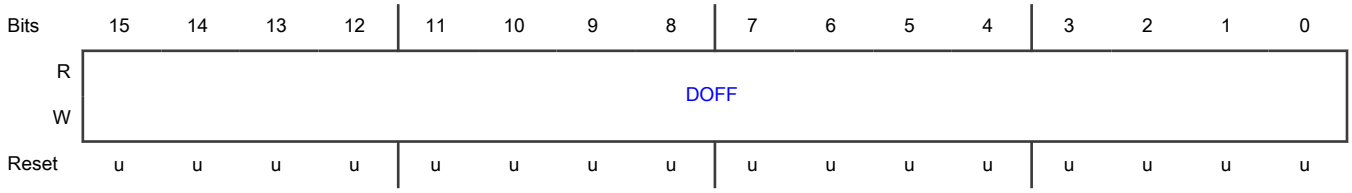
Specifies the sign-extended value added to TCD0_DADDR[DADDR] after each write transaction.

Offset

For n = 0 to 63:

Register	Offset
TCDn_DOFF	34h + (n × 1000h)

Diagram



Fields

Field	Description
15-0 DOFF	Destination Address Signed Offset Specifies the sign-extended offset that is applied to the current destination address to form the next-state value as each destination write is completed.

5.3.5.17 TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD63_CITER_ELINKNO)

Allows channel-to-channel linking and specifies the current major iteration count.

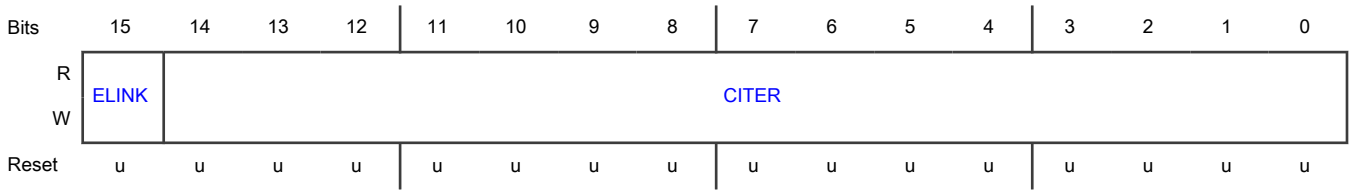
If TCD*n*_CITER[ELINK] is 0, the CITER field is defined as explained in this register.

Offset

For n = 0 to 63:

Register	Offset
TCDn_CITER_ELINKNO	36h + (n × 1000h)

Diagram



Fields

Field	Description
15 ELINK	Enable Channel-to-Channel Linking after Minor Loop Completion Enables linking to another channel as the channel completes the minor loop, defined by LINKCH. The link target channel initiates a channel service request through an internal mechanism that writes 1 to TCD0_CSR[START] of the specified channel. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed to support MAJORELINK channel linking.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field must be equal to BITER[ELINK]; otherwise, a configuration error is reported. 0 - Disable 1 - Enable
14-0 CITER	Current Major Iteration Count Represents the current major loop count (9-bit (ELINK = 1) or 15-bit (ELINK = 0)) for the channel. It is decremented each time the channel finishes a service request and is written back to the TCD memory. After the major iteration count is exhausted, the channel performs a number of operations, for example, final source and destination address calculations, optionally generating an interrupt to signal channel completion before reloading CITER from BITER. **NOTE** • When you initially load CITER, you must set this to the same value as that contained in BITER for proper operation of the optional interrupt (see TCD0_CSR[INTHALF]). • If the channel is configured to execute a single service request, the initial values of the BITER and CITER fields must be 0001h.

5.3.5.18 TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD63_CITER_ELINKYES)

Allows channel-to-channel linking and specifies the minor loop link channel number and current major iteration count.

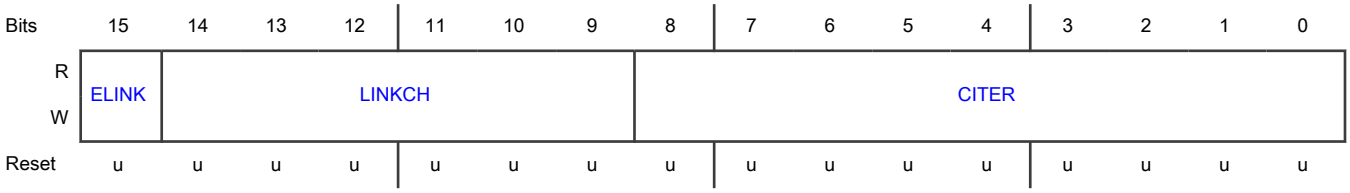
If TCD*n*_CITER[ELINK] is 1, the CITER field is defined as explained in this register.

Offset

For n = 0 to 63:

Register	Offset
TCDn_CITER_ELINKYES	36h + (n × 1000h)

Diagram



Fields

Field	Description
15 ELINK	Enable Channel-to-Channel Linking on Minor-Loop Complete Enables linking to another channel as the channel completes the minor loop, defined by LINKCH. When enabled, an internal mechanism sets TCD0_CSR[START] of the specified channel (LINKCH) after minor loop completion. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed to support MAJORELINK channel linking. **NOTE** This field must be equal to BITER[ELINK]; otherwise, a configuration error is reported. 0 - Disable 1 - Enable
14-9 LINKCH	Minor Loop Link Channel Number Specifies the minor loop link channel number. When channel-to-channel linking is enabled (ELINK = 1), after the minor loop is exhausted, the eDMA engine initiates a channel service request to the channel defined by this field by writing 1 to that channel's TCD0_CSR[START].
8-0 CITER	Current Major Iteration Count Represents the current major loop count (9-bit (ELINK = 1) or 15-bit (ELINK = 0)) for the channel. It is decremented each time the channel finishes a service request and is written back to the TCD memory. After the major iteration count is exhausted, the channel performs a number of operations (for example, final source and destination address calculations, optionally generating an interrupt to signal channel completion before reloading CITER from BITER). **NOTE** • When you initially load CITER, you must set this to the same value as that contained in BITER for proper operation of the optional interrupt (see TCD0_CSR[INTHALF]). • If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0001h.

5.3.5.19 TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD63_DLAST_SGA)

Specifies the value added to the destination address when the major loop is complete.

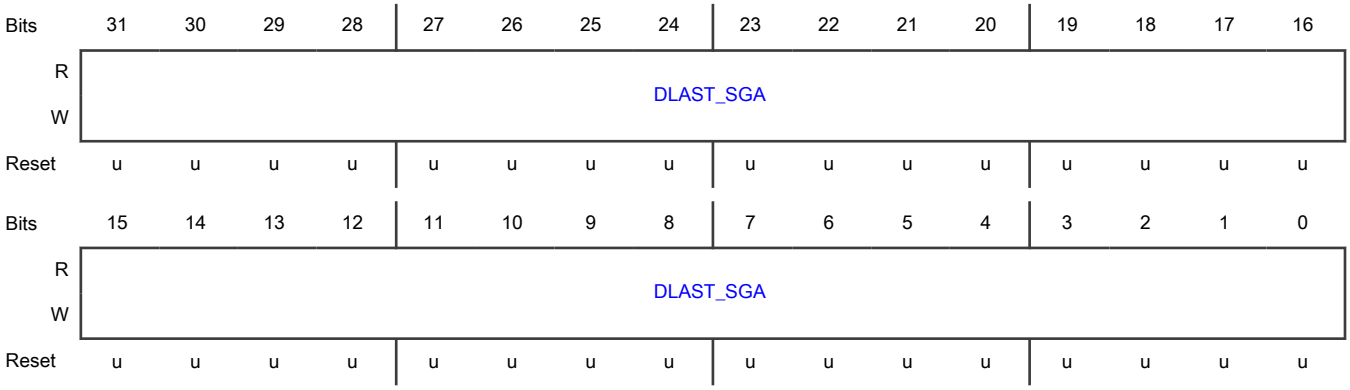
Provides a pointer to memory for fetching a TCD to reprogram the channel, if the scatter/gather option is enabled.

Offset

For n = 0 to 63:

Register	Offset
TCDn_DLAST_SGA	38h + (n × 1000h)

Diagram



Fields

Field	Description
31-0 DLAST_SGA	Final Destination Address Adjustment and Scatter/Gather Address Specifies the destination final address adjustment or the memory address for the next TCD to be loaded into this channel (scatter/gather). If TCD0_CSR[ESG] = 0, then: • Adjustment value is added to the destination address at the completion of the major iteration count. You can use this value to restore the destination address to the initial value or adjust the address to reference the next data structure. • This field uses the two's-complement notation for the final destination address adjustment. Otherwise: • This address points to the beginning of a 0-modulo-32-byte region containing the next TCD to be loaded into this channel, which is reloaded after the major iteration count completes. The scatter/gather address must be 0-modulo-32-byte, otherwise a configuration error is reported.

5.3.5.20 TCD Control and Status (TCD0_CSR - TCD63_CSR)

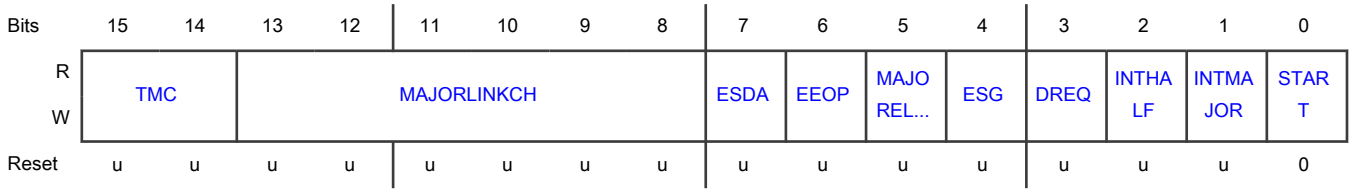
Provides status and control of TCD. It also enables optional features.

Offset

For n = 0 to 63:

Register	Offset
TCDn_CSR	3Ch + (n × 1000h)

Diagram



Fields

Field	Description
15-14 TMC	Transfer Mode Control Specifies the direction in which DMA transfers data. For normal data movement operation, Read/Write mode reads from the source address and writes to the destination address. For Read-Only mode, data is read from the source address and discarded. You can use Read-Only mode for calculating the CRC values on a block of memory. For Write-Only mode, the source address is read once and all subsequent writes to the destination use the data from the first and only read. You may use Write-Only mode to initialize a block of memory to a specific value. 00 - Read/Write 01 - Read-Only 10 - Write-Only 11 - Reserved
13-8 MAJORLINKCH	Major Loop Link Channel Number Specifies the major loop link channel number. If MAJORELINK = 0, you cannot perform any channel-to-channel linking, or chaining after the major loop counter is exhausted. If MAJORELINK is not 0, after the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by writing 1 to that channel's START.
7 ESDA	Enable Store Destination Address Enables the store destination address to system memory operation. This field enables writing the destination address (DADDR) to the address pointed to by TCD0_SLAST_SDA[SLAST_SDA] as the channel completes the major loop through either CITER decrementing to zero or, by receiving an enabled end-of-packet signal. When SDA = 1, TCD0_SLAST_SDA[SLAST_SDA] contains the write pointer instead of the final source address offset. The value written to the system memory specified by SLAST_SDA has the following possibilities: • For a normal successful transfer, the value is the last DADDR value prior to the DLAST_SGA offset that an enabled scatter/gather operation applies or overwrites. You can also compute this value using the TCD information. • For a transfer that EOP ends early, the value written to the memory is the last destination address written with the last good read data before EOP was issued.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Disable 1 - Enable
6 EEOP	Enable End-of-Packet Processing Enables the end-of-packet hardware input signal. When this field is 1, an end-of-packet hardware input signal directs eDMA to discontinue executing the active channel and to treat the shutdown as major loop completed. Additionally, the end-of-packet signal from supported peripherals is accepted. If this field is 0, the end-of-packet input is ignored. With an end-of-packet retirement, the current TCD destination address (or ESDA saved destination address) minus the software-saved initial address (DADDR) reflects the total amount of data transferred. For this calculation, the destination offset, DOFF, must match the natural boundary of the destination transfer size, DSIZE. For example, if DSIZE = 32-bit, DOFF = 4. 0 - Disable 1 - Enable
5 MAJORELINK	Enable Channel-to-Channel Linking on Major Loop Complete Enables linking to another channel (as the channel completes the major loop), defined by MAJORLINKCH. The link target channel initiates a channel service request through an internal mechanism that writes 1 to START of the specified channel. **NOTE** To support the dynamic linking coherency model, this field is forced to become 0 when you write to it while TCD*n*_CSR[DONE] is 1. Write to this field and MAJORLINKCH at the same time when attempting a dynamic channel link. 0 - Disable 1 - Enable
4 ESG	Enable Scatter/Gather Processing Specifies the current channel's TCD format. This field enables scatter/gather processing in the current channel as the channel completes the major loop. If this field is 1, the eDMA engine uses TCD0_DLAST_SGA[DLAST_SGA] as a memory pointer to a 0-modulo-32 address containing a 32-byte data structure loaded as TCD into the local memory. TCD0_DLAST_SGA[DLAST_SGA] provides a memory pointer to the next TCD to be loaded into this channel after the major loop completes its execution. **NOTE** Any changes made to this field while the channel is executing are not recognized until the next time the channel is activated. 0 - Normal format 1 - Scatter/gather format
3	Disable Request

Table continues on the next page...

Table continued from the previous page...

Field	Description
DREQ	Clears CH0_CSR[ERQ] after major loop completion, thus disabling hardware service requests. If this field is 0, no operation is carried out. The channel's ERQ {H,L} bit is not affected. If this field is 1, the channel's ERQ {H,L} bit becomes 0 when the major loop is complete. Also, the eDMA hardware automatically clears the corresponding ERQ field when the current major iteration count reaches 0. 0 - No operation 1 - Clear ERQ
2 INTHALF	Interrupt Enable after Major Counter is Half Complete Enables half-point interrupt. If this field is 1, the channel generates an interrupt request by writing 1 to the appropriate bit in INT when CITER reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER = (BITER/2)). This halfway point interrupt request is provided to support double-buffered, also known as ping-pong, schemes, or other types of data movement where the processor needs an early indication of the transfer's progress. **NOTE** If BITER = 1, do not use this field. Use INTMAJOR instead. 0 - Disable 1 - Enable
1 INTMAJOR	Interrupt Enable after Major Iteration Count Completion Enables end-of-major loop interrupt. If this field is 1, the channel generates an interrupt request by writing 1 to the appropriate bit in INT when CITER reaches zero. 0 - Disable 1 - Enable
0 START	Channel Start Specifies whether the channel is explicitly started through a software-initiated service request. If this field is 1, it indicates that the channel is requesting service. The eDMA hardware automatically writes 0 to this field after the channel begins execution. 0 - Do not start 1 - Start

5.3.5.21 TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD63_BITER_ELINKNO)

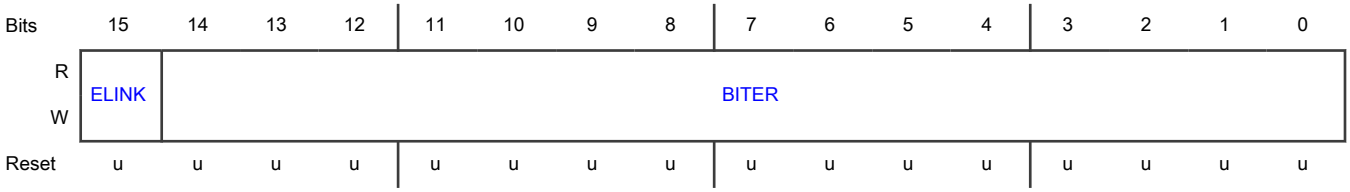
Allows channel-to-channel linking and specifies the major iteration count. If TCD*n*_BITER[ELINK] = 0, TCD*n*_BITER is defined as explained in this register.

Offset

For n = 0 to 63:

Register	Offset
TCDn_BITER_ELINKNO	3Eh + (n × 1000h)

Diagram



Fields

Field	Description
15 ELINK	Channel-to-Channel Linking after Minor Loop Completion Enables linking to another channel (after the channel completes the minor loop), defined by BITER[LINKCH]. The link target channel initiates a channel service request through an internal mechanism that writes 1 to TCD0_CSR[START] of the specified channel. If you disable the channel linking, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed to support MAJORELINK channel linking. **NOTE** When you load TCD, this field must be set equal to the corresponding CITER; otherwise, a configuration error is reported. When the major iteration count is exhausted, the contents of this field are reloaded into CITER. 0 - Disable 1 - Enable
14-0 BITER	Starting Major Iteration Count Specifies the starting major iteration count. When you first load TCD, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. When the major iteration count is exhausted, the contents of this field are reloaded into CITER. **NOTE** When you load TCD, this field must be set equal to the corresponding CITER field for the proper operation of the optional interrupt (see TCD0_CSR[INTHALF]). When the major iteration count is exhausted, the contents of this field are reloaded into CITER. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0001h.

5.3.5.22 TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD63_BITER_ELINKYES)

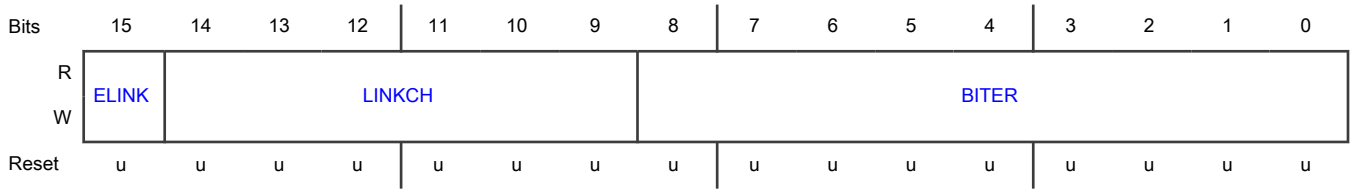
Allows channel-to-channel linking and specifies the link channel number and the starting major iteration count. If TCD*n*_BITER[ELINK] = 1, TCD*n*_BITER is defined as explained in this register.

Offset

For n = 0 to 63:

Register	Offset
TCDn_BITER_ELINKYES	3Eh + (n × 1000h)

Diagram



Fields

Field	Description
15 ELINK	Channel-to-Channel Linking after Minor Loop Completion Enables linking to another channel (as the channel completes the minor loop), defined by BITER[LINKCH]. The link target channel initiates a channel service request through an internal mechanism that writes 1 to TCD0_CSR[START] of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed to support MAJORELINK channel linking. **NOTE** When you load TCD, this field must be equal to the corresponding CITER; otherwise, a configuration error is reported. When the major iteration count is exhausted, the contents of this field are reloaded into CITER. 0 - Disable 1 - Enable
14-9 LINKCH	Link Channel Number Specifies the link channel number. If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field, by writing 1 to that channel's TCD0_CSR[START].

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE When the major iteration count is exhausted, the contents of this field are reloaded into CITER.
8-0 BITER	Starting Major Iteration Count Specifies the starting major iteration count. When you load TCD, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. When the major iteration count is exhausted, the contents of this field are reloaded into CITER. **NOTE** When you load TCD, this field must be set equal to the corresponding CITER for proper operation of the optional interrupt (see TCD0_CSR[INTHALF]). When the major iteration count is exhausted, the contents of this field are reloaded into CITER. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0001h.

5.4 Functional description

The eDMA operation is described in the following subsections.

5.4.1 Modes of operation

Table 23. Modes of operation

Mode	Description
Normal	eDMA transfers data between a source and destination. The source and destination can be a memory block or an I/O block capable of operation with eDMA. A service request initiates a transfer of a specific number of bytes (NBYTES) as specified in TCD. The minor loop is the sequence of read/write operations that transfers these NBYTES per service request. Each service request executes one iteration of the major loop, which transfers NBYTES of data.
Debug	eDMA operation is configurable in Debug mode through CSR: • When you write 0 to CSR[EDBG], eDMA continues to operate. • When you write 1 to CSR[EDBG], eDMA stops transferring data. If Debug mode is entered when a channel is active, eDMA continues operating until the channel retires.

5.4.2 eDMA basic data flow

The basic flow of a data transfer can be partitioned into three segments.

As shown in Figure 11, the first segment involves channel activation. This figure shows the assertion of the eDMA peripheral request signal to request service for channel *n*. When you activate the channel, TCD0_CSR[START] follows the same basic flow as peripheral requests. The eDMA request input signal is registered internally and then routed through the eDMA engine: first through the control module, and then into the program model and channel arbitration. In the next cycle, the channel arbitration begins using fixed-priority plus the optional round-robin algorithm. After arbitration is complete, the activated channel number is sent through the address path and converted into the required address to access the local memory for TCD*n*. Next, the TCD

memory is accessed, and the required descriptor is read from the local memory and loaded into the eDMA engine address path's primary or secondary channel execution registers. The TCD memory is 64-bit wide to minimize the time needed to fetch the activated channel descriptor and load it into the address path registers.

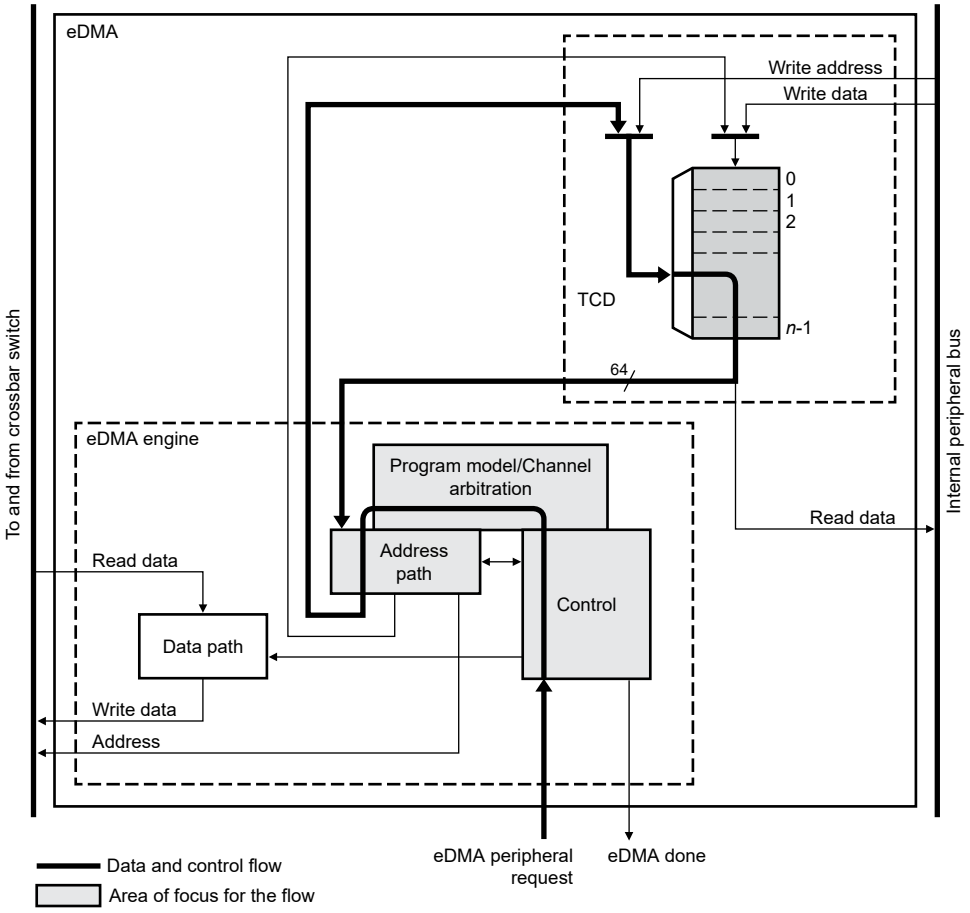


Figure 11. eDMA operation, part 1

Figure 12 illustrates the second part of the data flow to perform the data movement, from the source read to the destination write. The modules associated with the data flow include the address path, data path, and control block.

The source reads are initiated and the fetched data is temporarily stored in the data path block until it is gated onto the internal bus during the destination write. This source read or destination write processing continues until the byte count, NBYTES, is transferred.

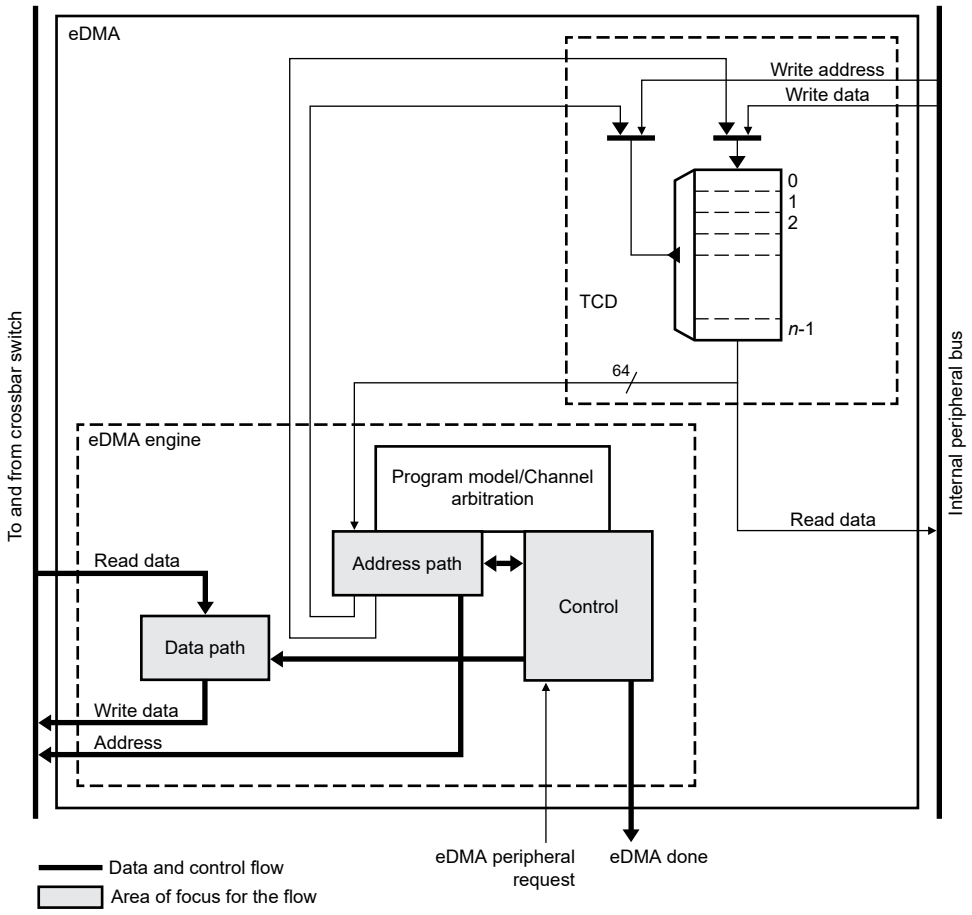


Figure 12. eDMA operation, part 2

After NBYTES of data is moved, the final phase of the basic data flow is performed. In this segment, the address path logic performs the required updates to some fields in the appropriate TCD, for example, SADDR, DADDR, and CITER. If the major iteration count is exhausted, additional operations are performed. These include the final address adjustments and reloading of BITER into CITER. Assertion of an optional interrupt request also occurs at this time, as does a possible fetch of a new TCD from memory using the scatter/gather address pointer included in the descriptor (if scatter/gather is enabled). Updates to the TCD memory and the assertion of an interrupt request are shown in Figure 13.

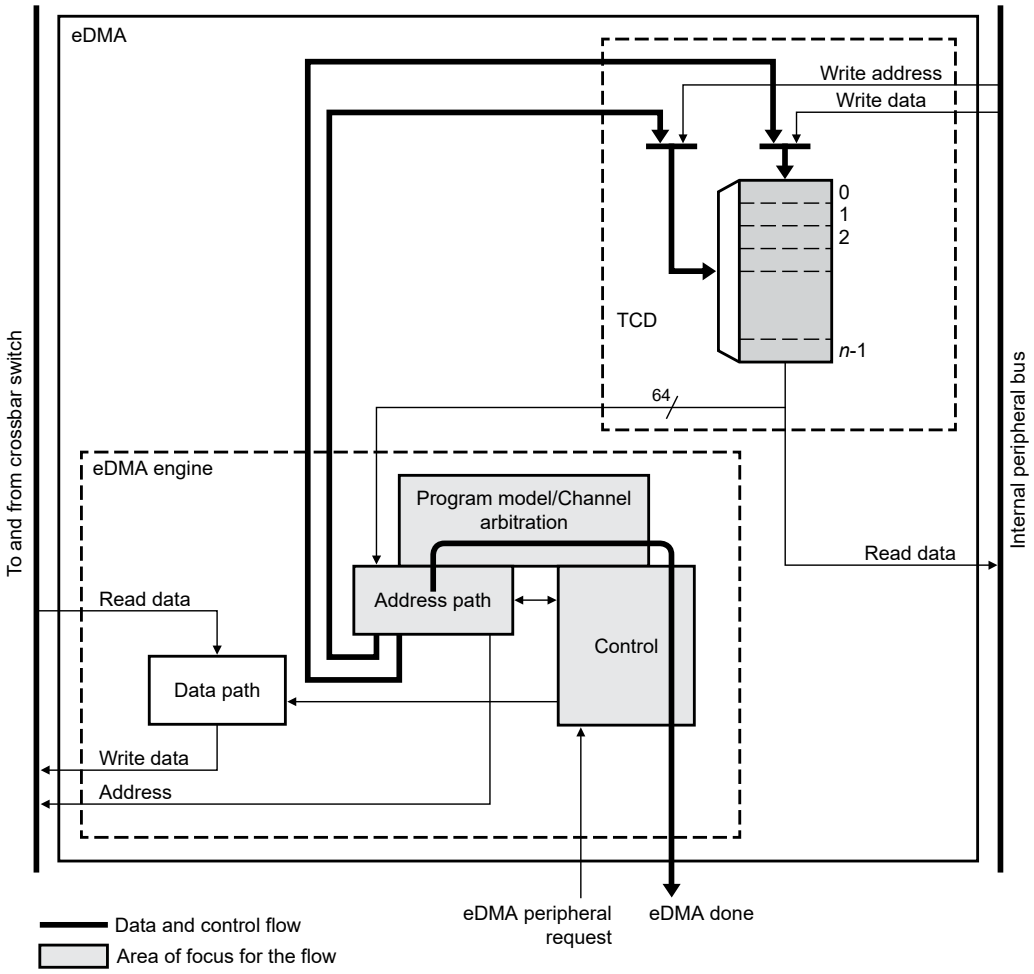


Figure 13. eDMA operation, part 3

5.4.3 Fault reporting and handling

Channel errors are reported in CH0_ES and can be caused by either of the following:

- A configuration error, which is an illegal setting in TCD
- An active channel canceled via a "cancel transfer with error" software request
- An error termination to a bus controller read or write cycle

eDMA reports a configuration error when the starting source or destination address, source or destination offsets, minor loop byte count, or the transfer size represents an inconsistent state. Following are the possible causes of a configuration error:

- The addresses and offsets are aligned on 0-modulo-transfer-size boundaries.
- The minor loop byte count must be a multiple of the source and destination transfer sizes.
- All source reads and destination writes are configured to the natural boundary of the programmed transfer size respectively.
- If you enable a scatter/gather operation after the channel completion, a configuration error is reported (if the scatter/gather address (DLAST_SGA) is not aligned on a 32-byte boundary).
- If you enable the minor loop channel linking after the channel completion, a configuration error is reported when the link is attempted (if TCD*n*_CITER[ELINK] is not equal to TCD*n*_BITER[ELINK]).

NOTE

To aid in debug, write 1 to CSR[HAE]. After an error condition occurs, eDMA is halted after the error is recorded. eDMA remains halted and does not process any channel service requests. After the error is fixed, you can enable the eDMA by writing 0 to CSR[HALT].

If enabled, all configuration error conditions (except the scatter/gather and minor-loop link errors) report as the channel activates and asserts an error interrupt request, in case it is enabled. A scatter/gather configuration error is reported when the scatter/gather operation begins on major loop completion when properly enabled. A minor loop channel link configuration error is reported when the link operation is serviced at minor loop completion.

When a system bus read or write is terminated with an error, the data transfer is stopped, and the appropriate bus error flag is set. In this case, the eDMA engine updates the state of the channel's TCD with the current source address, destination address, and the current iteration count at the point of the fault. When a system bus error occurs, the channel terminates the processing. Due to the pipeline effect, when eDMA receives a bus error, it does not abort the transfer in progress. Bus errors are imprecise. The source and destination addresses stored in TCD, after a bus error occurs, may not be the addresses that caused the error.

You may cancel a transfer by using CSR[CX]. When a cancel transfer request is recognized, the DMA engine stops processing the channel. Transactions in flight are allowed to complete. If you cancel a transfer on the last sequence of a major or minor loop, the canceled request is discarded, and the channel retires normally.

The error cancel transfer is the same as a cancel transfer except that ES is updated with the canceled channel number, and CSR[ECX] becomes 1. TCD of a canceled channel contains the source and destination addresses of the last transfer saved in TCD. If you want to restart the channel, you must reinitialize TCD because the aforementioned fields no longer represent the original parameters. When the error cancel transfer mechanism cancels a transfer, the channel number is loaded into ES[ERRCHN], and ES[ECX] and ES[VLD] become 1. In addition, it can generate an error interrupt if enabled.

NOTE

The cancel transfer request allows you to stop a large data transfer when you no longer need the full data transfer. CSR[CX] does not abort the channel. It stops the transferring of data and then retires the channel through its normal shutdown sequence. The cancel transfer is an imprecise operation. You must handle the context of the cancel. Depending on whether an interrupt is required or not, you must enable or disable the interrupt before the cancel request. You must clean up TCD because the complete transfer did not occur.

The occurrence of any error causes the eDMA engine to stop normal processing of the active channel immediately (it goes to its error processing states, and the transaction to the system bus has a pipeline effect), and the appropriate channel field in the eDMA error register is asserted. At the same time, the details of the error condition are loaded into ES. When an error is detected the following major loop indicators remain unaffected:

- Setting of CH0_CSR[DONE]
- Possible assertion of an interrupt request

After the error status is updated, the eDMA engine continues to operate by servicing the next appropriate channel. A channel that experiences an error condition is automatically disabled from hardware service requests. If an error terminates a channel, CH0_CSR[ERQ] becomes 0, preventing continuous errors from a peripheral's service request.

The fields in ES represent error indicators that are sticky and cannot be cleared. They show the last recorded error until DMA is reset. You can use ES[VLD] to determine whether a new error condition exists. ES[VLD] is the logical OR of each channel's CH0_ES[ERR]. After you resolve all the errors and clear all the error interrupt bits, ES[VLD] becomes 0, but the cause of the last error is still indicated.

5.4.4 Channel preemption

eDMA uses a priority vector value to determine the highest-priority channel requesting service.

The priority vector is a combination of:

- The channel's group priority, CH0_GRPRI[GRPRI]
- The channel's priority level, CH0_PRI[APL]

- The channel number

Priority vector = ((CH*n*_GRPRI << 8) + (CH*n*_PRI[APL] << 5) + CHn_*)

A channel requesting service with the highest-priority vector value receives the next execution slot, which is available:

- Immediately if eDMA is idle.
- When an active channel retires.
- When valid preemption conditions exist.

NOTE

Preemption is strictly priority based and is not bound by a specific group number as defined by CH*n*_GRPRI.

Write 1 to CH0_PRI[ECP] to enable channel preemption on a per-channel basis. Channel preemption allows the executing channel's data transfers to temporarily suspend in favor of starting a higher priority channel. After the preempting channel completes all its minor loop data transfers, the preempted channel restores and resumes execution. After the restored channel completes one read or write sequence, it is again eligible for preemption. If any higher priority channel requests service, the restored channel is suspended and the higher priority channel is serviced. Nested preemption, attempting to preempt a preempting channel, is not supported. After a preempting channel begins execution, it cannot preempt again.

You can disable the channel's ability to preempt another channel by writing 1 to CH0_PRI[DPA]. When you disable a channel's preempt ability, that channel cannot suspend a lower-priority channel's data transfer, regardless of the lower priority channel's CH*n*_PRI[ECP] being 1. This allows a pool of low priority, large data-moving channels to define. You can configure the low-priority channels not to preempt each other, therefore preventing a low-priority channel from consuming the preempt slot normally available to a true, high-priority channel. When you enable Round-Robin Channel Arbitration mode (CSR[ERCA] = 1), any channel with a priority level equal to zero (CH*n*_PRI[APL] = 0), has the preemption disabled and does not preempt another channel.

5.4.5 Clocking

This module has no clocking considerations.

5.4.6 Interrupts

This module has no interrupts.

5.5 External signals

This module has no external signals.

5.6 Initialization

The following sections discuss eDMA initialization and programming considerations.

5.6.1 Initialization procedure

Table 24. Initialization steps

Step	Purpose	Programming	Notes
1	Specify the configuration other than default.	Write to CSR.	—
2	Specify the priority levels to channel or group for a configuration other than default.	Write the channel priority levels to CH0_PRI and group priority levels to CH0_GRPRI.	—
3	Enable error interrupts if needed.	Write 1 to CH0_CSR[EEI].	—

Table continues on the next page...

Table 24. Initialization steps (continued)

Step	Purpose	Programming	Notes
4	Request service for each channel.	Write the 32-byte TCD for each channel.	—
5	Enable hardware service requests.	Write 1 to CH0_CSR[ERQ].	—
6	Request a channel service through software.	Write 1 to TCD0_CSR[START].	—
7	Request a channel service through hardware.	Target device asserts its eDMA peripheral request signal.	—

5.6.2 Channel selection

After any of the channel requests service, select a channel for execution based on the arbitration and priority levels written into the programmer's model. The eDMA engine reads the entire TCD, including the TCD control and status fields, as shown in Table 25, for the selected channel into its internal address path module.

As TCD is read, the first transfer is initiated on the internal bus, unless a configuration error is detected. Transfers from the source, as defined by TCD0_SADDR, to the destination, as defined by TCD0_DADDR, continue until the number of bytes specified by TCD*n*_NBYTES are transferred.

When the transfer is complete, write the eDMA engine's local TCD*n*_SADDR, TCD*n*_DADDR, and TCD*n*_CITER back to the main TCD memory and perform any minor loop channel linking, if enabled. If the major loop is exhausted, further postprocessing is performed, such as interrupts, major loop channel linking, and scatter/gather operations, if enabled.

Table 25. TCD*n*_CSR field descriptions

Field	Description
START	Control field to start channel explicitly when using a software-initiated DMA service (hardware clears it automatically)
EEOP	Control field to enable end-of-packet processing
ESDA	Control field to enable the storing of the destination address to system memory after the major loop completes
DREQ	Control field to disable hardware-initiated DMA service requests after major loop completion
ESG	Control field to enable the scatter-gather feature
INTHALF	Control field to enable the interrupt when the major loop is half complete
INTMAJOR	Control field to enable the interrupt when the major loop completes

Table 26. CH*n*_CSR field description

Field	Description
ACTIVE	Status field indicating the channel is in execution
DONE	Status field indicating major loop completion (you clear it when a channel begins execution)

Table continues on the next page...

Table 26. CH*n*_CSR field description (continued)

Field	Description
EEI	Control field to enable error interrupts
EARQ	Control field to enable an external, asynchronous wake-up event with the ERQ field
ERQ	Control field to enable hardware service requests

5.6.3 Multi-loop iterations

Figure 14 shows how each DMA request initiates one minor-loop transfer, or iteration, without CPU intervention. DMA arbitration can occur after each minor loop, allowing one level of minor loop DMA preemption. The beginning iteration count (BITER) specifies the number of minor loops in a major loop.

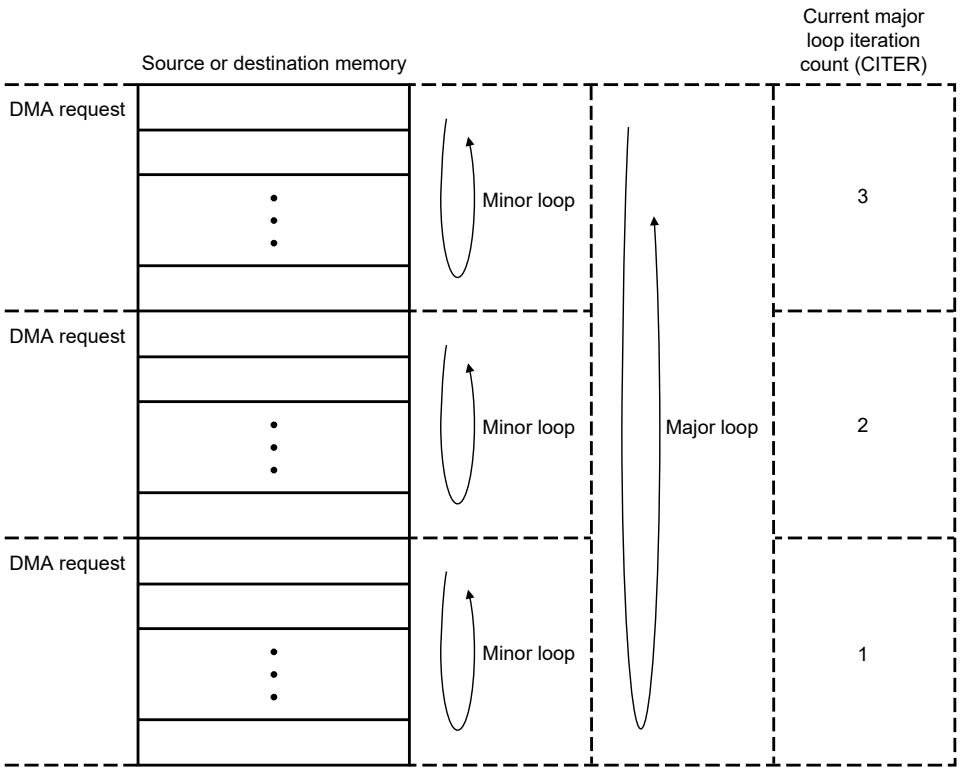


Figure 14. Example of multiple loop iterations

5.6.4 Memory array terms

Figure 15 shows the memory array terms and how the TCD settings interrelate.

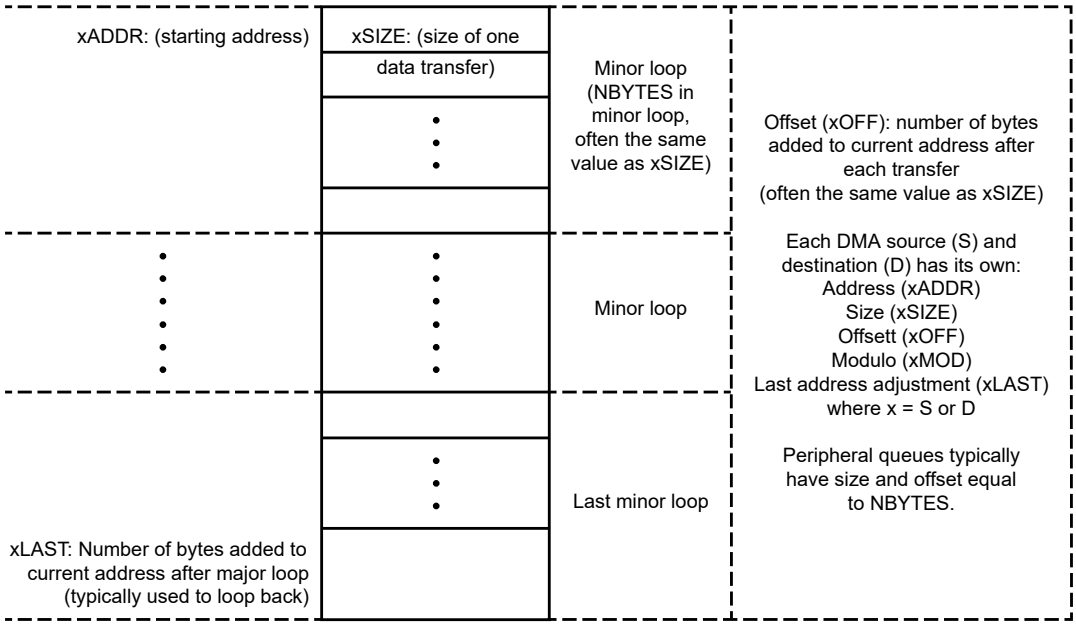


Figure 15. Memory array terms

5.7 Application information

The following sections discuss eDMA application information.

5.7.1 eDMA arbitration

eDMA uses a layered arbitration scheme consisting of multiple priority levels. It uses a fixed-priority arbitration scheme with an optional, round-robin arbitration under specific conditions. The priorities are evaluated in the following order:

1. Highest arbitration group priority: Each channel is assigned an arbitration group through CH0_GRPRI. Priority is given to the highest value (31 is the highest possible value) down to the lowest value (0 being the default).
2. Channel priority: Each channel is assigned a channel priority level through CH0_PRI. The channel priority is a relative priority level within an arbitration group. Priority is given to the highest value (7 being the highest possible value) down to the lowest value (0 being the default). Channel priorities within each arbitration group do not have to be unique. If multiple channels have the same channel priority level, the channel number is used to determine priority as defined in channel number.
3. Channel number: When two or more channels have the same arbitration group and channel priority, the channel number (CH_NUM) is used to determine the highest priority. Priority is given to the highest channel number. The lowest priority is channel 0. The channel numbers are static and you cannot change them in the programmer's model.
4. Lowest: When you enable round-robin, any channel configured for the round-robin operation has the lowest priority within an arbitration group. When you enable round-robin by writing 1 to CSR[ERCA], channels with a channel priority of zero (CH_PRI = 0) use round-robin arbitration, which rotates the channel selection among the channel's requesting service with CH_PRI = 0 within the arbitration group. Any nonzero channel within the arbitration group continues to use fixed-priority arbitration, and if requesting service, selects over any round-robin channels.

For fixed arbitration, you can consider the overall priority as a number composed of three concatenated priority levels (CH_GRPRI:CH_PRI:CH_NUM). The largest number has the highest priority and the lowest number has the lowest priority.

When you enable round-robin arbitration, channels are serviced in an order, per priority group (CH_GRPRI:0:X). Within a priority group, all channels with their channel priority set to zero are cycled through from the highest to lowest channel number.

The round-robin scheme can be overridden by a nonzero channel priority value (any channel with a nonzero CH*n*_PRI value automatically has a higher priority over the round-robin scheme).

5.7.2 Programming errors

eDMA performs various tests on TCD to verify consistency in the descriptor data.

The channel number causing the error is recorded in ES. If you do not remove the error source before the next activation of the problem channel, the error is detected and recorded again. Writing 1 to CSR[HAE] halts the DMA and prevents the recurrence of the error.

5.7.3 Arbitration mode considerations

This section discusses arbitration considerations for eDMA.

5.7.3.1 Fixed-group arbitration, fixed-channel arbitration

In this mode, the channel service request from the highest priority channel in the highest priority group is selected to execute. If you program eDMA in a way that the channels with a high-priority group have a high number of requests or large data transfers, that group may consume all the bandwidth of the eDMA controller. That is, no lower priority group is serviced if there is at least one DMA request pending on a channel in the highest priority group when the controller arbitrates the next DMA request. The advantage of this scenario is that latency can be small for channels that require immediate service.

5.7.3.2 Fixed-group arbitration, round-robin channel arbitration

The highest priority group with a request is serviced. Lower priority groups are serviced if no pending requests exist in the higher priority groups.

Within each group, channels are serviced starting with the highest nonzero channel priority. For all channels with a channel priority programmed to 0, the highest channel number requesting service is selected, rotating to the lowest channel number requesting service. The round-robin channel arbitration can provide a fairness mechanism to lower priority channels.

This scenario can lead to the same bandwidth consumption problem as indicated in Fixed-group arbitration, fixed-channel arbitration, but all channels in the highest priority group are serviced. Service latency is short on the highest priority group, but can become potentially longer as the group priority decreases.

5.7.4 Performing DMA transfers

This section presents examples on how to perform DMA transfers with eDMA.

5.7.4.1 Single request

To perform a simple transfer of *n* bytes of data with one activation, write 1 to the major loop (TCD*n*_CITER = TCD*n*_BITER = 1). The data transfer begins after you acknowledge the channel service request and select the channel to execute. After the transfer is complete, CH0_CSR[DONE] becomes 1 and an interrupt generates, if properly enabled.

The following TCD entry (see Code Listing 1) is configured to transfer 16 bytes of data. eDMA is programmed for one iteration of the major loop transferring 16 bytes per iteration. The source memory has a byte-wide memory port at 1000h. The destination memory has a 32-bit port at 2000h. Program the address offsets in increments to match the transfer size: one byte for the source and four bytes for the destination. The final source and destination addresses are adjusted to return to their initial values.

This generates the following event sequence:

1. You must write 1 to TCD0_CSR[START] to request channel service.
2. Arbitration selects the channel for servicing.
3. The eDMA engine writes 0 to CH0_CSR[DONE] and TCD*n*_CSR[START], and 1 to CH0_CSR[ACTIVE].
4. The eDMA engine reads channel TCD data from the local memory to internal register file.
5. The source-to-destination transfers are executed as follows:

a. Data is continuously read from the source addresses by issuing read transactions and adjusting the source address through the source address-offset after each transaction.

b. Data is continuously written to the destination addresses after sufficient data is buffered. Adjust the destination address through the destination address offset after each transaction.

6. The eDMA engine writes 1000h to TCD*n*_SADDR, 2000h to TCD*n*_DADDR, and 1 to TCD*n*_CITER(TCD*n*_BITER).
7. The eDMA engine writes 0 to CH*n*_CSR[ACTIVE], 1 to CH*n*_CSR[DONE], and 1 to CH0_INT[INT].
8. The channel retires and eDMA goes idle or services the next channel.

```
TCDn_CITER = TCDn_BITER = 1
TCDn_NBYTES = 16
TCDn_SADDR = 1000h
TCDn_SOFF = 1
TCDn_ATTR[SSIZE] = 0
TCDn_SLAST = -16
TCDn_DADDR = 2000h
TCDn_DOFF = 4
TCDn_ATTR[DSIZE] = 2
TCDn_DLAST_SGA= -16
TCDn_CSR[INTMAJ] = 1
TCDn_CSR[START] = 1 (You must write to this field after all other fields are initialized.)
All other TCDn fields = 0
```

Code Listing 1. Single request

5.7.4.2 Multiple requests

Code Listing 2 shows the transfer of 32 bytes through two hardware requests but is otherwise the same as the previous example (see Single request). The only fields that change are the major loop iteration count and final address offsets. Program eDMA for two iterations of the major loop transferring 16 bytes per iteration. After you enable the channel's hardware requests through CH0_CSR[ERQ], the target device initiates channel service requests.

This generates the following sequence of events:

1. First hardware, that is, eDMA peripheral, requests for channel service.
2. Arbitration selects the channel for servicing.
3. The eDMA engine writes 0 to CH0_CSR[DONE] and TCD0_CSR[START], and 1 to CH0_CSR[ACTIVE].
4. The eDMA engine reads channel TCD*n* data from the local memory to the internal register file.
5. The source to destination transfers are executed as follows:

   a. Data is continuously read from the source addresses by issuing read transactions and adjusting the source address through the source address offset value after each transaction.

   b. Data is continuously written to the destination addresses after sufficient data is buffered. Adjust the destination address through the destination address offset after each transaction.
6. The eDMA engine writes 1010h to TCD*n*_SADDR, 2010h to TCD*n*_DADDR, and 1 to TCD*n*_CITER.
7. The eDMA engine writes 0 to CH*n*_CSR[ACTIVE].
8. The channel retires → one iteration of the major loop. eDMA goes idle or services the next channel.
9. The second hardware, that is, eDMA peripheral, requests channel service.
10. Arbitration selects the channel for servicing.

11. The eDMA engine writes 0 to CH*n*_CSR[DONE] and TCD*n*_CSR[START], and 1 to CH*n*_CSR[ACTIVE].
12. The eDMA engine reads the channel TCD data from the local memory to the internal register file.
13. The source to destination transfers are executed as follows:
    a. Data is continuously read from the source addresses by issuing read transactions and adjusting the source address through the source address offset value after each transaction.
    b. Data is continuously written to the destination addresses after sufficient data is buffered. Adjust the destination address through the destination address offset after each transaction.
14. The eDMA engine writes 1000h to TCD*n*_SADDR, 2000h to TCD*n*_DADDR, and 2 to TCD*n*_CITER(TCD*n*_BITER).
15. The eDMA engine writes 0 to CH*n*_CSR[ACTIVE], and 1 to CH*n*_CSR[DONE] and CH0_INT[INT].
16. The channel retires → major loop complete. eDMA goes idle or services the next channel.

```
TCDn_CITER = TCDn_BITER = 2
TCDn_SLAST = -32
TCDn_DLAST_SGA = -32
```

Code Listing 2. Multiple request

5.7.4.3 Using the modulo feature

The modulo feature of eDMA provides the ability to implement a circular data queue in which the queue's size is a power of 2. MOD is a 5-bit field for the source and destination in TCD, and it specifies which lower address bits increment from their original value after the address plus offset calculation. All upper address bits remain the same as their original value. When you write 0 to this field, it disables the modulo feature. Modulo addressing applies to cases where the minor loop offset is enabled; that is, the upper address bits remain the same after the minor loop offset is added to the source or destination address.

Table 27 shows how to specify the transfer addresses based on the MOD field setting. A circular buffer is created where the address wraps to the original value while the 28 upper address bits (1234_567*x*h) retain their original value. In this example, the source address is set to 1234_5670h, the offset is set to 4 bytes, and MOD is set to 4, allowing for a 2^4 byte (16-byte) size queue.

Table 27. Modulo example

Transfer number	Address
1	1234_5670h
2	1234_5674h
3	1234_5678h
4	1234_567Ch
5	1234_5670h
6	1234_5674h

5.7.5 Monitoring TCD status

This section discusses how to monitor eDMA status.

5.7.5.1 Testing for minor loop completion

There are two methods to test minor loop completion when using software-initiated service requests. First is to read TCD*n*_CITER and test for a change. You may extract another method from Table 28. The second method is to test TCD0_CSR[START] and CH0_CSR[ACTIVE]. The minor-loop-complete condition is indicated by both fields reading zero after you write 1 to TCD*n*_CSR[START]. Only polling CH*n*_CSR[ACTIVE] can be inconclusive because you can miss the active status while the channel execution is of shorter duration.

The CH*n*_CSR and TCD*n*_CSR status fields execute the following sequence (as shown in Table 28) for a software-activated channel.

Table 28. Settings for software-activated channel

Stage	TCD*n*_CSR and CH*n*_CSR			State
	START	ACTIVE	DONE	
1	1	0	0	Channel service request through software.
2	0	1	0	Channel is executing.
3a	0	0	0	The channel is idle after completing its minor loop.
3b	0	0	1	The channel is idle after completing its major loop.

The best method to test for minor-loop completion when using eDMA (that is, peripheral, initiated service requests), is to read TCD*n*_CITER and test for a change. The hardware request and acknowledge handshake signals are not visible in the programmer's model.

The TCD status fields execute the following sequence (as shown in Table 29) for a hardware-activated channel.

Table 29. Settings for hardware-activated channel

Stage	TCD*n*_CSR and CH*n*_CSR			State
	START	ACTIVE	DONE	
1	0	0	0	Channel service request through hardware (peripheral request asserted).
2	0	1	0	Channel is executing.
3a	0	0	0	The channel is idle after completing its minor loop.
3b	0	0	1	The channel is idle after completing its major loop.

For both the activation types, the major-loop-completion status is explicitly indicated through CH0_CSR[DONE].

TCD0_CSR[START] becomes 0 automatically when the channel begins execution regardless of how the channel activates.

5.7.5.2 Checking channel preemption status

A preemptive situation is one in which a preempt-enabled channel is executing and a higher priority request becomes active. When you enable the round-robin channel arbitration mode, all the channels with their channel priority set to zero lose their preempt ability. Channel priorities of zero are treated as equal, that is, constantly rotating, when round-robin arbitration mode is enabled.

CH0_CSR[ACTIVE] remains 1 for the preempted channel throughout the preemption. The preempted channel is temporarily suspended when the preempting channel executes one major loop iteration. If you write 1 to two CH*n*_CSR[ACTIVE] fields simultaneously in the global TCD map, a higher priority channel actively preempts a lower priority channel.

5.7.6 Channel linking

Channel linking (or chaining) is a mechanism where one channel writes 1 to TCD0_CSR[START] of another channel (or itself), initiating a service request for that channel. When properly enabled, the eDMA engine automatically performs this operation at the major or minor loop completion.

The minor loop channel linking occurs after the minor loop (or one iteration of the major loop). TCD*n*_CITER[ELINK] determines whether a minor loop link is requested. When enabled, the channel link is made after each iteration of the major loop except for the last. After the major loop is exhausted, only the major loop channel link fields are used to determine whether a channel link should be created. For example, the initial fields listed in Code Listing 3 execute as:

1. Minor loop done → write 1 to TCD12_CSR[START]
2. Minor loop done → write 1 to TCD12_CSR[START]
3. Minor loop done → write 1 to TCD12_CSR[START]
4. Minor loop done, major loop done→ write 1 to TCD7_CSR[START]

```
TCDn_CITER[ELINK] = 1
TCDn_CITER[LINKCH] = Ch
TCDn_CITER[CITER] value = 4h
TCDn_CSR[MAJORELINK] = 1
TCDn_CSR[MAJORLINKCH] = 7h
```

Code Listing 3. Channel linking

After you enable minor loop linking (TCD*n*_CITER[ELINK] = 1), TCD*n*_CITER[CITER] uses a 9-bit vector to form the current iteration count. When you disable minor loop linking (TCD*n*_CITER[ELINK] = 0), TCD*n*_CITER[CITER] uses a 15-bit vector to form the current iteration count. Fields associated with TCD*n*_CITER[LINKCH] are concatenated onto the CITER value to increase the range of CITER.

NOTE

TCD*n*_CITER[ELINK] and TCD*n*_BITER[ELINK] must be equal; else, a configuration error is reported. CITER and BITER vector widths must be equal to calculate the major loop, halfway done interrupt point.

Table 30 summarizes how a DMA channel can link to another DMA channel, that is, use another channel's TCD at the end of a loop.

Table 30. Channel linking parameters

Desired link behavior	TCD control field name	Description
Link at the end of minor loop	TCD*n*_CITER[ELINK]	Enables channel-to-channel linking on minor loop completion (current iteration)
	TCD*n*_CITER[LINKCH]	Links channel number when linking at the end of minor loop (current iteration)
Link at the end of major loop	TCD0_CSR[MAJORELINK]	Enables channel-to-channel linking on major loop completion
	TCD0_CSR[MAJORLINKCH]	Links channel number when linking at the end of major loop

5.7.7 Dynamic channel linking

Dynamic channel linking is the process of writing 1 to TCD0_CSR[MAJORELINK] and TCD0_CSR[MAJORLINKCH] during channel execution. These fields are read from the TCD profile at the end of channel execution. This allows you to enable the feature during channel execution. When attempting a dynamic channel link, you must write to TCD*n*_CSR[MAJORELINK] and TCD*n*_CSR[MAJORLINKCH] in a single write transaction (that being a write size of at least 16 bits to TCD0_CSR).

Because you can change the configuration during execution, it requires a coherency model. Consider a scenario where you attempt to execute a dynamic channel link by writing 1 to TCD*n*_CSR[MAJORELINK] at the same time when the eDMA engine is retiring the channel. TCD*n*_CSR[MAJORELINK] is 1 in the programmer's model, but it is unclear whether the actual link was made before the channel retired.

When you execute a dynamic channel link request, NXP recommends using the following coherency model:

1. Write 1 to TCD*n*_CSR[MAJORELINK].
2. Read TCD*n*_CSR[MAJORELINK].
3. Test the TCD*n*_CSR[MAJORELINK] request status:
   - If TCD*n*_CSR[MAJORELINK] = 1, the dynamic link attempt is successful.
   - If TCD*n*_CSR[MAJORELINK] = 0, the attempted dynamic link does not succeed (the channel is retiring already).

The TCD local memory controller forces TCD*n*_CSR[MAJORELINK] to become 0 on any writes to a channel's TCD*n*_CSR after that channel's CH0_CSR[DONE] becomes 1, indicating that the major loop is complete.

NOTE

Write 0 to CH*n*_CSR[DONE] before writing to TCD*n*_CSR[MAJORELINK]. The eDMA engine automatically writes 0 to CH*n*_CSR[DONE] after a channel begins execution.

5.7.8 Suspend or resume a DMA channel with active hardware service requests

DMA allows you to move data from memory or peripheral registers to another location in memory or peripheral registers without CPU interaction. After you configure DMA and peripherals and they are active, it is rare to suspend a peripheral's service request dynamically. In this scenario, there are restrictions to disable a DMA hardware service request. Follow a specific procedure for coherency. This section provides guidance on how to coherently suspend and resume a DMA channel when a target module triggers the DMA.

5.7.8.1 Suspend an active DMA channel

Perform the following procedure to suspend an active DMA channel:

1. Stop the DMA service request at the peripheral.
2. Read back the appropriate register in the peripheral to confirm that the service request is disabled.
3. Check HRS to ensure that there is no suspended service request to the DMA channel.
4. Write 0 to CH0_CSR[ERQ] on the appropriate DMA channel to disable the hardware service request.

For example, assume the SPI is set as the controller for transmitting data through a DMA service request when DSPI_TXFIFO has an empty slot. DMA transfers the next command and data to the TXFIFO upon the request. If you require to suspend the DMA/SPI transfer loop, perform the following steps:

1. Disable the DMA service request at the source by writing 0 to DSPI_RSER[TFFF_RE]. Confirm that DSPI_RSER[TFFF_RE] is 0.
2. Ensure that there is no DMA service request from the SPI by verifying that HRS[HRS] is 0 for the appropriate channel. If no service request is present, disable the DMA channel by writing 0 to CH0_CSR[ERQ]. If a service request is present, wait until the request is processed and HRS becomes 0.

5.7.8.2 Resume a DMA channel

To resume a DMA channel:

1. Enable the DMA service request on the appropriate channel by writing 1 to CH0_CSR[ERQ].
2. Enable the DMA service request at the peripheral.

5.8 Glossary

TCD	Transfer control descriptor

Chapter 6
Security Overview

6.1 Security features

By controlling access to the device as well as providing trusted computing environment, this chip offers various security features and tools that allow customers to protect their assets.

Some of the basic features include:

- Secure boot ensuring that only authentic entity is allowed to access the device or execute code on it.
- Secure debug allowing customers to securely develop and debug software while the device is at manufacturing or deployed in the field.
- Secure lifecycle management guaranteeing that security mechanisms are enforced, and device reaches its highest security level after it has been deployed or returned to NXP for failure analysis.
- Hardware Security Module (HSM) and cryptographic services ensuring that cryptographic keys remain hidden from the application in all circumstances.

This chapter explains the main purpose and features of security-relevant SoC components:

- EdgeLock Secure Enclave (Advanced Profile) (ELE-AP) acts as a Root of Trust for platform security. It manages the SoC lifecycle, enforces secure boot, controls debug access, and provides full key management and key isolation from the rest of SoC.
- Trusted Resource Domain Controller (TRDC) is essential in enabling isolation between various processes running within the SoC. It is used to define and enforce separation between security critical and other processes by introducing the notion of domain IDs, as well as enforcing the trusted execution environment defined by ARM® TrustZone®.
- On-the-Fly AES Decryption (OTFAD) protects the confidentiality of code stored in external memory using AES in CTR mode. The key used for decryption is securely handled by ELE-AP and delivered to OTFAD using a dedicated private bus, ensuring that the key is not observable by any application running on the SoC. OTFAD supports up to four independent memory contexts, each having a unique 128-bit decryption key.
- Battery Backed Secure Memory (BBSM) is used as a non-volatile storage of security parameters including general purpose registers, run-time counter with a wake-up alarm, monotonic counters, as well as tamper detection inputs.

6.1.1 Immutable Root of Trust (RoT)

As defined by Trusted Computing Group, “an Immutable Root of Trust (RoT) is expected to remain identical across all devices within a set of device models based on a defined threat model. It is also expected not to change across time and, therefore, will behave the same during each device’s lifespan.”

Immutable RoT consists of truly immutable hardware logic, including analog and digital logic, read-only memory and one-time programmable memory. It is essential for guaranteeing any security feature, including secure boot, secure debug, life-cycle management, and a number of others. In this device, immutable RoT is embedded in the ELE-AP boot ROM (immutable boot loader) and it relies on the ELE-AP hardware cryptographic accelerators for its function.

6.1.2 Run time root of trust (RRoT)

The processor includes a Run Time Root of Trust (RRoT) Hardware State machine to ensure the Root of Trust is established before Hardware allows access to keys during runtime. The RRoT feature is enabled by the immutable ELE-AP ROM code and is used for features like attestation where keys are stored in memory protected by RRoT based secure access control.

6.1.3 Lifecycle management

During its lifespan, a typical device finds itself in various places around the world. It is manufactured in a semiconductor factory, tested and packaged in silicon manufacturer facilities, sold to various distributors, sold further to the Original Equipment

Manufacturer (OEM), assembled, tested and provisioned by their Contract Manufacturers and, finally, delivered to the end-customer. In the case of failure, a device is returned to OEM or even back to the silicon manufacturer for further failure analysis.

Device life cycle state is used to reflect the actual state of the device, which is further used to instruct the device on how exactly to protect the assets a device hosts during specific time. For example, when a device is being tested at a silicon manufacturer facility and no OEM or end-customer assets have been provisioned, then access to the device, in terms of debug or test, is less restrictive than when a device is with the end-customer.

Life cycle state is monotonic, meaning it can only always be increased. Immutable RoT is in charge of life cycle management and it enforces device access policies accordingly.

6.1.4 Secure boot

Secure boot ensures authenticity, integrity and confidentiality of the device boot loader, firmware, and other software during the boot process and ensures that the intended secure life-cycle state is reached. SHA with ECDSA or RSA with different key lengths are available to guarantee authenticity and integrity of the firmware image. AES-based memory encryption can be used to provide code/data confidentiality while the firmware image is stored in external memory. ELE-AP as an immutable RoT is in charge of enforcing secure boot and it does so according to the policies defined by the life-cycle state.

See i.MX 93 Applications Processor Security Reference Manual for more details about secure boot.

6.1.5 Secure debug

Secure debug is the process used to securely access external jtag debug, following the policy defined by the lifecycle state. When a debugger wants to debug a device, they indicate that through the jtag interface.

NXP provides the asymmetric signed message based mechanism.

6.1.6 IP protection

IP protection is a set of mechanisms used to protect confidentiality and integrity of valuable code and data stored on the device. During secure update, the firmware image is encrypted using AES. While at rest and during secure boot, the firmware image is encrypted and stored in external memory. Up to four independent memory contexts available through OTFAD using a unique 128-bit key per context.

6.1.7 Secure isolation

ELE-AP guarantees run-time key isolation from any process running on the rest of SoC. In addition, Arm® TrustZone® capable Cortex®-M33 and Cortex®-A55, in combination with the Trusted Resource Domain Controller (TRDC), enables secure isolation during run-time.

6.1.8 TrustZone®

Trustzone® (TZ) is a feature on the Cortex®-A55 and Cortex®-M33, which provides segmentation of memory arrays and peripherals into either Secure (S) or Non-Secure (NS). This feature allows the M33 core to execute in either S or NS mode, and provides a secure separation between various application software.

The TrustZone® security architecture is supported. Trustzone®-M is always enabled, as this is an integral feature of the Cortex®-M33 boot ROM and system security. For on-chip RAM, all of OCRAM have TrustZone® access control support. One region with configurable address range of the OCRAM can be set to TZ access only. The Trustzone® memory region protection of DRAM is supported through TRDC.

The majority of system initiators in the SoC do not natively support TrustZone® (and more specifically, do not support a concept of AxPROT[0] indicating controllable privileged vs normal access or AxPROT[1] indicating non-secure vs TrustZone® secure access). Therefore, the TRDC DAC provides controls to override IP initiator’s AxPROT[1:0] inputs to the subsystem NICs. However, initiators which do have concept of these AxPROT[1:0] signals, do not have their signals overridden by the TRDC DAC.

6.1.9 Trusted resource domain control (TRDC)

The chip uses a domain based resource control architecture for the memory/peripheral resource sharing and isolation between the Cortex®-A55 platform, Cortex®-M33 core, and other bus masters. A key feature of this architecture is that it must be possible for either Cortex-A55 platform or the Cortex®-M33 platform to program the resource domain control access lists to implement a customer-defined policy. Another key feature is that any peripheral can be assigned to any domain. The TRDC supports multiple regions with flexible access permission control for each memory space.

A core complex should not rely on TRDC and should use the internal MMU for memory management. In addition, initiators like Cortex®-A55, that implement the DSU architecture, do support AxPROT internally, but do not drive AxPROT[2] and AxPROT[0] on the system bus. For Cortex®-A55, after boot, it is recommended to use the internal MMU for access control.

For more details please refer Chip-specific TRDC information.

6.1.10 Secure key management

Secure key management is a process of securing valuable keys and various key material which are essential in maintaining security of the end-user, OEM and NXP assets. The process strongly relies on the Immutable RoT consisting of ELE-AP, including its logic, ROM as well as OTP fuse. A device-unique master key is only used for further key derivation. Part of the derivation data is supplied by the Immutable RoT, accurately reflecting the device's state, making sure different keys are derived in different states. For example, different life-cycle state will often yield a different derived key. All the platform keys reside within a security subsystem, hidden from the application core at all times.

6.1.11 Block control secure (BLK_CTRL_S_AONMIX)

The chip includes a block called BLK_CTRL_S_AONMIX which contains assorted configuration controls that can potentially have security implications. The BLK_CTRL_S_AONMIX registers are reset only by when ELE-AP is reset. These registers can only be accessed by TrustZone Secure World.

BLK_CTRL_S_AONMIX has the following functions:

- Cortex-M NVIC Interrupt Masking.
- Cortex-A GIC Interrupt Masking.
- WDOG Timer output masking for combined watchdog timer SoC output pin.

6.2 Edgelock secure enclave

The Edgelock Secure Enclave (ELE) supports the following features:

- SoC Life cycle management
- Cryptographic services
- Key Management
- Runtime Root of Trust
- One Time programmable Fuse controller
- Authenticated Debug with multiple debug domains
- Clock Frequency Monitoring

The ELE has a 32-bit bus with access to the SoC. The Edgelock secure enclave has multiple MU instances, and private buses to connect to security components outside EdgeLock Enclave such as BBSM, and FlexSPI's OTFAD.

Chapter 7
Fusemap

7.1 Overview

This section details the various fuses and their associated configuration settings. This chapter also describes various modes and the selection of the required boot devices.

NOTE

The fuses marked as “Reserved” are reserved for NXP internal (and future) use only, or are security related. Customers must not attempt to burn these, because the IC behavior may be unpredictable. The reserved fuses can be read as either '0' or '1'. Security-related fuses are described in the Security Reference Manual (SRM) for this device.

7.1.1 Bank redundancy

Fuses can generally only be set from 0 to 1 and the most common failure is a 1 appearing as a 0. The simplest mechanism to verify these failures is redundancy. Redundancy is when two bits are OR'd together so that if either is set, then the output value is a 1.

In case of fuses protected by redundancy, only 16 of the 32 bits are available (bits 15:0 are automatically copied to 31:16) and checked on read. This allows for multiple programming operations on a word, so bits can be configured at different times.

NOTE

In shadow registers, the inverted version of [15:0] will show in [31:16]. In the fuse array, [31:16] is the same as [15:0].

NOTE

The monotonic array requires redundancy, as it’s programmed individually

7.1.2 Lock bits

There are three kinds of lock bits, write lock, overwrite lock and read lock. If a word attribute of read / write / over-write lock is not set, then it can be read sensed, programmed and the shadow registers can be overwritten.

The write lock bits are used by the controller to control writing/programming of the fuse bits. A set of fuse words used for the same function can be write locked using a single fuse bit. If the fuse write lock is set, it indicates that the associated fuse bits can no longer be updated.

Read lock bits are a specific set of fuses that give you the ability to block bus access to specific words. When a fuse is read locked, it will be loaded to shadow registers during boot, but won't be readable via bus access.

Over-write lock bits are used to disable the feature of allowing SW to write to the shadow registers, enabling emulation of the function without committing to the fuse value.

Refer the attached spreadsheet for the chip fusemap descriptions.

7.1.3 API fuse access

Reading a fuse through an API needs the fuse word index. Fuse word index is calculated by: Fuse index / 32 (interer), or Bank number * 8 + word. Writing a fuse requires bit position, and the number of bits to be written. Please refer to the latest version of *EdgeLock® Secure Enclave (Advanced Profile) (ELE-AP) User Guide* for more details.

7.2 Boot Fusemap

This section describes the various modes and the selection of the required boot devices. The device select is specified by the BOOT_CFG0[3:0] fuses. See i.MX 93 Applications Processor Reference Manual for more details.

Table 31. Boot configuration fuses

Boot_CFGn[Bits]	7	6	5	4	3	2	1	0
0[7:0]	FlexSPI_NAND_4K_Page_Selection 0 - Not override the default selection (2K page) 1 - Override, 4K page is selected	FlexSPI_NOR_Boot_Override_Selection 00 - Flex SPI (Hyperflash 1.8V) 01 - Flex SPI (Flash with 4B READ(0x13)default supported) 10 - Default Octal mode (Micron, Nice to have) 11 - Default Octal mode (Mxic, Nice to have)		Override_FlexSPI_NOR_Boot_Selection 0 - Do not override 1 - Override	Boot_Mode_Fuses BootRom will retrieve boot mode from these fuses instead of Boot_Mode pins if (BT_FUSE_SEL=1 && Boot_Mode=Internal Fuses) Or (Force FORCE_BT_FROM_FUSE=1) 0000 - 1111			
0[15:8]	FLEXSPI_FCB_COPY_FLAG Indicate whether FCB copy is must-have for OEM closed parts: 0 - FCB copy is must-have; 1 - FCB copy is optional.	FLEXSPI_AUTOPROBE_FEQ_SEL Frequency options of FlexSPI NOR bus clock for autoprobe: 000 - 24MHz 001 - 133 MHz 010 - 166 MHz 011 - 200 MHz 100 - 80MHz 101 - 100MHz			FLEXSPI_NOR_FCB_OFFSET Select the offset of FlexSPI NOR FCB. 00 - 0x400B 01 - 0x600B 10 - 0x800B 11 - 0xA00B		FLEXSPI_NOR_DATA_IOMUX_SION_EN 0 - SION is disabled 1 - SION is enabled	uSDHC_Device_Type[1] 0 - eMMC 1 - SD
0[23:16]	FlexSPI_NOR_Reset_Pin_selections	uSDHC IO Voltage Selection 0 - 3.3 V 1 - 1.8 V	Reserved	eMMC fast boot 0 - Disabled 1 - Enabled	eMMC Speed Selection 0 - Normal Speed 1 - High Speed	SD Speed Selection: 00 - Normal/SDR12 01 - High/SDR25 10 - SDR50 11 - SDR104 eMMC/MMC Bus width: 00 - 4 bit 01 - 8 bit 10 - 4 bit DDR (eMMC 4.4) 11 - 8 bit DDR (eMMC 4.4)		SD Loopback Clock Source Sel (for SDR50 and SDR104 only) 0 - Through SD pad 1 - Direct

Table continues on the next page...

Table 31. Boot configuration fuses (continued)

Boot_CFGn[Bits]	7	6	5	4	3	2	1	0
0[31:24]	Force_Serial_Download Override all other Boot_Mode & Boot Mode Fuses and enter Serial Download	Force_Boot_from_Fuse 0 - Disable 1 - Override Boot_Mode pins by boot mode fuses	Boot Fuse Select (BT_FUSE_SEL) 0 - Boot fuses not programmed 1 - Boot fuses programmed yet	Boot Frequency select (BT_FREQ_SEL) 0 - Normal clock frequency 1 - Low clock frequency	FlexSPI NOR Reset Pin selections 00000 -- GPIO[x].0 11111 -- GPIO[x].31 **NOTE** Port is selected by "FlexSPI NOR Reset Port" fuses			
1[7:0]	SD Calibration Step 00 - 1 delay cell 01 - 2 delay cell 10 - 3 delay cell 11 - 5 delay cell		uSDHC Power Cycle Interval 00 - 20 ms 01 - 10 ms 10 - 5 ms 11 - 2.5 ms		uSDHC Power Cycle Delay 0 - 5 ms 1 - 2.5 ms	uSDHC Power On Polarity 0 - Low 1 - High	Reserved	Fast Boot Acknowledge Disable 0 - Boot Ack Disabled 1 - Boot Ack Enabled
1[15:8]	uSDHC PAD Settings Low bits The override value for uSDHC pad settings (Lowest 8 bits) 00000000 - 11111111							
1[23:16]	Reserved	uSDHC DLL Enable 0 - Disable DLL for SD/eMMC 1 - Enable DLL for SD/eMMC	uSDHC Override Pad Settings (using PAD_SETTINGS value)	Power Cycle Enable 0 - No power cycle 1 - Enabled	uSDHC PAD Setting High bits The override value for uSDHC pad settings (High bits) 0000 - 1111			
1[31:24]	uSDHC DLL Select 0 - DLL Slave Mode 1 - DLL Override Mode	MMC DLL DLY Delay target for uSDHC DL It is applied to slave mode target delay or override mode target delay depends on DLL Override fuse bit value 0000000 - 1111111						
2[7:0]	FlexSPI NAND Search Stride 00 - 64 pages 01 - 128 pages 10 - 256 pages		FlexSPI NAND Search Count 00 - 2 01 - 2 10 - 4		FlexSPI NAND CS Interval 00 - 100 ns 01 - 200 ns 10-400 ns 11-50 ns	FlexSPI NAND Column Address Width 00 - 12 01 - 13 10 - 14		

Table continues on the next page...

Table 31. Boot configuration fuses (continued)

Boot_CFGn[Bits]	7	6	5	4	3	2	1	0
	11 - 32 pages		11 - 8				11 - 15	
2[15:8]	FlexSPI DLL Value The delay line value only for HyperFlash 00000000 - 11111111							
2[23:16]	FlexSPI NAND Busy Bit Offset Override 0 - Do not override 1 - Override			FLEXSPI_NOR_AUTO_PROBE_EN 0 - Auto probe is disabled 1 - Auto probe is enabled	Recovery SPI Addressing 0 - 3-bytes (24-bit) 1 - 2-bytes (16-bit)	SPI SS Active 0 - Active low 1 - Active high	SPI SCLK Active 0 - Active high 1 - Active low	SPI Clock/ Data Phase Control 0 - Tx on falling, Rx on rising 1 - Tx on rising, Rx on falling
3[31:0]	Reserved			FLEXSPI_NOR_AUTO_PROBE type 00x - QuadSPI NOR 01x - MxicOctal 10x - MicronOctal 11x - AdestoOctal		FlexSPI NOR Reset Pin selection 000 - Not reset boot device 001 - GPIO port1 010 - GPIO port2 011 - GPIO port3 100 - GPIO port4		
4[31:0]	Reserved							
5[31:0]	Reserved							
6[31:0]	Reserved							
7[31:0]	Reserved							
8[7:0]	Reserved							
8[15:8]	Secondary_boot_offset Offset = 1 MB*(2^n). n = 2 if fuse is 0. n = 0 if fuse is 2. Secondary boot flow is disabled if fuse value is bigger than 10. 0000 - 4 MB 0001 - 2 MB 0010 - 1 MB 0011 - 8 MB 0100 - 8MB 0101 - 32MB ... 1001 - 512 MB EMMC :							

Table continues on the next page...

Table 31. Boot configuration fuses (continued)

Boot_CFGn[Bits]	7	6	5	4	3	2	1	0
	254 - last 128 M 255 - last 24K							
8[23:16]	M33_WDOG_Enable (WDOG0) 0 - Disabled 1 - Enabled	Reserved	USB1_SDP_Disable 0 - Not disabled 1 - Disable	Disable_M33_SPI_Recovery 0 - Disabled 1 - Enabled	Disable_A55_SPI_Recovery 0 - Disabled 1 - Enabled	USB2_SDP_Disable 0 - Not disabled 1 - Disable	M33_D_Cache_Disable 0 - Enabled 1 - Disabled	A55_D_Cache_Disable 0 - Enabled 1 - Disabled
8[31:24]	Reserved	PLL_LATE_EN 0 - Configure PLLs before installing ROM patch 1 - Configure PLLs after instaill ROM patch	USB_HID_EP1_Out_Disable On all USB Inst 0 - USB EP1 Transfer(EP 1 Out enable) 1 - USB EP0 Transfer(EP 1 Out disable)	A55_WDOG_Enable (WDOG3) 0 - Disabled 1 - Enabled	WDOG_Timeout_Select • 00 – 1s (Depends on the WDOG reset number) • 01 – 0.5s • 10 – 0.25s • 11 – 0.125s		M33_I_Cache_Dis 0 - Enable M33 I-Cache during ROM Boot 1 - Disable M33 I-Cache during ROM Boot	A55_I_Cache_Dis 0 - Enable A55 I-Cache during ROM Boot 1 - Disable A55 I-Cache during ROM Boot
9[31:0]	Reserved							
10[31:8]	Reserved							
11[31:0]	Reserved							

1. USDHC1 port supports both eMMC boot and SD boot, while USDHC2 only support SD boot. BOOT_CFG0[8] can be used to change the boot device type of USDHC1 from eMMC to SD

7.3 Fusemap descriptions table

Please refer the attached spreadsheet for the chip fusemap descriptions.

7.4 Programming Fuses

This section provides an overview on how to read and program eFuses. The OTP memory is part of the security subsystem and is controlled by the ELE. The ELE Firmware provides API functions which are used to interact with OCOTP CTRL block. Please refer the EdgeLock Enclave (ELE) API Reference Guide for more information.

Chapter 8
Fuse Status Block

8.1 Introduction

The Fuse and Status block (FSB) is an interface to allow the SoC to directly read public fuses and EdgeLock Secure Enclave status information.

FSB also provides SOC with write access to some of the OCOTP registers and shadow SRAM.

8.1.1 Features

The FSB includes the following features:

- Allow the SoC to read certain fuses and status registers
- Fuse values can be read from APB register interface
- FSB provides read/write access to OCOTP registers that hold status of fuse accesses
- FSB does not gate in any way accesses to fuse values and the OCOTP status registers
- FSB implements support for SoC to abort transfers on the FSB APB when accessing fuses
- FSB supports both read and write accesses to OCOTP fuses and status registers

8.2 Memory map and register definition

This section includes the FSB module memory map and detailed descriptions of all registers.

FSB does not by itself generate bus errors on accesses to its register space. However, FSB will return a bus error it might get from OCOTP on accesses to fuse and OCOTP status registers. Unsupported accesses or accesses to reserved space are ignored or return 0.

8.2.1 FSB register descriptions

8.2.1.1 FSB memory map

FSB1 base address: 4751_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	FSB Version ID Register (VERID)	32	R	0202_0001
1C	FSB Status Register (FSB_STATUS)	32	R	0000_0000
64	Access Count Register (ACCESS_COUNT)	32	RW	0000_0000
1000	Fuse Status Register (FUSE_STAT)	32	RW	0000_0000
1004	Fuse Event Register (FUSE_EVNT)	32	RW	0000_0000
1008	Fuse Interrupt Enable Register (FUSE_INT_EN)	32	RW	0000_0000
100C	Fuse Interrupt Register (FUSE_INT)	32	R	0000_0000
1080	Unprogrammed Fuse Counter (UNPROG_CTR)	32	R	0000_0000
8000 - 87FC	Fuse Value Registers (FUSE0 - FUSE511)	32	R	0000_0000

8.2.1.2 FSB Version ID Register (VERID)

This register records the specific version of the FSB module in the chip.

Offset

Register	Offset
VERID	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECO								MISC							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31-24 MAJOR	Major Version Number This read only field returns the major version number.
23-16 MINOR	Minor Version Number This read only field returns the minor version or revision number. Starting with minor version 2, FSB supports OCOTP status register space in addition to the fuse space and enables write transfers in these areas of the address map.
15-8 ECO	ECO Version Number This read only field returns the ECO version number.
7-0 MISC	Feature Specification This read only field returns the status of various configuration options indicating features present in this instance of FSB. xxxx_xx1x - MU IRQ steering is enabled. xxxx_xxx1 - Support for aborted transfers to OCOTP fuse space on the FSB slave APB is enabled.

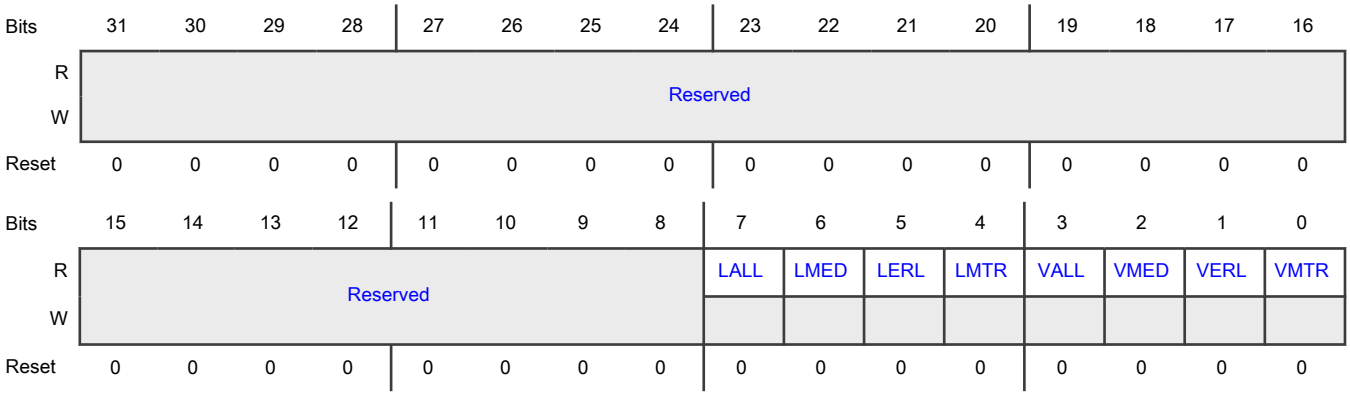
8.2.1.3 FSB Status Register (FSB_STATUS)

FSB status. For now the only status is whether or not fuse values are loaded and valid. At reset, the fuse values will not be valid. FSB does not gate accesses to OCOTP shadow RAM based on the state of these bits.

Offset

Register	Offset
FSB_STATUS	1Ch

Diagram



Fields

Field	Description
31-8 —	Reserved
7 LALL	All fuses loaded. The fuse values are loaded and can be read by the SoC. At boot, fuses may take some clock cycles to be available for SW to read.
6 LMED	Medium fuses loaded. The fuse values are loaded and can be read by the SoC. At boot, fuses may take some clock cycles to be available for SW to read.
5 LERL	Early fuses loaded. The fuse values are loaded and can be read by the SoC. At boot, fuses may take some clock cycles to be available for SW to read.
4 LMTR	MTR Trim fuses loaded. The fuse values are loaded and can be read by the SoC. At boot, fuses may take some clock cycles to be available for SW to read.
3 VALL	All fuses valid. The fuse values are valid and can be read by the SoC. At boot, fuses may take some clock cycles to be available for SW to read.
2 VMED	Medium fuses valid.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	The fuse values are valid and can be read by the SoC. At boot, fuses may take some clock cycles to be available for SW to read.
1 VERL	Early fuses valid. The fuse values are valid and can be read by the SoC. At boot, fuses may take some clock cycles to be available for SW to read.
0 VMTR	MTR Trim fuses valid. The fuse values are valid and can be read by the SoC. At boot, fuses may take some clock cycles to be available for SW to read.

8.2.1.4 Access Count Register (ACCESS_COUNT)

This register contains a count of the number of times an FSB register has been accessed.

Offset

Register	Offset
ACCESS_COUNT	64h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 COUNT	Access Count Register. Each time an FSB is read or written, this register will increment. Writing a value to this register will set a new starting value. Writing a new value takes precedence over incrementing this register.

8.2.1.5 Fuse Status Register (FUSE_STAT)

Reports the status and address of the last fuse shadow read. Writing to this register will clear the register contents.

If the fuse shadow read reports a RD_ERR or RLD_ERR then the Shadow Loading, Fuse Error, Unprogrammed, Zeroized, ECC and RED status bits will be 0 regardless of the actual fuse status.

Offset

Register	Offset
FUSE_STAT	1000h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ADDR_ERR	0	RD_ERR	RLD_ERR	LOADING	FUSE_ERR	SHDW_UN...	SHDW_ZE...	0							
W	0		0	0	0	0	0	0								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FUSE_ADDR															
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 ADDR_ERR	Address error High if the last fuse shadow read used an illegal address, including: • Writing to a fuse shadow • Accessing an address that is not 32-bit aligned • Accessing fuse which does not have a shadow in RAM • The Shadow RAM entry is not accessible from the FSB APB • The `soc_apb_fsb_access_disable` is asserted to disable FSB access • The SRAM controller FSM is in an error state preventing access to the SRAM
30 —	Reserved
29 RD_ERR	Read error High if the last fuse shadow read attempted to read a shadow that was read protected
28 RLD_ERR	Reload error High if the last fuse shadow was read during a Fuse Shadow Reload before it's section (early, mid or late) had finished loading and the Reload Control register setting for that fuse section is set to BLOCK

Table continues on the next page...

Table continued from the previous page...

Field	Description
27 LOADING	Shadow loading High if the last fuse shadow was read while the shadow was still loading from the fusebox. This prevents the ECC and Fuse Errors from being determined
26 FUSE_ERR	Fuse error High if the last fuse shadow read reported a serious error when the shadow was loaded from the fusebox. This includes: • Uncorrected Fuse ECC multi-bit error • Corrected Fuse RED multi-bit error • Transfer error from the fusebox This will also be high if the fuse shadow read reported a multi-bit error
25 SHDW_UNPROG	Shadow unprogrammed High if the last fuse shadow read was unprogrammed
24 SHDW_ZERO	Shadow zeroized High if the last fuse shadow read was zeroized
23-16 —	Reserved
15-0 FUSE_ADDR	Fuse address The address offset of the last fuse shadow read for which the status applies. Note that the 2 lsb will always be zero because the fuse address = fuse number left shifted by 2.

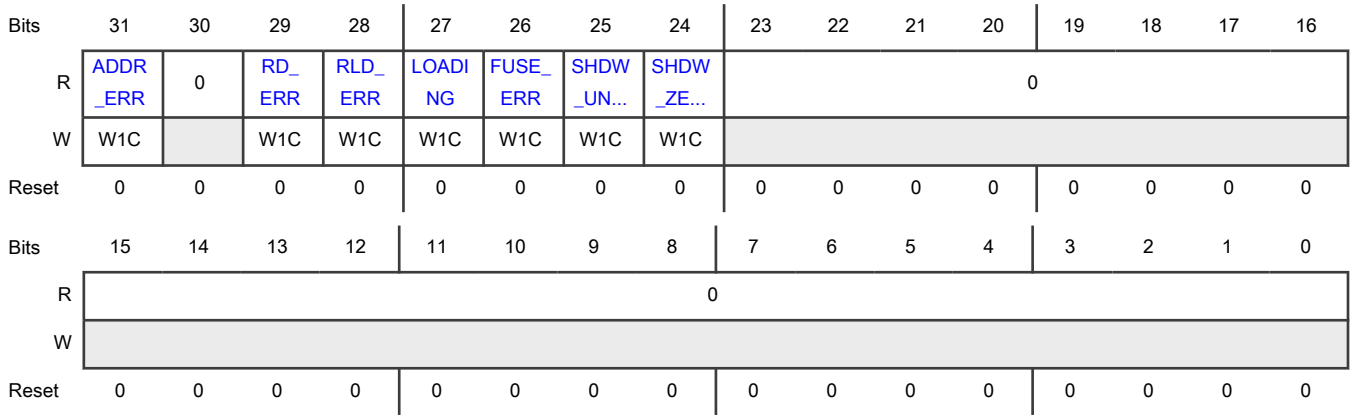
8.2.1.6 Fuse Event Register (FUSE_EVNT)

Accumulates the status from fuse shadow reads. Bits are set when the corresponding status bit is set and cleared by writing a '1' to the event bit.

Offset

Register	Offset
FUSE_EVNT	1004h

Diagram



Fields

Field	Description
31 ADDR_ERR	Address error Set if a shadow read used an illegal address, including: • Writing to a fuse shadow • Accessing an address that is not 32-bit aligned • Accessing fuse which does not have a shadow in RAM • The Shadow RAM entry is not accessible from the FSB APB • The `soc_apb_fsb_access_disable` is asserted to disable FSB access • The SRAM controller FSM is in an error state preventing access to the SRAM Write '1' to clear.
30 —	Reserved
29 RD_ERR	Read error Set if a fuse shadow read attempted to read a shadow that was read protected. Write '1' to clear.
28 RLD_ERR	Reload error Set if a fuse shadow was read during a Fuse Shadow Reload before it's section (early, mid or late) had finished loading and the Reload Control register setting for that fuse section is set to BLOCK. Write '1' to clear.
27 LOADING	Shadow loading Set if a fuse shadow was read while the shadow was still loading from the fusebox. This prevents ECC and Fuse Error from being determined. Write '1' to clear.
26 FUSE_ERR	Fuse error Set if a fuse shadow read reported a serious error when the shadow was being loaded from the fusebox. This includes:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Uncorrected Fuse ECC multi-bit error • Corrected Fuse RED multi-bit error • Transfer error from the fusebox This will also be set if a fuse shadow read reported a multi-bit error. Write '1' to clear.
25 SHDW_UNPROG	Shadow unprogrammed Set if a fuse shadow read was unprogrammed. Write '1' to clear.
24 SHDW_ZERO	Shadow zeroized Set if a fuse shadow read was zeroized. Write '1' to clear.
23-0 —	Reserved

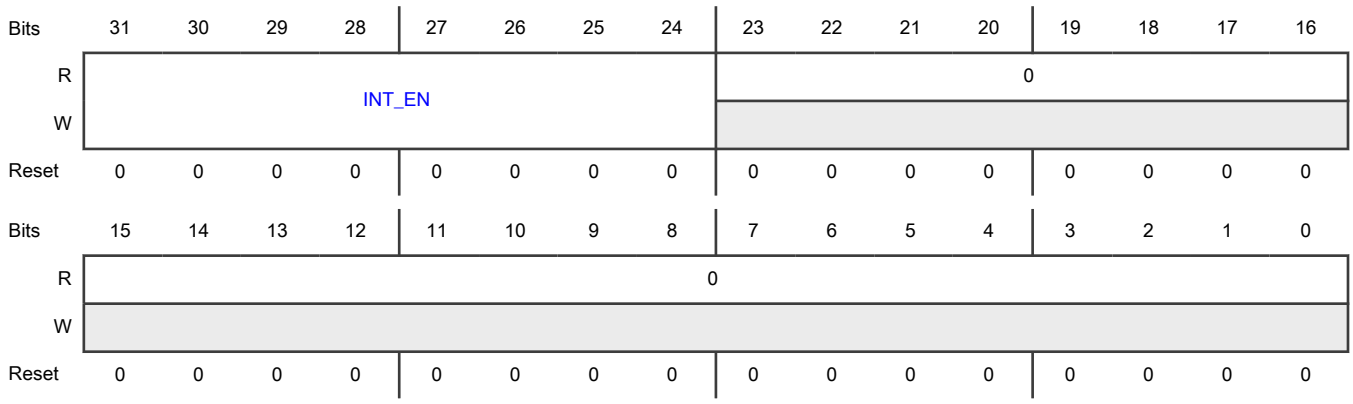
8.2.1.7 Fuse Interrupt Enable Register (FUSE_INT_EN)

Fuse Interrupt Enable Register

Offset

Register	Offset
FUSE_INT_EN	1008h

Diagram



Fields

Field	Description
31-24	Interrupt Enables

Table continues on the next page...

Table continued from the previous page...

Field	Description
INT_EN	AND-ed with the Fuse Event register value to generate the Fuse Interrupt bits. Setting interrupt enable register bits enables the corresponding interrupt bits. See the FUSE_EVNT register for a description of the interrupt signals.
23-0 —	Reserved

8.2.1.8 Fuse Interrupt Register (FUSE_INT)

Set when any of the Fuse Interrupt bits are non-zero. The CSSI ocotp_apb_fsb_irq output is high when this register is non-zero.

Offset

Register	Offset
FUSE_INT	100Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INT								0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 INT	Interrupts Reports the result of AND-ing the Fuse Event register with the Fuse Interrupt Enable register. See the FUSE_EVNT register for a description of the interrupt signals.
23-0 —	Reserved

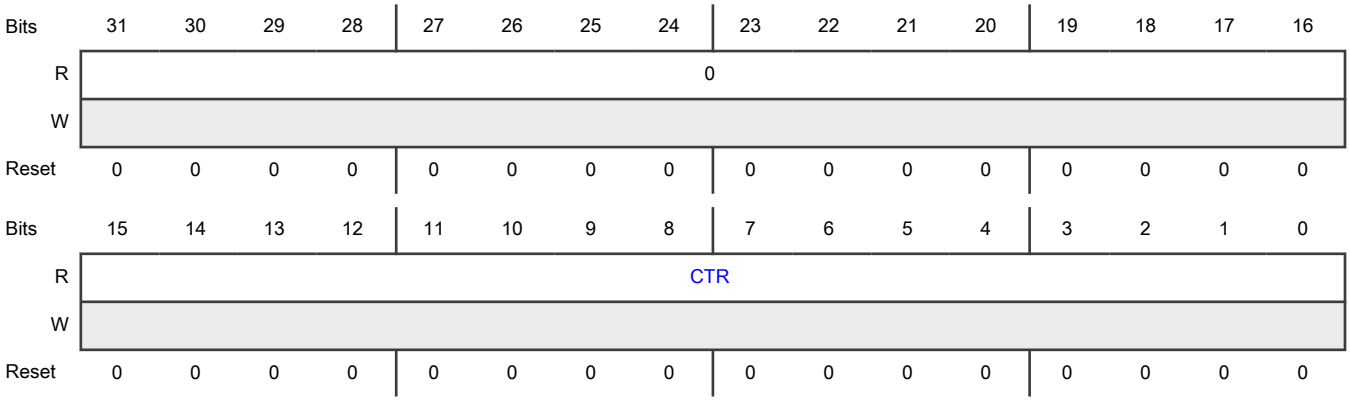
8.2.1.9 Unprogrammed Fuse Counter (UNPROG_CTR)

Reports the number of unprogrammed fuse words read in the last fuse shadow load, fuse shadow reload or CRC All operation.

Offset

Register	Offset
UNPROG_CTR	1080h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 CTR	Unprogrammed Fuse Counter The number of fuse words loaded that have not been programmed.

8.2.1.10 Fuse Value Registers (FUSE0 - FUSE511)

The various fuses shadow values can be read from this block of registers. The definition of the fuses is left to the SoC, and not defined here.

NOTE

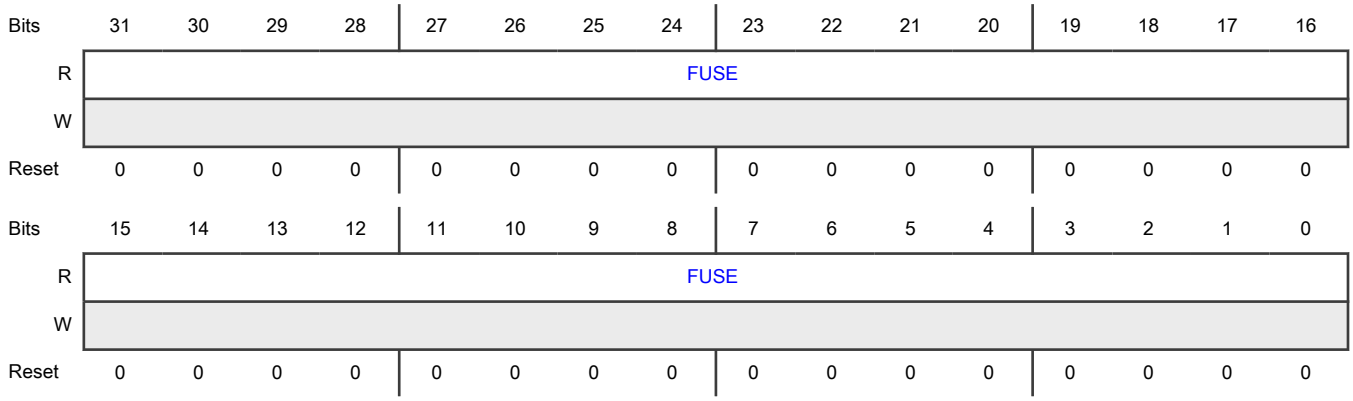
OCOTP will return a bus error if a write access is attempted to a read-only register.

Offset

For a = 0 to 511:

Register	Offset
FUSEa	8000h + (a × 4h)

Diagram



Fields

Field	Description
31-0 FUSE	Fuse Values SoC specific fuse information.

8.3 Functional description

The Fuse Status Block allows the SoC to read EdgeLock Secure Enclave fuse and status values without having to use the Messaging Unit.

Fuse values that are public are accessible via an APB interface between FSB and OCOTP. SoC accesses to fuse registers via FSB slave APB initiate transfers to OCOTP via FSB master APB.

FSB also provides access to OCOTP status registers that contain status of FSB's accesses to fuse values in OCOTP shadow RAM.

The FSB Status register provides state of various OCOTP status flags indicating when different types of fuses are valid and loaded. These flags do not affect behavior of FSB in any way.

FSB implements both read and write transfers to OCOTP status registers and shadow RAM. OCOTP and then FSB can return a bus error if a reserved location is accessed or when writes are not supported.

FSB implements support for SoC to abort transfers on the FSB APB when accessing fuses. This can happen if an access to the fuse area is attempted when OCOTP is not ready and the resulting excessive stall creates a timeout situation. An aborted transfer can be followed by a normal transfer without intervention of any kind. The presence of this feature is indicated via VERID.MISC[0]

The meaning of each fuse value is defined by EdgeLock Secure Enclave and the SoC. The Fuse Status Block does not imply specific meanings to any fuses.

Specific EdgeLock Secure Enclave status signals can also be connected to Fuse Status Block inputs and made available for reading by the SoC. The meaning of any status value is defined by EdgeLock Secure Enclave and the SoC. The Fuse Status Block does not imply specific meanings to any status inputs.

Because all the fuse and status values that are readable from the Fuse Status Block are public, there are no security mechanisms in this block.

8.3.1 Operating modes

Fuse Status registers are public values and can be read by any domain, at any time. There is no access control applied to the reading of any fuse or status registers.

8.3.2 APB errors

FSB asserts the Slave APB Error depending upon the memory region addressed.

8.3.2.1 Fuse Registers

When addressing the memory region containing the FUSE_STAT, FUSE_EVNT, FUSE_INT_EN, FUSE_INT and UNPROG_CTR registers APB bus errors will be reported under the following conditions:

- Reading an address that is not 32-bit aligned or does not match a control register
- Writing to an address that is not 32-bit aligned or does not match a writable control register

8.3.2.2 Fuse Values

When addressing the memory region containing the Fuse Values APB bus errors will be reported under the following conditions:

- Writing to a fuse shadow
- The address is not 32-bit aligned
- There is no fuse shadow RAM at this address
- There is a shadow RAM at this address, but it is not accessible by the FSB APB bus or the OCOTP SRAM Controller FSM is in an error state
- The OCOTP input `soc_apb_fsb_access_disable` is asserted to disable FSB access
- The shadow being read is read protected if the OP_CTRL[FSB_SHDW_ERR_EN] bit is set in OCOTP
- The shadow being accessed is reloading and the RLD_CTRL register in OCOTP is configured to block such accesses.

8.4 Acronyms and abbreviations

Table 32. Acronyms and abbreviated terms

Term	Meaning
FSB	Fuse and Status Block
MU	Messaging Unit - provides asynchronous communications between cores in different (and possibly protected) application domains.
MSC	Managed Security Component - an SoC resident security-related component which operates outside of the EdgeLock Secure Enclave boundary under SoC application core control, and for which EdgeLock Secure Enclave can manage/monitor security state as needed (via TROUT, using a TRIN connection). EdgeLock Secure Enclave can privately provision an MSC with a key information via a special-purpose masked bus (see TrustBus).
OCOTP	Secure On-Chip OTP Controller
TRIN	TRust INside - provides a monitoring interface for sensors or EdgeLock Secure Enclave for action.
Trust Bus	A private bus between EdgeLock Secure Enclave and Managed Security Components
TROUT	TRust OUTside - a logic block associated with a Managed Security Component which is capable of providing alert events to be reported to EdgeLock Secure Enclave for action.

Chapter 9
System Boot

9.1 Overview

The boot process begins at the Power-On Reset (POR) where the hardware reset logic forces the boot core to begin the execution starting from the on-chip boot ROM. The boot core is determined by a combination of boot mode pins and fuses. If the pins indicate boot from fuses, then the fuse value is selected. Boot mode pins and fuses have the same definition. The boot ROM code uses the state of BOOT_MODE[3:0] pins and fuses to determine the boot flow behavior of the device.

The platform boot is performed by the boot ROM and the EdgeLock secure enclave (ELE) ROM, working in concert to ensure that the intended and authorized set of images are loaded and allowed to execute on various processing domains contained by the SoC. The EdgeLock secure enclave ROM manages the secure boot flow based on ELE while the boot ROM mainly handles code fetch and other aspects of functional flow.

Please refer to the latest version of *EdgeLock® Secure Enclave (Advanced Profile) (ELE-AP) User Guide* for more details.

9.1.1 Features

The main features of the ROM include:

- Two boot types
  - Low Power Boot (Cortex-M33 ROM)
  - Single Boot (Cortex-A55 ROM)
- Support for booting from various boot devices
- Serial downloader support

The boot ROM supports the following boot devices:

- Serial NOR Flash via FlexSPI
  - Single (1-1-1 opcode-address-data) fast read
  - Quad (1-4-4 opcode-address-data) fast read
  - Octal (8-8-8 opcode-address-data) fast read
  - Auto probe for MxicOctal, MicronOctal and QuadSPINOR
- Serial NAND Flash via FlexSPI
  - 2K Page NAND
  - 4K Page NAND
- SD/eMMC via uSDHC
- Serial Downloader support on USB 2.0 only
- LPSPI (Recovery boot)

9.2 Boot Types

This chip supports two boot types:

- Single Boot
- Low Power Boot

NOTE

In low power boot mode, only Cortex-M33 core runs after POR. Cortex-A55 core cannot be triggered by Cortex-M33 firmware.

9.2.1 Boot Type Selection

The fuse settings for selecting a boot type is provided in the following table.

Table 33. Boot Type Selection

Boot Mode[3]	Result
0	Single Boot
1	Low Power Boot

9.2.2 Single Boot

In single boot, the Cortex-A55 ROM loads all containers and images, and then jumps to the A55 Firmware (FW).

If the container set has the Cortex-M33 firmware image, the Cortex-A55 ROM will read and place it in the Shared RAM (Cortex-M33 TCM) first, and then request the EdgeLock secure enclave to kick Cortex-M33. Refer to Single Boot Flow (Cortex-A55) for details.

NOTE

If there is a CM33 FW image in OEM container in single boot mode, the CM33 FW image must be prior to CA55 FW image

The EdgeLock secure enclave authenticates both the FW images if Secure Boot is enabled.

The following table provides the boot devices supported for single boot.

Table 34. Boot devices supported for single boot

Boot Stage	Cortex-A55 ROM	Cortex-M33 ROM
Primary and Secondary image set boot	eMMC on USDHC1 SD on USDHC1/2 FlexSPI NAND on FlexSPI1 FlexSPI NOR on FlexSPI1	N/A
Recovery boot	LPSPI 3/4	N/A
USB serial download	USB HID on USB1	N/A

9.2.3 Low Power Boot

In Low Power Boot (LPB), only the Cortex-M33 rom is running after Power-On Reset. Refer to Low Power Boot Flow (Cortex-M33) for the details.

The following table provides the boot devices supported for LP boot.

Table 35. Boot devices supported for LP boot

Boot Stage	Cortex-A55 ROM	Cortex-M33 ROM
Primary and Secondary image set boot	N/A	eMMC on USDHC1

Table continues on the next page...

Table 35. Boot devices supported for LP boot (continued)

Boot Stage	Cortex-A55 ROM	Cortex-M33 ROM
		SD on USDHC1/2 FlexSPI NOR on FlexSPI1 FlexSPI NAND on FlexSPI1
Recovery boot	N/A	LPSPI1/2
USB serial download	N/A	USB HID on USB1

9.3 Boot Modes

There are 3 different boot modes supported on this chip:

- Normal Boot Mode
- Boot from Internal Fuse Mode
- Serial Download Boot Mode

The boot mode can be selected via the Boot Mode pins or overridden by fuses.

9.3.1 Boot Mode pins

There are four boot mode pins used to select boot mode. The boot mode is selected based on the binary value stored in the internal BOOT_MODE register.

The boot mode is initialized by sampling the BOOT_MODE[3:0] inputs on the rising edge of the POR_B. After these inputs are sampled, their subsequent state does not affect the contents of the BOOT_MODE internal register. The state of the internal BOOT_MODE register may be read from the IPP_BOOT_MODE[3:0] field of the SRC Boot Mode Register (SRC_SBMR2).

The available boot modes are detailed in the table below.

Table 36. Boot mode selection

BOOT CODE	BOOT CORE	BOOT_MODE[3:0]	Function
0x0	Cortex-A55	0000	Boot From Internal Fuses
0x1		0001	Serial Download (USB1)
0x2		0010	USDHC1 8-bit eMMC 5.1
0x3		0011	USDHC2 4-bit SD 3.0
0x4		0100	FlexSPI Serial NOR
0x5		0101	FlexSPI Serial NAND 2K
0x8	Cortex-M33	1000	LPB: Boot From Internal Fuses
0x9		1001	LPB: Serial Downloader (USB1)
0xA		1010	LPB: USDHC1 8-bit 1.8V eMMC 5.1

Table continues on the next page...

Table 36. Boot mode selection (continued)

BOOT CODE	BOOT CORE	BOOT_MODE[3:0]	Function
0xB		1011	LPB: USDHC2 4-bit SD 3.0
0xC		1100	LPB: FlexSPI Serial NOR
0xD		1101	LPB: FlexSPI Serial NAND 2K

9.3.2 Normal Boot Mode

In this mode, the boot flow is controlled by the boot type selection, boot device selection, and the boot mode settings.

In this mode, the boot code does the following:

- The Cortex-M33 ROM or/and Cortex-A55 ROM perform hardware initialization, and loads the images from the primary boot device which is selected by the Boot Mode pins or fuses.
- Performs image validation - it requests the EdgeLock secure enclave ROM to do this.
- Then jumps to the Cortex-M33/Cortex-A55 FW entry point specified in the Image Container.

If any error occurs during normal boot, the Cortex-M33/Cortex-A55 ROM will try to boot from the Secondary Image Container Set which is in the same boot device (primary boot device). If this fails, the ROM will try to boot from the Recovery Boot device. If Recovery Boot also fails, the Cortex-M33/Cortex-A55 ROM will jump to the Serial Downloader Mode (see Serial Download Mode).

9.3.3 Boot from Internal Fuses Mode

This mode is similar to the Normal Boot mode but with one difference. In this mode, the Boot Mode pins are ignored. The boot ROM code uses the boot eFUSE settings only. If set to Boot from eFuses, the boot flow is controlled by the BT_FUSE_SEL eFUSE value. If BT_FUSE_SEL = 0, indicating that the boot settings have not been provisioned into the fuse yet, the Cortex-M33 ROM and Cortex-A55 ROM both jump directly to the Serial Downloader. If BT_FUSE_SEL = 1, ROM retrieves the Boot Mode setting from the BT_MODE_FUSES fuse, and then attempts to boot from the selected boot device. The first time a board is used, the default eFUSEs may be configured incorrectly for the hardware on the platform. In such a case, the Boot ROM code may try to boot from a device that does not exist. This may cause an electrical/logic violation on some pads. Using Boot from Fuses mode addresses this problem.

The first time the BT_FUSE_SEL = 0 eFUSE is encountered, it is not blown (thus setting BT_FUSE_SEL). This forces the ROM code to jump directly to the Serial Downloader. This allows a bootloader to be downloaded which can then provision the boot device with a Program Image and blow the BT_FUSE_SEL fuse and other boot configuration eFUSEs.

After reset, the Boot ROM code determines that the BT_FUSE_SEL fuse is blown (BT_FUSE_SEL = 1) and the ROM code performs the internal boot according to the new eFUSE settings. This allows a user to set BOOT_MODE[3:0] =4'b0000 on a production device and burn fuses on the same device (by forcing entry to the Serial Downloader), without pullups/pulldowns on the BOOT_MODE pins.

BT_FUSE_SEL is only an indicator of BOOT_MODE fuse been blown. It has no impact for Internal Boot Mode.

If the fuse, Force_BT_FROM_FUSE, has been blown, the Boot Mode pins will be ignored, and the BootROM will always retrieve the boot mode from the internal fuses.

9.3.4 Serial Download Mode

The Serial Downloader provides a means to download a Program Image to the chip over USB serial connection.

Both, the Cortex-M33 (LP Boot) and Cortex-A55 ROM (Single Boot) support serial download mode on USB 2.0, and only on USB OTG1. This mode can be disabled by fuse.

9.4 Boot Stages

Both the Cortex-M33 and Cortex-A55 ROMs support the following four boot stages.

1. Primary Image Container Set Boot (Primary Boot): Boot from the Primary Image Set Primary Boot Device.
2. Secondary Image Set Boot (Secondary Boot): Boot from the Secondary Image Set in Primary Boot Device.
3. Recovery Boot (if enabled via fuse): Boot from the image set in Recovery Boot Device (only one Image Set supported).
4. Serial Download Boot: load the image set via USB.

9.5 Image Container Set

9.5.1 Image Container Format

The image container format consists of the following parts.

- Container header
- Image array entry
- Signature block
- User program images and data

The high-level view of the container format is shown in the following figure.

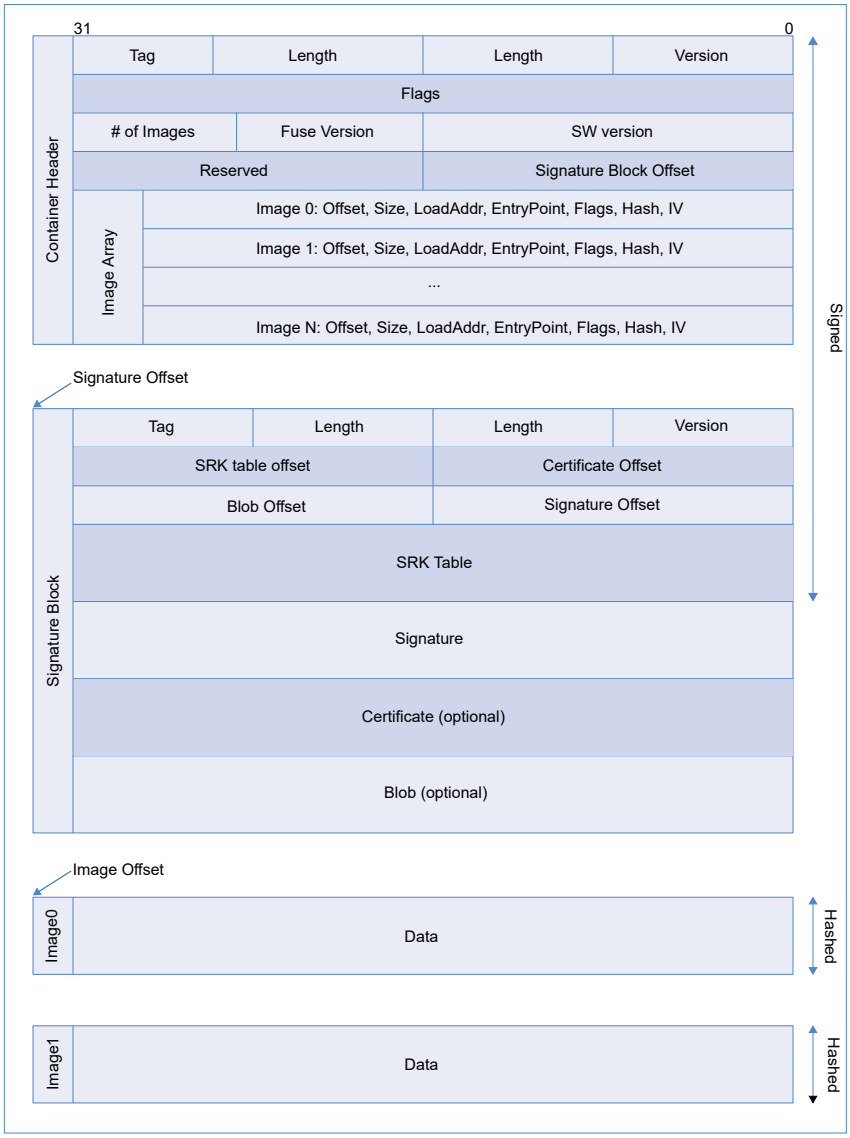


Figure 16. High level view of the container format

ROMs view of the offset/address in the Image container:

- **Blob/Certificate/Signature offset:** Relative to the beginning of Container Header start address, in bytes. 8 bytes alignment.
- **ImageArray.Imagex.Offset:** Relative to the container's start address (beginning of the 1st Container Header start address in the SD card (32KB), for instance), in bytes. Needs to be page aligned.
- **ImgeArray.Imagex.LoadAddr:** The address map is from the Cores view (required to be 4 bytes aligned):
  - For the images booted by Cortex-M33 ROM, the address map is from Cortex-M33 view.
  - For the images booted by Cortex-A55 ROM, the address map is from Cortex-A55 view.
  - 4 bytes aligned
- **ImageArray.Imagex.Entry:** This is from target core's local view. For example,
  - EdgeLock secure enclave FW entry address is from EdgeLock secure enclave view.
  - Cortex-M33 FW entry address is from Cortex-M33 view.

— Cortex-A55 FW entry address is from Cortex-A55 view.

9.5.1.1 Container Header

The container header contains essential information about the entire container image which is used by the ROM.

Table 37. Container header format

Offset	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
0x00	Tag	Length (MSB)	Length (LSB)	Version
0x04	Flags			
0x08	# of Images	Fuse Version	SW Version	
0x0c	Reserved		Signature Block Offset	
0x10	Image array (Please see Image Array Entry)			
	Signature Header (Please see Signature Block)			

The container header format fields are defined below.

- **TAG**: Must be 0x87
- **Length**: Size of the container in bytes, from header, up through and including signature block.
- **Version**: Must be 0x00
- **Flags**

Table 38. Flags bitfield description

Bit Field	Description
0-1	SRK Set 0x0 – Container not authenticated 0x1 – NXP SRK 0x2 – OEM SRK 0x3 – Reserved
2-3	Reserved (must be 0)
4-5	SRK Selection 0x0 – SRK1 is used 0x1 – SRK2 is used 0x2 – SRK3 is used 0x3 – SRK4 is used
6-7	Reserved (must be 0)
8-11	SRK Revoke Mask

Table continues on the next page...

Table 38. Flags bitfield description (continued)

Bit Field	Description
	bitmask used to indicate which SRKs to revoke – zero bits set means no SRKs are to be revoked, bit0 – SRK1 is to be revoked, bit1 – SRK2 is to be revoked, bit2 – SRK3 is to be revoked, bit3 – SRK4 is to be revoked,
12-15	Reserved (must be 0)
16-31	Reserved (must be 0)

- **# of images**: Number of images in the image array
- **Fuse Version**: This value must be equal to or greater than the version stored in the fuses to allow loading this container
- **SW Version**: Used to select between multiple images with same Fuse version field.
- **Signature Block Offset**: Offset in bytes from beginning of header to signature block (keys, signature, blob)

9.5.1.2 Container Header Offsets

The following table provides the offset for the 1st image set container header on various boot devices.

Table 39. The offset of the 1st image set

Boot Device	1st Image Container Offset
SD	32KB
eMMC	0 if located in Boot Partition and 32K if in User area
SPI NAND	Right behind FCB/DBBT
FlexSPI NOR	0x1000

The 2nd image set container header and its signature block will be right after the 1st image set container header and it's signature block, with 1KB alignment.

9.5.1.3 Signature Block

Table 40. Signature block format

Offset	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
0x00	Tag	Length (MSB)	Length (LSB)	Version
0x04	SRK Table Offset		Certificate Offset	
0x08	Blob Offset		Signature Offset	
0x0C	Reserved (for 64bit alignment purposes)			

Table continues on the next page...

Table 40. Signature block format (continued)

Offset	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
	32 bit			
0x10	SRK Table (Please see SRK Table)			
…	Signature (Please see Signature Header)			
…	Certificate (Please see Certificate Header)			
…	Blob (Please see Blob (DEK))			

The signature block format fields are defined below.

- **TAG**: 0x90
- **Version**: 0x00
- **SRK Table Offset**: Offset in bytes from beginning of signature block to SRK table. If 0x0000, then SRK table is not present. This should be the case only if the container is not authenticated, and the SRK key set field in the Container header should be 0.
- **Certificate Offset**: Offset in bytes from beginning of signature block to signing key certificate. If 0x0000, then only a SRK is used in signature verification
- **Signature Offset**: Offset in bytes from beginning of signature block to signature
- **Blob Offset**: Offset in bytes from beginning of signature block to DEK Blob. If 0x0000, then no images are encrypted

NOTE

1. SRK Table, Signature, Certificate and Blob must be in the order shown
2. Certificate and Blob are optional
3. SRK Table, Certificate, Blob, and Signature must start on 64 bit boundaries. Any padding needed to make this happen must be zeros.
4. The container signature is calculated of all data from beginning of container header to the last byte before the signature starts. In other words, most of the signature block itself is included in the signature calculations.

9.5.1.3.1 Certificate Header

The table below provides details about the Certificate Header structure:

Table 41. Certificate format

Offset	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
0x00	Tag	Length (MSB)	Length (LSB)	Version
0x04	Permissions	Permissions(inverted)	Signature Offset	
0x08	Public Key			
	Signature			

The certificate format fields are defined below.

- **TAG**: 0xaf

- **Version**: 0x00
- **Permissions**: Bitmask used to indicate what a certificate can be used for
- **Permissions (inverted)**: Must be the same as the Permissions field bitwise inverted
- **Signature Offset**: Offset in bytes from beginning of certificate header to signature
- **Public Key**: Public key. Format matches SRK table “SRK record” entry, for both the RSA as well as ECDSA type keys.
- **Signature**: Signature of certificate. The signature is calculated over all data from the beginning of the certificate up to one byte before the signature itself: from the “Tag” byte through and including the public key SRK record.

9.5.1.3.2 Signature Header

The table below provides details about the Signature Header structure:

Table 42. Signature format

Offset	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
0x00	Tag	Length (MSB)	Length (LSB)	Version
0x04	Reserved			
0x08	Signature Data			

The signature format fields are defined below.

- **TAG**: 0xD8
- **Version**: 0x00
- **Signature Data**:Signature.
  - RSA-PSS encoding
  - ECDSA: The r and s components are padded with leading zeros to make them each 8 bit aligned, then concatenated – r|s

9.5.1.3.2.1 SRK Table

The SRK table determines the signature type and length, and consists of 1 to 4 SRK records which contains the public keys. The format of the SRK table is inherited from HAB v4.x

Table 43. SRK table format

Offset	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
0x00	Version	Length of SRK Table		Tag
0x04	SRK Record 1			
-	SRK Record 2			
-	SRK Record 3			
-	SRK Record 4			

The SRK table fields are defined below.

- **TAG**: 0xD7
- **Length**: The size of the SRK table in bytes

- **Version**: 0x42

9.5.1.3.2.2 SRK Record

Table 44. SRK record format

Offset	Bits 31-24	Bits 23-16		Bits 15-8	Bits 7-0
0x00	Signing Algorithm	Length of SRK			Tag
0x04	SRK Flags	Reserved	Key Size/Curve		Hash Algorithm
0x08	RSA: Exponent Length (bytes)	RSA: Modulus Length (bytes)			
	ECDSA: Y Length (bytes)	ECDSA: X Length (bytes)			
0x0C	RSA: Modulus (Big endian)				
	ECDSA: X (Big endian)				
-	RSA: Exponent (Big endian)				
	ECDSA: Y (Big endian)				

The SRK record fields are defined below.

- **TAG**: 0xE1
- **Length**: The size of the SRK record
- **Signing Algorithm**:
  - RSA-PSS: 0x22
  - ECDSA: 0x27
- **Hash Algorithm**:
  - SHA-256: 0x00
  - SHA-384: 0x01
  - SHA-512: 0x02
- **Key Size/Curve**:
  - PRIME256V1: 0x1
  - SEC384R1: 0x2
  - SEC521R1: 0x3
  - RSA2048: 0x5
  - RSA3072: 0x6
  - RSA4096: 0x7
- **SRK Flags**:
  - None: 0x00
  - CA Flag: 0x80

NOTE

The keys in the SRK table must match the signing key in algorithm and key length

9.5.1.3.2.3 SRK Revoke

The EdgeLock secure enclave sets fuses based on the fields in the container or a message. If an SRK revoke bit or field return flag is set, the EdgeLock secure enclave will set that fuse. Since the containers are issued by the same entity that owns the SRKs, or is authorized by the SRKs owner, it is expected that the SRK revoke bits will correctly reflect the SRK status registered in the fuses, or revoke additional keys. The EdgeLock secure enclave will check that the SRK revoke bits in a parsed container match what is currently set in the fuses, or will revoke the additional keys corresponding to the revoke bits by updating the fuses.

9.5.1.3.2.4 SRK Fuses

The hash for each of the entire SRK table is stored in the SRK fuses. This allows the boot code to validate the keys in the SRK table from each container. For any of the 4 keys in SRK table that are unused, the corresponding key bits will be set to 0.

NOTE

SHA256 is used to hash the SRK tables

9.5.1.3.3 Image Array Entry

The boot ROM loads images in the order that the images appear in the array.

Table 45. Image Array Entry format

Offset	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
0x00	Image Offset			
0x04	Image Size			
0x08	Load Address (64 bits)			
0x0c				
0x10	Entry point (64 bits)			
0x14				
0x18	Flags			
0x1c	Image meta data			
0x20	Hash (512 bits)			
…				
0x60	IV (256 bits)			

The image array entry format fields are defined below.

- **Image Offset**: Offset in bytes from start of container header to beginning of the image
- **Image Size**: Size of the image in bytes
- **Load Address**: Address where the image is copied to in memory (absolute address in system memory map) by the ROM
- **Entry Point**: Entry point of the image (absolute address). Only valid for executable image types.
- **Flags**:

Table 46. Flags bitfield description

Bit Field	Description
0-3	Type of image 0x3 Executable 0x4 Data 0x7 Provisioning image 0x9 Provisioning data All other values reserved (These values extend HAB IVT V1)
4-7	Core ID 0x1 Cortex-M33 0x2 Cortext-A55 All other values reserved
8-10	Hash type (all images in the container have the same HASH size) 0x0 – SHA256 0x1 – SHA384 0x2 – SHA512
11	Encrypted Indicate if this image is encrypted or not
12-31	Reserved

- **Image meta data**: For NXP FW private usage.
- **Hash**: SHA hash of image. Fixed size at 512 bits. Left aligned and padded with zero for hash sizes below 512 bits.
- **IV**: Used only for encrypted images (zero otherwise): SHA256 of the plain text image. Fixed size at 256 bits. The lower 128-bit part of the SHA256 value will be retained as IV in the encryption/decryption process.

9.5.1.3.4 Blob (DEK)

The Blob holds the Wrapped Data Encryption Key (DEK). It is optional if no images are encrypted. The format of the Blob is inherited from HAB v4.x.

Table 47. Blob format

Offset	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
0x00	Tag	Length (MSB)	Length (LSB)	Version
0x04	Reserved	Algorithm	Size	Flags
0x08	Wrapped key			

The blob fields are defined below.

- **TAG**: 0x81

- **Length**: Size of the blob in bytes
  - 0x48: 72 bytes, corresponding to a 128 bit AES key blob.
  - 0x50: 80 bytes, corresponding to a 192 bit AES key blob.
  - 0x58: 88 bytes, corresponding to a 256 bit AES key blob.
- **Version**: 0x00
- **Algorithm**: 0x03: AES CBC
- **Size**:
  - 0x10: 128 bit AES key
  - 0x18: 192 bit AES key
  - 0x20: 256 bit AES key
- **Flags**:

  0x80: KEK key flag

  0x01: DEK

9.5.2 Primary Boot Container Set

The images that will be loaded as part of the boot process are stored in containers. A container can contain one or multiple images, as blocks of data specifying the load address, target address, and so on.

9.5.2.1 Secondary Boot Container Set

The Secondary Boot Container Set is a copy of the Primary Container Set. ROM will always try to load the Primary Container Set first and if the Primary Container Set fails, ROM will then attempt to load the Secondary Container Set.

9.5.2.2 Primary Image Set and Secondary Image Set

The Primary Image Set is used in the Primary boot stage while the Secondary Image Set is used in the Secondary boot stage.

If there are two valid image container sets in the primary boot device:

- Primary Image Container Set: The one that has the newer SW Version. If the SW Versions are the same, or for eMMC fast boot Scenario, Image Container Set0 will be the primary one.
- Secondary Image Container Set: The one that has the older SW Version. If the SW Versions are the same, or for eMMC fast boot Scenario, Image Container Set1 will be the secondary one.

If there is only one valid image container set in the primary boot device:

- Both, Primary Image Container Set and Secondary Image Container Set are this only one valid image container set.

Please refer Boot Image Container Sets Selection for information on primary and secondary image sets selection.

9.5.2.3 Typical Boot Image Sets Layout

The typical image layout for the single boot (handled by the Cortex-A55 ROM) is provided in the following table.

Table 48. Typical Cortex-A55 image layout for single boot

Primary Image Container Set	1st Container Header
	1st Signature Block
	Padding for 1KB alignment

Table continues on the next page...

Table 48. Typical Cortex-A55 image layout for single boot (continued)

	2nd Container Header (Optional)
	2nd Signature Block (Optional)
	EdgeLock secure enclave FW (Optional)
	FCB Copy Image (Optional)
	Cortex-M33 Image
	Cortex-A55 Image
…	
Secondary Image Container Set	

The typical image layout for the LP boot (handled by the Cortex-M33 ROM) is provided in the following table.

Table 49. Typical Cortex-M33 image layout for LP boot

Primary Image Container Set	1st Container Header
	1st Signature Block
	Padding for 1KB alignment
	2nd Container Header (Optional)
	2nd Signature Block (Optional)
	EdgeLock secure enclave FW (Optional)
	FCB Copy Image (Optional)
	Cortex-M33 Image
…	
Secondary Image Container Set	

Restrictions to Images:

- The length of all container header(s) in a boot image set is up to 8KB.
- If EdgeLock secure enclave FW exists, it must be in the EdgeLock secure enclave/NXP container, and the EdgeLock secure enclave/NXP container must be the first container in the boot image set.
- The target address of EdgeLock secure enclave FW must be in TCM.
- The target address of Cortex-M33 FW must be in TCM free space or FlexSPI1 NOR space (XIP).
- The target address of Cortex-A55 FW must be in OCRAM free space or FlexSPI1 NOR space (XIP).
- The images cannot overlap with each other.
- The image offset (EdgeLock secure enclave FW, Cortex-M33 FW, and Cortex-A55 FW) must be MAX[1KB, PAGE_SIZE_OF_BOOT_DEVICE] aligned.

- Except EdgeLock secure enclave FW, the target loading address must be MAX[1KB, PAGE_SIZE_OF_BOOT_DEVICE] aligned. For the EdgeLock secure enclave FW, the image size must be 4 bytes aligned.
- For eMMC fast boot or USB boot, images (including container headers) need to be continuous to its former.
- For Single Boot, Cortex-A55 FW is mandatory for the boot image set.
- For Low Power boot, Cortex-M33 FW is mandatory for the boot image set.
- For Low Power boot, Cortex-A55 FW is disallowed to be present in the boot image set.

NOTE

PAGE_SIZE_OF_BOOT_DEVICE value:

- For SPI NAND, it is 2KB/4KB.
- For SD/eMMC, it is 512B.
- For USB, it is 1KB.

9.6 Boot Flow

Although there are two system ROMs (A55 ROM and CM33 ROM), some concepts of ROM boot flow are common:

1. Up to two containers (NXP and OEM containers) are expected to handle by system ROM code while only OEM container is must-have and NXP container is optional.
2. In case NXP container is available in the boot image container set, the NXP container must be the 1st container followed by OEM container. System ROM code always handles the containers one by one based on their container headers order in the header of the boot image container set.
3. If a container has multiple boot images (OEM container may contain multiple images while only EdgeLock secure enclave FW is allowed in NXP container), the images in the container will be handled by ROM code one by one based on the images order in the container header in most cases. The exceptions are the FCB region copy image and the fast boot image, if these two images are available in the OEM container, they will be handled by ROM code prior to other OEM images no matter their orders in OEM container.

9.6.1 High Level Boot Flow for A55 ROM and CM33 ROM

1. System ROM code should initialize all HW/SW resources which system ROM code is needed in ROM boot phase, including stack/data sections, PLLs, clock root/source and peripherals.
2. The boot type (single boot/LP boot) is decided the setting on BT_MODE[3], and it's HW/EdgeLock secure enclave responsibility to release the boot core based on the setting of BT_MODE[3]. The logics of boot type selection are transparent to system ROM code.
3. System ROM code needs to select the one boot image container set based on both the SW versions of NXP/OEM container in the boot image container set and the boot stage.
4. System ROM code needs to address all containers in the selected boot image container set, includes load them into the target (XIP is also supported on XIP device), and request EdgeLock secure enclave to authenticate/verify the containers and their images.
5. If any failure happens, ROM code will go to the error handling, please refer to the Failure Handling in ROM for the details.
6. In case everything goes well, ROM code eventually jumps to the core image: CM33 FW in LP boot mode and CA55 FW in single boot mode.

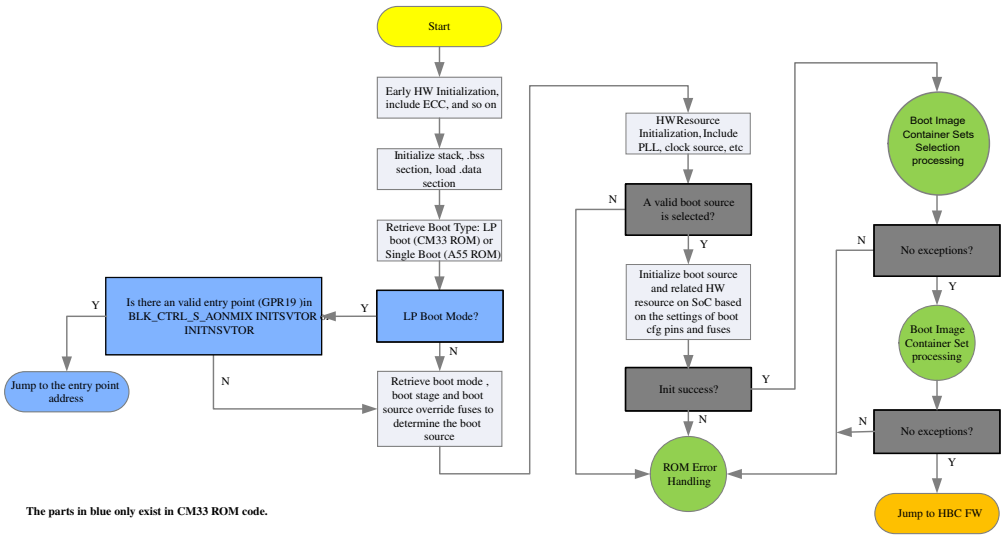


Figure 17. High Level ROM Boot Flow

9.6.2 ROM Early HW Initialization

9.6.2.1 RAM ECC

Because the ECCs of CM33 TCM and NICMIX OCRAM are enabled by default, so CM33 ROM should take care of CM33 TCM initialization while A55 ROM code needs to take care of NIXMIX OCRAM to avoid ECC error which may be caused by accessing to un-initialized RAM space. Please refer to ECC Support section for the details of how ROM code to initialize CM33 TCM/NICMIX OCRAM.

9.6.3 Boot Image Container Sets Selection

For primary boot stage and secondary boot stage, HBC ROM code boots from the selected primary boot device, and it is allowed up to two boot image container sets on primary boot device.

The offset of the 1st boot image container set is fixed on the primary boot device and hard code in ROM code, please refer to The Offsets of the Container Header section for the details. The offset of the 2nd boot image container set on the primary boot device is specified by the fuses “Secondary_boot_offset”:

Offset = 1MB * (2^n)

If Secondary_boot_offset = 0 → n = 2;

If Secondary_boot_offset = 2 → n = 0;

If Secondary_boot_offset = others and Secondary_boot_offset <= 10 → n = Secondary_boot_offset

If Secondary_boot_offset > 10 → Means there is no the 2nd boot image container set

In case there is only one boot image container set on boot source, ROM code regard it as primary boot image container set and will be used in primary boot stage; In case there is two boot image container sets on boot source, ROM code will pick up the one as primary boot image container set which will be used in primary boot stage and another→ one will be the secondary boot image set which will be used in secondary boot stage. It is determined by the SW version of NXP container and OEM container in two boot image container sets, below diagram shows the flow about how ROM code determine the primary and secondary boot image set.

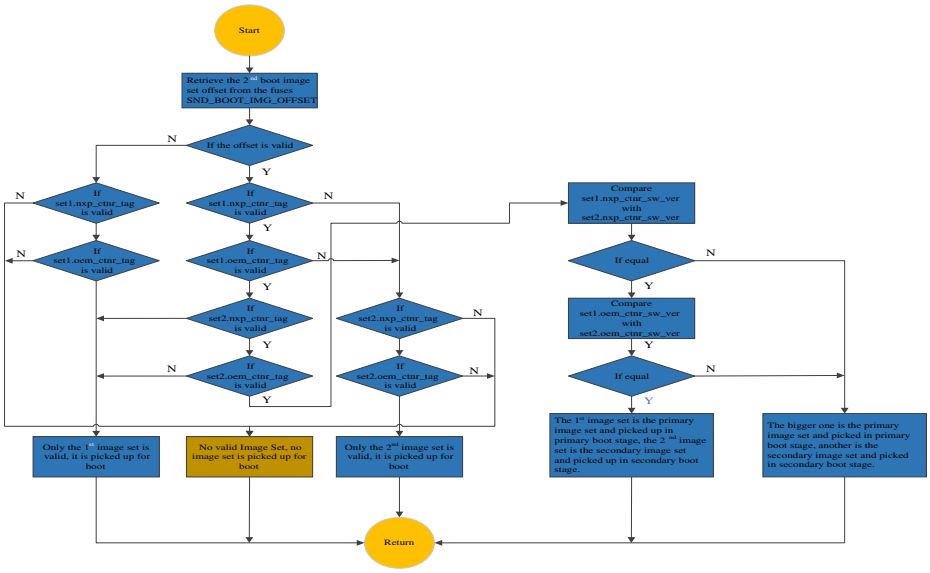


Figure 18. Boot Image Container Sets Selection

9.6.4 Boot Image Container Set Processing

Besides of the legacy regular processing of handling the containers in boot image set, a new AP Fast Boot processing is supported in single boot mode. In order to improve the security level of ROM boot, a FCB check processing is supported in CM33/CA55 ROM code. This processing is used in FlexSPI NOR boot, and to protect the FCB date from the attacks. Please refer to the FCB Check for details.

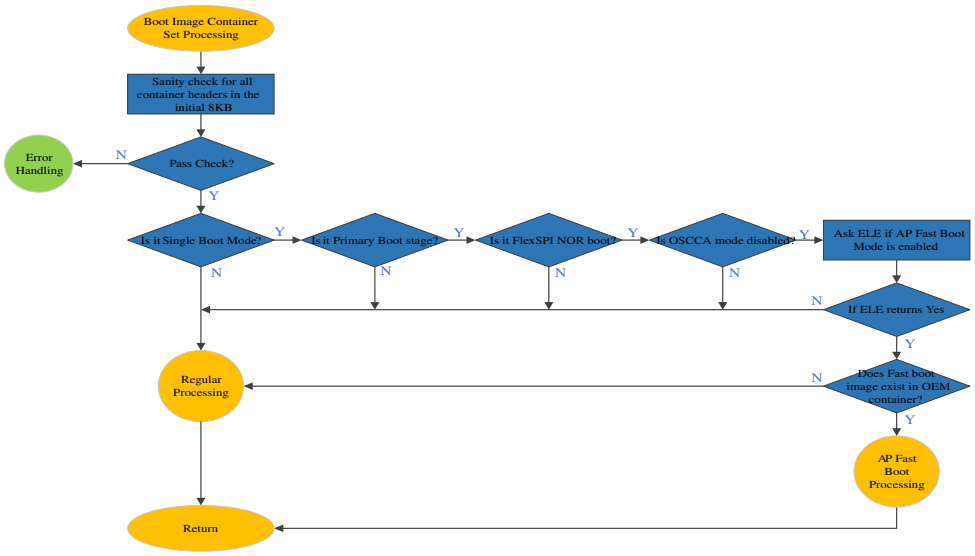


Figure 19. Boot Image Container Sets Selection

9.6.4.1 Regular Processing

- System ROM code loads the headers of the containers in boot image set (the headers are placed in the first 8KB space of the boot image set), the headers are analysed.

- If NXP container exists in boot image container set, system ROM code needs to handle it prior to OEM container. System ROM code loads the NXP container (both header and EdgeLock Enclave FW) and request EdgeLock Enclave to authenticate NXP container via issuing “AUTH_NXP_CTNR” command to EdgeLock Enclave. If EdgeLock Enclave replies failure, break ROM boot flow, update boot stage to next stage then reset system.
- OEM container is must-have, if it doesn’t exist in the boot image container set, system ROM code will break ROM boot flow, update boot stage to next stage then reset system.
- For OEM container, system ROM code requests EdgeLock Enclave to authenticate its container header via issuing “AUTH_OEM_CTNR” command to EdgeLock Enclave, If EdgeLock Enclave replies failure, breaks ROM boot flow, the boot stage is updated to next stage and then the system is reset.
- In the regular boot flow, if boot source is FlexSPI NOR device and FCB is available, the FCB check processing will be executed prior to other OEM images; if FCB check fails, break ROM boot flow, update boot stage to next stage then reset system.
- ROM code will address OEM images one by one based the order in the OEM container (except FCB region copy image since it has been addressed yet), include loading them into target and request EdgeLock Enclave to verify via issuing “Image Verify” command to EdgeLock Enclave. If any failure, break ROM boot flow, update boot stage to next stage then reset system. In single boot mode, if the image is CM33 FW, A55 ROM code needs to requests EdgeLock Enclave to release CM33 TROUT then starts CM33 core after this image pass verify.

9.6.4.2 AP Fast Boot Processing

The performance improvement to meet AP Fast Boot requirement:

1. A55 core hashes the “fast boot image” instead of EdgeLock secure enclave.
2. The hash number calculated by A55 core is passed to EdgeLock secure enclave with command “FAST_VERIFY”, EdgeLock secure enclave takes responsibility to check if the signature is damaged.
3. A new command “Generic Auth” is added in EdgeLock secure enclave ROM, which requests EdgeLock secure enclave to authenticate both NXP container and OEM container header. Correspondingly, A55 ROM code is requested to load both NXP container (includes EdgeLock secure enclave FW) and OEM container header into target before issuing this comman. The information of NXP container and OEM container header should be passed with this command. EdgeLock secure enclave needs to reply twice for this command, the 1st reply means it received the request and starts the processing, and the 2nd reply is to return the result of NXP/OEM container authentication: pass or not pass.

The AP Fast Boot Flow in Single/A55 ROM boot:

1. The AP Fast boot flow will not be executed unless all of the following conditions are met:
    a. The Boot Type is single boot mode (CA55 ROM boot);
    b. The boot stage is primary boot;
    c. The boot source is FlexSPI NOR;
    d. OSCCA mode is disabled;
    e. The AP fast boot mode is enabled: the fuse FASTBOOT_ENABLE[1:0] = 2’b01;
    f. The AP fast boot image exists in OEM container. Although does limit the AP fast boot image must be CM33 FW, normally it should be CM33 image in typical case, and should be running with XIP mode since TCM room is not big enough to place a 3MB image.
2. A55 ROM code loads both NXP container (includes EdgeLock secure enclave FW) and OEM container header, then issues “Generic Auth” command to EdgeLock secure enclave. If EdgeLock secure enclave returns failure, breaks ROM boot flow, the boot stage needs to be updated to next stage. The system must be reset then. If EdgeLock secure enclave returns success, it just means EdgeLock secure enclave received the request and starts to execute Generic Auth processing.
3. A55 ROM code finds out the AP fast boot image, and calculated its hash number;

4. After hash is done, A55 ROM code receives reply from EdgeLock secure enclave to check if both NXP container and OEM container pass authentication (the authentication was requested via “Generic Auth” command). If not pass, update boot stage to next stage then reset system;
5. If pass “Generic Auth” check, if boot source is FlexSPI NOR device and FCB is available, the FCB check processing will be executed prior to other OEM images; if FCB check fails, break ROM boot flow, update boot stage to next stage then reset system. Please refer to the “FCB Check” section for the details;
6. A55 ROM code issues “Fast Verify” command to EdgeLock secure enclave , the hash number of the AP fast boot image will be passed via this command as well, to request EdgeLock secure enclave to check the signature of the AP fast boot image. If not pass check, break ROM boot flow, update boot stage to next stage then reset system;
7. Typically, the AP fast boot image should be CM33 FW, if yes, A55 ROM code needs to requests EdgeLock secure enclave to release CM33 TROUT then starts CM33 core after this image pass verify;
8. A55 ROM addresses other OEM images except the image with fast boot flag set and FCB region copy image, the flow is same with the regular processing.

9.6.4.2.1 FCB Check

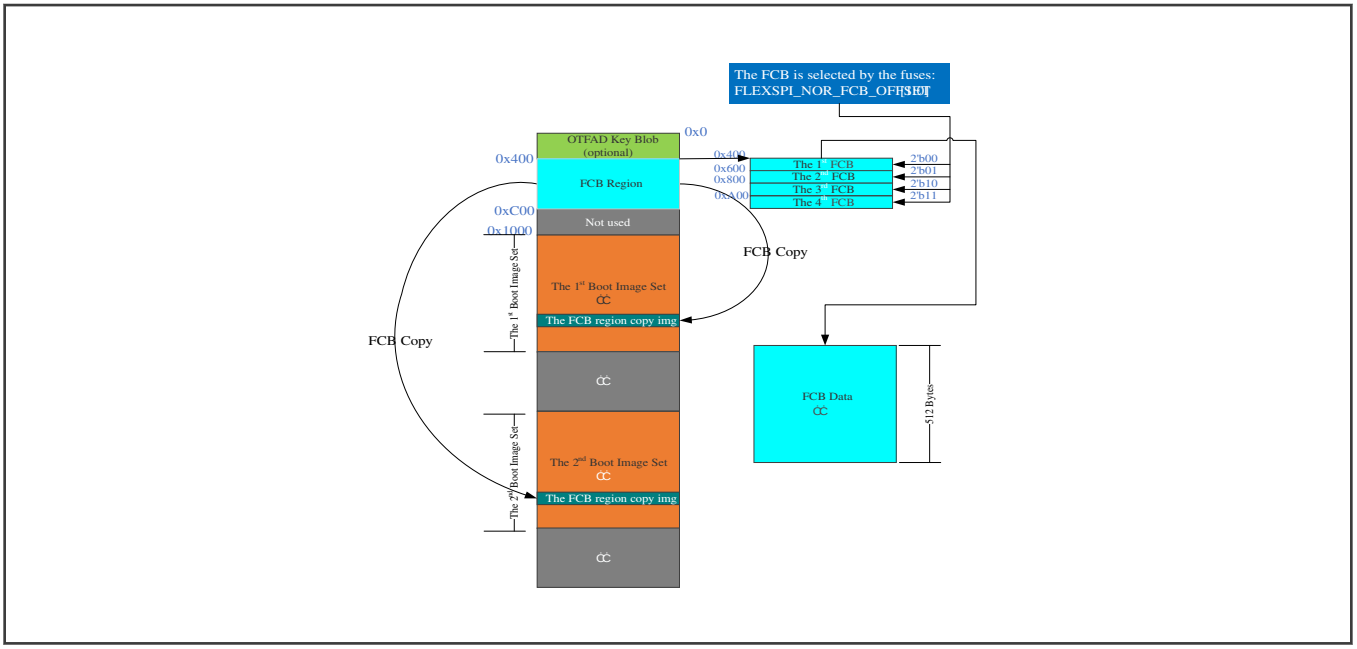
FCB Check processing is to protect FCB content from unexpected modifications because it may introduce the secure issue. For example, if an attacker modifies the FCB on device, then the corrupted FCB data could make the timing marginal, such that the images may pass authentication, but then when we try to execute the code with XIP mode, might not read the correct value from device.

The change request to boot image set:

Needs to build an extra image, names it as FCB region copy image, this image should be placed into OEM container. The content of the FCB region copy image should be exactly same with the content of FCB region at 0x400 on FlexSPI NOR device. Please refer to the “FCB Region” section for the details of FCB region. This FCB region copy image will be packaged into OEM container, and similar with other OEM images, the FCB region copy image should be signed with OEM SRK.

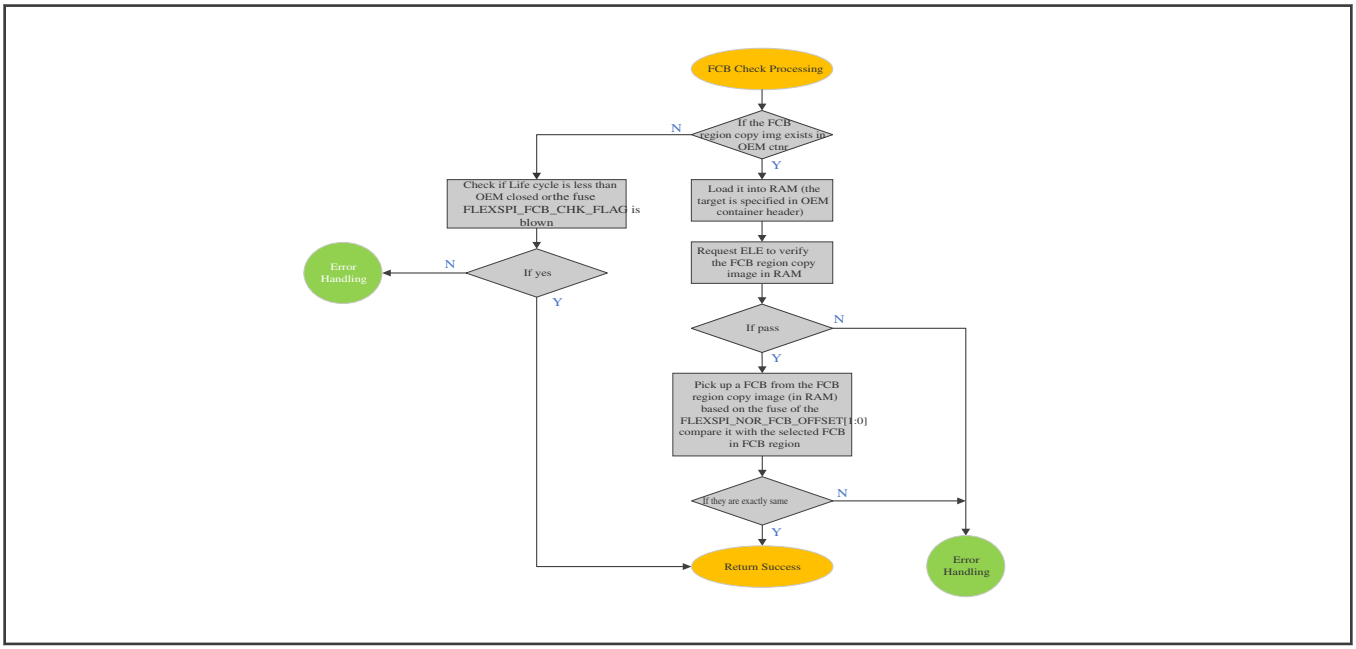
NOTE

The dst of this image must be located in RAM free space (TCM for LP boot and OCRAM for Single boot).



9.6.4.2.2 FCB Check Flow

1. Based on the value of the fuses FLEXSPI_NOR_FCB_OFFSET[1:0], ROM code picks up a FCB from FCB region, then retrieve the configuration parameters from this FCB, and use them to initialize FlexSPI NOR device;
2. Address NXP container;
3. Address OEM container:
    a. Request EdgeLock secure enclave to authenticate OEM container header;
    b. Checks if the FCB region copy image exists in OEM container. If yes, load it to RAM (the dst is specified in OEM container header), then request EdgeLock secure enclave to verify it;
    c. If the FCB region copy image pass the image verify, picks up the FCB from the FCB region copy image based on the value of the fuses FLEXSPI_NOR_FCB_OFFSET[1:0], the compare the content of this FCB with the FCB picked up in step #1. If they are exactly same, continue ROM boot flow; if not, break ROM boot flow, move boot stage then reset system.



The table below shows whether the FCB region copy image is must-have or optional.

Fuse FLEXSPI_FCB_CHK_FLAG	Life Cycle	FCB region copy image
X	Less than OEM Closed	Optional
0	Equal or bigger than OEM Closed	Must-have
1		Optional

NOTE

Once the FCB region copy image exists in OEM container, the FCB check is mandatory and the result of FCB check impacts the ROM boot, means the ROM boot flow will be broken in case any failure in FCB check.

FCB check is needed or not:

1. If the FCB region copy image exists in OEM container, the FCB check processing must be executed, and the result will impact the ROM boot processing; (FCB check is needed)
2. If the FCB region copy image doesn't exists in OEM container,

a. If life cycle is less than OEM closed, it doesn't impact ROM boot; (avoid FCB check)

b. If life cycle is >= OEM closed:

i. In case FLEXSPI_FCB_CHK_FLAG is blown (=1), it doesn't impact ROM boot; (avoid FCB check)

ii. In case FLEXSPI_FCB_CHK_FLAG is not blown (=0), ROM boot flow will be broken and move to next boot stage; (FCB check is needed)

9.6.5 Single Boot Flow (Cortex-A55)

In single boot, up to 2 containers (NXP and OEM containers) are expected to be handled by A55 ROM code.

- NXP container (optional):
  - EdgeLock secure enclave FW only (must-have in case NXP container present)
- OEM container (mandatory), contains:
  - CM33 FW (optional)
  - CA55 FW (must-have)
  - FCB Region Copy Image (optional)

When the CM33 FW is present in the OEM container, A55 ROM loads it, and requests EdgeLock secure enclave to verify image. It then writes the entry point address of the CM33 FW into the SRC GPR19 register then kicks of CM33 core, and continues A55 ROM boot processing. In this case, after CM33 ROM code starts, it is expected to retrieve the entry point from that SRC GPR19 register then directly jumps there.

1. For OTFAD case, A55 ROM needs to initialize FlexSPI controller and device with the safe settings, so that EdgeLock secure enclave can access key blob from FlexSPI NOR device. A55 ROM doesn’t care whether the OTFAD enabled fuse blown, EdgeLock secure enclave needs to take care of this fuse. A55 will always send the message to EdgeLock secure enclave to enable the OTFAD and check the status from EdgeLock secure enclave. From system ROM perspective, the expected behaviors of EdgeLock secure enclave are: IF (OTFAD Enabled fuse set) THEN EdgeLock secure enclave enable the OTFAD and send status to A55 ROM. ELSE EdgeLock secure enclave does not enable OTFAD and always send success to A55 ROM.
2. Before loading EdgeLock secure enclave FW, A55 ROM needs to check the corresponding bits in the GPR register. If the bit set, means the EdgeLock secure enclave FW is in running, A55 ROM skips to load this FW, move to load next image.
3. There is a fuse indicating if EdgeLock secure enclave FW is necessary. System ROMs (both M33 and A55 ROMs) don’t check this fuse. In case the EdgeLock secure enclave/NXP container exists in the boot image set, M33/A55 always load the EdgeLock secure enclave/NXP container and the EdgeLock secure enclave FW, then request EdgeLock secure enclave to authenticate the EdgeLock secure enclave container. EdgeLock secure enclave will decide whether to authenticate it or not. In the MUST NOT NEED EdgeLock secure enclave FW scenario, if M33/A55 ROM find a EdgeLock secure enclave FW container, then send the message to EdgeLock secure enclave. EdgeLock secure enclave should bypass this request and reply success to M33/A55 ROM.

9.6.5.1 Kick Off Cortex-M33 Flow

When the Cortex-A55 finds the Cortex-M33 image in the container.

- The Cortex-A55 ROM sets the SCR bit to release the Cortex-M33 reset to let the TCM accessible.
- After the Cortex-A55 loads the Cortex-M33 image, the Cortex-M33 image is verified.
- The Cortex-A55 ROM sends a request to EdgeLock secure enclave to release the Cortex-M33 TROUT (See BLK_CTRL_S_AONMIX.
- The Cortex-A55 ROM releases the Cortex-M33 CPUWAIT signal. See BLK_CTRL_S_AONMIX[M33_CFG] register.

9.6.5.2 AARCH32 A55 ROM

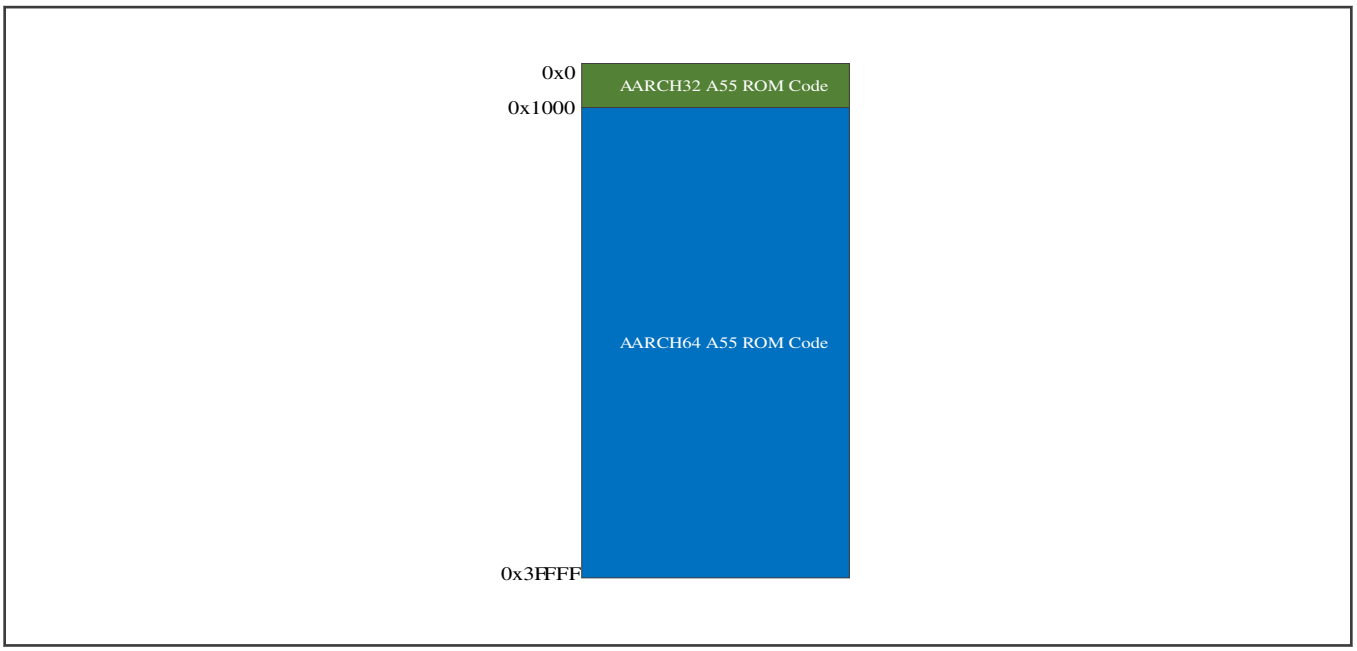
There are two pieces of A55 ROM code:

- AARCH64 A55 ROM code
- AARCH32 A55 ROM code

The purpose to have a AARCH32 CA5 ROM code is to meet the A55 resume usage in AARCH32 mode.

In AARCH64 mode, the start address of CA55 core is decided by RVBAR setting. The reset value of RVBAR is point to 0x1000, means the AARCH64 CA55 ROM code will be executed after reset. And SW is able to update the start address of CA55 core via updating the RVBAR register, for example, place a resume SW in OCRAM space before suspend system, and update the RVBAR register with the entry point of the resume SW in OCRAM, then system is resume, CA55 core starts from OCRAM that the address is specified by RVBAR.

But in AARCH32 mode, CA55 core must starts at 0x0 address, so is not able to update the start address of CA55 core in AARCH32 mode via updating RVBAR. So has to have a AARCH32 CA55 ROM code and place it at 0 address. In AARCH32 mode, after system resume, CA55 core starts from 0 address, it's AARCH32 CA55 ROM code, this AARCH32 CA55 ROM code will retrieve the entry point from RVBAR register, if the entry point is valid, then jumps to it. The entry point reading from RVBAR must be within OCRAM or TCM or DRAM or FlexSPI NOR memory-mapped space, otherwise the entry point will be regarded as invalid, then re-run from 0 address.



9.6.6 Low Power Boot Flow (Cortex-M33)

In LP boot, up to three containers (NXP and OEM containers) are expected to be handled by CM33 ROM code.

- NXP container (optional): EdgeLock Enclave FW only (must-have in case NXP container present)
- OEM container (mandatory), contains:
  - CM33 FW (must-have)
  - FCB Region Copy Image (optional)

When the Cortex-M33 reboots (such as when waking from a low power mode in which Cortex-M33 was off, or if the Cortex-M33 is warm reset without resetting the entire rest of the SoC) in Low Power Boot mode, the Cortex-M33 Boot ROM validates the resume entry point is valid before jumping to the address pointed to by the BLK_CTRL_S_AONMIX INITSVTOR[31:7] and/or INITNSVTOR[31:7] registers.

1. For OTFAD case, M33 ROM needs to initialize FlexSPI controller and device with the safe settings, so that EdgeLock secure enclave can access key blob from FlexSPI NOR device. M33 ROM doesn't care whether the OTFAD enabled fuse blown, EdgeLock secure enclave needs to take care of this fuse. M33 will always send the message to EdgeLock secure

enclave to enable the OTFAD and check the status from EdgeLock secure enclave. From system ROM perspective, the expected behaviors of EdgeLock secure enclave are:

IF (OTFAD Enabled fuse set) THEN EdgeLock secure enclave enable the OTFAD and send status to M33 ROM.

ELSE

EdgeLock secure enclave does not enable OTFAD and always send success to M33 ROM.

2. There is a fuse indicating if EdgeLock secure enclave FW is necessary. System ROMs (both M33 and A55 ROMs) don't check this fuse. In case the EdgeLock secure enclave /NXP container exists in the boot image set, M33/A55 always load the EdgeLock secure enclave /NXP container and the EdgeLock secure enclave FW, then request EdgeLock secure enclave to authenticate the EdgeLock secure enclave container. EdgeLock secure enclave will decide whether to authenticate it or not.
3. In the MUST NOT NEED EdgeLock secure enclave FW scenario, if M33/A55 ROM find a EdgeLock secure enclave FW container, then send the message to EdgeLock secure enclave. EdgeLock secure enclave should bypass this request and reply success to M33/A55 ROM.
4. In LP boot mode, CM33 firstly starts to run while A55 core isn't kicked off by EdgeLock secure enclave during CM33 ROM boot time.

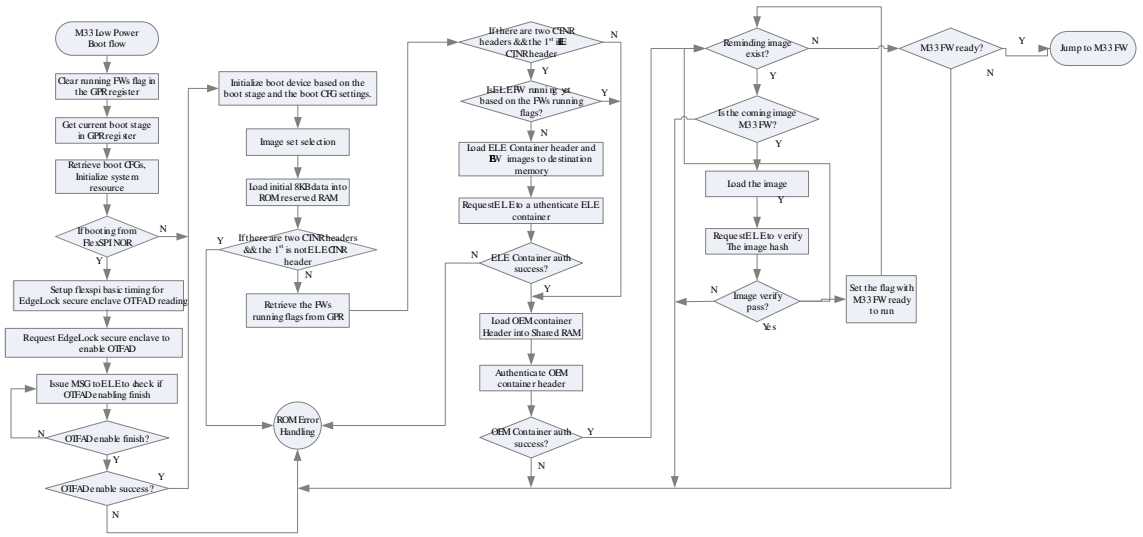
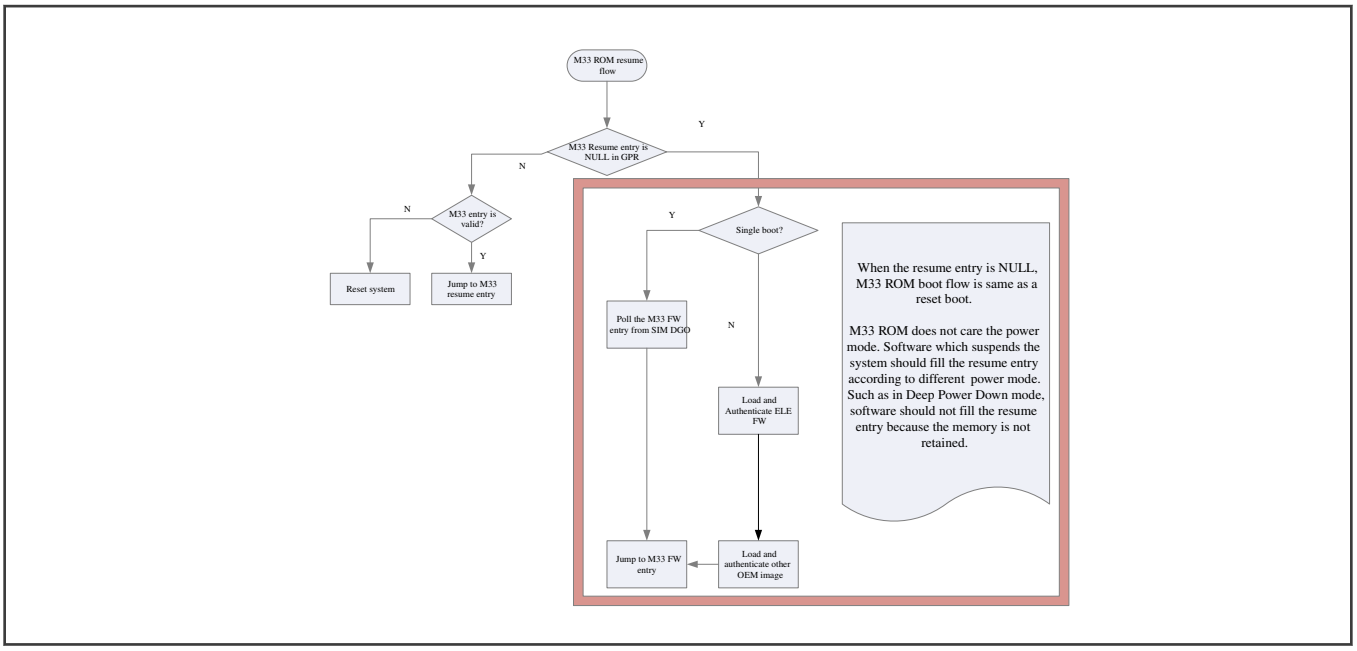


Figure 20. Cortex-M33 ROM Sequence for LP Boot

9.6.6.1 M33 ROM Resume Flow



9.6.7 Failure Handling in ROM

There are two categories of failures: Fatal and non-Fatal.

Fatal:

- Attack been detected by HW or Cortex-M33/Cortex-A55 ROM itself (stack cookie, SW secure counter).
- Hard Fault in ROM code.

Non-Fatal:

- Boot device initialization failed.
- Boot device read failure (ECC error, timeout, etc.)
- image failure.

For Fatal failures, ROM trigger a WDOG reset.

For non-fatal failures, ROM sets next boot stage, and then trigger a WDOG reset. After the reset, ROM will try to boot from the secondary image container set. If this fails, it moves to recovery boot. If recovery boot also fails, ROM will move to USB SDP boot. If Booting from the primary boot image set fails, ROM updates the boot stage information stored in the SRC GPR16 register to move to next boot stage, then resets the system. In next boot cycle after system is out of reset, ROM moves to the next boot stage and tries again. The sequence of boot stage moving is Primary image set boot => Secondary Image Set boot => Recovery boot(If enable in fuse) => USB serial download boot;

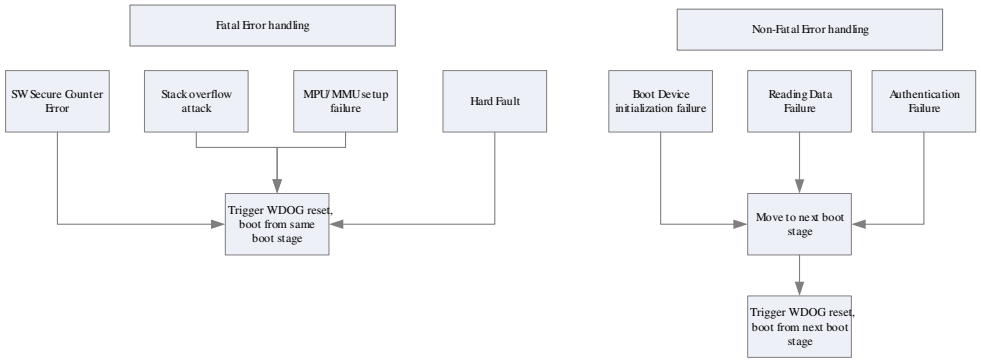


Figure 21. Failure Handling

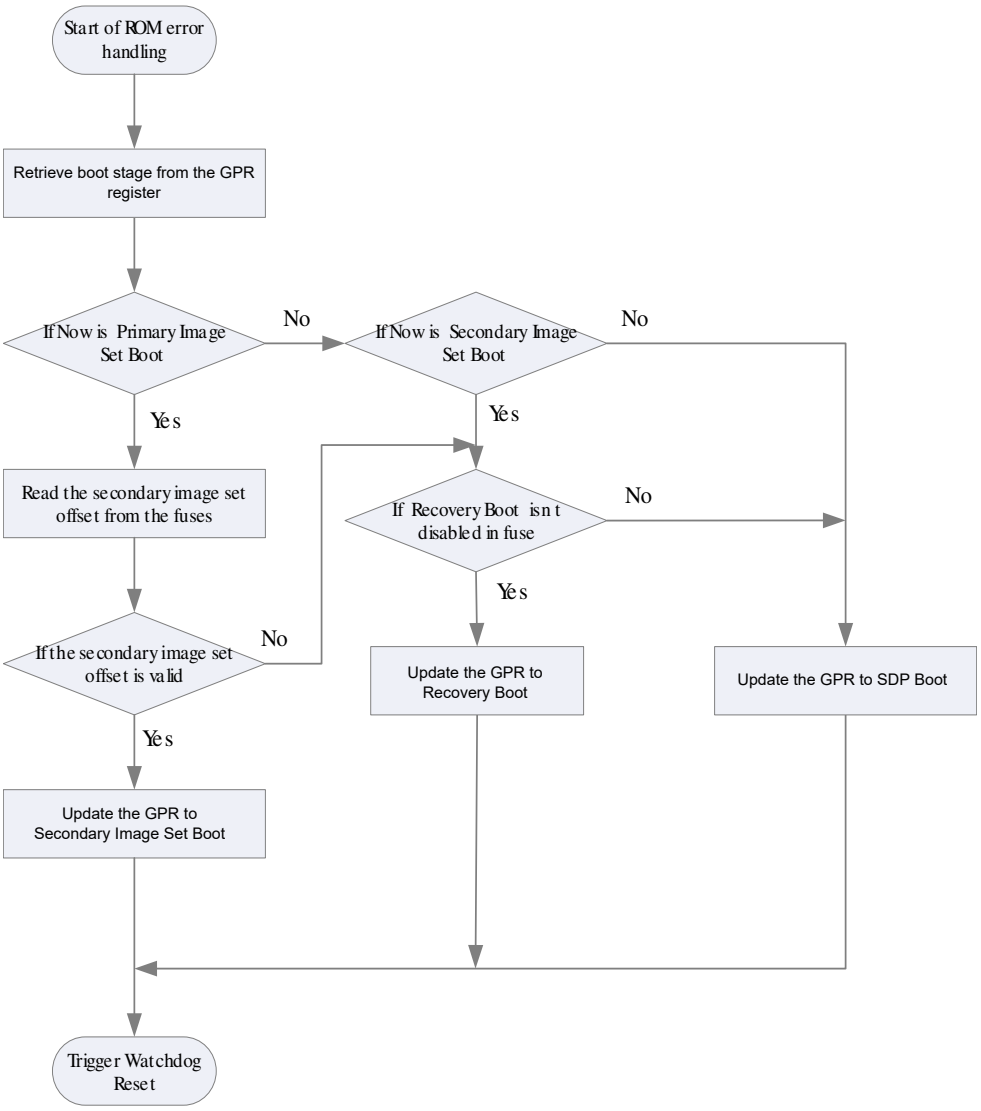


Figure 22. ROM Error Handling

9.7 Platform Initialization In ROM

This section describes the details of the ROM and provides the initialization details. This includes details on:

- ROM memory usage map
- On-chip blocks that the ROM must enable.
- Clock initialization
- Watchdog initialization
- Timer initialization
- Exception handling and interrupt handling
- Failure Handling

9.7.1 Memory Usage

9.7.1.1 Cortex-M33 Memory Usage Map

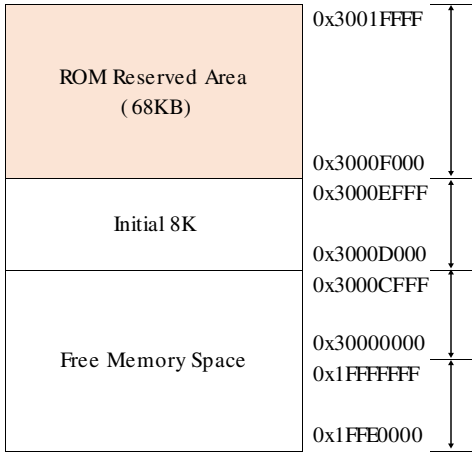


Figure 23. TCM Usage in Cortex-M33 ROM Code (Secure View)

The initial 8KB is used to store the initial image and can be cleared by post boot firmware.

9.7.1.2 Cortex-A55 Memory Usage Map

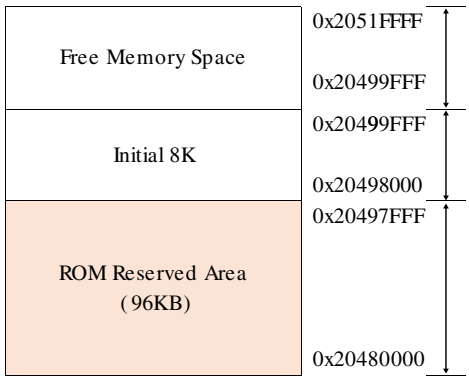


Figure 24. OCRAM Usage in Cortex-A55 ROM Code (Non-secure View)

The initial 8KB is used to store the initial image and can be cleared by post boot firmware.

For single boot, where loads image to Cortex-M33, there is no reserved area on Cortex-M33 TCM.

9.7.2 System HW Settings in ROMs

Table 50. ROM Hardware Usage Overview

Cortex-M33 ROM	Cortex-A55 ROM
EdgeLock secure enclave AHB	EdgeLock secure enclave AHB
CCM/System PLL	CCM/System PLL/ARM PLL
WDOG1	WDOG3
FlexSPI1	FlexSPI1
IOMUXC	IOMUXC
LPSPI1, LPSPI2	LPSPI3, LPSPI4
Cortex-M33 ICache	Cortex-A55 ICache
USDHC1, USDHC2	USDHC1, USDHC2
USB1	USB1
USB1 PHY	USB1 PHY
TCM	OCRAM

9.7.3 Peripheral Clock Enable

The ROM code enables the clocks for the peripherals listed in the following table.

Table 51. Peripheral Clock Enable

Peripheral	Status after ROM exits
WDOG1/3	Enabled
FLEXSPI1	Enabled if FLEXSPI1 NOR/NAND boot
USDHC1/2	Enabled if USDHC1/2 boot
LPSPI1/2/3/4	Enabled if LPSPI1/2/3/4 recovery boot
USB_OTG1	Enabled if SDP boot

9.7.4 Watchdog

By default, the Watchdog (WDOG) hardware is disabled. Unless the fuse WDOGn_ENABLE is blown, ROM will not enable the watchdog. The WDOG timeout can be selected by the fuse 'WDOG Timeout Select'. The fuse setting details are provided below.

'WDOG Timeout Select' configuration options:

- 00 – 1s (Depends on the WDOG reset number)
- 01 – 0.5s

- 10 – 0.25s
- 11 – 0.125s

WDOG1_ENABLE and WDOG3_ENABLE configuration options:

- 0 – Disabled
- 1 – Enabled”

The Cortex-M33 ROM only uses WDOG1 and the Cortex-A55 only uses WDOG3.

9.7.5 Timer

After the necessary internal initialization tasks have been completed, ROM will enable the timer. Once the timer has been enabled, it is in free running mode and ROM will not disable it again.

The Cortex-M33 ROM uses the System Counter, and the Coretx-A55 ROM uses the Generic Timer.

9.7.6 ECC support

There is a HW ECC initialization feature in MECC256. ROM will take advantage of this. This HW engine zeroize all area of OCRAM. ROM polling the register bit until the initialize finishes.

9.7.7 Interrupt and Exception Handling

All interrupts are masked by both the Cortex-M33 and Cortex-A55 ROM.

There are three exception handles in the Cortex-M33 ROM to handle various exceptions:

1. SVC
   - ROM SVC handler
2. NMI
   - Reconfigures NMI_b then return
3. Others
   - Triggers a watchdog reset.

There are two exception handles in the Cortex-A55 ROM to handle various exceptions:

1. SVC
2. Others
   - Trigger a watchdog reset

9.7.8 SRC GPR Registers Usage in System ROM Code

Table 52. SRC GPR Registers Usage in System ROM

GPR16	ROM flags	Bit0: EdgeLock secure enclave FW Loaded flag Bit1: CM33 FW Loaded flag Bit2: CA55 FW Loaded flag Running flag: 0 – The FW is not loaded 1 – The FW is loaded

Table continues on the next page...

Table 52. SRC GPR Registers Usage in System ROM (continued)

		Bit8: SDP on Demand 0 – USB SDP is not on-demand 1 – USB SDP mode is on-demand Bit[27:24]: CM33 ROM Boot Stage Bit[31:28]: CA55 ROM Boot Stage
GPR17	The argument of the CM33 resume FW	Provided by A55 image in Single Boot mode
GPR19	The address of the CM33 resume FW	Provided by A55 image in Single Boot mode. This is optional and can be used a s other purpose if argument is not needed when resuming

9.8 Boot Devices

9.8.1 FlexSPI

The ROM supports auto probe (SFDP (JESD-216) discoverable parameters) feature.

9.8.1.1 IOMUX configuration for FlexSPI devices

The following table shows the FlexSPI IOMUX pin configuration.

Table 53. FlexSPI IOMUX Pin Configuration

Signal Name	PAD Name
FlexSPI_A_DATA0	SD3_DATA0 (ALT1)
FlexSPI_A_DATA1	SD3_DATA1 (ALT1)
FlexSPI_A_DATA2	SD3_DATA2 (ALT1)
FlexSPI_A_DATA3	SD3_DATA3 (ALT1)
FlexSPI_A_DQS	SD1_STROBE (ALT1)
FlexSPI_A_SS0_B	SD3_CMD (ALT1)
FlexSPI_A_SS1_B	SD1_DATA3 (ALT1)
FlexSPI_A_SCLK	SD3_CLK (ALT1)
FlexSPI_A_DATA4	SD1_DATA4 (ALT1)
FlexSPI_A_DATA5	SD1_DATA5 (ALT1)
FlexSPI_A_DATA6	SD1_DATA6 (ALT1)
FlexSPI_A_DATA7	SD1_DATA7 (ALT1)

9.8.1.2 FlexSPI eFUSE Configuration

The boot ROM determines the configuration of external Serial NAND/NOR flash by parameters, either provided by eFUSE, or sampled on GPIO pins during boot. See below table for parameters details.

Table 54. FlexSPI Boot eFUSE Configuration

Fuse	Definition	Settings
BOOT_MODE[3:0]	Boot device selection	Coretx-A55 (Single boot): • 0100 - FlexSPI Serial NOR • 0101 - FlexSPI Serial NAND 2K page Coretx-M33 (LP boot): • 1100 - FlexSPI Serial NOR • 1101 - FlexSPI Serial NAND 2K page
BOOT_CFG0[4]	Override_FlexSPI_NOR_Boot_Selection	0 - Do not override 1 - Override
BOOT_CFG0[6:5]	FlexSPI_NOR_Boot_Override_Selection	00 - Flex SPI (Hyperflash 1.8V) 01 - Flex SPI (Flash with 4B READ(0x13)default supported) 10 - Default Octal mode (Micron) 11 - Default Octal mode (Mxic)
BOOT_CFG0[7]	FlexSPI_NAND_4K_Page_Selection	0 - Not override the default seleciton (2K page) 1 - Override, 4K page is selected
BOOT_CFG2[1:0]	FlexSPI NAND Column Address Width	00 - 12 01 - 13 10 - 14 11 - 15
BOOT_CFG2[3:2]	FlexSPI NAND CS Interval	00 - 100 ns 01 - 200 ns 10 - 400 ns 11 - 50 ns
BOOT_CFG2[5:4]	FlexSPI NAND Search Count	00 - 2 01 - 2 10 - 4 11 - 8
BOOT_CFG2[7:6]	FlexSPI NAND Search Stride	00 - 64 pages 01 - 128 pages 10 - 256 pages 11 - 32 pages

Table continues on the next page...

Table 54. FlexSPI Boot eFUSE Configuration (continued)

Fuse	Definition	Settings
BOOT_CFG2[15:8]	FlexSPI DLL Value	The delay line value only for HyperFlash 00000000 - 11111111
BOOT_CFG2[23]	FlexSPI NAND Busy Bit Offset Override	0 - Do not override 1 - Override

9.8.1.3 FlexSPI Initialization Flow

A reset pin can be selected via the “FlexSPI NOR Reset Pin Port selection” fuses and “FlexSPI NOR Reset Pin selection” fuses. If a valid number has been programmed in these fuses then ROM code will toggle the selected GPIO pin to execute a cold reset to FlexSPI NOR device. The selected GPIO pin should be able to power down the FlexSPI NOR device on low polarity and power up the device on high polarity.

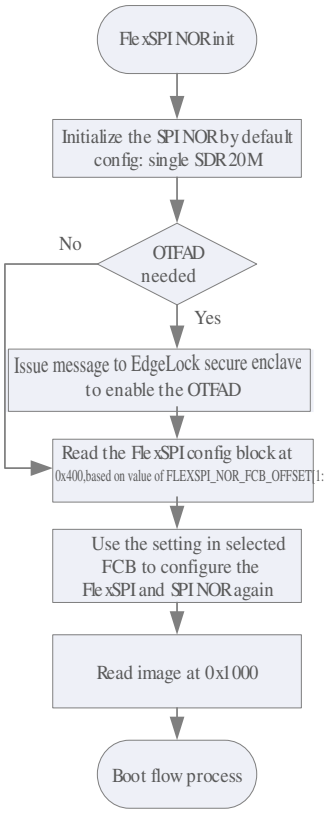


Figure 25. FlexSPI Initialization Flow

9.8.1.4 FlexSPI NOR Memory Map

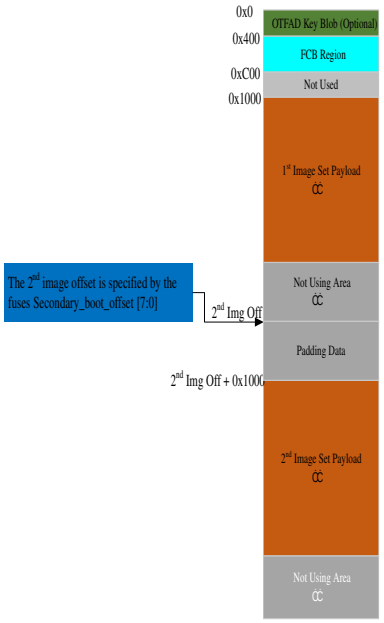


Figure 26. FlexSPI NOR Memory Map

9.8.1.5 FlexSPI NOR FCB region

The FCB region is located at offset 0x400 on FlexSPI NOR device, the size of FCB region is 2KB, so it can have up to 4 FCBs (FlexSPI NOR Configuration Block).

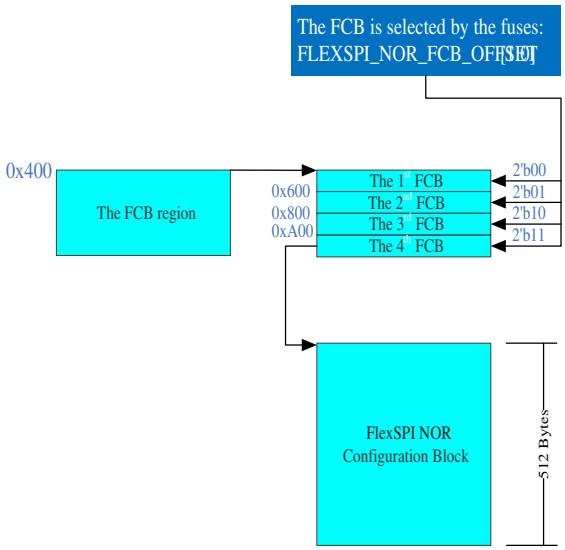
The change request to boot image set:

Two fuses FLEXSPI_NOR_FCB_OFFSET[1:0] are used to select the FCB which will be picked up ROM code to initialize FlexSPI NOR device:

NOTE

the dst of this image must be located in RAM free space (TCM for LP boot and OCRAM for Single boot).

FLEXSPI_NOR_CONFIG_BLOCK_OFFSET[1:0]	The selected FCB
2'b00	The 1st FCB (offset 0x400)
2'b01	The 2nd FCB (offset 0x600)
2'b10	The 3rd FCB (offset 0x800)
2'b11	The 4th FCB (offset 0xA00)



Name	Offset	Size in bytes	Description
Tag	0x000	4	0x42464346, ascii:" FCFB"
Version	0x004	4	0x56010000
			[07:00] bugfix = 0
			[15:08] minor = 0
			[23:16] major = 1
			[31:24] ascii 'V'
-	0x008	4	Reserved to 0
readSampleClkSrc	0x00c	1	0 - internal loopback
			1 - loopback from DQS pad
			2 - loopback from SCK pad
			3 - Flash provided DQS
dataHoldTime	0x00d	1	Serial Flash CS Hold Time
			Recommend default value is 0x03
dataSetupTime	0x00e	1	Serial Flash CS setup time
			Recommended default value is 0x03
columnAdressWidth	0x00f	1	3 - For HyperFlash
			0 - Other devices
deviceModeCfgEnable	0x010	1	Device Mode Configuration Enable feature

Table continues on the next page...

Table continued from the previous page...

			0 - Disabled
			1 - Enabled
-	0x011	3	Reserved to 0
deviceModeSeq	0x014	4	Sequence parameter for device mode configuration **0 seqNum Sequence count** **1 seqId Sequence starting index** **2-3 Reserved**
deviceModeArg	0x018	4	Device Mode argument, effective only when deviceModeCfgEnable = 1
configCmdEnable	0x01c	1	Config Command Enable feature
			0 - Disabled
			1 - Enabled
-	0x01d	3	Reserved to 0

Name	Offset	Size in bytes	Description
configCmdSeqs	0x020	16	Sequences for Config Command, allow 4 separate configuration command sequences.
			The structure of each entry is the same with devceModeSeq
cfgCmdArgs	0x030	16	Arguments for each separate configuration command sequence.
controllerMiscOption	0x040	4	0Differential clock enable 1. - Differential clock is not supported 2. 1- Differential clockis supported 1 CK2 Enable Must set to 0 for this silicon 2Parallel Mode Enable Must set to 0 for this silicon 3 Word Addressable Enable

Table continues on the next page...

Table continued from the previous page...

			1. - Device is not word addressable 2. 1- Device is word addressable 4 Safe Configuration Frequency Enable 1. - Configure external device using specifiedfrequency 2. 1- Configure external device using30MHz 5 Reserved 6 DDR Mode Enable 1. - External device works using SDRcommands 2. - External device works using DDRcommands
deviceType	0x044	1	1 - Serial NOR
sflashPadType	0x045	1	1 - Single pad
			2 - Dual pads
			4 - Quad pads
			8 - Octal pads
serialClkFreq	0x046	1	Chip specific value, for this silicon
			1 - 20MHz
			2 - 50MHz
			3 - 62MHz for SDR and 67MHz for DDR

Name	Offset	Size in bytes	Description
			4 - 80MHz
			5 - 100MHz
			6 - 133MHz
			7 - 166MHz for SDR and 200MHz for DDR
			Other values: 20MHz

Table continues on the next page...

Table continued from the previous page...

lutCustomSeqEnable	0x047	1	0 - Use pre-defined LUT sequence index and number
			1 - Use LUT sequence parameters provided in this block
-	0x048	8	Reserved
sflashA1Size	0x050	4	For SPI NOR, need to fill with actual size, in terms of bytes
sflashA2Size	0x054	4	The same as above
sflashB1Size	0x058	4	The same as above
sflashB2Size	0x05c	4	The same as above
csPadSettingOverride	0x060	4	Set to 0 if it is not supported
sclkPadSettingOverride	0x064	4	Set to 0 if it is not supported
dataPadSettingOverride	0x068	4	Set to 0 if it is not supported
dqsPadSettingOverride	0x06c	4	Set to 0 if it is not supported
timeoutInMs	0x070	4	Maximum wait time during read busy status
			Not Used in ROM
commandInterval	0x074	4	Interval of CS deselected period.
			Set to 0.
dataValidTime	0x078	4	Time from clock edge to data valid edge, unit ns.
			This field is used when the FlexSPI Root clock is less than 100MHz and the read sample clock source is device provided DQS signal without CK2 support.
			[31:16] data valid time for DLLB in terms of 0.1ns
			[15:0] data valid time for DLLA in terms of 0.1ns
busyOffset	0x07c	2	busy bit offset, valid range :0-31

Table continues on the next page...

Table continued from the previous page...

busyBitPolarity	0x07e	2	0 – busy bit is 1 if device is busy
			1 – busy bit is 0 if device is busy
lookupTable	0x080	256	Lookup table
lutCustomSeq	0x180	48	Customized LUT sequence, see below table for details.
-	0x1b0	16	Reserved for future use
pageSize	0x1c0	4	Page size of Serial NOR Flash.
			Not used in ROM
sectorSize	0x1c4	4	Sector size of Serial NOR Flash.

9.8.1.6 FlexSPI NAND

A typical FlexSPI NAND memory map contains two copies of Firmware Config Blocks (FCBs), two copies of discovered bad-block tables (DBBTs) and two image sets. Two image sets share same FCB and DBBT. The first FCB located at the start of the SPI NAND Flash, which is block0, page0. The second FCB located at (0 + page_index * FCB0->searchStride). Using FCB data structure, the DBBT0 and DBBT1 can be found. Normally, ROM use the first FCB and DBBT. If the blocks for FCB and DBBT are corrupted, ROM will search for the next FCB and DBBT using searchCount.

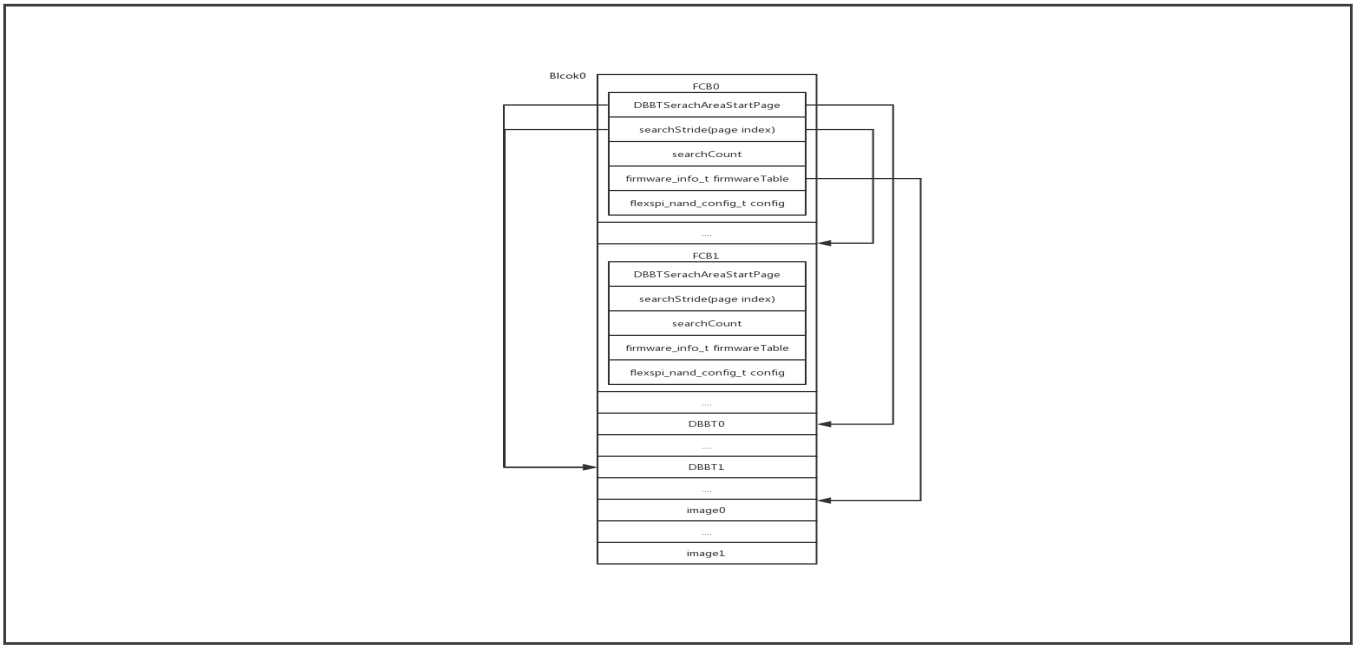


Table 55. FlexSPI NAND Configuration

Name	Offset	Size in bytes	Descriptions
flexspi_mem_config_t memConfig	0x0	448	Generic Memory configuration block over FlexSPI
pageDataSize	0x1c0	4	Data Size in one page, usually it is 2048 or 4096
pageTotalSize	0x1c4	4	Total size in one page, usually, it equals 2 ^ width of column address
pagesPerBlock	0x1c8	4	Pages per block
bypassReadStatus	0x1cc	1	Not used in ROM
bypassEccRead	0x1cd	1	Not used in ROM
hasMultiPlanes	0X1ce	1	Has multiple planes
skipOddBlocks	0x1cf	1	Skip odd blocks
eccCheckCustomEnable	0x1d0	1	Determine if use the customized ECC check related masks
ipCmdSerialClkFreq	0x1d1	1	IP Command Serial Clock Frequency, keep 0
readPageTimeUs	0x1d2	2	Not used in ROM
eccStatusMask	0x1d4	4	ECC status mask
eccFailureMask	0x1d8	4	ECC failure mask
blocksPerDevice	0x1dc	4	Blocks per device, not used in ROM
reserved	0x1e0	32	

Secondary image offset

In the above picture, the image1 offset is decided on the SECONDARY_BOOT_OFFSET fuse bits.

0 = 4MB

1 = 2MB

2 = 1MB

3 = 8MB

4 = 16MB

5 = 32MB

6 = 64MB

7 = 128MB

8 = 256MB

9.8.1.6.1 SPI NAND flash boot flow and Boot Control Blocks(BCB)

There are two BCB data structures:

- FCB
- DDBT

As a part of the SPI NAND media initialization, the ROM driver uses safe SPI NAND timings to search for the Firmware Configuration Block(FCB) that contains the optimum NAND timings, the page address of the Discovered Bad Block Table (DBBT) Search Area, and the start page address of image table. If the FCB is found, the optimum NAND timings are loaded for further reads. If the ECC fails, or the fingerprints do not match, the Block Search state machine increments the page number to the Search Stride number of pages to read for the next BCB until the SearchCount pages have been read. If the search fails to find a valid FCB, the NAND driver responds with an error and the boot ROM enters the serial download mode. The FCB contains the page address of the DBBT Search Area, and the page address for primary and secondary boot images. The DBBT is searched in the DBBT Search Area, just like the FCB is searched. After the FCB is read, the DBBT is loaded, and the primary or secondary boot image is loaded using the starting page address from the FCB.

9.8.1.6.2 Firmware configuration block (FCB) and Discovered Bad Block Table (DBBT)

The FCB is the first sector in the first good block. The FCB must be present at each search stride of the search area. The search area contains copies of the FCB at each stride distance, so, in case the first NAND block becomes corrupted, ROM finds its copy in the next NAND block. The search area must span over at least two NAND blocks. The location information for the DBBT search area, FW1, and FW2 are all specified in the FCB. This table shows the flash control block structure:

Table 56. FLEXSPI NAND FCB

Name	Offset	Size in bytes	Descriptions
reserved	0x0	4	Not used
fingerprint	0x4	4	32-bit word with a value of 0x20424346, in ascii \"FCB\"
version	0x8	4	32-bit version number; this version of FCB is 0x00000001
DBBTSerachAreaStartPage	0xc	4	page index for the first DBBT
searchStride	0x10	2	FCB and DBBT search stride
searchCount	0x12	2	Search time for both FCB and DBBT
firmwareCopies	0x14	4	not used in ROM
reserved	0x18	40	
firmwareTable	0x40	80	Firmware table, each firmware consist two word. - uint32_t startPage - uint32_t pagesInFirmware ROM only use the first firmware to do primary boot. Put this field here is to be compatible with other projects
reserved1	0x80	128	
flexspi_nand_config_t config	0x100	512	Another config structure for flexspi host driver

Table 57. Discovered Bad Block Table

Name	Offset	Size in bytes	Descriptions
reserved	0x0	4	-
fingerprint	0x4	4	32-bit word with a value of 0x44424254,in ascii \"DBBT\"
version	0x8	4	32-bit version number; this version of DBBT is 0x00000001
reserved	0xc	4	
badBlockNumber	0x10	4	Total number of bad blocks
reserved	0x14	12	
badBlockTable	0x20	1024	block number

Bad Block handling in ROM - During the firmware boot, at the block boundary, the Bad Block table is searched for a match to the next block. If no match is found, the next block can be loaded. If a match is found, the block must be skipped and the next block checked.

9.8.2 Expansion Device

The ROM supports booting from the eMMC and SD compliant devices.

9.8.2.1 Clock Settings for SD/eMMC

For USDHC1/2, the clock source is SYS_PLL_PFD1, which is 200MHz.

9.8.2.2 IOMUX Configuration for SD/eMMC

Table 58. SD/eMMC IOMUX pin configuration

Signal Name	USDHC1	USDHC2
CMD	SD1_CMD (ALT0)	SD2_CMD (ALT0)
CLK	SD1_CLK (ALT0)	SD2_CLK (ALT0)
DATA0	SD1_DATA0 (ALT0)	SD2_DATA0 (ALT0)
DATA1	SD1_DATA1 (ALT0)	SD2_DATA1 (ALT0)
DATA2	SD1_DATA2 (ALT0)	SD2_DATA2 (ALT0)
DATA3	SD1_DATA3 (ALT0)	SD2_DATA3 (ALT0)
DATA4	SD1_DATA4 (ALT0)	-
DATA5	SD1_DATA5 (ALT0)	-
DATA6	SD1_DATA6 (ALT0)	-
DATA7	SD1_DATA7 (ALT0)	-
RESET	-	SD2_RESET_B (ALT0)
VSEdgeLock secure enclaveCT	-	SD2_VSEdgeLock secure enclaveCT (ALT0)

9.8.2.3 Expansion Device eFUSE Configuration

SD/eMMC boot can be performed using the USDHC ports, based on setting of the BOOT_MODE.

Table 59. Expansion Device eFUSE Configuration

Fuse	Definition	Settings
BOOT_MODE[3:0]	Boot device selection	Coretx-A55 (Single boot): • 0010 - USDHC1 8-bit eMMC 5.1 • 0011 - USDHC2 4-bit SD3.0 Coretx-M33 (LP boot): • 1010 - USDHC1 8-bit eMMC 5.1 • 1011 - USDHC2 4-bit SD3.0
BOOT_CFG0[8]	uSDHC Device Type	0 - eMMC 1 - SD
BOOT_CFG0[16]	SD Loopback Clock Source Sel (for SDR50 and SDR104 only)	0 - Through SD pad 1 - Direct
BOOT_CFG0[18:17]	SD Speed Selection / eMMC/MMC Bus width	SD Speed Selection • 00 - Normal/SDR12> • 01 - High/SDR25 • 10 - SDR50 • 11 - SDR104 eMMC/MMC Bus width • 00 - 4 bit • 01 - 8 bit • 10 - 4 bit DDR (eMMC 4.4) • 11 - 8 bit DDR (eMMC 4.4)
BOOT_CFG0[19]	eMMC Speed Selection	0 - Normal Speed 1 - High Speed
BOOT_CFG0[20]	eMMC fast boot	0 - Disabled 1 - Enabled
BOOT_CFG0[22]	uSDHC IO Voltage Selection	0 - 3.3 V 1 - 1.8 V
BOOT_CFG1[0]	Fast Boot Acknowledge Disable	0 - Boot Ack Disabled 1 - Boot Ack Enabled
BOOT_CFG1[2]	uSDHC Power On Polarity	0 - Low

Table continues on the next page...

Table 59. Expansion Device eFUSE Configuration (continued)

Fuse	Definition	Settings
		1 - High
BOOT_CFG1[3]	uSDHC Power Cycle Delay	0 - 5 ms 1 - 2.5 ms
BOOT_CFG1[5:4]	uSDHC Power Cycle Interval	00 - 20 ms 01 - 10 ms 10 - 5 ms 11 - 2.5 ms
BOOT_CFG1[7:6]	SD Calibration Step	00 - 1 delay cell 01 - 2 delay cell 10 - 3 delay cell 11 - 5 delay cell
BOOT_CFG1[15:8]	uSDHC PAD Settings Low bits	The override value for uSDHC pad settings (Lowest 8 bits) 00000000 - 11111111
BOOT_CFG1[19:16]	uSDHC PAD Setting High bits	The override value for uSDHC pad settings (High bits) 0000 - 1111
BOOT_CFG1[20]	Power Cycle Enable	0 - No power cycle 1 - Enabled
BOOT_CFG1[21]	uSDHC Override Pad Settings (using PAD_SETTINGS value)	-
BOOT_CFG1[22]	uSDHC DLL Enable	0 - Disable DLL for SD/eMMC 1 - Enable DLL for SD/eMMC
BOOT_CFG1[30:24]	MMC DLL DLY	Delay target for uSDHC DLL, it is applied to slave mode target delay or override mode target delay depends on DLL Override fuse bit value 0000000 - 1111111
BOOT_CFG1[31]	uSDHC DLL Select	0 - DLL Slave Mode 1 - DLL Override Mode

9.8.2.4 SD/eMMC Memory Map

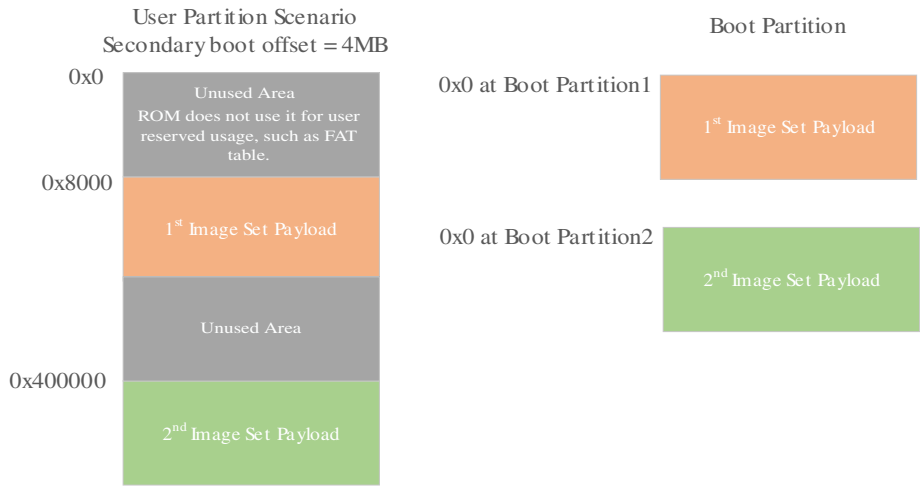


Figure 27. SD/eMMC Memory Map

9.8.3 LPSPI

The chip supports recovery devices, such as serial EEPROM and NOR flash, using the LPSPI. If the primary boot device fails, the boot ROM tries to boot from the recovery device using one of the LPSPI ports.

9.8.3.1 IOMUX Configuration for LPSPI

The following table shows the LPSPI pin configuration.

Table 60. LPSPI pin configuration

Signal Name	LPSPI1	LPSPI2	LPSPI3	LPSPI4
PCS1	PDM_BIT_STREAM0 (ALT2)	PDM_BIT_STREAM1 (ALT2)	GPIO_IO07 (ALT1)	GPIO_IO17 (ALT5)
SIN	SAI1_TXC (ALT2)	UART1_RXD (ALT2)	GPIO_IO09 (ALT1)	GPIO_IO19 (ALT5)
SOUT	SAI1_RXD0 (ALT2)	UART2_RXD (ALT2)	GPIO_IO10 (ALT1)	GPIO_IO20 (ALT5)
SCK	SAI1_TXD0 (ALT2)	UART2_TXD (ALT2)	GPIO_IO11 (ALT1)	GPIO_IO21 (ALT5)
PCS0	SAI1_TXFS (ALT2)	UART1_TXD (ALT2)	GPIO_IO08 (ALT1)	GPIO_IO18 (ALT5)

9.8.3.2 Expansion Device LPSPI fuse Configuration

LPSPI boot can be performed based on setting of the BOOT_CFG.

Table 61. Expansion Device LPSPI FUSE Configuration

Fuse	Definition	Settings
BOOT_CFG8[19]	Disable_A55_SPI_Recovery	0 - Disabled 1 - Enabled
BOOT_CFG8[20]	Disable_M33_SPI_Recovery	0 - Disabled

Table continues on the next page...

Table 61. Expansion Device LPSPI FUSE Configuration (continued)

Fuse	Definition	Settings
		1 - Enabled
BOOT_CFG2[16]	SPI Clock/Data Phase Control	0 - Tx on falling, Rx on rising 1 - Tx on rising, Rx on falling
BOOT_CFG2[17]	SPI SCLK Active	0 - Active high 1 - Active low
BOOT_CFG2[18]	SPI SS Active	0 - Active low 1 - Active high
BOOT_CFG2[19]	Recovery SPI Addressing	0 - 3-bytes (24-bit) 1 - 2-bytes (16-bit)
BOOT_CFG2[24]	Recovery CS Select	0 - CS0 1 - CS1
BOOT_CFG2[27:26]	Recovery Port Select	For single boot 0 - LPSPI3 1 - LPSPI4 For LP boot (CM33 ROM boot) 0 - LPSPI1 1 - LPSPI2

9.9 Serial Downloader Boot

9.9.1 USB Serial Download

The Serial Downloader provides a means to download a Program Image to the chip over USB serial connection.

USB serial download boot is supported on USB 2.0 OTG, as HID device. Serial download is only supported via the USB1 instance, for both boot types (LP Boot and Single boot).

USB boot in ROM code works as a HID device. The ROM will view the USB as a normal boot device, similar to the SD card. The difference from the SD card here is that the USB is a 'stream' device. ROM must 'read' it one byte at a time, and it can not specify the offset for the reading. This means that the image for USB boot must be continuous.

The USB boot flow is as follows:

1. ROM gets into USB boot mode and has been enumerated by Host.
2. The Host sends 8KB (initial image header) through USB. ROM reads 8KB from USB connection.
3. ROM parses the header and reads all SPL into OCRAM, per the information provisioned in the IVT header.
4. ROM authenticates the SPL and then jumps to it, if authentication passes.

USB Serial Download boot can be disabled by fuse.

The USB HID device VID/PID and strings are listed in the following table.

Table 62. VID/PID and strings for USB HID device

Descriptor	Value
VID	0x1FC9
PID	0x14E
PID for Azure	0x150
bcdDevice	0x0001

9.9.2 USB Serial Download Descriptor on Chip Lifecycle

ROM provides the current lifecycle (LC) status in the USB String Descriptor 2.

Table 63. USB string descriptor 2 lifecycle values

Lifecycle	USB String Descriptor 2 Value
FAB	FA
INVALID/UNKNOWN	IV
BLANK	BA
NXP_PROV	NP
OEM_OPEN	OO
OEM_SWC	OS
OEM_CLOSED	OC
FR_OEM	FO
FR_NXP	FN
OEM_LOCKED	OL

NOTE

For information on life cycle management of this chip, see the chip's Security Reference Manual.

9.9.3 Serial Downloader eFUSE Configuration

The SDP eFUSE settings are provided in the following table.

Table 64. SDP eFUSE Configuration

Fuse	Definition	Settings
BOOT_MODE[3:0]	Boot device selection	Coretx-A55 (Single boot): • 0001 - Serial Downloader Coretx-M33 (LP boot): • 1001 - Serial Downloader
BOOT_CFG10[2]	SDP_Disable	0 - Enabled 1 - Disabled

Table continues on the next page...

Table 64. SDP eFUSE Configuration (continued)

Fuse	Definition	Settings
BOOT_CFG10[3]	SDP_WO	0 - Enabled 1 - Disabled
BOOT_CFG10[4]	SDP_READ_DISABLE	0 - Enabled 1 - Disabled

9.10 Secure boot

The boot ROM uses the Advanced High Assurance Boot (AHAB) library to provide a secure boot environment for the Cortex-M33 low power boot and Cortex-A55 single boot procedures.

A key feature of the boot ROM is the ability to perform a secure boot. The boot ROM is also called Advanced High-Assurance Boot (AHAB), which protects the system from executing unauthorized program images. Before the AHAB allows the user image to execute, the image must be signed. The signing process is done during the image build process by the private key holder and the signatures are then incuded as a part of the final program image. If configured to do so, the AHAB verifies the signatures using the public keys included in the program image. The corresponding public key hash is embedded in the fuse.

In addition to supporting the digital signature verification to authenticate the program images, encrypted boot is also supported. The encrypted boot can be used to prevent the cloning of the program image directly off the boot device. A secure boot with AHAB can be performed on all boot devices supported on the chip in addition to the serial downloader. The out-of-fab setting for the LIFE_CYLCE is the open configuration, in which the ROM/AHAB performs the image authentication, but all authentication errors are ignored and the image is still allowed to execute.

NOTE

Refer to the chip's Security Reference Manual for additional details on secure boot.

9.11 ROM API

The main purpose is for SPL calling to load/authentication images. The ROM API table is as follows:

Table 65. ROM API Table

Offset	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
0x0	Tag: 0xEA91		Version: 0x0001	
0x4	Reserved as 0x0			
0x8-0xF	Pointer to address of download_image			
0x10 – 0x17	Pointer to address of query_boot_infor			
0x18-0x3F	Reserved as 0x0			

9.11.1 API download_image

The SPL calls this API to load images from the current boot device being active. The definition of this API is as follows.

```
UINT32 load_image(UINT8 *dst, UINT32 offset, UINT32 size, UINT32 xor)
```

The parameters are defined below:

- dst - The physical memory address to be loaded to.

- size - The image size to load. In bytes.
- offset - Where to read from boot device. In bytes. Relative to the beginning of the boot device
- xor - The xor of dst, size, and offset. To address fault injection attack.

The return values are defined below:

- 0xF0 for success others for failure.
- 0x01 – Invalid para. XOR wrong, invalid dest/sizeRead, invalid offset (not page aligned).
- 0x02 – low level device driver read failure (NAND ECC error, SD CRC error, etc.).
- 0x03 – Device not ready (not initialized yet or no valid device at all).
- 0x04 – Out of memory (such as loading data from FlexSPI flash but space is out of FlexSPI NOR device).

NOTE

For USB, the offset will be ignored by ROM.

For eMMC Fast boot, there are two scenarios.

1. Primary boot succeeds, eMMC card in Fast Boot Mode, offset will be ignored by load_image API.
2. Primary boot failed, ROM moves card into normal boot mode, then boots from Secondary Image, offset needs to be provisioned then.

The SPL can call query_boot_infor (BOOT_DEVICE) to check if the card is in Fast Boot Mode. Provisioning "offset" in SPL is recommended since it works for both scenario #1 and #2 above.

NOTE

The API does not need the caller to specify which interface it needs to download.

When booting from SPI NAND and RawNAND, the SPL must pass the good page offset (skip all the bad block) of the NAND device.

9.11.2 API query_boot_infor

```
UINT32 query_boot_infor(UINT32 infor_type, UINT32 *infor, UINT32 xor)
```

The return values are defined below:

- 0xF0 for success and other for failure.
- 1 Invalid para. Invalid Infor type or Invalid Infor type.

The paramaters are defined below:

- infor_type:
  - 01 - BootROM Version
  - 02 - Boot device (see Boot Device Definition).
  - 03 - Device page size (in byte).
  - 04 - The offset (in byte, relative to beginning of the boot image) of the current image's IVT. The intent here is SPL can calculate the physical address (in the boot device) of SW image to be loaded.
  - 05 - Boot Stage (see Boot Stage definition).
  - 06 - The offset (in byte, relative to beginning of the boot device) of the current image. The intent here is SPL can calculate the physical address (in the boot device) of SW image to be loaded.
- Infor - Points to the word to save the infor being queried, valid if returns 0xF0. Must be in TCM/OCRAM free space.
- Xor - The xor of infor_type and Info. To address fault injection attack.

9.11.2.1 Boot Device Definition

Table 66. Boot Device Definition

Bit 31-24	Bit 23-16	Bit 15-8	Bit 7-0
Reserved	Boot Interface	Instance	Device_State

The field definitons are provides below.

- Boot Interface -Indicating the boot interface from which SPL been boot.
  - 0x1 – SD
  - 0x2 – eMMC
  - 0x4 – FlexSPI NOR flash
  - 0x6 – LPSPI EEPROM
  - 0x8 – FlexSPI NAND flash
  - 0xE – USB
- Instance - to specify which port the SPL boots from, from 0 (for example, for SD/eMMC boot, 0 means booting from USDHC1 port).
- Device State - to specify the state of the boot device, if needed. Usually Boot Interface dependent.
  - For eMMC: Bit0 – Card in Fast Boot Mode if set.
  - For other Boot Interfaces, this field shall be 0.

9.11.2.2 Boot Stage definition

The Boot Stage values are as defined below:

- 0x6 – Primary Boot
- 0x9 – Secondary Boot
- 0xA – Recovery Boot
- 0x5 – USB boot

Chapter 10
IOMUXC GPR

10.1 Overview

The IOMUXC GPR module includes a configuration register.

10.2 Memory map and register definition

This section includes the IOMUXC-GPR memory map and detailed description of Configuration register.

10.2.1 IOMUXC GPR register descriptions

10.2.1.1 IOMUXC GPR memory map

IOMUXC_GPR.IOMUXC_GPR base address: 443C_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	IOMUXC GPR Configuration (CONFIG)	32	RW	0000_8920

10.2.1.2 IOMUXC GPR Configuration (CONFIG)

Lower 16 bits [15:0] are data bits and upper 16 bits [31:16] are lock bits. They are one-to-one mapped and control the access of IOMUXC module. Once a lock bit is set, it and its corresponding data bit cannot be cleared unless a reset. Each field can be assigned to any master ID through configuration.

Table 67. Master IDs

Module	4-bit Master ID
Cortex-A55 cores	000n
Reserved	0010 - 0111
Cortex-M33	1000
Reserved	1001
Edgelock	1010
eDMA3_1(AHB)	1011
eDMA3_1(AXI)	1100
Reserved	1101 - 1110
All other bus masters	1111

Offset

Register	Offset
CONFIG	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				MASTERID2_LOCK				MASTERID1_LOCK				MASTERID0_LOCK			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				MASTERID2_DATA				MASTERID1_DATA				MASTERID0_DATA			
Reset	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-24 MASTERID2_LOCK	Lock bits
23-20 MASTERID1_LOCK	Lock bits
19-16 MASTERID0_LOCK	Lock bits
15-12 —	Reserved
11-8 MASTERID2_DATA	Data bits
7-4 MASTERID1_DATA	Data bits
3-0 MASTERID0_DATA	Data bits

Chapter 11
Arm Modules Overview

11.1 Overview

The following Arm modules are supported in this chip:

- Arm Cortex®-A55 MPCore
- Cortex-M33 Platform
- Neural-network Processing Unit (NPU)
- TrustZone

11.1.1 Arm Cortex®-A55 MPCore

The Cortex®-A55 MPCore has the following features:

- Two Cortex®-A55 processors:
  - 32 kByte L1 Instruction Cache
  - 32 kByte L1 Data Cache
  - 64 kByte per-core L2 cache
  - Media Processing Engine (MPE) with NEONTM technology supporting the Advanced Single Instruction Multiple Data architecture
  - Floating Point Unit (FPU) with support of the VFPv4-D16 architecture
- Support of 64-bit Arm®v8.2-A architecture
- 256 kByte cluster L3 cache

In addition, it also includes various components composing the CoreSightTM debug and trace technology, such as ETM, CTM and CTI.

The Cortex®-A55 platform supports following power gating modes:

Table 68. Cortex-A55 Platform Power Modes

-	CPU0	CPU1	PLATFORM	L3 CACHE Memory array	VDD
ALL_ON	ON	ON	ON	ON	ON
ONE_CPU_ON	ON	OFF	ON	ON	ON
ALL_CPU_OFF	OFF	OFF	ON	ON	ON
L3_RETENTION	OFF	OFF	OFF	RET	ON
ALL_OFF	OFF	OFF	OFF	OFF	ON
POWER_DOWN	OFF	OFF	OFF	OFF	OFF

11.1.2 Cortex-M33 Platform

- Stand by monitoring with Cortex®-A55 and other high-power modules power gated
- IoT device control (for example running uWeave)
- ML applications

- ECC support for both cache and TCM

The Cortex-CM33 core

- 16KB L1 Instruction Cache
- 16KB L1 Data Cache
- 256 KByte TCM, also accessible as SRAM by the rest of the system

.

11.1.3 Neural-network Processing Unit (NPU)/ML (Machine Learning)

This chip includes a Neural-network Processing Unit (NPU)/ML (Machine Learning) module. It supports the following features:

- 256 MACs operating at up to 1.0GHz and 2 OPS/MAC
- 2x 128-bit AXI master interfaces
- 32-bit APB slave interface
  - The 128-bit M0 port is used by the NPU to access OCRAM.
  - The 128-bit M1 port is used by the NPU to access DRAM.
- Q-channel slave port for interaction with the CCM and GPC clock and power control.

11.1.4 GIC-600

GIC-600 is used as the interrupt controller for the Cortex®-A55 processor. It supports a distributed microarchitecture containing several individual blocks that are used to provide a flexible GIC implementation.

- Forward compatibility with ARMv8.2 Cores supporting increased 512 CPU threads/chip and multi-chip capabilities
  - Distributed architecture for reduced power, congestion and improved latency
  - End-to-End credited protocols allows bridging communication to any free-flowing transport layer. Transport layer will be native AXI4 stream.
  - Clock and power control for each block for improved power savings
- QoS improvements
  - Increased ITS caching to reduce memory accesses
  - More flexible use of LPI cache and Pending Tables for reduced memory accesses and improved latency bounding
- Improved trace and debug
  - PMU with full snapshot functionality
  - Full ITS command error logging and reporting via ARMv8.2 RAS error records
  - ECC error logging and containment to allow software recovery
- Improved control over 1-of-N interrupt routing
- SPI collators enabling Q-Channel support on main GIC

11.1.5 TrustZone

The TrustZone architecture provides means to help secure systems, using the TrustZone Security Extensions, and Secure peripherals. TrustZone is a security feature on ARM Cortex M and A series that enables execution separation of trusted (secure) software, and enables access control isolation of trusted resources from non-trusted (non-secure) software and resources, while running on the same CPU. Memory arrays and peripherals are segmented into either Secure (S) or Non-Secure (NS). TrustZone is optimized for energy-efficient, embedded applications that require real-time responsiveness.

More details on TrustZone can be found in “TrustZone® technology can be found at arm.com.

NOTE

"S" = Secure; "NS" = Non-Secure.

Trustzone is always enabled on this chip and is supported through TRDC.

If TrustZone functionality is enabled, then the following restrictions apply to the cores:

- The CPU in a Secure state (CPU-S) can execute instructions from Secure memory (S-memory) only; the CPU cannot execute instructions from Non-Secure memory (NS-memory).
- CPU-S can access data in both S-memory and NS-memory, which means that CPU-S can execute data reads from both S- and NS-memory, and also execute data writes to S or NS-memory.
- CPU-NS can execute instructions only from NS-memory, and cannot execute instructions from S-memory.
- CPU-NS can access data only in NS-memory, which means that CPU-NS can execute data reads from NS-memory only, and can execute data writes to NS-memory only. CPU-NS cannot access data from S-memory.

Chapter 12
ARM Cortex A55

12.1 Overview

The Arm Cortex-A55 implements the Armv8-A architecture. The Cortex-A55 is a high efficiency processor with Media Processing Engine (MPE) with NEON technology supporting Advanced Single Instruction Multiple Data architecture. The Cortex-A55 also includes a floating-point unit (FPU), L1 instruction cache, and L1 data cache, and L2 cache.

There are 2 A55 cores on this chip.

The core also includes various components composing the Arm CoreSight debug and trace technology, such as ETM, CTM and CTI. The debug interface is connected to the DAP and CoreSight debug modules in the SoC to allow debug through JTAG interface.

12.1.1 Block Diagram

The A55 block diagram is shown below.

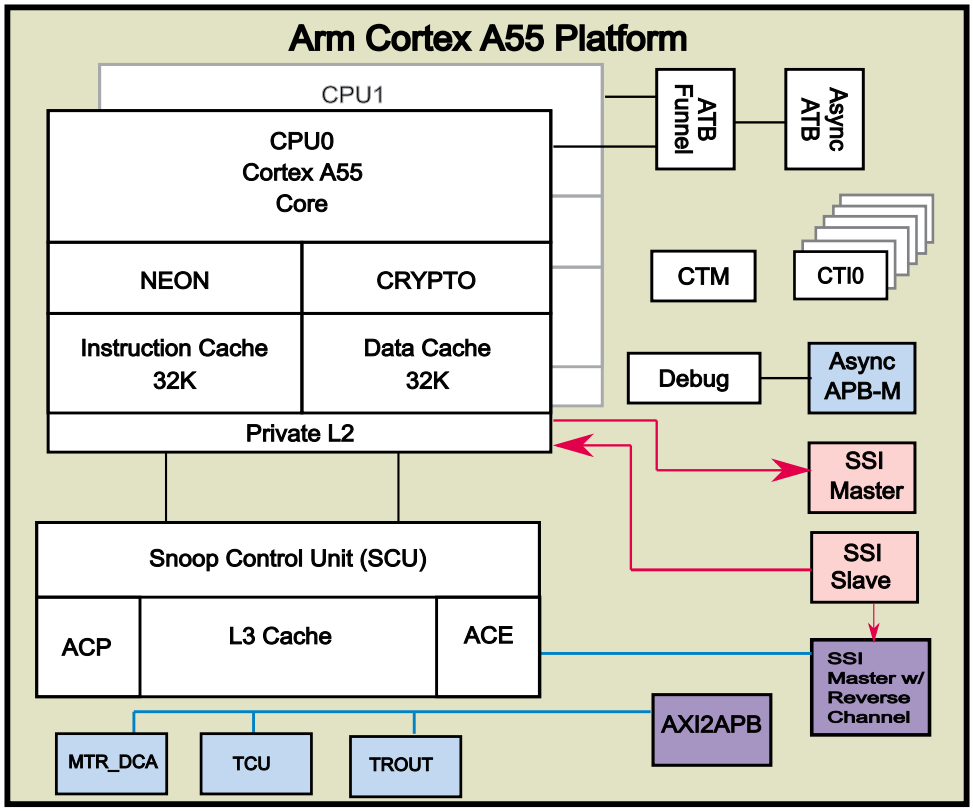


Figure 28. A55 Block Diagram

12.1.2 Features

The Arm Cortex A55 MPCore has the following features:

- Two Cortex-A55 processors:
  - 32 kByte L1 Instruction Cache

  - 32 kByte L1 Data Cache
  - 64 kByte per-core L2 cache
  - Media Processing Engine (MPE) with NEON technology supporting the Advanced Single Instruction Multiple Data architecture
  - Floating Point Unit (FPU) with support of the VFPv4-D16 architecture
- Support of 64-bit Arm v8.2-A architecture
- 256 kByte cluster L3 cache
- Various components composing the CoreSightTM debug and trace technology, such as ETM, CTM and CTI
- The DynamIQ Shared Unit (DSU) shared between cores, includes Snoop Control Unit (SCU), L3 cache, ACE interface, etc.

12.2 Functional Description

The sections below discuss the high-level overview of the Arm Cortex-A55 Platform components.

12.2.1 Arm Cortex-A55 MPCore Processor

The Arm Cortex-A55 processor implements the Armv8-A architecture. the A55 complex has 2. Each core has a 32 kByte L1 Instruction Cache, a 32 kByte L1 Data Cache and a 64 kByte per-core L2 cache. Each core also has a Media Processing Engine (MPE) with NEON technology supporting the Advanced Single Instruction Multiple Data architecture and a Floating Point Unit (FPU) with support of the VFPv4-D16 architecture.

12.2.2 DynamIQ Shared Unit (DSU)

The DynamIQ Shared Unit (DSU) is the portion of the A55 complex that is shared between the cores. The DSU includes the Snoop Control Unit (SCU), L3 cache, etc. The SCU arbitrates between the cores requesting L3 cache access.

12.2.3 Advanced SIMD (NEON)

The Cortex-A55 processor supports the Advanced SIMD and Scalar Floating-point instructions.

See Arm Cortex-A55 documentation for more information.

12.2.4 Atomic Instructions

The Arm Cortex-A55 Core Technical Reference Manual mentions that atomics are supported to cacheable memory but requires a coherent interconnect to support atomics to device or non-cacheable memory. This product does not support such a coherent interconnect. If such an atomic instruction is executed when the interconnect does not support them, it will result in a synchronous Data Abort (for load atomics) or an asynchronous Data Abort (for store atomics)

12.2.5 Debug

The Cortex-A55 platform includes the debug and trace sub-modules. The debug interface is connected to the DAP and CoreSight debug modules in the SoC to allow debug through JTAG interface.

See the System Debug chapter for more information about the A55 Platform debug components, such as ETM, CTI and CTM.

12.2.6 Clocking

The Cortex-A55 platform takes in the following clocks, all of which may be fully asynchronous to each other.

Table 69. A55 Clocks

A55 Platform Clock Name	Source	Frequency (Nominal Voltage)	Description
PERIPHCLK	CCM: Arm_a55_periphclk_root	333MHz	Timers, clock and power management logic in Cortex-A55 platform.
PCLK			Debug Block and DSU Debug APB interfaces.
ATCLK			ATB trace buses output from DSU
GICCLK			GIC AXI-stream interface between DSU and SoC's GIC-600
SCLK	Mux between either Arm_pll_clk or Arm_a55_clk_root	1.4GHz	SCU and L3 subsystem, ACE interface and A55 cores

12.2.7 Interrupts

The Generic Interrupt Controller (GIC) supports and manages interrupts in the cores.

The GIC provides registers for managing:

- Interrupt sources
- Interrupt behavior
- Interrupt routing to one or more cores

GIC-600 is supported

See Arm documentation for more information.

12.3 External Signals

The A55 Platform has no external signals.

Chapter 13
ARM Cortex CM33 TCM MCM and ECC MCM (CM33_CACHE_TCM_ECC_MCM)

13.1 Chip-specific CM33 information

Table 70. Reference links to related information

Topic	Related module	Reference
Full description	Arm Cortex CM33 Platform (CM33)	CM33 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The Cortex-CM33 accesses its Code TCM and Data TCM through its C-AHB and S-AHB buses.

By default, the Code TCM and System TCM are 128KB each. Note that the BLK_CTRL_S_AONMIX register bitfield M33_CFG[TCM_SIZE] is a 2-bit field with 3 valid encodings:

- 00 (default) = REGULAR_TCM: 128kB Code TCM and 128kB Sys TCM
- 01 = DOUBLE_CODE_TCM. 256kB Code TCM and 0 Sys TCM
- 10 = DOUBLE_SYS_TCM. 256kB Sys TCM and 0 Code TCM
- 11 = Reserved

NOTE

To avoid unexpected SW issues, writes to this M33_CFG[TCM_SIZE] register should only be performed when the Cortex-M33 core is in reset.

Code and System TCM are mapped to addresses:

- Code TCM => 0x0FFE_0000 (non-secure) and 0x1FFE_0000 (secure) (plus the additional 128kByte below this if M33_CFG[TCM_SIZE] == DOUBLE_CODE_TCM)
- Data TCM => 0x2000_0000 (non-secure) and 0x3000_0000 (secure) (plus the additional 128kByte above this if M33_CFG[TCM_SIZE] == DOUBLE_DATA_TCM)

The system access to the TCM is mapped to address:

- Code TCM => 0x201E_0000 .. 0x201F_FFFF (plus the additional 128kByte below this if TCM_SIZE == DOUBLE_CODE_TCM)
- System TCM => 0x2020_0000 .. 0x2021_FFFF (plus the additional 128kByte above this if TCM_SIZE == DOUBLE_DATA_TCM)

The valid range of these address ranges depends on the value of M33_CFG[TCM_SIZE].

See the M33_CFG register in the BLK_CTRL_S_AONMIX chapter for more information about CM33 and TCM configuration options.

NOTE

The i.MX93W does not support the Cortex-M33 external NMI pin.

13.2 Overview

CM33 implements the Armv8-M architecture with main extensions. The Arm Cortex-M33 is a high-efficiency processor with security considerations. CM33 includes:

- A single-precision floating-point unit (FPU)
- A nested vectored interrupt controller (NVIC)
- A flash patch breakpoint (FPB)
- The data watchpoint and trace (DWT) unit
- An instrumentation trace macrocell (ITM)

As shown in Figure 29, the CM33 platform also includes:

- Miscellaneous control module CM33_TCM_MCM, which contains programming registers to control the ECC features for CM33 code TCM and system TCM
- Miscellaneous control module CM33_CACHE_ECC_MCM, which contains programming registers to control the ECC features for TAG0, TAG1, DATA0, and DATA1 in the CM33 code cache and system cache

13.2.1 Block diagram

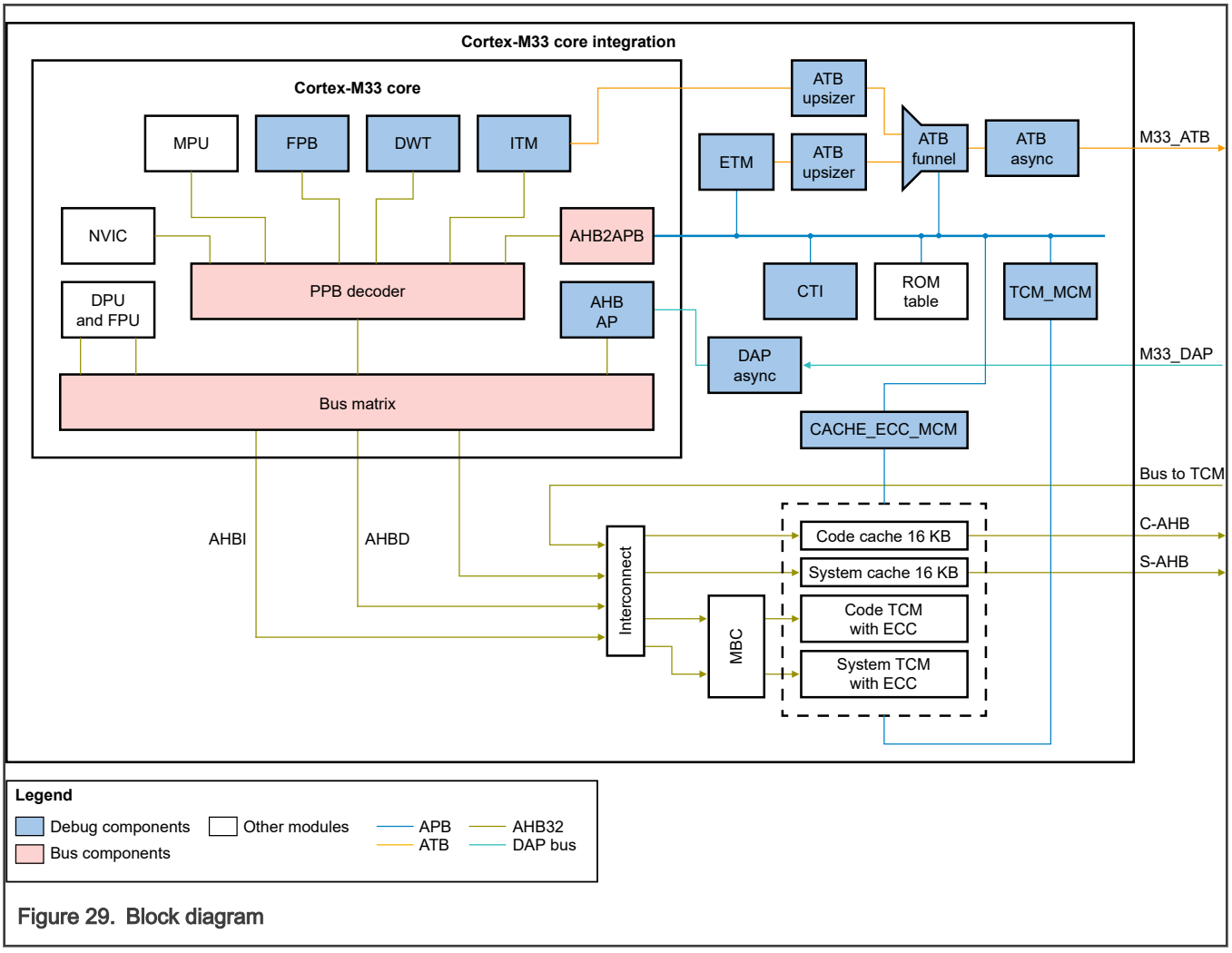


Figure 29. Block diagram

13.2.2 Features

- Cortex-M33 processor
- Single-precision FPU
- NVIC support
- FPB support (does not include patching)
- DWT and ITM support
- ETM support
- Two-way set-associative 16-KB system cache
- Two-way set-associative 16-KB code cache
- Configurable 256-KB tightly coupled memory (TCM):
  - 128-KB system TCM on system bus
  - 128-KB code TCM on code bus
- TCM ECC support
- Cache ECC support
- DSP extensions
- Security extensions
- Wake-up interrupt controller (WIC) support
- Nonsecure memory protection unit (MPU) with 16 regions
- Secure MPU with 16 regions
- Cross trigger interface (CTI) support

The AON domain, which incorporates the Cortex-M33 core, includes one messaging unit (MU) and one hardware semaphore (SEMA42) for inter-core communication with the Cortex-M33 core.

For the Cortex-M33 core, the CM33 platform also supports execution in place (XIP) to facilitate fast power-on wake-up. See the "System Boot" chapter for more details.

The Cortex-M33 debug interface is connected to the DAP and CoreSight debug modules in the chip to allow debugging through the JTAG interface. See the "System Debug" chapter for more details.

13.3 Functional description

The following sections contain a high-level overview of the CM33 components.

13.3.1 Cortex-M33 processor

The Cortex-M33 processor is based on the Armv8-M architecture and has a low gate count and high energy efficiency.

13.3.2 NVIC

NVIC supports 6 priority levels for interrupts. The NVIC IPR registers define 4 bits per IRQ.

13.3.3 FPU

The FPU implements FPv5 floating-point extensions, which support these single-precision operations: add, subtract, multiply, divide, multiply and accumulate, and square root. The FPU also provides conversions between fixed-point and floating point data formats, and it provides floating-point constant instructions.

The FPU is compliant with ANSI/IEEE Std 754-2008.

13.3.4 MPU

The MPU has independent attribute settings for each region and can export memory attributes.

CM33 also supports a secure MPU.

When a core access hits an overlapping memory region, the processor generates a fault.

Instruction accesses and data accesses in a given region have the same region settings.

13.3.5 Debug

See the "System Debug" chapter for more information about the CM33 platform debug components.

13.3.6 Code and system TCM

CM33 accesses its code TCM and system TCM through its C-AHB and S-AHB buses, respectively.

By default, the code TCM and system TCM are 128 KB each. The BLK_CTRL_S_AONMIX register field M33_CFG[TCM_SIZE] is a 2-bit field with three valid encodings:

- 00 (default) = REGULAR_TCM: 50% code TCM and 50% system TCM
- 01 = DOUBLE_CODE_TCM: 100% code TCM and 0% system TCM
- 10 = DOUBLE_SYS_TCM: 100% system TCM and 0% code TCM
- 11 = Reserved

NOTE

To avoid unexpected software issues, perform writes to BLK_CTRL_S_AONMIX.M33_CFG[TCM_SIZE] only when the Cortex-M33 core is in reset.

Code TCM is mapped to 0FFE0000h (nonsecure) and 1FFE0000h (secure) (plus the additional 128 KB below this if M33_CFG[TCM_SIZE] = DOUBLE_CODE_TCM)

System TCM is mapped to 20000000h (nonsecure) and 30000000h (secure) (plus the additional 128 KB above this address if M33_CFG[TCM_SIZE] = DOUBLE_DATA_TCM)

The system bus can access both TCM memories as a continuous address space via their aliases:

- Code TCM is mapped to the aliased code TCM base address (plus the additional 128 KB below this if TCM_SIZE = DOUBLE_CODE_TCM)
- System TCM is mapped to the aliased system base address (plus the additional 128 KB above this address if TCM_SIZE = DOUBLE_DATA_TCM)

See the chip's memory map information for aliased code and system TCM base address details.

The valid range of these address ranges depends on the value of BLK_CTRL_S_AONMIX.M33_CFG[TCM_SIZE].

See the M33_CFG register in the BLK_CTRL_S_AONMIX chapter for more information about the CM33 and TCM configuration options.

13.3.6.1 CM33_TCM_MCM

CM33_TCM_MCM is a miscellaneous control module for the Cortex-M33 core. It contains programming registers to control the ECC features of CM33 TCM.

13.3.6.2 CM33 TCM ECC support

You can use the syndrome value for single-bit ECC to locate the error bit in the CM33 system and code TCM. See the following lookup table for these values.

Table 71. Lookup table for 32-bit data single-bit ECC syndrome

Syndrome value	Error bit
07h	31
7Fh	30
6Eh	29
6Dh	28
6Bh	27
67h	26
5Eh	25
5Dh	24
5Bh	23
57h	22
4Fh	21
7Ch	20
7Ah	19
79h	18
76h	17
75h	16
73h	15
70h	14
68h	13
64h	12
62h	11
61h	10
58h	9
54h	8
52h	7
51h	6
4Ch	5
4Ah	4
49h	3
46h	2
45h	1
43h	0

NOTE

- You cannot use the syndrome value for multibit ECC to locate the error bit in the CM33 system and code TCM.
- You cannot use the syndrome value for single-bit and multibit ECC to locate the error bit in the CM33 system and code cache.

13.3.7 CM33 code and system cache

The following caches are included in the CM33 platform:

- Two-way set-associative 16-KB code cache with ECC and write path error injection support
- Two-way set-associative 16-KB system cache with ECC and write path error injection support

13.3.7.1 CM33_CACHE_ECC_MCM

CM33_CACHE_ECC_MCM is a miscellaneous control module for the Arm Cortex-M33 cache ECC. It contains programming registers to control the ECC features of CM33 TAG0, TAG1, DATA0, and DATA1.

13.3.8 Clocking

See the chip's clocking sections for more information on CM33 clocks.

Table 72. Clocks

Clock name	Description
m33_clk	Main M33 clock
m33_systick_clk	M33 Systick clock

13.3.9 Reset

You can use a CM33 reset to reset the CM33_TCM_MCM and CM33_CACHE_ECC_MCM submodules. Performing a cold or soft reset on the CM33 core also resets the submodules.

13.3.10 Interrupts

Table 73. CM33_TCM_MCM interrupt requests

Interrupt request	Asserted when
tcm_ecc1bit_irq	CM33 detects a single-bit ECC error on TCM (including code and system TCM) and the corresponding interrupt is enabled.
tcm_ecc2bit_irq	CM33 detects a multi-bit ECC error on TCM (including code and system TCM) and the corresponding interrupt is enabled.

Table 74. CM33_CACHE_ECC_MCM interrupt requests

Interrupt request	Asserted when
ecc1bit_irq	CM33 detects a single-bit ECC error on cache (including cache TAG0, cache TAG1, cache DATA0, and cache DATA1) and the corresponding interrupt is enabled.
ecc2bit_irq	CM33 detects a multibit ECC error on cache (including cache TAG0, cache TAG1, cache DATA0, and cache DATA1) and the corresponding interrupt is enabled.

13.4 External signals

The CM33 platform has no external signals.

13.5 Initialization

To initialize CM33:

1. Verify that the clock for CM33 is enabled.
2. Enable the ECC features by writing 0 to the following fields before checking the cache or TCM ECC on read or write data:
   - CACHE_ECCR[RECC_DIS]
   - CACHE_ECCR[WECC_DIS]
   - TCMECCR[RECC_DIS]
   - TCMECCR[WECC_DIS]

13.6 Memory map and register definition

This section includes the memory map and detailed descriptions of all registers.

NOTE

For CM33_TCM_MCM:

- When you configure the code TCM to be double sized (BLK_CTRL_S_AONMIX.M33_CFG[TCM_SIZE] = 01b), all of the TCM is used as code TCM.
- When you configure the system TCM to be double sized (BLK_CTRL_S_AONMIX.M33_CFG[TCM_SIZE] = 10b), all the TCM is used as system TCM.
- However, the code injection and status register always correspond to the TCM space that starts at FFE0000h, and the system injection and status register always correspond to the TCM space that starts at 20000000h.

13.6.1 CM33_CACHE_ECC_MCM register descriptions

13.6.1.1 CM33_CACHE_ECC_MCM memory map

CM33_CACHE_ECC_MCM base address: 4440_1000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	CACHE ECC Control (CACHE_ECCR)	32	RW	0000_0000
20	Interrupt Status (INT_STATUS)	32	RW	0000_0000
24	Interrupt Status Enable (INT_STAT_EN)	32	RW	0000_0000
28	Interrupt Enable (INT_SIG_EN)	32	RW	0000_0000
5C	Code Cache Single-Bit ECC Error Information (CODE_CACHE_ECC_SINGLE_ERROR_INFO)	32	R	0000_0000
60	Code Cache Single-Bit ECC Error Address (CODE_CACHE_ECC_SINGLE_ERROR_ADDR)	32	R	0000_0000
68	Code Cache Multibit ECC Error Information (CODE_CACHE_ECC_MULTI_ERROR_INFO)	32	R	0000_0000
74	System Cache Single-Bit ECC Error Information (SYSTEM_CACHE_ECC_SINGLE_ERROR_INFO)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
78	System Cache Single-Bit ECC Error Address (SYSTEM_CACHE_ECC_SINGLE_ERROR_ADDR)	32	R	0000_0000
80	System Cache Multibit ECC Error Information (SYSTEM_CACHE_ECC_MULTI_ERROR_INFO)	32	R	0000_0000
84	System Cache Multibit ECC Error Data (SYSTEM_CACHE_ECC_MULTI_ERROR_DATA)	32	R	0000_0000
8C	Code Cache TAG0 ECC Error Injection (CODE_CACHE_TAG0_ECC_ERROR_INJEC)	32	RW	0000_0000
90	Code Cache TAG1 ECC Error Injection (CODE_CACHE_TAG1_ECC_ERROR_INJEC)	32	RW	0000_0000
94	Code Cache DATA0 ECC Error Injection (CODE_CACHE_DATA0_ECC_ERROR_INJEC)	32	RW	0000_0000
98	Code Cache DATA1 ECC Error Injection (CODE_CACHE_DATA1_ECC_ERROR_INJEC)	32	RW	0000_0000
9C	System Cache TAG0 ECC Error Injection (SYTEM_CACHE_TAG0_ECC_ERROR_INJEC)	32	RW	0000_0000
A0	System Cache TAG1 ECC Error Injection (SYSTEM_CACHE_TAG1_ECC_ERROR_INJEC)	32	RW	0000_0000
A4	System Cache DATA0 ECC Error Injection (SYSTEM_CACHE_DATA0_ECC_ERROR_INJEC)	32	RW	0000_0000
A8	System Cache DATA1 ECC Error Injection (STSTEM_CACHE_DATA1_ECC_ERROR_INJEC)	32	RW	0000_0000

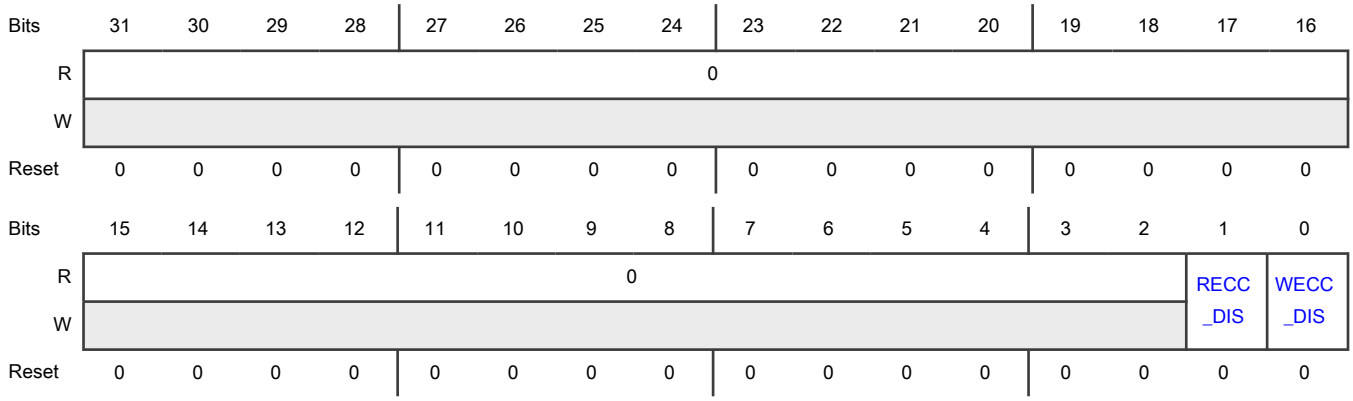
13.6.1.2 CACHE ECC Control (CACHE_ECCR)

Enables the ECC logic for the Cortex-M33 cache. The cache ECC is enabled by default.

Offset

Register	Offset
CACHE_ECCR	0h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 RECC_DIS	Disable Cache ECC Read Check Disables the ECC check on cache read data. 0 - Enable 1 - Disable
0 WECC_DIS	Disable CACHE ECC Write Generation Disables ECC generation on cache write data. 0 - Enable 1 - Disable

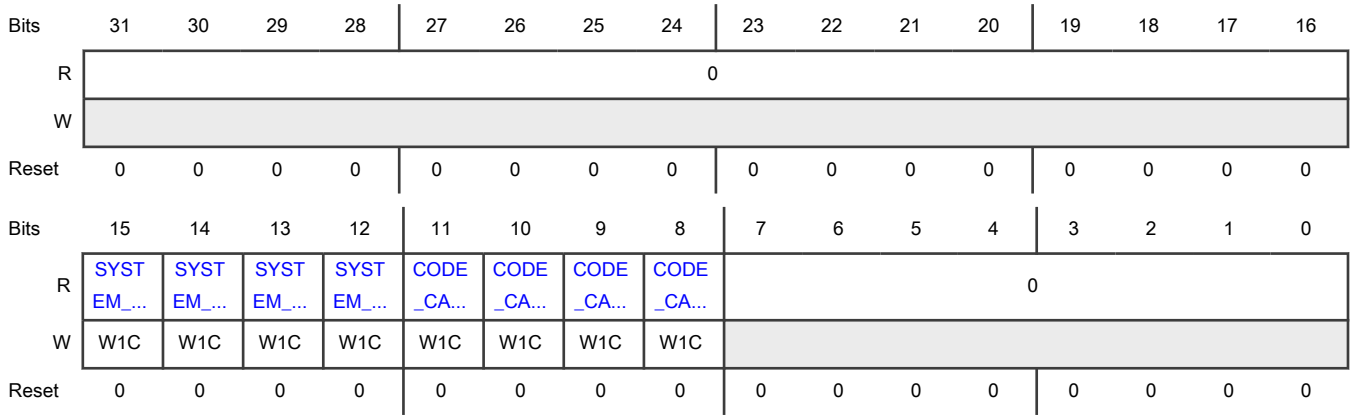
13.6.1.3 Interrupt Status (INT_STATUS)

Contains the ECC interrupt status for the code and system cache.

Offset

Register	Offset
INT_STATUS	20h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 SYSTEM_CACHE_ECC_ERRS_OVER_INT	System Cache Access Multiple Single-Bit ECC Error Interrupt Status Indicates more than one single-bit ECC error in system cache read data when INT_STAT_EN[SYSTEM_CACHE_ECC_ERRS_OVER_INT_EN] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not more than one error 1 - Multiple errors When writing 0 - No effect 1 - Clear the flag
14 SYSTEM_CACHE_ECC_ERRM_OVER_INT	System Cache Access Multiple Multibit ECC Error Interrupt Status Indicates more than one multibit ECC error in system cache read data when INT_STAT_EN[SYSTEM_CACHE_ECC_ERRM_OVER_INT_EN] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not more than one error 1 - Multiple errors When writing 0 - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Clear the flag
13 SYSTEM_CACHE_ECC_ERRS_INT	System Cache Access Single-Bit ECC Error Interrupt Status Indicates a single-bit ECC error in system cache read data when INT_STAT_EN[SYSTEM_CACHE_ECC_ERRS_INT_EN] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
12 SYSTEM_CACHE_ECC_ERRM_INT	System Cache Access Multibit ECC Error Interrupt Status Indicates a multibit ECC error in system cache read data when INT_STAT_EN[SYSTEM_CACHE_ECC_ERRM_INT_EN] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
11 CODE_CACHE_ECC_ERRS_OVER_INT	Code Cache Access Multiple Single-Bit ECC Error Interrupt Status Indicates more than one single-bit ECC error in code cache read data when INT_STAT_EN[CODE_CACHE_ERRS_OVER_INT_EN] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not more than one error 1 - Multiple errors When writing 0 - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Clear the flag
10 CODE_CACHE_ECC_ERRM_OVER_INT	Code Cache Access Multiple Multibit ECC Error Interrupt Status Indicates more than one multibit ECC error in code cache read data when INT_STAT_EN[CODE_CACHE_ERRM_OVER_INT_EN] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not more than one error 1 - Multiple errors When writing 0 - No effect 1 - Clear the flag
9 CODE_CACHE_ECC_ERRS_INT	Code Cache Access Single-Bit ECC Error Interrupt Status Indicates a single-bit ECC error in code cache read data when INT_STAT_EN[CODE_CACHE_ERRS_INT_EN] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
8 CODE_CACHE_ECC_ERRM_INT	Code Cache Access Multibit ECC Error Interrupt Status Indicates a multibit ECC error in code cache read data when INT_STAT_EN[CODE_CACHE_ERRM_INT_EN] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Clear the flag
7-0 —	Reserved

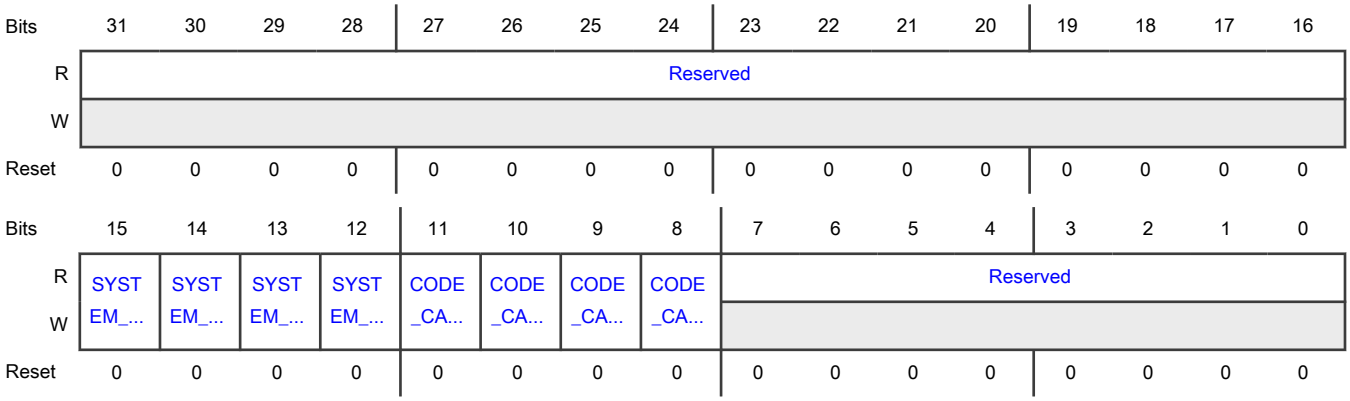
13.6.1.4 Interrupt Status Enable (INT_STAT_EN)

Enables the interrupt statuses in Interrupt Status (INT_STATUS).

Offset

Register	Offset
INT_STAT_EN	24h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 SYSTEM_CACHE_ECC_ERRS_OVER_INT_EN	System Cache Access Multiple Single-Bit ECC Error Interrupt Status Enable Enables INT_STATUS[SYSTEM_CACHE_ECC_ERRS_OVER_INT]. 0 - Mask 1 - Enable
14	System Cache Access Multiple Multibit ECC Error Interrupt Status Enable Enables INT_STATUS[SYSTEM_CACHE_ECC_ERRM_OVER_INT].

Table continues on the next page...

Table continued from the previous page...

Field	Description
SYSTEM_CACHE_ECC_ERRM_OVER_INT_EN	0 - Mask 1 - Enable
13 SYSTEM_CACHE_ECC_ERRS_INT_EN	System Cache Access Single-Bit ECC Error Interrupt Status Enable Enables INT_STATUS[SYSTEM_CACHE_ECC_ERRS_INT]. 0 - Mask 1 - Enable
12 SYSTEM_CACHE_ECC_ERRM_INT_EN	System Cache Access Multibit ECC Error Interrupt Status Enable Enables INT_STATUS[SYSTEM_CACHE_ECC_ERRM_INT]. 0 - Mask 1 - Enable
11 CODE_CACHE_ERRS_OVER_INT_EN	Code Cache Access Multiple Single-Bit ECC Error Interrupt Status Enable Enables INT_STATUS[CODE_CACHE_ECC_ERRS_OVER_INT]. 0 - Mask 1 - Enable
10 CODE_CACHE_ERRM_OVER_INT_EN	Code Cache Access Multiple Multibit ECC Error Interrupt Status Enable Enables INT_STATUS[CODE_CACHE_ECC_ERRM_OVER_INT]. 0 - Mask 1 - Enable
9 CODE_CACHE_ERRS_INT_EN	Code Cache Access Single-Bit ECC Error Interrupt Status Enable Enables INT_STATUS[CODE_CACHE_ECC_ERRS_INT]. 0 - Mask 1 - Enable
8 CODE_CACHE_ERRM_INT_EN	Code Cache Access Multibit ECC Error Interrupt Status Enable Enables INT_STATUS[CODE_CACHE_ECC_ERRM_INT]. 0 - Mask 1 - Enable
7-0 —	Reserved

13.6.1.5 Interrupt Enable (INT_SIG_EN)

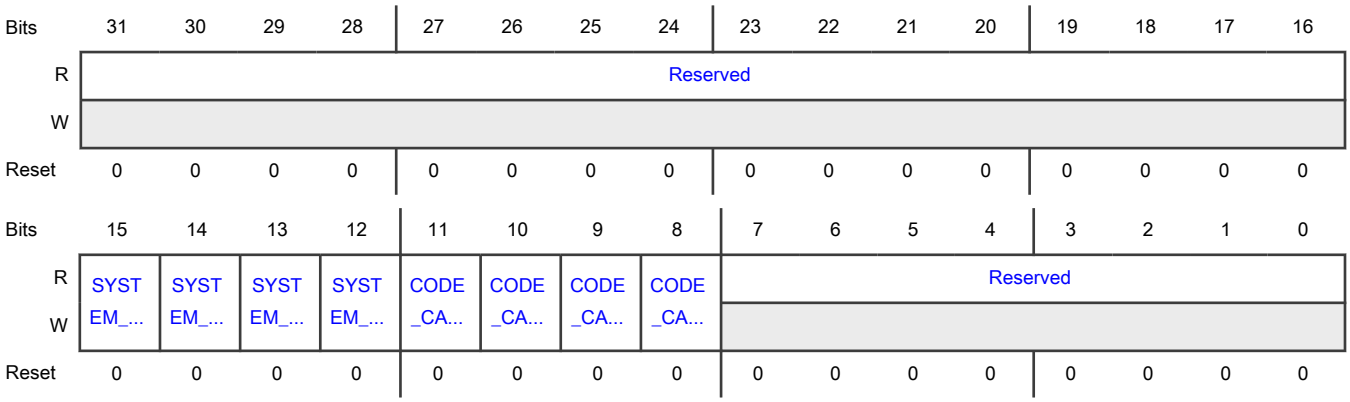
Selects which interrupt statuses are available to interrupt the host system. All these status fields share the same interrupt line (that is, all interrupts are combined with a logical OR). Writing 1 to any of these fields enables interrupt generation. The

corresponding field in Interrupt Status (INT_STATUS) generates an interrupt when the corresponding field in Interrupt Status Enable (INT_STAT_EN) is 1.

Offset

Register	Offset
INT_SIG_EN	28h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 SYSTEM_CACHE_ERRS_OVER_INT_SIG_EN	System Cache Access Multiple Single-Bit ECC Error Interrupt Signal Enable Selects INT_STATUS[SYSTEM_CACHE_ECC_ERRS_OVER_INT] as one of the interrupt statuses available to interrupt the host system. 0 - Mask 1 - Enable
14 SYSTEM_CACHE_ERRM_OVER_INT_SIG_EN	System Cache Access Multiple Multibit ECC Error Interrupt Signal Enable Selects INT_STATUS[SYSTEM_CACHE_ECC_ERRM_OVER_INT] as one of the interrupt statuses available to interrupt the host system. 0 - Masked 1 - Enabled
13 SYSTEM_CACHE_ERRS_INT_SIG_EN	System Cache Access Single-Bit ECC Error Interrupt Signal Enable Selects INT_STATUS[SYSTEM_CACHE_ECC_ERRS_INT] as one of the interrupt statuses available to interrupt the host system. 0 - Mask

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable
12 SYSTEM_CACHE_ERRM_INT_SIG_EN	System Cache Access Multibit ECC Error Interrupt Signal Enable Selects INT_STATUS[SYSTEM_CACHE_ECC_ERRM_INT] as one of the interrupt statuses available to interrupt the host system. 0 - Mask 1 - Enable
11 CODE_CACHE_ERRS_OVER_INT_SIG_EN	Code Cache Access Multiple Single-Bit ECC Error Interrupt Signal Enable Selects INT_STATUS[CODE_CACHE_ECC_ERRS_OVER_INT] as one of the interrupt statuses available to interrupt the host system. 0 - Mask 1 - Enable
10 CODE_CACHE_ERRM_OVER_INT_SIG_EN	Code Cache Access Multiple Multibit ECC Error Interrupt Signal Enable Selects INT_STATUS[CODE_CACHE_ECC_ERRM_OVER_INT] as one of the interrupt statuses available to interrupt the host system. 0 - Mask 1 - Enable
9 CODE_CACHE_ERRS_INT_SIG_EN	Code Cache Access Single-Bit ECC Error Interrupt Signal Enable Selects INT_STATUS[CODE_CACHE_ECC_ERRS_INT] as one of the interrupt statuses available to interrupt the host system. 0 - Mask 1 - Enable
8 CODE_CACHE_ERRM_INT_SIG_EN	Code Cache Access Multibit ECC Error Interrupt Signal Enable Selects INT_STATUS[CODE_CACHE_ECC_ERRM_INT] as one of the interrupt statuses available to interrupt the host system. 0 - Mask 1 - Enable
7-0 —	Reserved

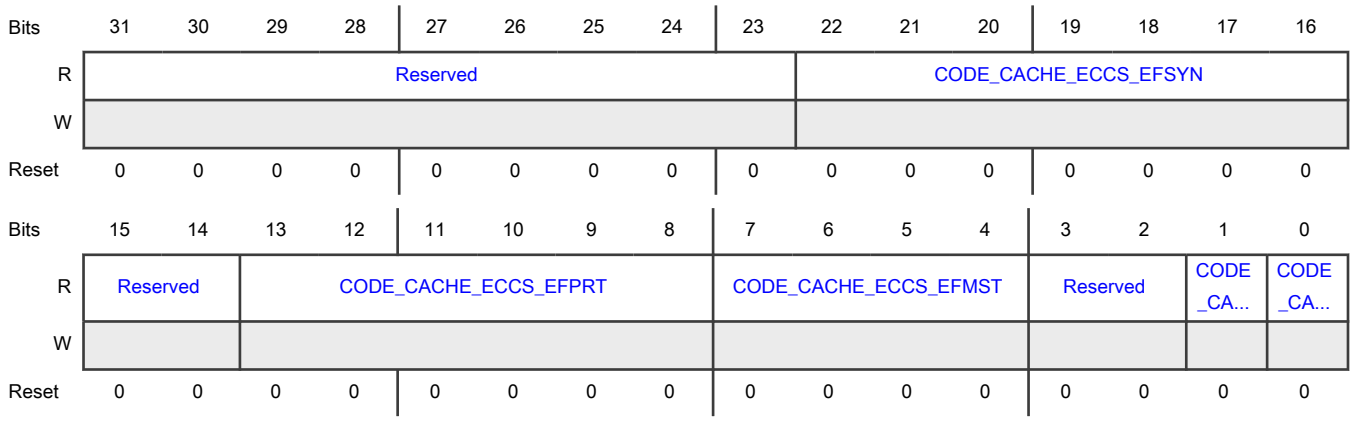
13.6.1.6 Code Cache Single-Bit ECC Error Information (CODE_CACHE_ECC_SINGLE_ERROR_INFO)

Stores detailed error information for the corresponding code cache single-bit ECC interrupt when the interrupt is enabled.

Offset

Register	Offset
CODE_CACHE_ECC_SINGLE_ERROR_INFO	5Ch

Diagram



Fields

Field	Description
31-23 —	Reserved
22-16 CODE_CACHE_ECCS_EFSYN	Code Cache Single-Bit ECC Error Corresponding Syndrome Indicates the corresponding syndrome when a single-bit error is reported. You cannot use the syndrome for single-bit ECC to locate the error bit.
15-14 —	Reserved
13-8 CODE_CACHE_ECCS_EFPRT	Code Cache Single-Bit ECC Error Protection Indicates the master access protection attribute of the access that created the error when a single-bit error is reported. For details about the protection attribute decoding, see the Arm AMBA AHB specification.
7-4 CODE_CACHE_ECCS_EFMST	Code Cache Single-Bit ECC Error Master Number Indicates the ECC error master number when a single ECC error is reported.
3-2 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 CODE_CACHE_ECCS_CMD	Code Cache Single-Bit ECC Error on Cache Command Indicates a code cache single-bit ECC error on the cache command when a single ECC error is reported. 0 - No error 1 - Error
0 CODE_CACHE_ECCS_TAG	Code Cache Single-Bit ECC Error Indicates whether the single-bit ECC error is in the cache tag or the cache data. 0 - Data 1 - Tag

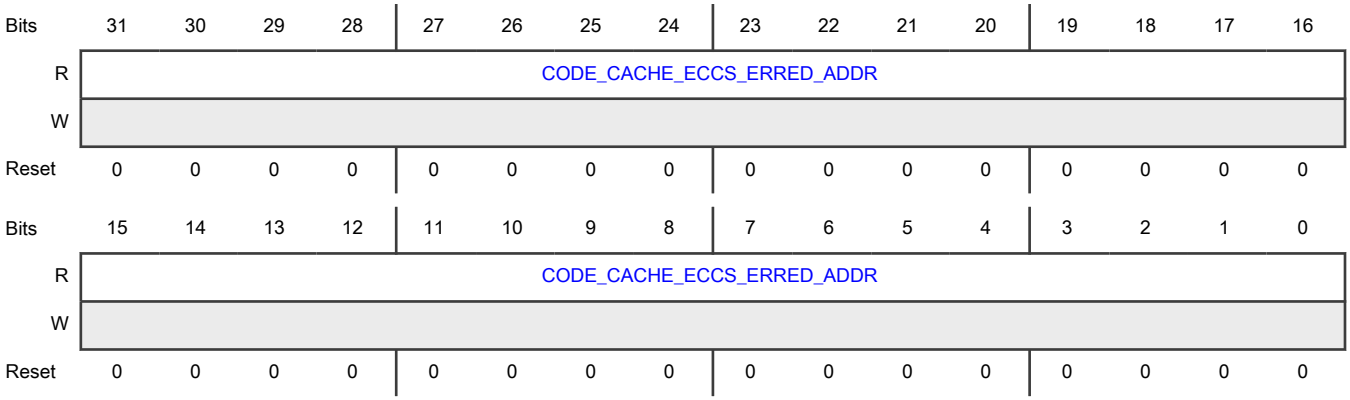
13.6.1.7 Code Cache Single-Bit ECC Error Address (CODE_CACHE_ECC_SINGLE_ERROR_ADDR)

Stores the error address for a single-bit ECC interrupt.

Offset

Register	Offset
CODE_CACHE_ECC_SINGLE_ERROR_ADDR	60h

Diagram



Fields

Field	Description
31-0 CODE_CACHE_ECCS_ERRED_ADDR	Code Cache Single-Bit ECC Error Address Stores the error address for the corresponding code cache single-bit ECC interrupt when the interrupt is enabled.

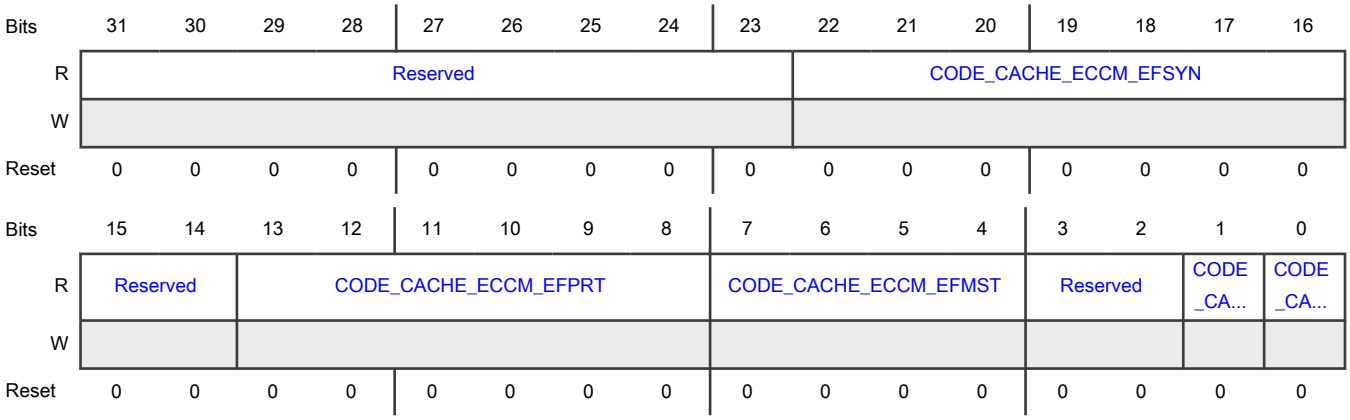
13.6.1.8 Code Cache Multibit ECC Error Information (CODE_CACHE_ECC_MULTI_ERROR_INFO)

Stores detailed error information for the corresponding code cache multibit ECC interrupt when the interrupt is enabled.

Offset

Register	Offset
CODE_CACHE_ECC_MULTI_ERROR_INFO	68h

Diagram



Fields

Field	Description
31-23 —	Reserved
22-16 CODE_CACHE_ECCM_EFSYN	Code Cache Multibit ECC Error Corresponding Syndrome Indicates the corresponding syndrome when a multibit error is reported. You cannot use the syndrome for multibit ECC to locate the error bit.
15-14 —	Reserved
13-8 CODE_CACHE_ECCM_EFPRT	Code Cache Multibit ECC Error Protection Indicates the master access protection attribute of the access creating the error when a multibit error is reported. For details about the protection attribute decoding, see the Arm AMBA AHB specification.
7-4	Code Cache Multibit ECC Error Master Number Indicates the ECC error master number when a multibit ECC error is reported.

Table continues on the next page...

Table continued from the previous page...

Field	Description
CODE_CACHE_ECCM_EFMST	
3-2 —	Reserved
1 CODE_CACHE_ECCM_CMD	Code Cache Multibit ECC Error on Code Cache Command Indicates whether the error has occurred on a code cache command when a multibit ECC error is reported. 0 - No error 1 - Error
0 CODE_CACHE_ECCM_TAG	Code Cache Multibit ECC Error Indicates whether the multibit ECC error is in the cache tag or the cache data. 0 - Data 1 - Tag

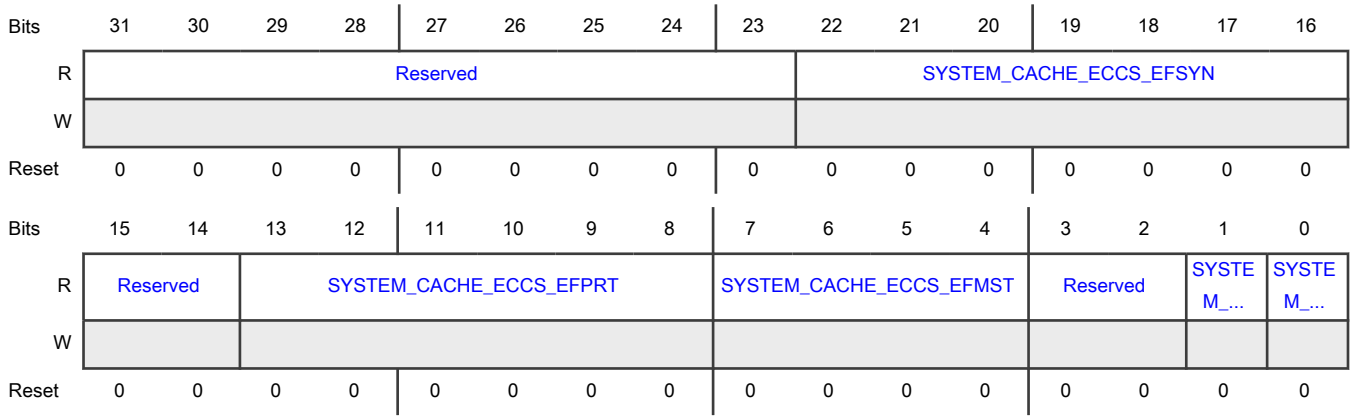
13.6.1.9 System Cache Single-Bit ECC Error Information (SYSTEM_CACHE_ECC_SINGLE_ERROR_INFO)

Stores error information for the corresponding system cache single-bit ECC interrupt when the interrupt is enabled.

Offset

Register	Offset
SYSTEM_CACHE_ECC_SINGLE_ERROR_INFO	74h

Diagram



Fields

Field	Description
31-23 —	Reserved
22-16 SYSTEM_CACHE_ECCS_EFSYN	System Cache Single-Bit ECC Error Corresponding Syndrome Indicates the corresponding syndrome when a single-bit error is reported. You cannot use the syndrome for single-bit ECC to locate the error bit.
15-14 —	Reserved
13-8 SYSTEM_CACHE_ECCS_EFPRT	System Cache Single-Bit ECC Error Protection Indicates the master access protection attribute of the access creating the error when a single-bit error is reported. For details about the protection attribute decoding, see the Arm AMBA AHB specification.
7-4 SYSTEM_CACHE_ECCS_EFMST	System Cache Single-Bit ECC Error Master Number Indicates the ECC error master number when a single-bit ECC error is reported.
3-2 —	Reserved
1 SYSTEM_CACHE_ECCS_CMD	System Cache Single-Bit ECC Error on Cache Command Indicates whether a system cache single-bit ECC error has occurred on the cache command when a single ECC error is reported. 0 - No error 1 - Error
0 SYSTEM_CACHE_ECCS_TAG	System Cache Single-Bit ECC Error Indicates whether the single-bit ECC error is in the system tag or the system data. 0 - Data 1 - Tag

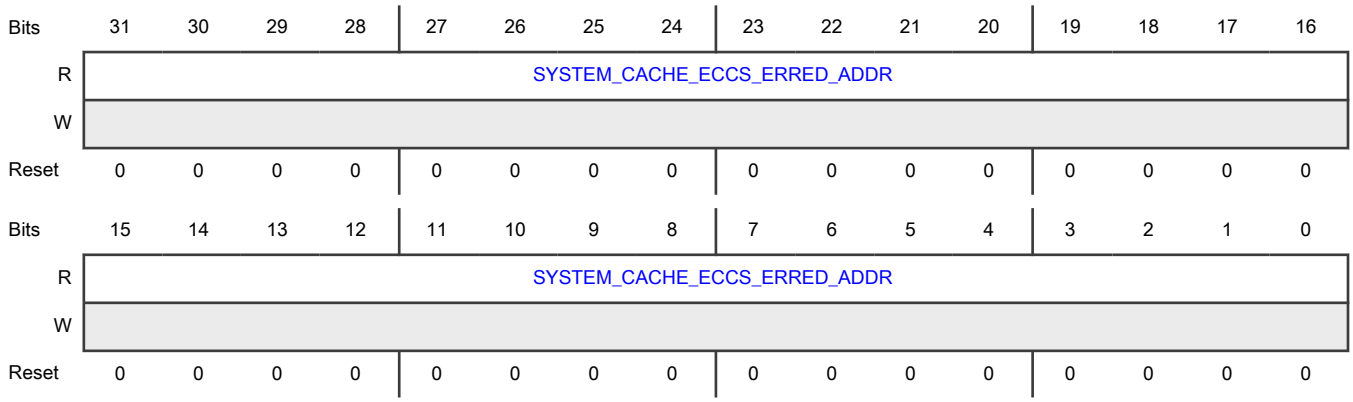
13.6.1.10 System Cache Single-Bit ECC Error Address (SYSTEM_CACHE_ECC_SINGLE_ERROR_ADDR)

Stores the error address for a system cache single-bit ECC interrupt.

Offset

Register	Offset
SYSTEM_CACHE_ECC_SINGLE_ERROR_ADDR	78h

Diagram



Fields

Field	Description
31-0 SYSTEM_CACHE_ECCS_ERRED_ADDR	System Cache Single-Bit ECC Error Address Stores the error address for the corresponding system cache single-bit ECC interrupt when the interrupt is enabled.

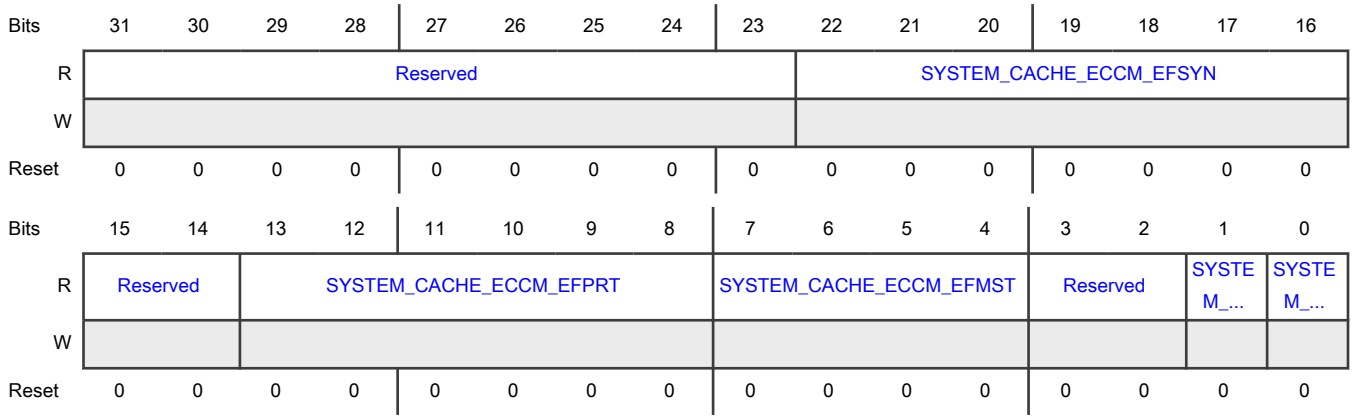
13.6.1.11 System Cache Multibit ECC Error Information (SYSTEM_CACHE_ECC_MULTI_ERROR_INFO)

Stores error information for the corresponding system cache multibit ECC interrupt when the interrupt is enabled.

Offset

Register	Offset
SYSTEM_CACHE_ECC_MULTI_ERROR_INFO	80h

Diagram



Fields

Field	Description
31-23 —	Reserved
22-16 SYSTEM_CACHE_ECCM_EFSYN	System Cache Multibit ECC Error Corresponding Syndrome Indicates the corresponding syndrome when a multibit error is reported. You cannot use the syndrome for multibit ECC to locate the error bit.
15-14 —	Reserved
13-8 SYSTEM_CACHE_ECCM_EFPRT	System Cache Multibit ECC Error Protection Indicates the master access protection attribute of the access that created the error when a multibit error is reported. For details about the protection attribute decoding, refer to the Arm AMBA AHB specification.
7-4 SYSTEM_CACHE_ECCM_EFMST	System Cache Multibit ECC Error Master Number Indicates the ECC error master number when a multibit ECC error is reported.
3-2 —	Reserved
1 SYSTEM_CACHE_ECCM_CMD	System Cache Multibit ECC Error on System Cache Command Indicates whether a system cache multibit ECC error has occurred on a cache command when a multibit ECC error is reported. 0 - No error 1 - Error

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 SYSTEM_CACHE_ECCM_TAG	System Cache Multibit ECC Error Indicates whether the multibit ECC error is in the system tag or the system data. 0 - Data 1 - Tag

13.6.1.12 System Cache Multibit ECC Error Data (SYSTEM_CACHE_ECC_MULTI_ERROR_DATA)

Stores error data for the system cache multibit ECC interrupt.

Offset

Register	Offset
SYSTEM_CACHE_ECC_MULTI_ERROR_DATA	84h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SYSTEM_CACHE_ECCM_ERRED_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SYSTEM_CACHE_ECCM_ERRED_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SYSTEM_CACHE_ECCM_ERRED_DATA	System Cache Multibit ECC Error Data Stores error data for the corresponding system cache multibit ECC interrupt when the interrupt is enabled.

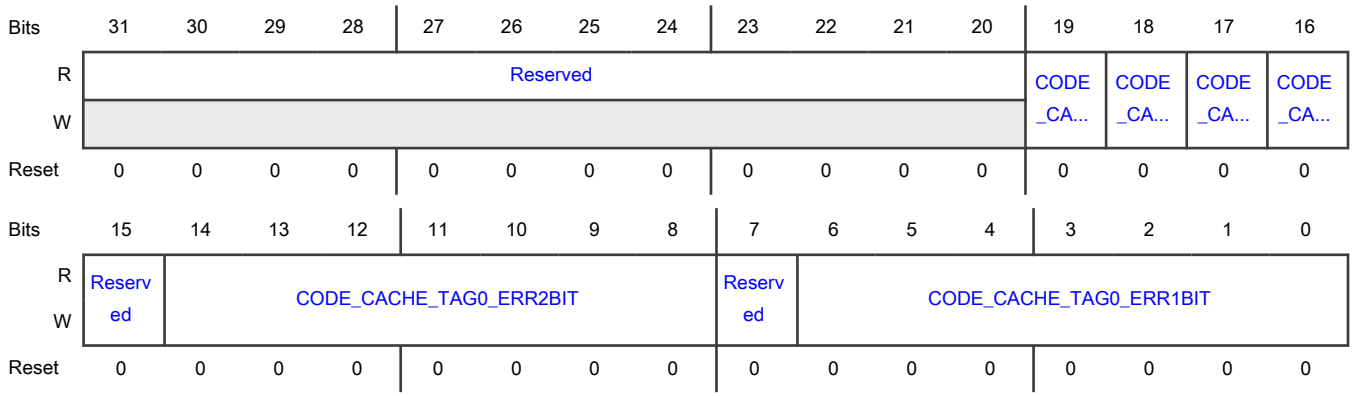
13.6.1.13 Code Cache TAG0 ECC Error Injection (CODE_CACHE_TAG0_ECC_ERROR_INJEC)

Injects an ECC error into code cache TAG0. This register is used only for verification and debugging.

Offset

Register	Offset
CODE_CACHE_TAG0_ECC_ERROR_INJEC	8Ch

Diagram



Fields

Field	Description
31-20 —	Reserved
19 CODE_CACHE_TAG0_FRCNCI	Force Continuous Noncorrectable Data Inversions on Code Cache TAG0 Write Access Injects a multibit ECC error on the bits specified by CODE_CACHE_TAG0_ERR1BIT and CODE_CACHE_TAG0_ERR2BIT into every write access to code cache TAG0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
18 CODE_CACHE_TAG0_FRC1BI	Force Continuous 1-Bit Data Inversions on Code Cache TAG0 Write Access Injects a single-bit ECC error on the bit specified by CODE_CACHE_TAG0_ERR1BIT into every write access to code cache TAG0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
17 CODE_CACHE_TAG0_FR1NCI	Force One Noncorrectable Data Inversion on Code Cache TAG0 Write Access Injects a multibit ECC error on the bits specified by CODE_CACHE_TAG0_ERR1BIT and CODE_CACHE_TAG0_ERR2BIT into the first write access to code cache TAG0 that occurs after you write 1 to this field. 0 - Disable injection

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable injection
16 CODE_CACHE_TAG0_FR11BI	Force One 1-Bit Data Inversion on Code Cache TAG0 Write Access Injects a single-bit ECC error on the bit specified by CODE_CACHE_TAG0_ERR1BIT into the first write access to code cache TAG0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
15 —	Reserved
14-8 CODE_CACHE_TAG0_ERR2BIT	Position of Second Bit to Inject ECC Error Determines the position of the second bit into which CM33 injects an ECC error. This field is used only for multibit ECC error injection.
7 —	Reserved
6-0 CODE_CACHE_TAG0_ERR1BIT	Position of First Bit to Inject ECC Error Determines the position of the first bit into which CM33 injects an ECC error. This field can be used for both single-bit and multibit ECC error injection.

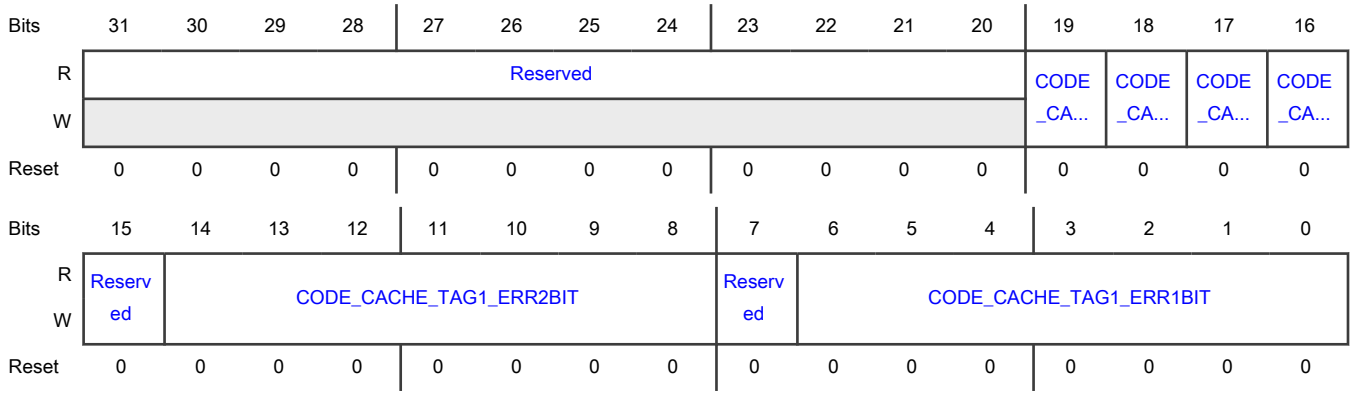
13.6.1.14 Code Cache TAG1 ECC Error Injection (CODE_CACHE_TAG1_ECC_ERROR_INJEC)

Injects an ECC error into code cache TAG1. This register is used only for verification and debugging.

Offset

Register	Offset
CODE_CACHE_TAG1_ECC_ERROR_INJEC	90h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 CODE_CACHE_TAG1_FRCNCI	Force Continuous Noncorrectable Data Inversions on Code Cache TAG1 Write Access Injects a multibit ECC error on the bits specified by CODE_CACHE_TAG1_ERR1BIT and CODE_CACHE_TAG1_ERR2BIT into every write access to code cache TAG1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
18 CODE_CACHE_TAG1_FRC1BI	Force Continuous 1-Bit Data Inversions on Code Cache TAG1 Write Access Injects a single-bit ECC error on the bit specified by CODE_CACHE_TAG1_ERR1BIT into every write access to code cache TAG1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
17 CODE_CACHE_TAG1_FR1NCI	Force One Noncorrectable Data Inversion on Code Cache TAG1 Write Access Injects a multibit ECC error on the bits specified by CODE_CACHE_TAG1_ERR1BIT and CODE_CACHE_TAG1_ERR2BIT into the first write access to code cache TAG1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
16 CODE_CACHE_TAG1_FR11BI	Force One 1-Bit Data Inversion on Code Cache TAG1 Write Access Injects a single-bit ECC error on the bit specified by CODE_CACHE_TAG1_ERR1BIT into the first write access to code cache TAG1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection

Table continues on the next page...

Table continued from the previous page...

Field	Description
15 —	Reserved
14-8 CODE_CACHE_TAG1_ERR2BIT	Position of Second Bit to Inject ECC Error Determines the position of the second bit into which CM33 injects an ECC error. This field is used only for multibit ECC error injection.
7 —	Reserved
6-0 CODE_CACHE_TAG1_ERR1BIT	Position of First Bit to Inject ECC Error Determines the position of the first bit into which CM33 injects an ECC error. This field can be used for both single-bit and multibit ECC error injection.

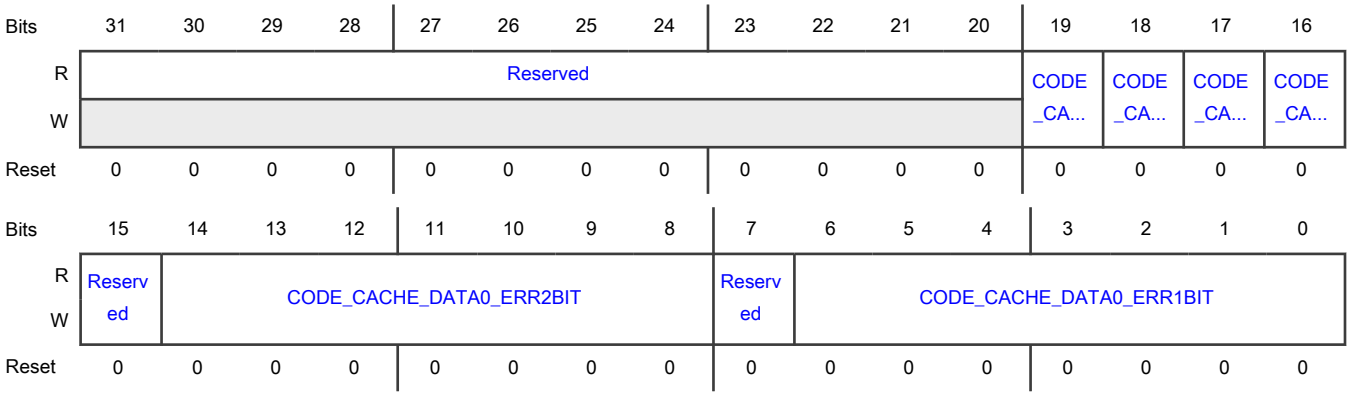
13.6.1.15 Code Cache DATA0 ECC Error Injection (CODE_CACHE_DATA0_ECC_ERROR_INJEC)

Injects an ECC error into code cache DATA0. This register is used only for verification and debugging.

Offset

Register	Offset
CODE_CACHE_DATA0_ECC_ERROR_INJEC	94h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 CODE_CACHE_DATA0_FRCNCI	Force Continuous Noncorrectable Data Inversions on Code Cache DATA0 Write Access Injects a multibit ECC error on the bits specified by CODE_CACHE_DATA0_ERR1BIT and CODE_CACHE_DATA0_ERR2BIT into every write access to code cache DATA0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
18 CODE_CACHE_DATA0_FRC1BI	Force Continuous 1-Bit Data Inversions on Code Cache DATA0 Write Access Injects a single-bit ECC error on the bit specified by CODE_CACHE_DATA0_ERR1BIT into every write access to code cache DATA0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
17 CODE_CACHE_DATA0_FR1NCI	Force One Noncorrectable Data Inversion on Code Cache DATA0 Write Access Injects a multibit ECC error on the bits specified by CODE_CACHE_DATA0_ERR1BIT and CODE_CACHE_DATA0_ERR2BIT into the first write access to code cache DATA0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
16 CODE_CACHE_DATA0_FR11BI	Force One 1-Bit Data Inversion on Code Cache DATA0 Write Access Injects a single-bit ECC error on the bit specified by CODE_CACHE_DATA0_ERR1BIT into the first write access to code cache DATA0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
15 —	Reserved
14-8 CODE_CACHE_DATA0_ERR2BIT	Position of Second Bit to Inject ECC Error Determines the position of the second bit into which CM33 injects an ECC error. This field is used only for multibit ECC error injection.
7 —	Reserved
6-0	Position of First Bit to Inject ECC Error

Table continues on the next page...

Table continued from the previous page...

Field	Description
CODE_CACHE_DATA0_ERR1BIT	Determines the position of the first bit into which CM33 injects an ECC error. This field can be used for both single-bit and multibit ECC error injection.

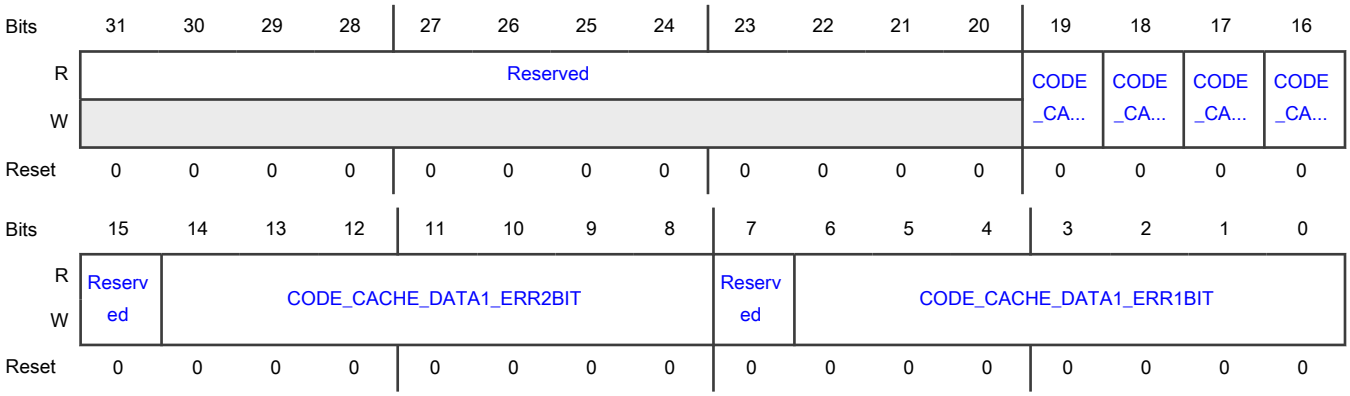
13.6.1.16 Code Cache DATA1 ECC Error Injection (CODE_CACHE_DATA1_ECC_ERROR_INJEC)

Injects an ECC error into code cache DATA1. This register is used only for verification and debugging.

Offset

Register	Offset
CODE_CACHE_DATA1_ECC_ERROR_INJEC	98h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 CODE_CACHE_DATA1_FRCNCI	Force Continuous Noncorrectable Data Inversions on Code Cache DATA1 Write Access Injects a multibit ECC error on the bits specified by CODE_CACHE_DATA1_ERR1BIT and CODE_CACHE_DATA1_ERR2BIT into every write access to code cache DATA1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
18	Force Continuous 1-Bit Data Inversions on Code Cache DATA1 Write Access

Table continues on the next page...

Table continued from the previous page...

Field	Description
CODE_CACHE_DATA1_FRC1BI	Injects a single-bit ECC error on the bit specified by CODE_CACHE_DATA1_ERR1BIT into every write access to code cache DATA1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
17 CODE_CACHE_DATA1_FR1NCI	Force One Noncorrectable Data Inversion on Code Cache DATA1 Write Access Injects a multibit ECC error on the bits specified by CODE_CACHE_DATA1_ERR1BIT and CODE_CACHE_DATA1_ERR2BIT into the first write access to code cache DATA1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
16 CODE_CACHE_DATA1_FR11BI	Force One 1-Bit Data Inversion on Code Cache DATA1 Write Access Injects a single-bit ECC error on the bit specified by CODE_CACHE_DATA1_ERR1BIT into the first write access to code cache DATA1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
15 —	Reserved
14-8 CODE_CACHE_DATA1_ERR2BIT	Position of Second Bit to Inject ECC Error Determines the position of the second bit into which CM33 injects an ECC error. This field is used only for multibit ECC error injection.
7 —	Reserved
6-0 CODE_CACHE_DATA1_ERR1BIT	Position of First Bit to Inject ECC Error Determines the position of the first bit into which CM33 injects an ECC error. This field can be used for both single-bit and multibit ECC error injection.

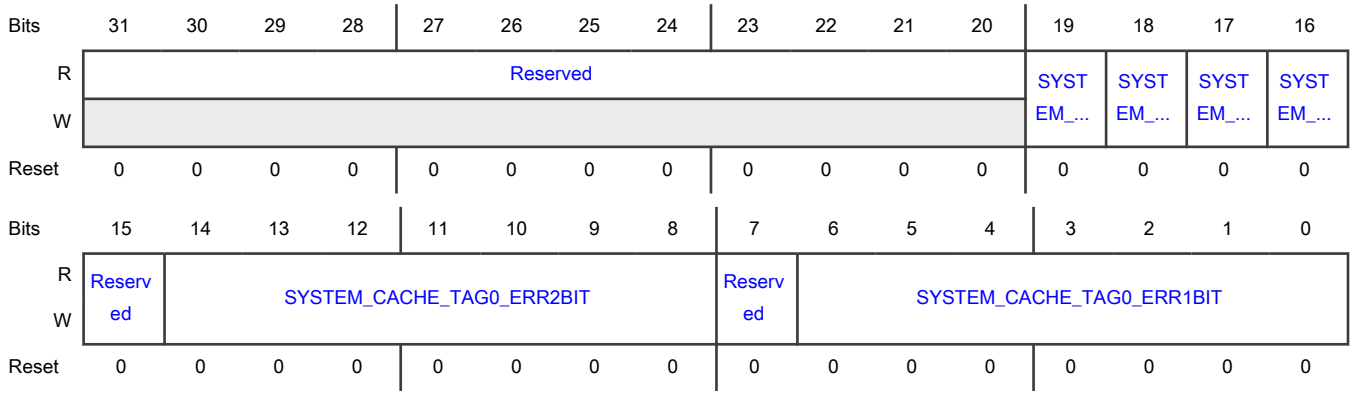
13.6.1.17 System Cache TAG0 ECC Error Injection (SYTEM_CACHE_TAG0_ECC_ERROR_INJEC)

Injects an ECC error into system cache TAG0. This register is used only for verification and debugging.

Offset

Register	Offset
SYTEM_CACHE_TAG0_ECC_ERROR_INJEC	9Ch

Diagram



Fields

Field	Description
31-20 —	Reserved
19 SYSTEM_CACHE_TAG0_FRCNCI	Force Continuous Noncorrectable Data Inversions on System Cache TAG0 Write Access Injects a multibit ECC error on the bits specified by SYSTEM_CACHE_TAG0_ERR1BIT and SYSTEM_CACHE_TAG0_ERR2BIT into every write access to system cache DATA1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
18 SYSTEM_CACHE_TAG0_FRC1BI	Force Continuous 1-Bit Data Inversions on System Cache TAG0 Write Access Injects a single-bit ECC error on the bit specified by SYSTEM_CACHE_TAG0_ERR1BIT into every write access to system cache TAG0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
17 SYSTEM_CACHE_TAG0_FR1NCI	Force One Noncorrectable Data Inversion on System Cache TAG0 Write Access Injects a multibit ECC error on the bits specified by SYSTEM_CACHE_TAG0_ERR1BIT and SYSTEM_CACHE_TAG0_ERR2BIT into the first write access to system cache TAG0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
16 SYSTEM_CACHE_TAG0_FR11BI	Force One 1-Bit Data Inversion on System Cache TAG0 Write Access Injects a single-bit ECC error on the bit specified by SYSTEM_CACHE_TAG0_ERR1BIT into the first write access to system cache TAG0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection

Table continues on the next page...

Table continued from the previous page...

Field	Description
15 —	Reserved
14-8 SYSTEM_CACHE_TAG0_ERR2BIT	Position of Second Bit to Inject ECC Error Determines the position of the second bit into which CM33 injects an ECC error. This field is used only for multibit ECC error injection.
7 —	Reserved
6-0 SYSTEM_CACHE_TAG0_ERR1BIT	Position of First Bit to Inject ECC Error Determines the position of the first bit into which CM33 injects an ECC error. This field can be used for both single-bit and multibit ECC error injection.

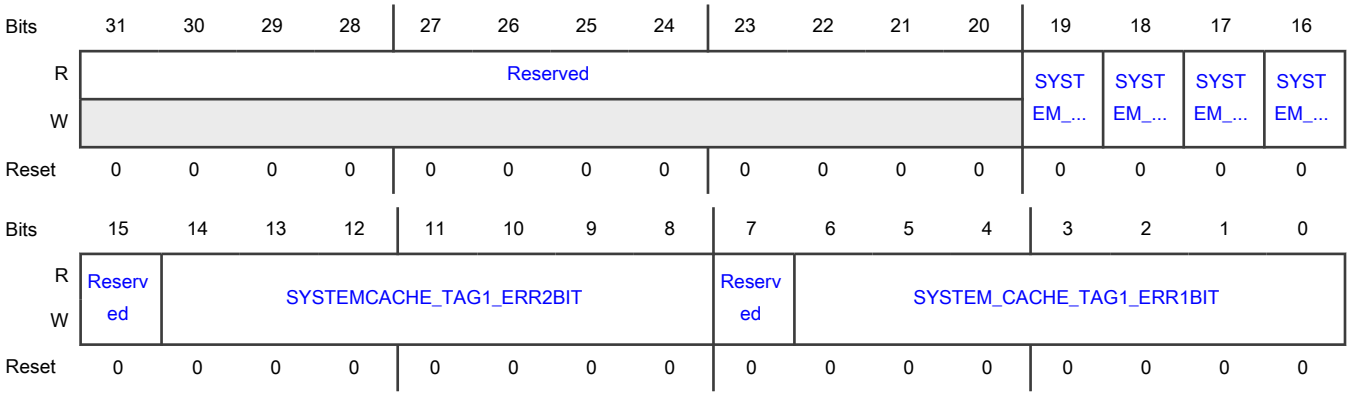
13.6.1.18 System Cache TAG1 ECC Error Injection (SYSTEM_CACHE_TAG1_ECC_ERROR_INJEC)

Injects an ECC error into system cache TAG1. This register is used only for verification and debugging.

Offset

Register	Offset
SYSTEM_CACHE_TAG1_ECC_ERROR_INJEC	A0h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 SYSTEM_CACHE_TAG1_FRCNCI	Force Continuous Noncorrectable Data Inversions on System Cache TAG1 Write Access Injects a multibit ECC error on the bits specified by SYSTEM_CACHE_TAG1_ERR1BIT and SYSTEM_CACHE_TAG1_ERR2BIT into every write access to system cache TAG1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
18 SYSTEM_CACHE_TAG1_FRC1BI	Force Continuous 1-Bit Data Inversions on System Cache TAG1 Write Access Injects a single-bit ECC error on the bit specified by SYSTEM_CACHE_TAG1_ERR1BIT into every write access to system cache TAG1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
17 SYSTEM_CACHE_TAG1_FR1NCI	Force One Noncorrectable Data Inversion on System Cache TAG1 Write Access Injects a multibit ECC error on the bits specified by SYSTEM_CACHE_TAG1_ERR1BIT and SYSTEM_CACHE_TAG1_ERR2BIT into the first write access to system cache TAG1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
16 SYSTEM_CACHE_TAG1_FR11BI	Force One 1-Bit Data Inversion on System Cache TAG1 Write Access Injects a single-bit ECC error on the bit specified by SYSTEM_CACHE_TAG1_ERR1BIT into the first write access to system cache TAG1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
15 —	Reserved
14-8 SYSTEMCACHE_TAG1_ERR2BIT	Position of Second Bit to Inject ECC Error Determines the position of the second bit into which CM33 injects an ECC error. This field is used only for multibit ECC error injection.
7 —	Reserved
6-0	Position of First Bit to Inject ECC Error

Table continues on the next page...

Table continued from the previous page...

Field	Description
SYSTEM_CACHE_TAG1_ERR1BIT	Determines the position of the first bit into which CM33 injects an ECC error. This field can be used for both single-bit and multibit ECC error injection.

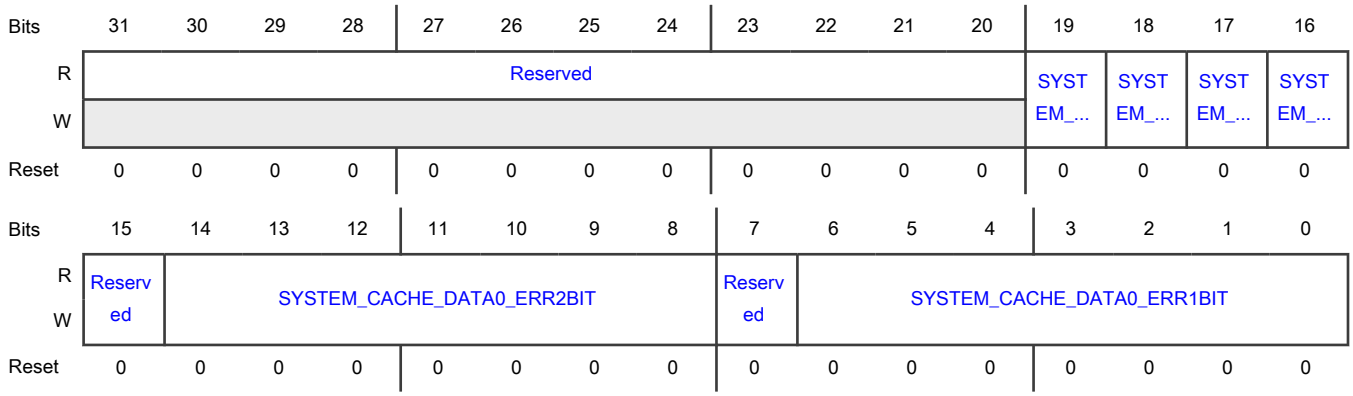
13.6.1.19 System Cache DATA0 ECC Error Injection (SYSTEM_CACHE_DATA0_ECC_ERROR_INJEC)

Injects an ECC error into system cache DATA0. This register is used only for verification and debugging.

Offset

Register	Offset
SYSTEM_CACHE_DATA0_ECC_ERROR_INJEC	A4h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 SYSTEM_CACHE_DATA0_FRCNCI	Force Continuous Noncorrectable Data Inversions on System Cache DATA0 Write Access Injects a multibit ECC error on the bits specified by SYSTEM_CACHE_DATA0_ERR1BIT and SYSTEM_CACHE_DATA0_ERR2BIT into every write access to system cache DATA0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection

Table continues on the next page...

Table continued from the previous page...

Field	Description
18 SYSTEM_CACHE_DATA0_FRC1BI	Force Continuous 1-Bit Data Inversions on System Cache DATA0 Write Access Injects a single-bit ECC error on the bit specified by SYSTEM_CACHE_DATA0_ERR1BIT into every write access to system cache DATA0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
17 SYSTEM_CACHE_DATA0_FR1NCI	Force One Noncorrectable Data Inversion on System Cache DATA0 Write Access Injects a multibit ECC error on the bits specified by SYSTEM_CACHE_DATA0_ERR1BIT and SYSTEM_CACHE_DATA0_ERR2BIT into the first write access to system cache DATA0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
16 SYSTEM_CACHE_DATA0_FR11BI	Force One 1-Bit Data Inversion on System Cache DATA0 Write Access Injects a single-bit ECC error on the bit specified by SYSTEM_CACHE_DATA0_ERR1BIT into the first write access to system cache DATA0 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
15 —	Reserved
14-8 SYSTEM_CACHE_DATA0_ERR2BIT	Position of Second Bit to Inject ECC Error Determines the position of the second bit into which CM33 injects an ECC error. This field is used only for multibit ECC error injection.
7 —	Reserved
6-0 SYSTEM_CACHE_DATA0_ERR1BIT	Position of First Bit to Inject ECC Error Determines the position of the first bit into which CM33 injects an ECC error. This field can be used for both single-bit and multibit ECC error injection.

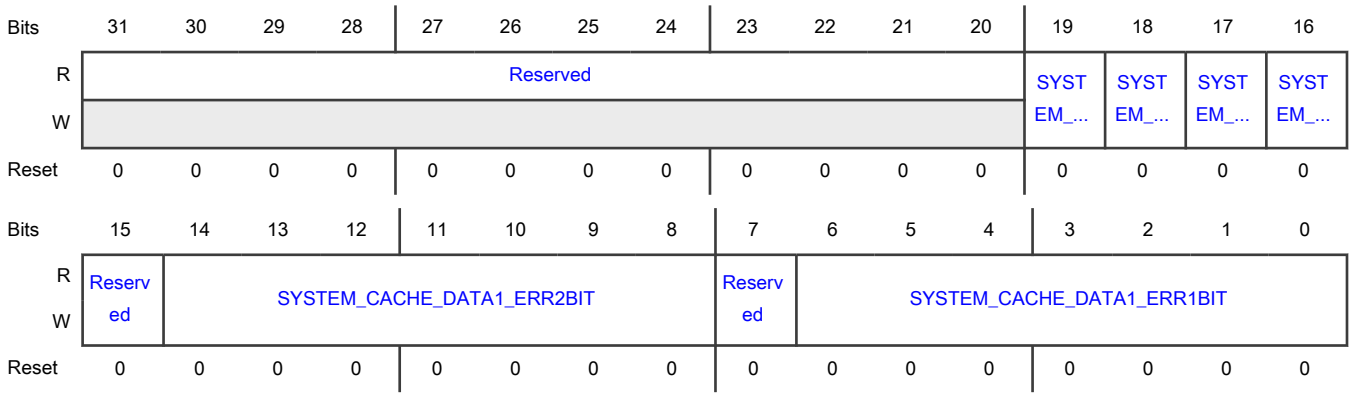
13.6.1.20 System Cache DATA1 ECC Error Injection (STSTEM_CACHE_DATA1_ECC_ERROR_INJEC)

Injects an ECC error into system cache DATA1. This register is used only for verification and debugging.

Offset

Register	Offset
STSTEM_CACHE_DATA1_ECC_ERROR_INJEC	A8h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 SYSTEM_CACHE_DATA1_FRCNCI	Force Continuous Noncorrectable Data Inversions on System Cache DATA1 Write Access Injects a multibit ECC error on the bits specified by SYSTEM_CACHE_DATA1_ERR1BIT and SYSTEM_CACHE_DATA1_ERR2BIT into every write access to system cache DATA1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
18 SYSTEM_CACHE_DATA1_FRC1BI	Force Continuous 1-Bit Data Inversions on System Cache DATA1 Write Access Injects a single-bit ECC error on the bit specified by SYSTEM_CACHE_DATA1_ERR1BIT into every write access to system cache DATA1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
17 SYSTEM_CACHE_DATA1_FR1NCI	Force One Noncorrectable Data Inversion on System Cache DATA1 Write Access Injects a multibit ECC error on the bits specified by SYSTEM_CACHE_DATA1_ERR1BIT and SYSTEM_CACHE_DATA1_ERR2BIT into the first write access to system cache DATA1 that occurs after you write 1 to this field. 0 - Disable injection

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable injection
16 SYSTEM_CACHE_DATA1_FR11BI	Force One 1-Bit Data Inversion on System Cache DATA1 Write Access Injects a single-bit ECC error on the bit specified by SYSTEM_CACHE_DATA1_ERR1BIT into the first write access to system cache DATA1 that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
15 —	Reserved
14-8 SYSTEM_CACHE_DATA1_ERR2BIT	Position of Second Bit to Inject ECC Error Determines the position of the second bit into which CM33 injects an ECC error. This field is used only for multibit ECC error injection.
7 —	Reserved
6-0 SYSTEM_CACHE_DATA1_ERR1BIT	Position of First Bit to Inject ECC Error Determines the position of the first bit into which CM33 injects an ECC error. This field can be used for both single-bit and multibit ECC error injection.

13.6.2 CM33_TCM_MCM register descriptions

13.6.2.1 CM33_TCM_MCM memory map

CM33_TCM_MCM base address: 4442_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	TCM ECC Control (TCMECCR)	32	RW	0000_0000
20	Interrupt Status (INT_STATUS)	32	RW	0000_0000
24	Interrupt Status Enable (INT_STAT_EN)	32	RW	0000_0000
28	Interrupt Enable (INT_SIG_EN)	32	RW	0000_0000
5C	Code TCM Single-Bit ECC Error Information (CODE_TCM_ECC_SINGLE_ERROR_INFO)	32	R	0000_0000
60	Code TCM Single-Bit ECC Error Address (CODE_TCM_ECC_SINGLE_ERROR_ADDR)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
68	Code TCM Multibit ECC Error Information (CODE_TCM_ECC_MULTI_ERROR_INFO)	32	R	0000_0000
6C	Code TCM Multibit ECC Error Address (CODE_TCM_ECC_MULTI_ERROR_ADDR)	32	R	0000_0000
74	System TCM Single-Bit ECC Error Information (SYS_TCM_ECC_SINGLE_ERROR_INFO)	32	R	0000_0000
78	System TCM Single-Bit ECC Error Address (SYS_TCM_ECC_SINGLE_ERROR_ADDR)	32	R	0000_0000
80	System TCM Multibit ECC Error Information (SYS_TCM_ECC_MULTI_ERROR_INFO)	32	R	0000_0000
84	System TCM Multibit ECC Error Address (SYS_TCM_ECC_MULTI_ERROR_ADDR)	32	R	0000_0000
94	Code TCM ECC Error Injection (CODE_TCM_ECC_ERROR_INJEC)	32	RW	0000_0000
98	System TCM ECC Error Injection (SYS_TCM_ECC_ERROR_INJEC)	32	RW	0000_0000

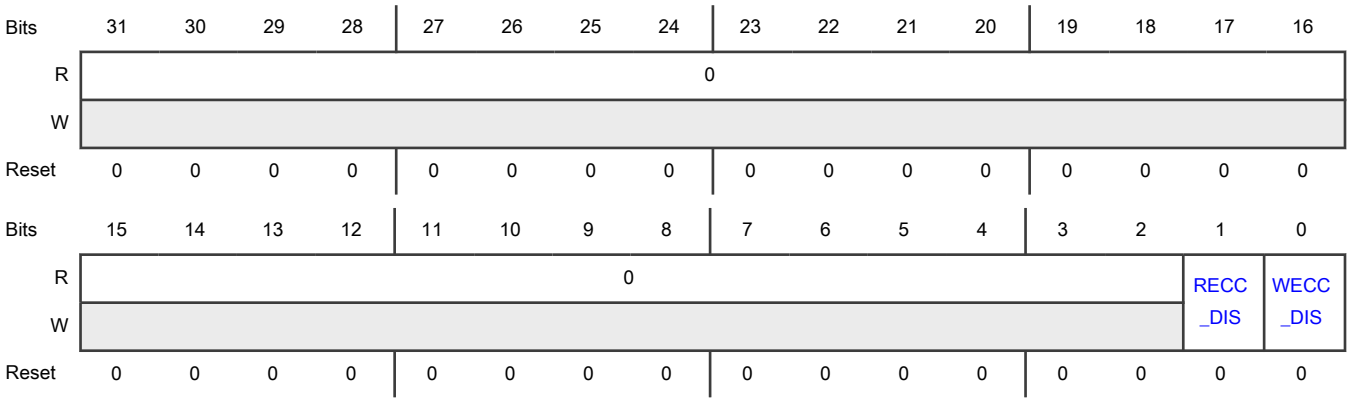
13.6.2.2 TCM ECC Control (TCMECCR)

Disables the ECC logic for Cortex-M33 TCM. TCM ECC is enabled by default.

Offset

Register	Offset
TCMECCR	4h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 RECC_DIS	TCM ECC Read Check Disable Disables the ECC check on TCM read data. 0 - Enable 1 - Disable
0 WECC_DIS	TCM ECC Write Generation Disable Disables ECC generation on TCM write data. 0 - Enable 1 - Disable

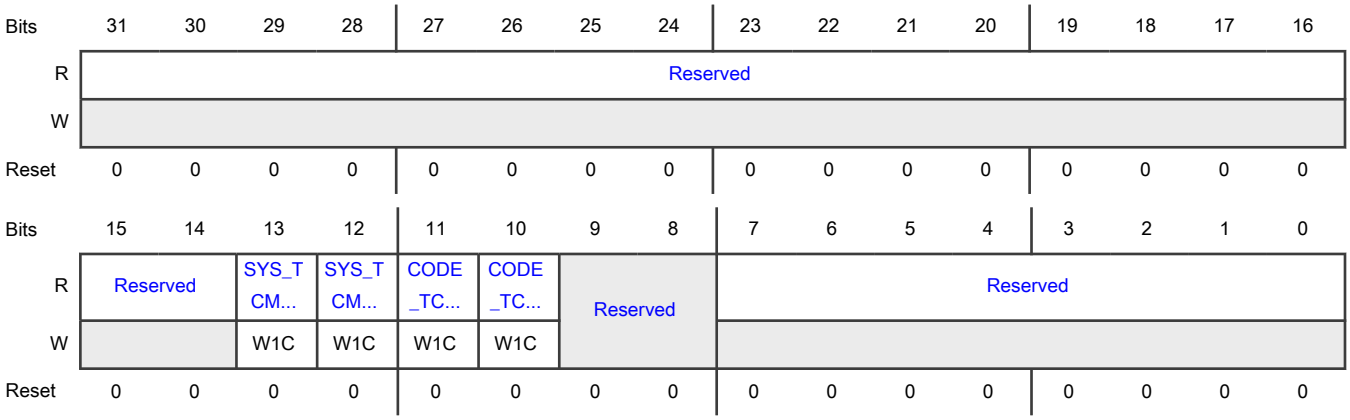
13.6.2.3 Interrupt Status (INT_STATUS)

Contains the status of ECC errors for system TCM and code TCM.

Offset

Register	Offset
INT_STATUS	20h

Diagram



Fields

Field	Description
31-14	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
13 SYS_TCM_ECC_ERRS_INT	System TCM Access Single-Bit ECC Error Interrupt Status Indicates a single-bit ECC error in system TCM read data when INT_STAT_EN[SYS_TCM_ERRS_INT_EN] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
12 SYS_TCM_ECC_ERRM_INT	System TCM Access Multibit ECC Error Interrupt Status Indicates a multibit ECC error in system TCM read data when INT_STAT_EN[SYS_TCM_ERRM_INT_EN] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
11 CODE_TCM_ECC_ERRS_INT	Code TCM Access Single-Bit ECC Error Interrupt Status Indicates a single-bit ECC error in code TCM read data when INT_STAT_EN[CODE_TCM_ERRS_INT_EN] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Clear the flag
10 CODE_TCM_ECC_ERRM_INT	Code TCM Access Multibit ECC Error Interrupt Status Indicates a multibit ECC error in code TCM read data when INT_STAT_EN[CODE_TCM_ERRM_INT_EN] = 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
9-8 —	Reserved
7-0 —	Reserved

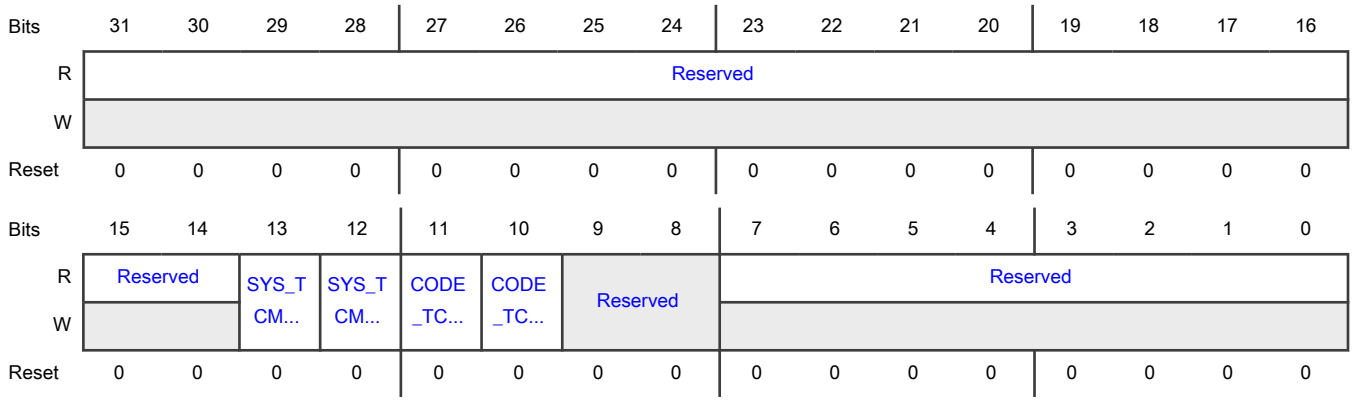
13.6.2.4 Interrupt Status Enable (INT_STAT_EN)

Enables the interrupt statuses in Interrupt Status (INT_STATUS).

Offset

Register	Offset
INT_STAT_EN	24h

Diagram



Fields

Field	Description
31-14 —	Reserved
13 SYS_TCM_ERRS_INT_EN	System TCM Access Single-Bit ECC Error Interrupt Status Enable Enables INT_STATUS[SYS_TCM_ECC_ERRS_INT]. 0 - Mask 1 - Enable
12 SYS_TCM_ERRM_INT_EN	System TCM Access Multibit ECC Error Interrupt Status Enable Enables INT_STATUS[SYS_TCM_ECC_ERRM_INT]. 0 - Mask 1 - Enable
11 CODE_TCM_ERRS_INT_EN	Code TCM Access Single-Bit ECC Error Interrupt Status Enable Enables INT_STATUS[CODE_TCM_ECC_ERRS_INT]. 0 - Mask 1 - Enable
10 CODE_TCM_ERRM_INT_EN	Code TCM Access Multibit ECC Error Interrupt Status Enable Enables INT_STATUS[CODE_TCM_ECC_ERRM_INT]. 0 - Mask 1 - Enable
9-8 —	Reserved
7-0 —	Reserved

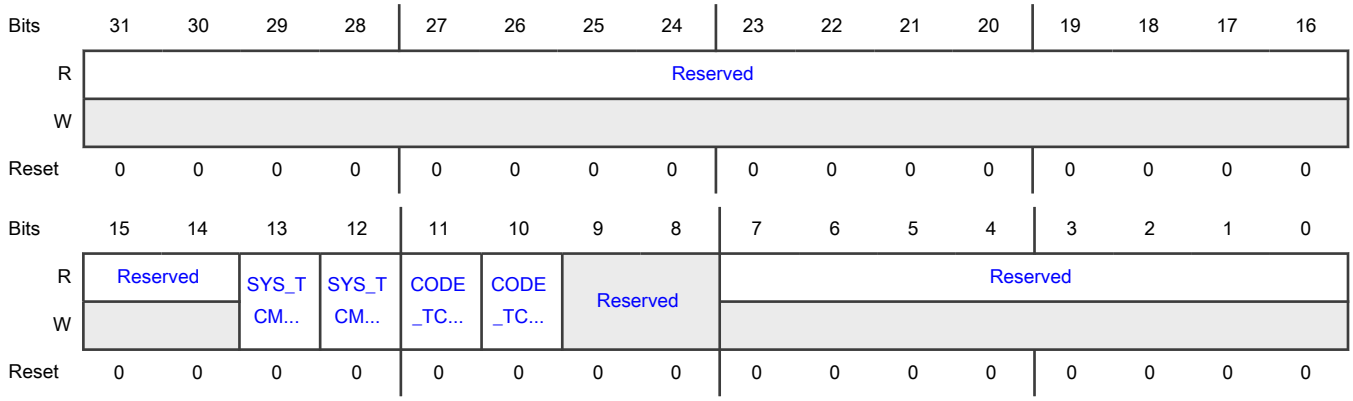
13.6.2.5 Interrupt Enable (INT_SIG_EN)

Selects which interrupt statuses are available to interrupt the host system. All these status fields share the same interrupt line (that is, all interrupts are combined with a logical OR). Writing 1 to any of these fields enables interrupt generation. The corresponding field in Interrupt Status (INT_STATUS) generates an interrupt when the corresponding field in Interrupt Status Enable (INT_STAT_EN) is 1.

Offset

Register	Offset
INT_SIG_EN	28h

Diagram



Fields

Field	Description
31-14 —	Reserved
13 SYS_TCM_ERRS_INT_SIG_EN	System TCM Access Single-Bit ECC Error Interrupt Signal Enable Selects INT_STATUS[SYS_TCM_ECC_ERRS_INT] as one of the interrupt statuses available to interrupt the host system. 0 - Mask 1 - Enable
12 SYS_TCM_ERRM_INT_SIG_EN	System TCM Access Multibit ECC Error Interrupt Signal Enable Selects INT_STATUS[SYS_TCM_ECC_ERRM_INT] as one of the interrupt statuses available to interrupt the host system. 0 - Mask 1 - Enable
11 CODE_TCM_ERRS_INT_SIG_EN	Code TCM Access Single-Bit ECC Error Interrupt Signal Enable Selects INT_STATUS[CODE_TCM_ECC_ERRS_INT] as one of the interrupt statuses available to interrupt the host system. 0 - Mask 1 - Enable
10 CODE_TCM_ERRM_INT_SIG_EN	Code TCM Access Multibit ECC Error Interrupt Signal Enable Selects INT_STATUS[CODE_TCM_ECC_ERRM_INT] as one of the interrupt statuses available to interrupt the host system. 0 - Mask 1 - Enable
9-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7-0 —	Reserved

13.6.2.6 Code TCM Single-Bit ECC Error Information (CODE_TCM_ECC_SINGLE_ERROR_INFO)

Stores detailed error information for the corresponding code TCM single-bit ECC interrupt when the interrupt is enabled.

Offset

Register	Offset
CODE_TCM_ECC_SINGLE_ERROR_INFO	5Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved													CODE_TCM_ECCS_EFSYN		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CODE_TCM_ECCS_EFSYN				CODE_TCM_ECCS_EFPRT				CODE_TCM_ECCS_EFMST				CODE_TCM_ECCS_EFSIZ			0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-19 —	Reserved
18-12 CODE_TCM_ECCS_EFSYN	Code TCM Single-Bit ECC Error Corresponding Syndrome Contains the corresponding syndrome when a single-bit error is reported. You can use the ECC syndrome to locate the error bit via a lookup table. See CM33 TCM ECC support for more information.
11-8	Code TCM Single-Bit ECC Error for Corresponding TCM Access Protection

Table continues on the next page...

Table continued from the previous page...

Field	Description
CODE_TCM_ECCS_EFPRT	Contains the master access protection attribute (tcm_priv) of the access that created the error when a single-bit error is reported. For details about the protection attribute decoding, see the Arm AMBA AHB specification.
7-4 CODE_TCM_ECCS_EFMST	Code TCM Single-Bit ECC Error for Corresponding TCM Master Contains the code TCM single-bit ECC error for master number (manager identifier).
3-1 CODE_TCM_ECCS_EFSIZ	Code TCM Single-Bit ECC Error for Corresponding TCM Access Size Contains the code TCM single-bit ECC error for TCM access size. For code TCM, this value must be 2 (32 bits).
0 —	Reserved

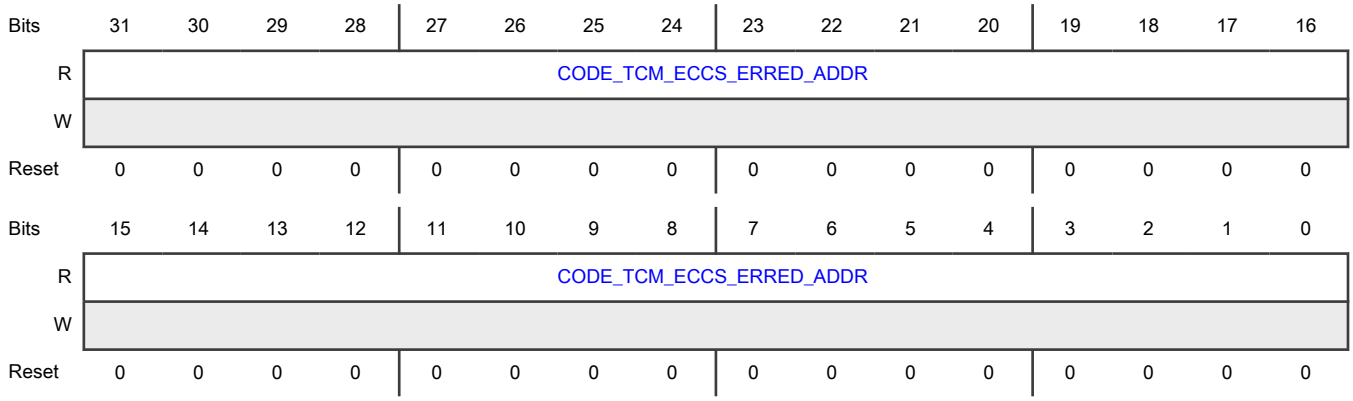
13.6.2.7 Code TCM Single-Bit ECC Error Address (CODE_TCM_ECC_SINGLE_ERROR_ADDR)

Stores the relative error address for the corresponding code TCM single-bit ECC interrupt.

Offset

Register	Offset
CODE_TCM_ECC_SINGLE_ERROR_ADDR	60h

Diagram



Fields

Field	Description
31-0 CODE_TCM_ECCS_ERRED_ADDR	Code TCM Single-Bit ECC Error Address Stores the relative error address for the corresponding code TCM single-bit ECC interrupt when the interrupt is enabled.

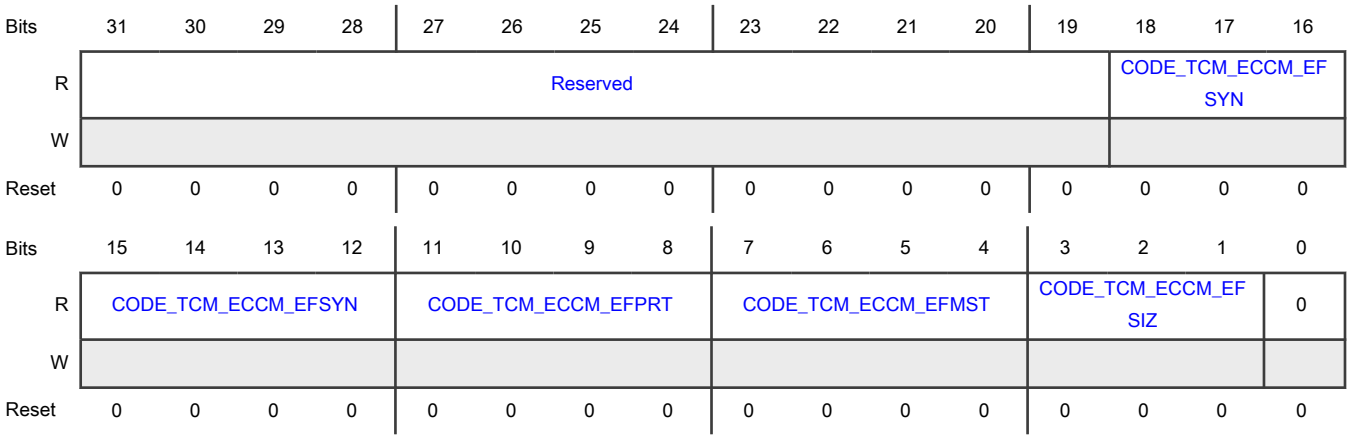
13.6.2.8 Code TCM Multibit ECC Error Information (CODE_TCM_ECC_MULTI_ERROR_INFO)

Stores detailed error information for the corresponding code TCM multibit ECC interrupt when the interrupt is enabled.

Offset

Register	Offset
CODE_TCM_ECC_MULTI_ERROR_INFO	68h

Diagram



Fields

Field	Description
31-19 —	Reserved
18-12 CODE_TCM_ECCM_EFSYN	Code TCM Multibit ECC Error for Corresponding Syndrome Contains the corresponding syndrome when a multibit error is reported. You cannot use the syndrome for multibit ECC to locate the error bit.
11-8	CODE_TCM Multibit ECC Error for Corresponding Access Protection Attribute

Table continues on the next page...

Table continued from the previous page...

Field	Description
CODE_TCM_ECCM_EFPRT	Contains the master access protection attribute (tcm_priv) of the access that created the error when a multibit error is reported. For details about the protection attribute decoding, see the Arm AMBA AHB specification.
7-4 CODE_TCM_ECCM_EFMST	Code TCM Multibit ECC Error for Corresponding TCM Master Contains the code TCM multibit ECC error master number (manager identifier).
3-1 CODE_TCM_ECCM_EFSIZ	Code TCM Multibit ECC Error for Corresponding TCM Access Size Contains the code TCM multibit ECC error for TCM access size. For code TCM, this value must be 2 (32 bits).
0 —	Reserved

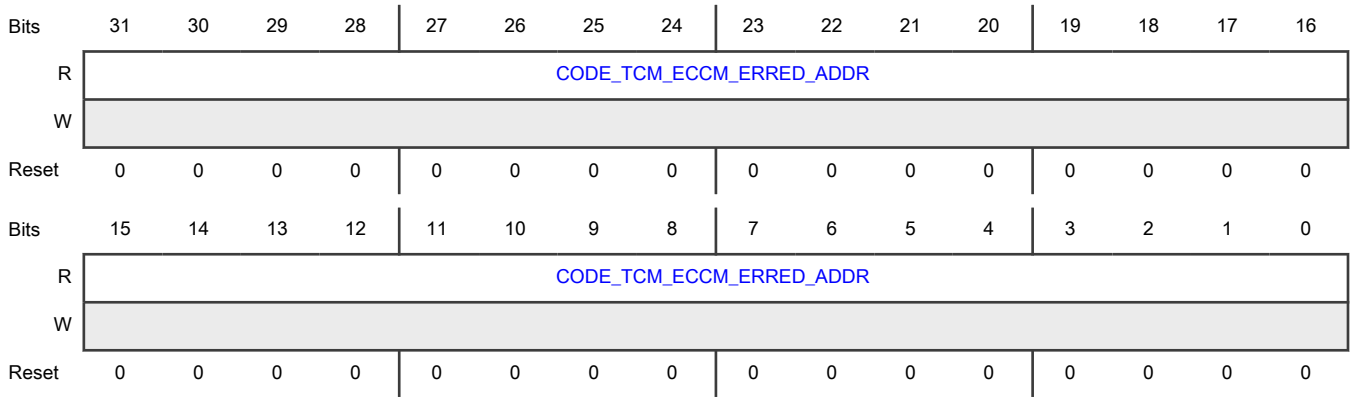
13.6.2.9 Code TCM Multibit ECC Error Address (CODE_TCM_ECC_MULTI_ERROR_ADDR)

Stores the relative error address for the corresponding code TCM multibit ECC interrupt.

Offset

Register	Offset
CODE_TCM_ECC_MULTI_ERROR_ADDR	6Ch

Diagram



Fields

Field	Description
31-0 CODE_TCM_ECCM_ERRED_ADDR	Code TCM Multibit ECC Error Address Stores the relative error address for the corresponding code TCM multibit ECC interrupt when the interrupt is enabled.

13.6.2.10 System TCM Single-Bit ECC Error Information (SYS_TCM_ECC_SINGLE_ERROR_INFO)

Stores detailed error information for the corresponding system TCM single-bit ECC interrupt when the interrupt is enabled.

Offset

Register	Offset
SYS_TCM_ECC_SINGLE_ERROR_INFO	74h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved													SYS_TCM_ECCS_EFSYN		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SYS_TCM_ECCS_EFSYN				SYS_TCM_ECCS_EFPRT				SYS_TCM_ECCS_EFMST				SYS_TCM_ECCS_EFSIZ			0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-19 —	Reserved
18-12 SYS_TCM_ECCS_EFSYN	System TCM Single-Bit ECC Error for Corresponding Syndrome Contains the corresponding syndrome when a single-bit error is reported. You can use the ECC syndrome to locate the error bit via a lookup table. See CM33 TCM ECC support for more information.
11-8	System TCM Single-Bit ECC Error for Corresponding Access Protection Attribute

Table continues on the next page...

Table continued from the previous page...

Field	Description
SYS_TCM_ECCS_EFPRT	Contains the master access protection attribute (tcm_priv) of the access that created the error when a single-bit error is reported. For details about the protection attribute decoding, see the Arm AMBA AHB specification.
7-4 SYS_TCM_ECCS_EFMST	System TCM Single-Bit ECC Error for Corresponding TCM Master Contains the system TCM single-bit ECC error master number (manager identifier).
3-1 SYS_TCM_ECCS_EFSIZ	System TCM Single-Bit ECC Error for Corresponding TCM Access Size Contains the code TCM single-bit ECC error for TCM access size. For system TCM, this value must be 2 (32 bits).
0 —	Reserved

13.6.2.11 System TCM Single-Bit ECC Error Address (SYS_TCM_ECC_SINGLE_ERROR_ADDR)

Stores the relative error address for the corresponding system TCM single-bit ECC interrupt.

Offset

Register	Offset
SYS_TCM_ECC_SINGLE_ERROR_ADDR	78h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SYS_TCM_ECCS_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SYS_TCM_ECCS_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SYS_TCM_ECCS_ERRED_ADDR	System TCM Single-Bit ECC Error Address Stores the relative error address for the corresponding system TCM single-bit ECC interrupt when the interrupt is enabled.

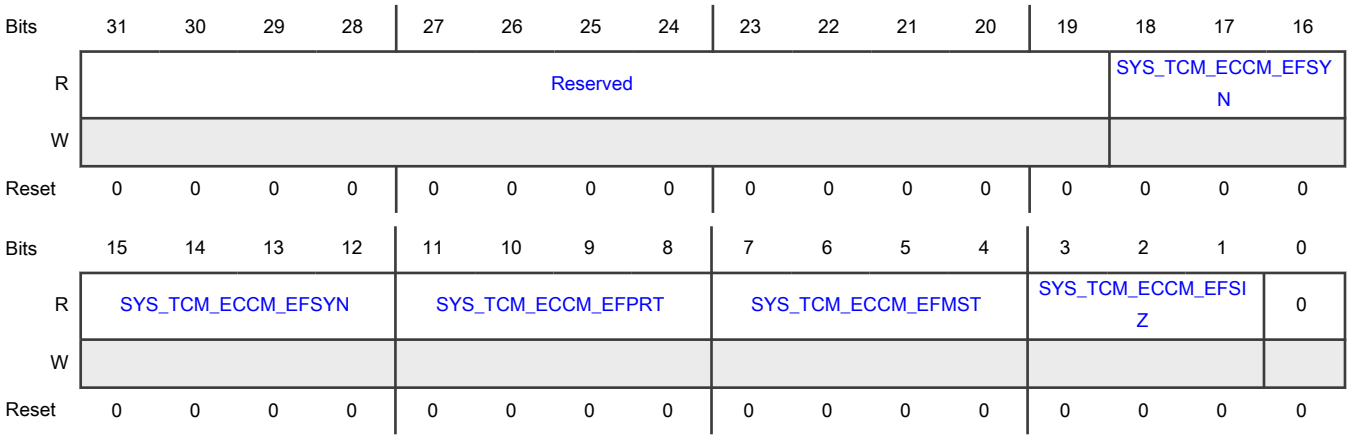
13.6.2.12 System TCM Multibit ECC Error Information (SYS_TCM_ECC_MULTI_ERROR_INFO)

Stores detailed error information for the corresponding system TCM multibit ECC interrupt when the interrupt is enabled.

Offset

Register	Offset
SYS_TCM_ECC_MULTI_ERROR_INFO	80h

Diagram



Fields

Field	Description
31-19 —	Reserved
18-12 SYS_TCM_ECCM_EFSYN	System TCM Multibit ECC Error for Corresponding Syndrome Contains the corresponding syndrome when a multibit error is reported. You cannot use the syndrome for multibit ECC to locate the error bit.
11-8	System TCM Multibit ECC Error for Corresponding Access Protection Attribute

Table continues on the next page...

Table continued from the previous page...

Field	Description
SYS_TCM_ECCM_EFPRT	Contains the master access protection attribute (tcm_priv) of the access that created the error when a multibit error is reported. For details about the protection attribute decoding, see the Arm AMBA AHB specification.
7-4 SYS_TCM_ECCM_EFMST	System TCM Multibit ECC Error for Corresponding TCM Master Contains the code TCM multibit ECC error master number (manager identifier).
3-1 SYS_TCM_ECCM_EFSIZ	System TCM Multibit ECC Error for Corresponding TCM Access Size Contains the system TCM multibit ECC error for TCM access size. For system TCM, this value must be 2 (32 bits).
0 —	Reserved

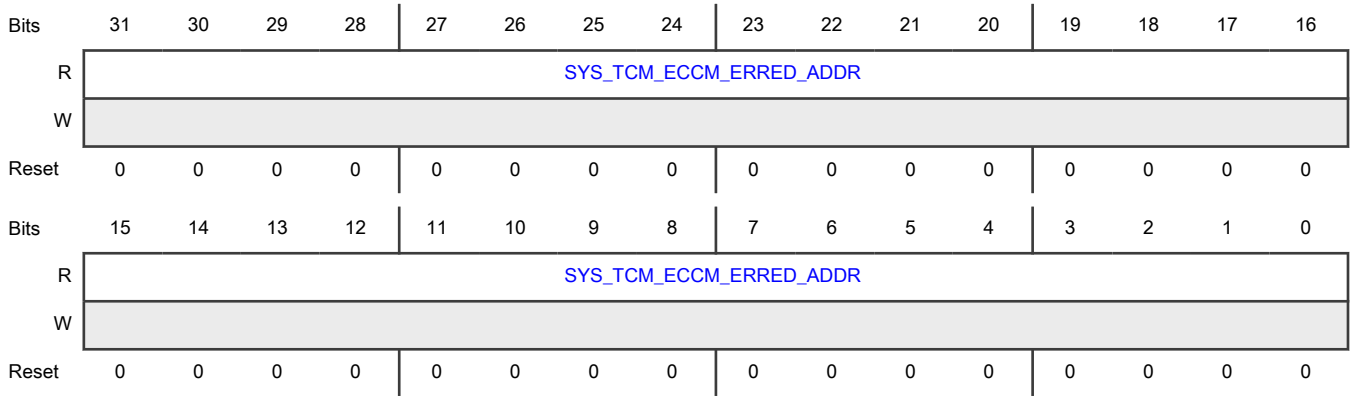
13.6.2.13 System TCM Multibit ECC Error Address (SYS_TCM_ECC_MULTI_ERROR_ADDR)

Stores the relative error address for the corresponding system TCM multibit ECC interrupt.

Offset

Register	Offset
SYS_TCM_ECC_MULTI_ERROR_ADDR	84h

Diagram



Fields

Field	Description
31-0 SYS_TCM_ECCM_ERRED_ADDR	System TCM Multibit ECC Error Address Stores the relative error address for the corresponding system TCM multibit ECC interrupt when the interrupt is enabled.

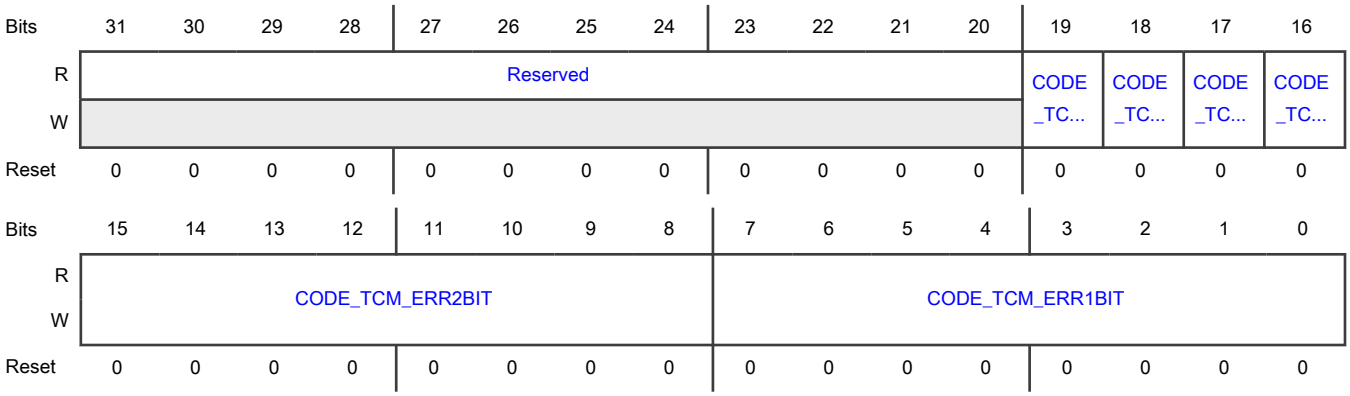
13.6.2.14 Code TCM ECC Error Injection (CODE_TCM_ECC_ERROR_INJEC)

Injects an ECC error into code TCM. This register is used only for verification and debugging.

Offset

Register	Offset
CODE_TCM_ECC_ERROR_INJEC	94h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 CODE_TCM_FRCNCI	Force Continuous Noncorrectable Data Inversions on Code TCM Write Access Injects a multibit ECC error on the bits specified by CODE_TCM_ERR1BIT and CODE_TCM_ERR2BIT into every write access to code TCM that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
18	Force Continuous 1-Bit Data Inversions on Code TCM Write Access

Table continues on the next page...

Table continued from the previous page...

Field	Description
CODE_TCM_FRC1BI	Injects a single-bit ECC error on the bit specified by CODE_TCM_ERR1BIT into every write access to code TCM that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
17 CODE_TCM_FR1NCI	Force One Noncorrectable Data Inversion on Code TCM Write Access Injects a multibit ECC error on the bits specified by CODE_TCM_ERR1BIT and CODE_TCM_ERR2BIT into the first write access to code TCM that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
16 CODE_TCM_FR11BI	Force One 1-Bit Data Inversion on Code TCM Write Access Injects a single-bit ECC error on the bit specified by CODE_TCM_ERR1BIT into the first write access to code TCM that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
15-8 CODE_TCM_ERR2BIT	Position of Second Bit to Inject ECC Error Determines the position of the second bit into which CM33 injects an ECC error. This field is used only for multibit ECC error injection.
7-0 CODE_TCM_ERR1BIT	Position of First Bit to Inject ECC Error Determines the position of the first bit into which CM33 injects an ECC error. This field can be used for both single-bit and multibit ECC error injection.

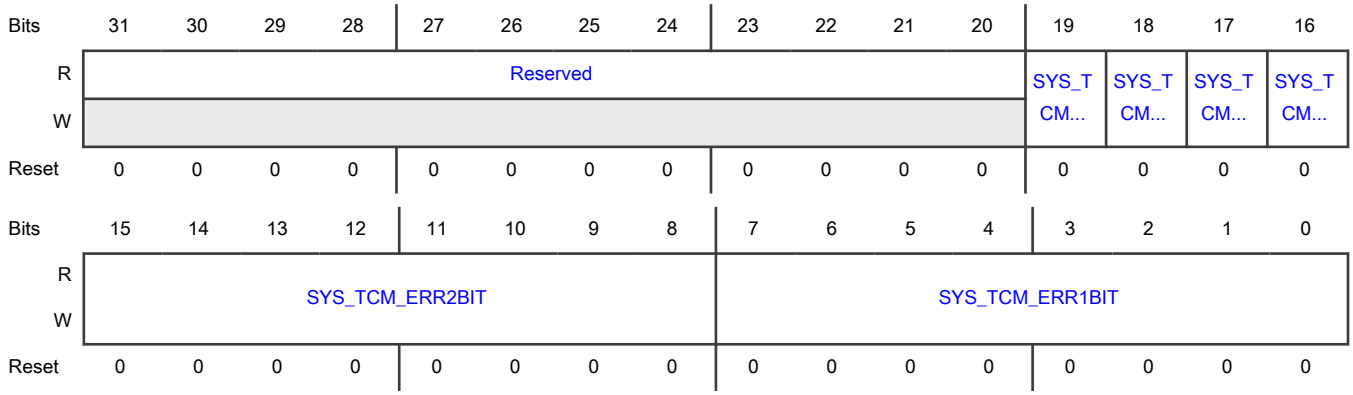
13.6.2.15 System TCM ECC Error Injection (SYS_TCM_ECC_ERROR_INJEC)

Injects an ECC error into system TCM. This register is used only for verification and debugging.

Offset

Register	Offset
SYS_TCM_ECC_ERROR_INJEC	98h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 SYS_TCM_FRCNCI	Force Continuous Noncorrectable Data Inversions on System TCM Write Access Injects a multibit ECC error on the bits specified by SYS_TCM_ERR1BIT and SYS_TCM_ERR2BIT into every write access to system TCM that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
18 SYS_TCM_FRC1BI	Force Continuous 1-Bit Data Inversions on System TCM Write Access Injects a single-bit ECC error on the bit specified by SYS_TCM_ERR1BIT into every write access to system TCM that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
17 SYS_TCM_FR1NCI	Force One Noncorrectable Data Inversion on System TCM Write Access Injects a multibit ECC error on the bits specified by SYS_TCM_ERR1BIT and SYS_TCM_ERR2BIT into the first write access to system TCM that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
16 SYS_TCM_FR11BI	Force One 1-Bit Data Inversion on System TCM Write Access Injects a single-bit ECC error on the bit specified by SYS_TCM_ERR1BIT into the first write access to system TCM that occurs after you write 1 to this field. 0 - Disable injection 1 - Enable injection
15-8	Position of Second Bit to Inject ECC Error

Table continues on the next page...

Table continued from the previous page...

Field	Description
SYS_TCM_ERR2BIT	Determines the position of the second bit into which CM33 injects an ECC error. This field is used only for multibit ECC error injection.
7-0 SYS_TCM_ERR1BIT	Position of First Bit to Inject ECC Error Determines the position of the first bit into which CM33 injects an ECC error. This field can be used for both single-bit and multibit ECC error injection.

Chapter 14
ARM Cortex CM33 MCM (CM33_MCM)

14.1 Overview

MCM provides many miscellaneous control functions.

NOTE

The terminology in this chapter has been updated to align with NXP's inclusive language standards, as shown in the table below.

Table 75. Updated terms

Updated term	Deprecated term
Initiator	Master
Target	Slave

14.1.1 Features

- Control and counting logic used when the embedded trace buffer (ETB) is almost full
- Error status and interrupts for the cache write buffer
- Error status and interrupts for the core's floating-point unit (FPU)

14.2 Functional description

14.2.1 Clocking

This module has no clocking considerations.

14.2.2 Interrupts

MCM generates an interrupt if any of the following is true:

- FPU input denormal interrupt is enabled (ISCR[FIDCE] = 1) and an input is denormalized (ISCR[FIDC] = 1).
- FPU inexact interrupt is enabled (ISCR[FIXCE] = 1) and a number is inexact (ISCR[FIXC] = 1).
- FPU underflow interrupt is enabled (ISCR[FUFCE] = 1) and an underflow occurs (ISCR[FUFC] = 1).
- FPU overflow interrupt is enabled (ISCR[FOFCE] = 1) and an overflow occurs (ISCR[FOFC] = 1).
- FPU divide-by-zero interrupt is enabled (ISCR[FDZCE] = 1) and a divide-by-zero occurs (ISCR[FDZC] = 1).
- FPU invalid operation interrupt is enabled (ISCR[FIOCE] = 1) and an invalid operation occurs (ISCR[FIOC] = 1).
- A cache write buffer error occurs.

See Determining the source of the interrupt for more details.

14.3 External signals

This module has no external signals.

14.4 Application information

14.4.1 Determining the source of the interrupt

To determine the exact source of the interrupt, check the interrupt status flags with the corresponding interrupt enable fields.

1. Read the fields located in ISCR[31:16] and in ISCR[15:0].
2. Search the result for asserted flags, which indicate the exact interrupt sources.

14.5 Memory map and register descriptions

The memory map and register descriptions below describe the registers using byte addresses.

NOTE

All MCM registers support only 32-bit writes. Any attempted 8-bit or 16-bit writes will result in undefined behavior.

14.5.1 MCM register descriptions

14.5.1.1 MCM memory map

ppb_mcm2_cm4 base address: E008_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
10	Interrupt Status and Control (ISCR)	32	RW	0000_0000
14	ETB Counter Control (ETBCC)	32	RW	0000_0000
18	ETB Reload (ETBRR)	32	RW	0000_0000
1C	ETB Counter Value (ETBCV)	32	R	0000_0000
20	Write Buffer Fault Address (FADR)	32	R	See section
24	Store Buffer Fault Attributes (FATR)	32	R	See section
28	Store Buffer Fault Data (FDR)	32	R	See section

14.5.1.2 Interrupt Status and Control (ISCR)

Configures and reports the status of core-related interrupt exception conditions. The individual event indicators are first checked with their exception enables and then logically summed to form an interrupt request sent to the core's NVIC.

The fields in bits 15–8 are indicator flags based on the processor's FPSCR register. Attempted writes to these fields are ignored. After the flags are set, they remain set until you write 0 to the corresponding FPSCR field.

The fields in bits 31–24 control the interrupt enablement. If you write 1 to an interrupt enablement control field, MCM generates an interrupt when the corresponding status field is 1.

Offset

Register	Offset
ISCR	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FIDCE	0		FIXCE	FUFCE	FOFCE	FDZCE	FIOCE	0		0	CWBEE	0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIDC	0		FIXC	FUFC	FOFC	FDZC	FIOC	0		0	CWBER	DHREQ	NMI	IRQ	0
W												W1C		W1C	W1C	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 FIDCE	FPU Input Denormal Interrupt Enable 0 - Disable 1 - Enable
30-29 —	Reserved
28 FIXCE	FPU Inexact Interrupt Enable 0 - Disable 1 - Enable
27 FUFCE	FPU Underflow Interrupt Enable 0 - Disable 1 - Enable
26 FOFCE	FPU Overflow Interrupt Enable 0 - Disable 1 - Enable
25 FDZCE	FPU Divide-by-Zero Interrupt Enable 0 - Disable 1 - Enable
24 FIOCE	FPU Invalid Operation Interrupt Enable 0 - Disable 1 - Enable
23-22	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
21 —	Reserved
20 CWBEE	Cache Write Buffer Error Enable Enables the generation of an interrupt in response to a bus error termination reported on a system bus transfer that has been initiated from the cache's write buffer. 0 - Disable 1 - Enable
19-16 —	Reserved
15 FIDC	FPU Input Denormal Interrupt Status Contains a copy of the core's FPSCR[IDC] and indicates that an input denormalized number has been detected in the processor's FPU. After this field becomes 1, it remains 1 until you write 0 to FPSCR[IDC]. 0 - No interrupt 1 - Interrupt occurred
14-13 —	Reserved
12 FIXC	FPU Inexact Interrupt Status Contains a copy of the core's FPSCR[IXC] and indicates that an inexact number has been detected in the processor's FPU. After this field becomes 1, it remains 1 until you write 0 to FPSCR[IXC]. 0 - No interrupt 1 - Interrupt occurred
11 FUFC	FPU Underflow Interrupt status Contains a copy of the core's FPSCR[UFC] and indicates that an underflow has been detected in the processor's FPU. After this field becomes 1, it remains 1 until you write 0 to FPSCR[UFC]. 0 - No interrupt 1 - Interrupt occurred
10 FOFC	FPU Overflow Interrupt Status Contains a copy of the core's FPSCR[OFC] and indicates that an overflow has been detected in the processor's FPU. After this field becomes 1, it remains 1 until you write 0 to FPSCR[OFC]. 0 - No interrupt 1 - Interrupt occurred

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 FDZC	FPU Divide-by-Zero Interrupt Status Contains a copy of the core's FPSCR[DZC] and indicates that a divide-by-zero has been detected in the processor's FPU. After this field becomes 1, it remains 1 until you write 0 to FPSCR[DZC]. 0 - No interrupt 1 - Interrupt occurred
8 FIOC	FPU Invalid Operation Interrupt Status Contains a copy of the core's FPSCR[IOC] and indicates that an illegal operation has been detected in the processor's FPU. After this field becomes 1, it remains 1 until you write 0 to FPSCR[IOC]. 0 - No interrupt 1 - Interrupt occurred
7-6 —	Reserved
5 —	Reserved
4 CWBER	Cache Write Buffer Error Status Indicates whether a data transfer from the core's cache write buffer was terminated because of a bus error. This field can become 1 only when the corresponding enable field (ISCR[CWBEE]) = 1. The corresponding core fault address, attributes, and write data are typically retrieved from Write Buffer Fault Address (FADR), Store Buffer Fault Attributes (FATR), and Store Buffer Fault Data (FDR) registers during the interrupt handler before clearing the CWBER flag. 0 - No error 1 - Error occurred
3 DHREQ	Debug Halt Request Indicator Indicates whether a debug halt request is initiated because of an ETB counter expiration. This field becomes 1 when: • ETBCC[RSPT] = 11b. • ETBCC[CNTEN] = 1b. • ETBCV[COUNTER] = 0h. This field becomes 0 when the ETB counter disables or reloads. 0 - Not initiated 1 - Initiated
2 NMI	Nonmaskable Interrupt Pending Specifies whether a pending NMI exists. If ETBCC[RSPT] = 10b, this field becomes 1 to indicate a pending NMI when the ETB counter expires.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - No pending NMI 1 - Pending NMI
1 IRQ	ETB-Related Interrupt Pending Specifies whether a pending IRQ exists. If ETBCC[RSPT] = 01b, this field becomes 1 to indicate a pending IRQ when the ETB counter expires. 0 - No pending IRQ 1 - Pending IRQ
0 —	Reserved

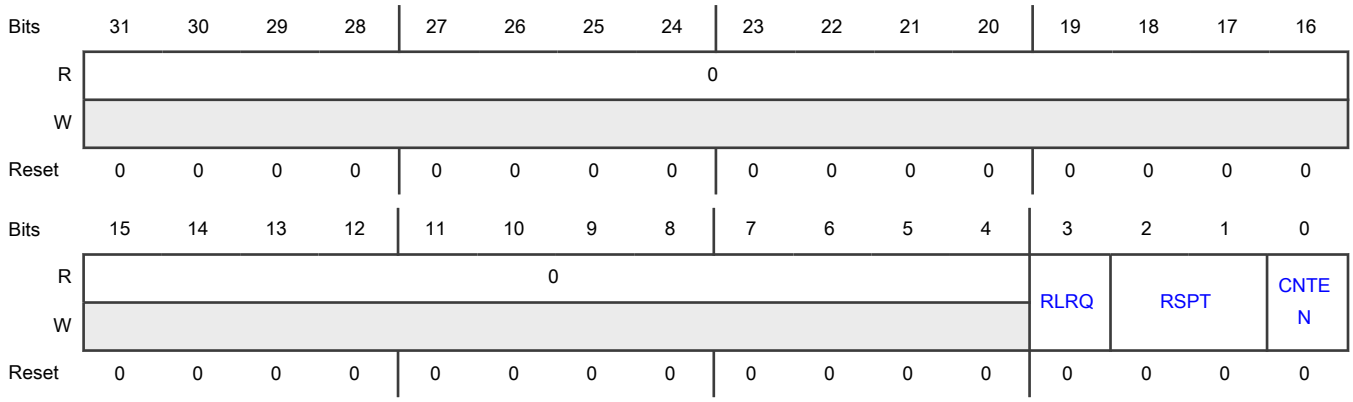
14.5.1.3 ETB Counter Control (ETBCC)

Configures ETB features.

Offset

Register	Offset
ETBCC	14h

Diagram



Fields

Field	Description
31-4 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
3 RLRQ	Reload Request Reloads the ETB packet counter with the value defined in ETBRR[RELOAD]. Writing 1 to this field also clears the pending debug halt, NMI, or IRQ interrupt requests: • Clears the pending NMI or IRQ interrupt request if IRQ or NMI interrupts were enabled and an NMI or IRQ interrupt was generated on counter expiration. • Clears the debug halt request if debug halt was enabled and a debug halt request was asserted on counter expiration. 0 - No effect 1 - Reload the counter (clear pending interrupt requests)
2-1 RSPT	Response Type Specifies the response to the ETB count expiration. 00 - No response 01 - Generate a normal interrupt 10 - Generate an NMI 11 - Generate a debug halt
0 CNTEN	Counter Enable Enables the ETB counter. 0 - Disable 1 - Enable

14.5.1.4 ETB Reload (ETBRR)

Configures the 0-mod-4 value that reloads into the ETB packet counter.

Offset

Register	Offset
ETBRR	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				RELOAD											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-11 —	Reserved
10-0 RELOAD	Byte Count Reload Value Specifies the reload value for the ETB packet counter. When ETBCC[RLRQ] = 1b, the ETB packet counter reloads with the 0-mod-4 value that you configure in this field. Writing a non-0-mod-4 value to this field results in a bus error.

14.5.1.5 ETB Counter Value (ETBCV)

Indicates the current 0-mod-4 value of the ETB counter.

Offset

Register	Offset
ETBCV	1Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				COUNTER											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-11 —	Reserved
10-0 COUNTER	Byte Count Counter Value Indicates the current 0-mod-4 value of the ETB counter.

14.5.1.6 Write Buffer Fault Address (FADR)

Captures the faulty address if a properly enabled cache write buffer error interrupt event is detected.

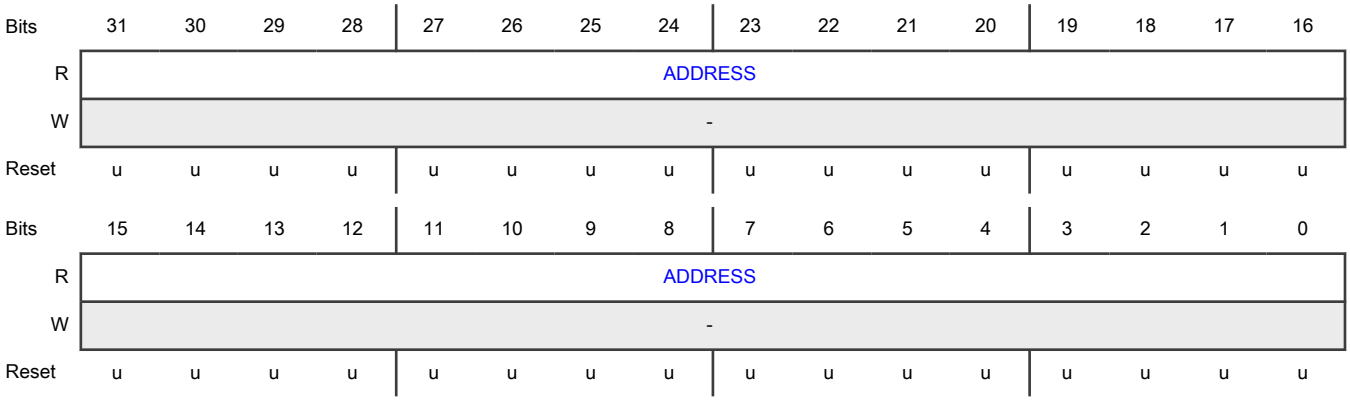
MCM captures a single cache write buffer bus error event. If a subsequent error is detected before the captured error information has been read from the corresponding registers and ISCR[CWBER] is 0, then FATR[BEOVR] becomes 1. However, no additional information is captured.

MCM writes 1 to the field in this register when ISCR[CWBER] is 1. Attempted writes have no effect.

Offset

Register	Offset
FADR	20h

Diagram



Fields

Field	Description
31-0 ADDRESS	Fault Address Contains the faulty address if a properly enabled cache write buffer error interrupt event is detected.

14.5.1.7 Store Buffer Fault Attributes (FATR)

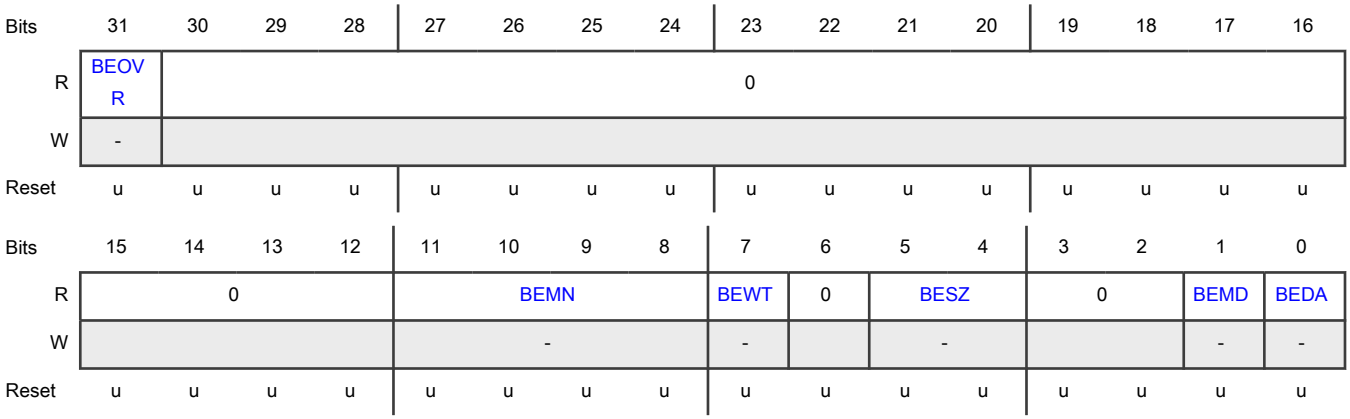
Indicates the faulty attributes if a properly enabled cache write buffer error interrupt event is detected.

MCM writes values to the fields in this register when ISCR[CWBER] is 1b. Your attempted writes have no effect.

Offset

Register	Offset
FATR	24h

Diagram



Fields

Field	Description
31 BEOVR	Bus Error Overrun Indicates if another cache write buffer bus error is detected before system software has retrieved all the error information from the original event. The time window is from the detection of the original cache write buffer error termination to the time when you write 1 to ISCR[CWBER] to clear it and rearm the capture logic. MCM writes 1 to this field and it becomes 0 whenever you write 1 to ISCR[CWBER]. Write Buffer Fault Address (FADR), Store Buffer Fault Data (FDR), and the other FATR fields do not update to reflect this new bus error. 0 - No overrun 1 - Overrun
30-12 —	Reserved
11-8 BEMN	Bus Error Initiator Number Indicates the crossbar switch bus initiator number of the captured cache write buffer bus error.
7 BEWT	Bus Error Write Indicates the type of system bus access when the cache write buffer error was detected. 0 - Read 1 - Write

Table continues on the next page...

Table continued from the previous page...

Field	Description
6 —	Reserved
5-4 BESZ	Bus Error Size Indicates the size of the cache write buffer access when the error was detected. 00 - 8-bit 01 - 16-bit 10 - 32-bit 11 - Reserved
3-2 —	Reserved
1 BEMD	Bus Error Privilege Level Indicates the privilege level of the cache write buffer access when the error was detected. 0 - User mode 1 - Supervisor or privileged mode
0 BEDA	Bus Error Data Access Type Indicates the type of cache write buffer access when the error was detected. 0 - Instruction 1 - Data

14.5.1.8 Store Buffer Fault Data (FDR)

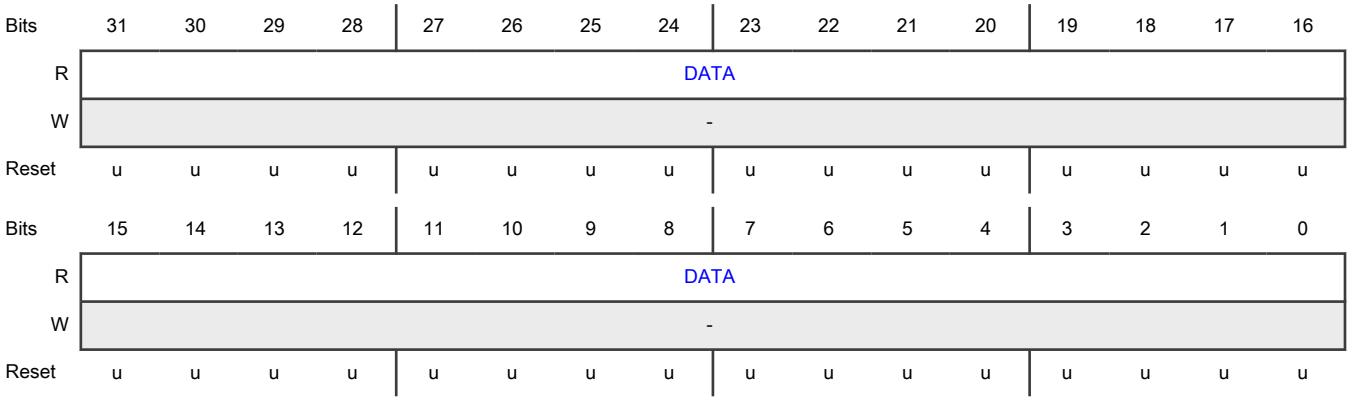
Indicates the captured faulty data, when a properly enabled cache write buffer error interrupt event is detected.

MCM writes 1 to DATA when ISCR[CWBER] is 1. Only the accessed byte lanes contain valid data for byte and halfword writes. Contents of the other bytes are undefined. Attempted writes have no effect.

Offset

Register	Offset
FDR	28h

Diagram



Fields

Field	Description
31-0 DATA	Fault Data Contains the captured faulty data, when a properly enabled cache write buffer error interrupt event is detected.

Chapter 15
CM33 Cache Memory Controller (XCACHE)

15.1 Overview

XCACHE is a general-purpose AMBA AHB bus protocol cache that provides low-latency access to slower memories.

15.1.1 Block diagram

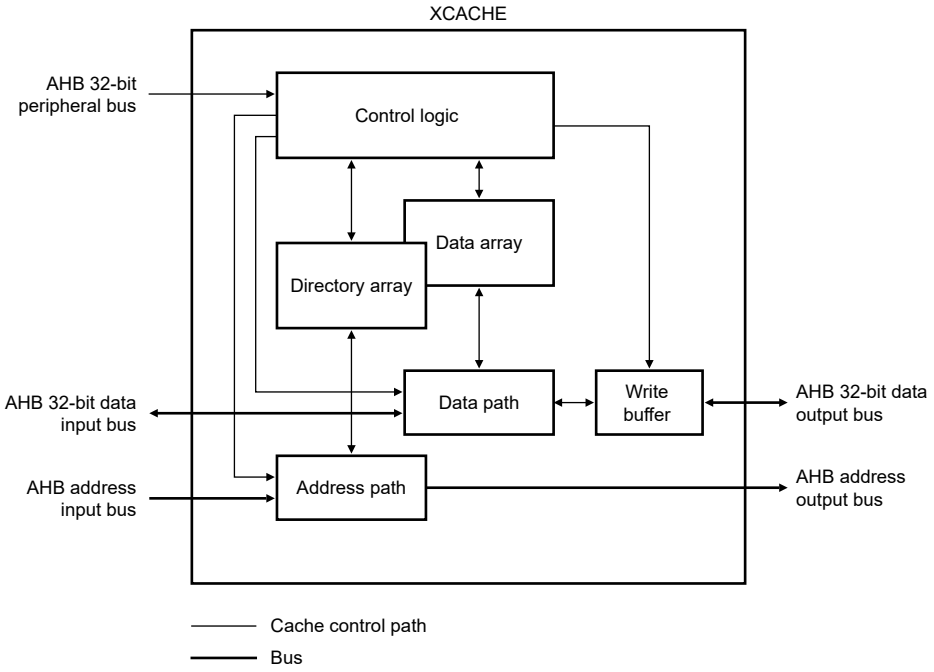


Figure 30. Block diagram

15.1.2 Features

XCACHE is designed to work with AMBA AHB bus systems. It includes:

- An AHB input bus. Accesses have cache mode attributes for cacheable copyback, cacheable write-through, or noncacheable mode operation and for write bufferable mode operation on an access-by-access basis.
- 32-bit data path AHB output bus for downstream cache read miss, cache write-through write, and noncacheable accesses.
- Connections to tag and data RAMs for implementing the cache functions.
- A peripheral interface for accessing the cache's programming model.

XCACHE includes the following AMBA_AHB buses:

- AHB input bus
- AHB output bus

15.2 Functional description

XCACHE is a block of high-speed memory locations that contains address information (commonly known as a tag) and its associated data. It decreases the average time of a memory access.

When data loads into a cache, the access times for subsequent loads and stores are reduced, resulting in overall performance benefits. Access to information already in a cache is known as a cache hit, and other accesses are known as cache misses.

Normally, caches are self-managing, with the updates occurring automatically. Whenever the input bus wants to access a cacheable location, the cache is checked. If the access is a cache hit, the access occurs immediately. Otherwise, a location is allocated and the cache line is loaded from memory. Different cache topologies and access policies are possible. However, they must comply with the memory coherency model of the underlying architecture.

15.2.1 Principles of locality

XCACHE operates using the following two principles of locality.

Table 76. Type of locality

Type of locality	Description	Access
Spatial	Groups several locations together under the same tag to minimize the quantity of control information stored. This logical block is commonly known as a cache line.	Access to one location is possibly followed by accesses from adjacent locations (for example, sequential instruction execution or usage of a data structure).
Temporal	Retains recently accessed lines in the cache.	Access to an area of memory possibly repeats within a short period of time (for example, execution of a code loop).

15.2.2 Modes of operation

Table 77. Modes of operation

Mode	Description
Write-Through	Access to address spaces with this cache mode are cacheable: • If all cacheable spaces are read-only spaces, then cache contains read-only data and all writes to the cache cause a fault. See chip-specific cacheable space information. • A read miss on the input bus causes a line read on the output bus of a 16-byte-aligned memory address that contains the desired address. This missed data is loaded into the cache and is marked as valid and not modified. • A read hit to a valid cache location returns data from the cache with no output bus access. • A write miss bypasses the cache and writes to the output bus (no allocation on write-miss policy for Write-Through mode spaces). • A write hit updates the cache hit line with the write data and writes to the output bus. • Caches are input-bus-local and do not support hardware cache coherency. If the input bus accesses write-through regions and an external input bus (such as DMA) need to update these regions, you must first perform explicit cache clears to any needed cache memory range. This ensures all modified cache lines update their associated memories before external inputs modify them, and subsequent cache input bus accesses get the updated memory.
Write-back	Access to address spaces with this cache mode are cacheable:

Table continues on the next page...

Table 77. Modes of operation (continued)

Mode	Description
	• A read miss on the input bus causes a line read on the output bus of a 16-byte-aligned memory address that contains the desired address. This missed data is loaded into the cache and is marked as valid and not modified. • A read hit to a valid cache location returns data from the cache with no output bus access. • A write miss performs a read-to-write (allocate on write miss policy for write-back mode spaces). You perform a line read on the output bus of a 16-byte-aligned memory address containing the desired write address. This missed data is loaded into the cache and marked as valid and modified. Then the write data updates the appropriate cache data locations. • A write hit updates the cache hit line with the write data and marks the line as modified. • Caches do not support hardware cache coherency. If the cache's input bus has written to write-back regions and another input bus that can access the same target memory without going through the cache needs to see these updates, you must perform explicit cache pushes to any needed cache memory range to ensure all modified cache lines update their associated memories before being read by this input. Likewise, if the cache's input bus has accessed write-back or write-through regions and another input bus that can access the same target memory without going through the cache needs to update these regions, you must first perform explicit cache clears to any needed cache memory range. This ensures all modified cache lines update their associated memories before this input modifies them, and subsequent cache input bus accesses get the updated memory.
Non-cacheable	Access to address spaces with this cache mode are not cacheable. These accesses bypass the cache and access the output bus.

There is an optional ECC (Error correcting code) support of the cache tag and data storage RAMs. The optional ECC is SECDED (Single error correct, double error detect).

15.2.3 XCACHE function

XCACHE receives bus requests on the input bus.

You can access the programming model for the cache via XCACHE's AMBA APB bus.

XCACHE then processes the cacheable accesses as needed, when bypassing the noncacheable, cache write-through, cache miss, and cache maintenance accesses to its output bus.

The cache on this chip is structured as follows. It has a 2-way set-associative cache structure with a total size of 16 KB for the cache. The caches have 32-bit addresses and data paths and a 16-byte line size. The cache tags and data storage use single-port, synchronous RAMs.

The tag cache contains the required upper address bits, a modify bit, a valid bit, and up to 8 bits of extended address attributes. The extended address attribute field has information such as the supervisor, user, and/or domain ID of the accesses loading this tag. If any extended address attribute bits are provided, for subsequent accesses that hit a valid tag but have different attributes, the cache coherently forces a miss so that the downstream memory protection logic can check the access.

For this 16 KB cache, each cache tag function uses two 512 x 21-bit RAM arrays or optionally two up to 512 x 29-bit RAM arrays (if all 8 bits of the extended address attribute are used). The cache data function uses two 2048 x 32-bit RAM arrays. The cache

tag entries store 19 bits of upper address as well as a modified and valid bit per cache line, and any optional extended address attribute bits. The cache data entries store four bytes of code or data.

The tag and data RAM ECC codes are stored in RAM arrays with their associated tag and data. The 7Bits tag ECC code word per cache way is stored along side the associated tag way. The 7Bit data ECC code word per cache way is stored along side the associated data way.

All normal cache accesses use physical addresses. This leads to the use of the following cache address:

CACHE - 16 KB size = (512 sets) x (16-byte lines) x (2-way set associative)

Table 78. Tag cache address use

Tag cache address use	Cache
Tag hit address range	Address[31:13]
Tag set select address range	Address[12:4] selects 1 of the512 sets
Not used	Address[3:0]

Table 79. Data cache address use

Data cache address use	System cache
Not used	Address[31:13]
Data set select address range	Address[12:4] used to select one of the 512 sets
32-bit word select	Address[3:2] used to select one of the four 32-bit words within a set
Byte select	Address[1:0] used to select the byte within the 32-bit word

15.2.4 XCACHE control

The cache is disabled at reset. Cache tag and data arrays are not cleared at reset. Therefore, to enable and configure the cache, you must clear the cache commands and initialize the required tag array.

15.2.4.1 XCACHE set commands

XCACHE set commands may operate on:

- All of way 0
- All of way 1
- All of both ways (complete cache)

You can initiate cache set commands by using the upper bits in Cache Control (CCR). Cache set commands perform their operation on the cache independent of CCR[ENCACHE].

You initiate a cache set command by writing 1 to CCR[GO]. This field also acts as a busy bit for set commands and remains 1 when the command is active. XCACHE writes 0 to it after the completion of the set command.

Table 80 shows a list of supported cache set commands:

- Invalidate − Unconditionally clears valid and modify bits of a cache entry.
- Push − Pushes a cache entry if it is valid and modified, and then clears the modify bit. If the entry is not valid or not modified, leave it as is.
- Clear − Pushes a cache entry if it is valid and modified, and then clears the valid and modify bits. If the entry is not valid or not modified, clear the valid bit.

Table 80. Cache set commands

Cache Control (CCR)[27:24] settings				Command
CCR[PUSHW1]	CCR[INVW1]	CCR[PUSHW0]	CCR[INVW0]	
0	0	0	0	NOP
0	0	0	1	Invalidate all way 0
0	0	1	0	Push all way 0
0	0	1	1	Clear all way 0
0	1	0	0	Invalidate all way 1
0	1	0	1	Invalidate all way 1; invalidate all way 0 (invalidate cache)
0	1	1	0	Invalidate all way 1; push all way 0
0	1	1	1	Invalidate all way 1; clear all way 0
1	0	0	0	Push all way 1
1	0	0	1	Push all way 1; invalidate all way 0
1	0	1	0	Push all way 1; push all way 0 (push cache)
1	0	1	1	Push all way 1; clear all way 0
1	1	0	0	Clear all way 1
1	1	0	1	Clear all way 1; invalidate all way 0
1	1	1	0	Clear all way 1; push all way 0
1	1	1	1	Clear all way 1 and way 0 (clear cache)

After a reset, complete an invalidate cache command before using the cache. It is possible to combine the cache invalidate command with the cache enable. That is, writing 8500_0001h to Cache Control (CCR) invalidates the cache and enables the cache.

15.2.4.2 XCACHE line commands

XCACHE line commands operate on a single line in the cache at a time. You can perform cache line commands using a physical or cache address that consists of a set address and a way select. The line command acts on the specified cache line.

Cache line commands with physical addresses first search both ways of the cache set specified by the following physical address bits (see Table 81). If they hit, the commands perform their action on the hit way. For cache-[12:4]

You specify cache line commands using the upper bits in Cache Line Control (CLCR). Cache line commands perform their operation on the cache independent of CCR[ENCACHE]. Using a cache address, the command can be completely specified using Cache Line Control (CLCR). Using a physical address, the command must also use Cache Search Address (CSAR) to specify the physical address.

You can initiate a line cache command by writing 1 to CLCR[LGO] or CSAR[LGO]. These fields also act as busy bits for line commands and remain 1 when the line cache command is active and XCACHE writes 0 to them after the completion of the command.

Table 81. Cache line commands

Cache Line Control (CLCR)[27:24] settings			Command
CLCR[LACC]	CLCR[LADSEL]	CLCR[LCMD]	
0	0	00	Search by cache address and way
0	0	01	Invalidate by cache address and way
0	0	10	Push by cache address and way
0	0	11	Clear by cache address and way
0	1	00	Search by physical address
0	1	01	Invalidate by physical address
0	1	10	Push by physical address
0	1	11	Clear by physical address
1	0	00	Write by cache address and way
1	0	01	Reserved, NOP
1	0	10	Reserved, NOP
1	0	11	Reserved, NOP
1	1	*xx*	Reserved, NOP

15.2.4.2.1 Executing a series of line commands using cache addresses

You can perform a series of line commands with incremental cache addresses by writing to Cache Line Control (CLCR):

1. Place the command in the following fields:
   - CLCR[LACC]
   - CLCR[LADSEL]
   - CLCR[LCMD]
2. Write 1 to CLCR[WSEL] and CLCR[TDSEL] as required.
3. Place the cache address in CLCR[CACHEADDR].
4. Write 1 to CLCR[LGO].

After one line command completes, perform the following procedure to initiate the next command:

1. Increment the cache address (at bit 2 to step through data or at bit 4 to step through lines).
2. Write 1 to CLCR[LGO].

15.2.4.2.2 Executing a series of line commands using physical addresses

You can perform a series of line commands with incremental physical addresses by writing to Cache Line Control (CLCR):

1. Place the command in the following fields:
   - CLCR[LACC]
   - CLCR[LADSEL]
   - CLCR[LCMD]
2. Write 1 to CLCR[TDSEL].
3. Place the physical address in CSAR[PHYADDR] and write 1 to CSAR[LGO].

After one line command completes, perform the following procedure to initiate the next command:

1. Increment the physical address (at bit 2 to step through data or at bit 4 to step through lines).
2. Write 1 to CSAR[LGO].

Cache Line Control (CLCR) and Cache Search Address (CSAR) both contain a field called LGO. Therefore, you can complete the aforementioned steps in a single write to CSAR.

15.2.4.2.3 Line command results

After the completion of a line command, Cache Line Control (CLCR) contains information on the initial state of the line targeted by the command. For line commands with cache addresses, read this information before performing a line command action from the targeted cache line. For line commands with physical addresses, read this information on a hit before performing a line command action from the hit cache line, or, clear the initial valid bit if the command misses. In general, if CLCR[LCIVB] becomes 0, the targeted line is invalid at the start of the line command and no line operation is performed.

Table 82. Line command results

Cache Line Control (CLCR)[22:20] settings			For cache address commands	For physical address commands
CLCR[LCWAY]	CLCR[LCIMB]	CLCR[LCIVB]		
0	0	0	Way 0 line is invalid	No hit
0	0	1	Way 0 valid and not modified	Way 0 valid and not modified
0	1	0	Way 0 line is invalid	No hit
0	1	1	Way 0 valid and modified	Way 0 valid and modified
1	0	0	Way 1 line is invalid	No hit
1	0	1	Way 1 valid and not modified	Way 1 valid and not modified
1	1	0	Way 1 line is invalid	No hit
1	1	1	Way 1 valid and modified	Way 1 valid and modified

After the completion of a line command other than a write, Cache Read/Write Value (CCVR) contains information on the initial state of the line tag or data targeted by the command. For line commands, CLCR[TDSEL] selects between tag and data. If the line command uses a physical address and misses, the data is don't care. For write commands, Cache Read/Write Value (CCVR) holds the write data.

The cache does not have line lock capability.

15.2.4.3 XCACHE ECC enable

There are two ECC control enable inputs - one to enable ECC generation on writes and another to enable ECC checking and correction on reads. These inputs are controlled in an external program model. These signals are:

- ecc_wr_enable - Enable ECC generation on writes.
- ecc_rd_enable - Enable ECC checking and correction on reads.

At reset, the cache is disabled. After reset, to enable the cache with ECC enabled, perform the following:

1. Assert ecc_wr_enable and negate ecc_rd_enabled.
2. Perform the required cache "invalidated all" command. This initializes the TAG ECC.
3. After the invalidate completes, assert both ecc_wr_enable and ecc_rd_enabled.
4. Enable the cache - now the cache is enabled with ECC protection on.

15.2.4.4 XCACHE ECC function

When XCACHE ECC is enabled, for cache tag writes:

- All cache tag write generate the required tag ECC word.
- The cache tag and the associated tag ECC word are written to the tag RAM and tag ECC word RAM simultaneously.

When THE cache ECC is enabled, for cache data writes:

- All cache data writes of less than 32 bits are converted to a 32-bit read-modify-write sequence.
- All cache data writes generate the required data ECC word.
- The cache data and associated data ECC word are written to the data RAM and data ECC word RAM simultaneously.

When cache ECC is enabled, for cache tag reads:

- The cache tag and its ECC word are read in parallel.
- ECC is checked, any single bit error is corrected on-the-fly, any single, or multi-bit error is flagged.

When cache ECC is enabled, for cache data reads:

- For every data 32-bit word read, the data 32-bit word and its ECC word are read in parallel.
- ECC is checked, any single-bit error is corrected on-the-fly. Any single-, or multi-bit error is flagged.

The cache has single- and multi-bit ECC error indicator outputs.

Because all single-bit ECC errors are corrected on-the-fly, cache operations continue as normal when only single-bit ECC errors occur. Note it is possible to get multiple single- or multi-bit ECC errors on one cache access.

Cache read accesses that get multi-bit ECC errors are handled as follows:

- Any access that gets a tag multi-bit error terminates with a bus fault.
- Any access that gets a data multi-bit error on its hit data terminates with a bus fault.
- Data multi-bit error on miss or invalidate ways are ignored.

NOTE

Cache allocate pushes read the cache data arrays and may get ECC errors. Because the cache access that caused the allocate push may have completed before the ECC error, bus errors do not occur on multi-bit ECC errors. Only single or multi-bit ECC error indicators provide indications for an ECC error.

NOTE

Cache commands, which initiate via the cache's program model, may also access the cache tags and cache data RAMs. If these accesses get a multi-bit ECC error, the peripheral bus access that initiated the cache command does not terminate with a bus error. Only single or multi-bit ECC error indicators provide indications for an ECC error.

15.2.5 Clocks

XCACHE uses the AHB input bus clock.

15.2.6 Interrupts

XCACHE has no interrupts.

15.3 External signals

XCACHE has no external signals.

15.4 Initialization

To configure and enable the cache, you must perform XCACHE commands to clear and initialize the required tag array fields.

Table 83. Initialization sequence

Purpose	Programming	Notes
Enable the cache after reset, with the write buffer enabled.	Write 8500_0003h to Cache Control (CCR).	After you write to CCR[GO], it initiates the cache command specified by the following fields: • CCR[INVW1] • CCR[INVW0] The aforementioned fields indicate a command to clear all entries in both cache ways. CCR[ENWRBUF] enables the write buffer and CCR[ENCACHE] enables the cache. See XCACHE control for more information.
Enable the cache after reset, with the write buffer disabled.	Write 8500_0001h to Cache Control (CCR).	After you write to CCR[GO], it initiates the cache command specified by the following fields: • CCR[INVW1] • CCR[INVW0] The aforementioned fields indicate a command to clear all entries in both cache ways, and CCR[ENCACHE] enables the cache. See XCACHE control for more information.

15.5 Application information

XCACHE provides a low-latency access to instructions or data. This decouples processor performance from system memory performance, thereby increasing bus availability for other modules and improving system performance.

15.6 XCACHE register descriptions

The XACHE programming model provides various registers for the following functions:

- Configuring and controlling the cache
- Providing indirect access paths to all cache tag and data storage

NOTE

You can access XCACHE registers only in Supervisor mode.

15.6.1 XCACHE memory map

LPCAC_PC base address: 4440_0000h

LPCAC_PS base address: 4440_0800h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Cache Control (CCR)	32	RW	0000_0000
4	Cache Line Control (CLCR)	32	RW	0000_0000
8	Cache Search Address (CSAR)	32	RW	0000_0000
C	Cache Read/Write Value (CCVR)	32	RW	0000_0000

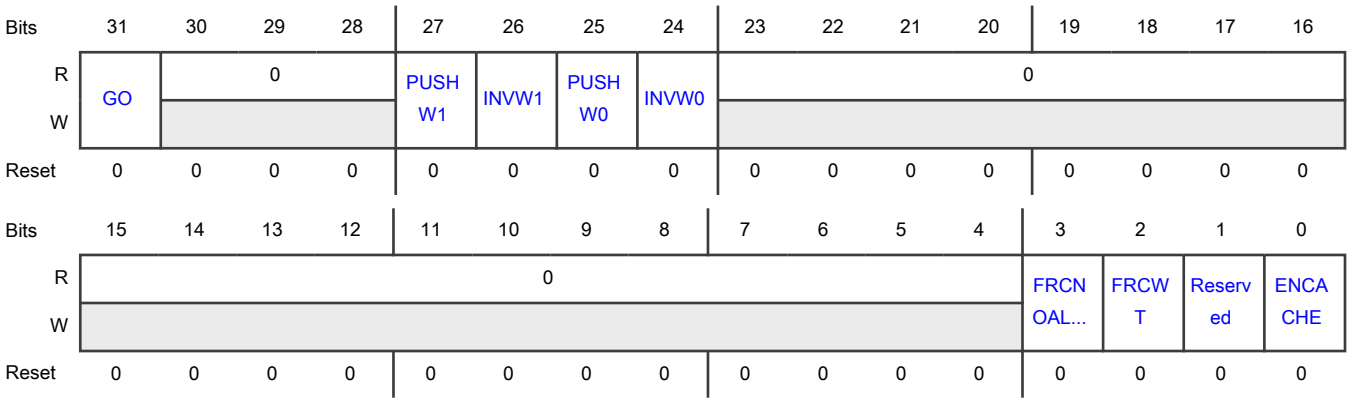
15.6.2 Cache Control (CCR)

Configures the cache control.

Offset

Register	Offset
CCR	0h

Diagram



Fields

Field	Description
31 GO	Initiate Cache Command Initiates the cache command that PUSHW1, INVW1, PUSHW0, and INVW0 indicate if you write 1 to this field. Reading this field indicates whether a command is active. **NOTE** This field remains 1 until the command completes. Writing 0 to it has no effect. 0 - Write: no effect; Read: no cache command active 1 - Write: initiate command; Read: cache command active
30-28	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
27 PUSHW1	Push Way 1 Pushes all modified lines in way 1 when you write 1 to GO. 0 - No operation 1 - Push all modified lines in way 1
26 INVW1	Invalidate Way 1 Invalidates all lines in way 1 when you write 1 to GO. **NOTE** If PUSHW1 and INVW1 are 1, then after you write 1 to GO, this field pushes all modified lines in way 1 and invalidates all lines in way 1 (clear way 1). 0 - No operation 1 - Invalidate all lines in way 1
25 PUSHW0	Push Way 0 Pushes all modified lines in way 0 when you write 1 to GO. 0 - No operation 1 - Pushes all modified lines in way 0
24 INVW0	Invalidate Way 0 Invalidates all lines in way 0 when you write 1 to GO. **NOTE** If PUSHW0 and INVW0 are 1, then after you write 1 to GO, this field pushes all modified lines in way 0 and invalidates all lines in way 0 (clear way 0). 0 - No operation 1 - Invalidate all lines in way 0
23-4 —	Reserved
3 FRCNOALLC	Force No Allocation on Cache Misses Forces no allocation on cache misses when FRCWT is 1. 0 - Allocate on cache misses 1 - Force no allocation on cache misses
2 FRCWT	Force Write Through Mode Specifies whether all cacheable spaces to write through are forced. 0 - Not force

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Force
1 —	Reserved You must always write 0 to this field to maintain compatibility.
0 ENCACHE	Cache Enable Enables the cache. 0 - Disable 1 - Enable

15.6.3 Cache Line Control (CLCR)

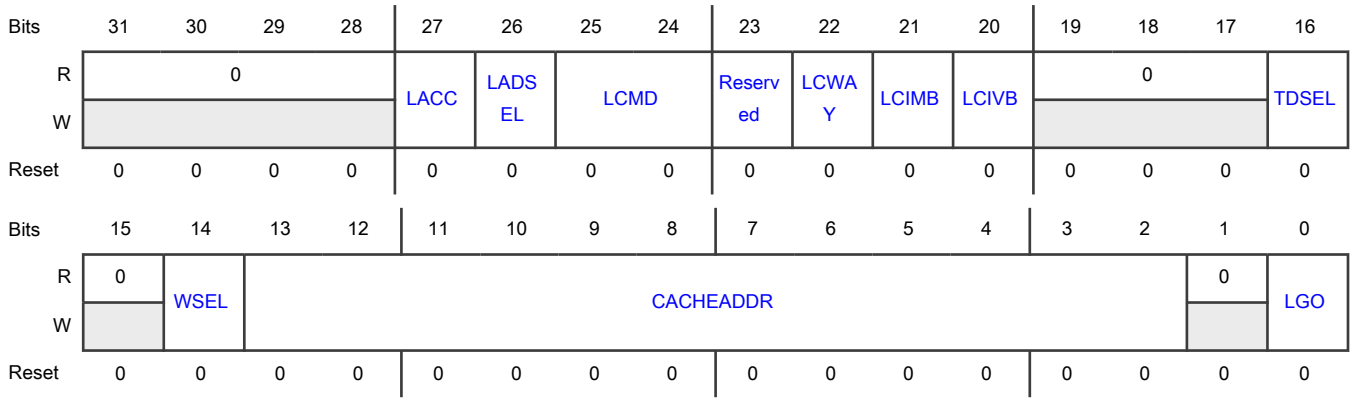
Defines the specific line-sized cache operations that you perform using a specific cache line address or physical address.

If you specify a physical address, both ways of the cache are searched, and the command only performs on the way that hits.

Offset

Register	Offset
CLCR	4h

Diagram



Fields

Field	Description
31-28 —	Reserved
27	Line Access Type

Table continues on the next page...

Table continued from the previous page...

Field	Description
LACC	Specifies the line access type. 0 - Read 1 - Write
26 LADSEL	Line Address Select Specifies the address (cache or physical). When you use the cache address, you must also specify the way in CLCR[WSEL]. When you use the physical address, both ways are searched and the command performs only if there is a hit. 0 - Cache address 1 - Physical address
25-24 LCMD	Line Command Specifies the line command. 00 - Search and read or write 01 - Invalidate 10 - Push 11 - Clear
23 —	Reserved
22 LCWAY	Line Command Way Indicates the way that the line command uses. Applies only if CLCR[LCIVB] = 1.
21 LCIMB	Line Command Initial Modified Specifies the initial state of the modified bit. If the command uses cache address and way, then this field shows the initial state of the modified bit. If the command uses physical address and a hit, then this field shows the initial state of the modified bit. If the command uses a miss, then this field reads zero.
20 LCIVB	Line Command Initial Valid Specifies the initial state of the valid bit. If the command uses cache address and way, then this field shows the initial state of the valid bit. If the command uses physical address and a hit, then this field shows the initial state of the valid bit. If the command uses a miss, then this field reads zero.
19-17	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
16 TDSEL	Tag or Data Select Selects tag or data for search and read or write commands. 0 - Data 1 - Tag
15 —	Reserved
14 WSEL	Way Select Selects the way for line commands. 0 - Way 0 1 - Way 1
13-2 CACHEADDR	Cache Address Represents bits [13:2] of the cache address: CLCR [12:4] accesses the tag arrays. CLCR [12:2] accesses the data arrays.
1 —	Reserved
0 LGO	Initiate Cache Line Command Initiates the cache command that CLCR[LCMD], CLCR[LADSEL], and CLCR[LACC] indicate if you write 1 to this field. Reading this field indicates whether a line command is active. **NOTE** • This field remains 1 until the command completes. Writing 0 to it has no effect. • This field is shared with CSAR[LGO]. 0 - Write: no effect; Read: no line command active 1 - Write: initiate line command; Read: line command active

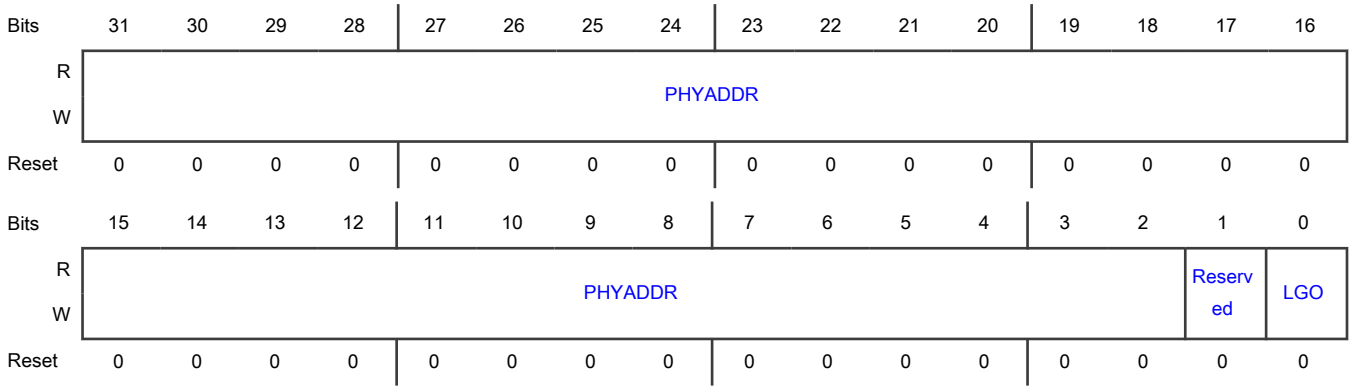
15.6.4 Cache Search Address (CSAR)

Defines the explicit cache address or physical address for line-sized commands specified in CLCR[LADSEL].

Offset

Register	Offset
CSAR	8h

Diagram



Fields

Field	Description
31-2 PHYADDR	Physical Address Represents bits [31:2] of the system address. CSAR [31:13] is used for tag compare. CSAR [12:4] accesses the tag arrays. CSAR [12:2] accesses the data arrays.
1 —	Reserved
0 LGO	Initiate Cache Line Command Initiates the cache line command that CLCR[LCMD], CLCR[LADSEL], and CLCR[LACC] indicate. Reading this field indicates whether a line command is active. **NOTE** • This field remains 1 until the command completes. Writing 0 to it has no effect. • This field is shared with CLCR[LGO]. 0 - Write: no effect; Read: no line command active 1 - Write: initiate line command; Read: line command active

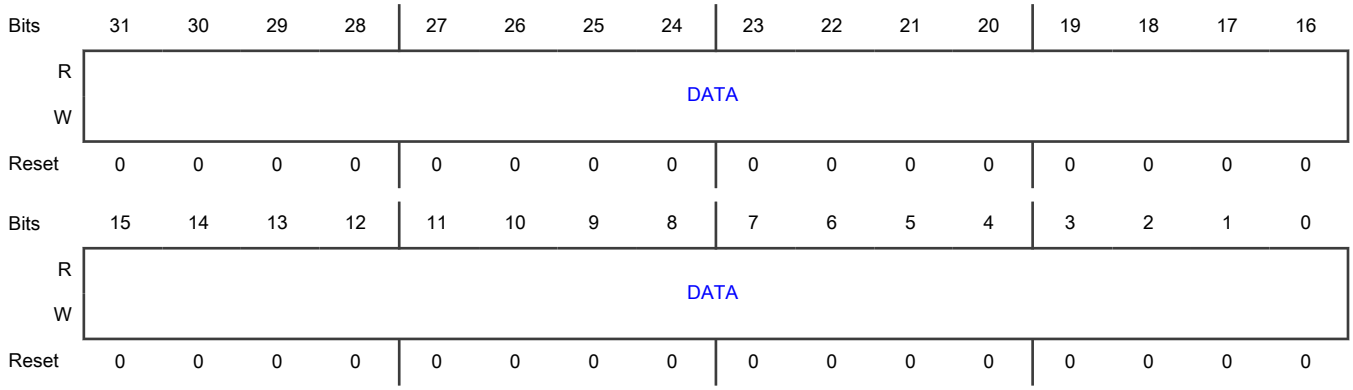
15.6.5 Cache Read/Write Value (CCVR)

Sources write data or returns read data for the commands specified in Cache Line Control (CLCR).

Offset

Register	Offset
CCVR	Ch

Diagram



Fields

Field	Description
31-0 DATA	Cache Read/Write Data Specifies the data that you want to read or write during tag search or data search. For tag search: • Read or write to CCVR [31:13] for tag array R/W value. • Read or write to CCVR [12:4] for tag set address on reads; unused on writes. • CCVR[3:2] are reserved. • CCVR[1] is tag modify bit. • CCVR[0] is tag valid bit. For data search: • Read or write to CCVR[31:0] for data array R/W value.

Chapter 16
CM33 Performance Monitor (CMX-PERFMON)

16.1 Chip-specific CMX-PERFMON information

Table 84. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The chip includes two instances of Performance Monitor (CMX-PERFMON).

16.2 Overview

CMX_PERFMON contains counters that you can configure to count events used to calculate the performance of a CPU, cache, or memory. You can configure CMX_PERFMON to select the events to be counted for each counter. An additional instruction counter is available to count CPU instruction fetches.

Block diagram provides an overview of CMX_PERFMON, which includes three event counters and an instruction counter.

16.2.1 Block diagram

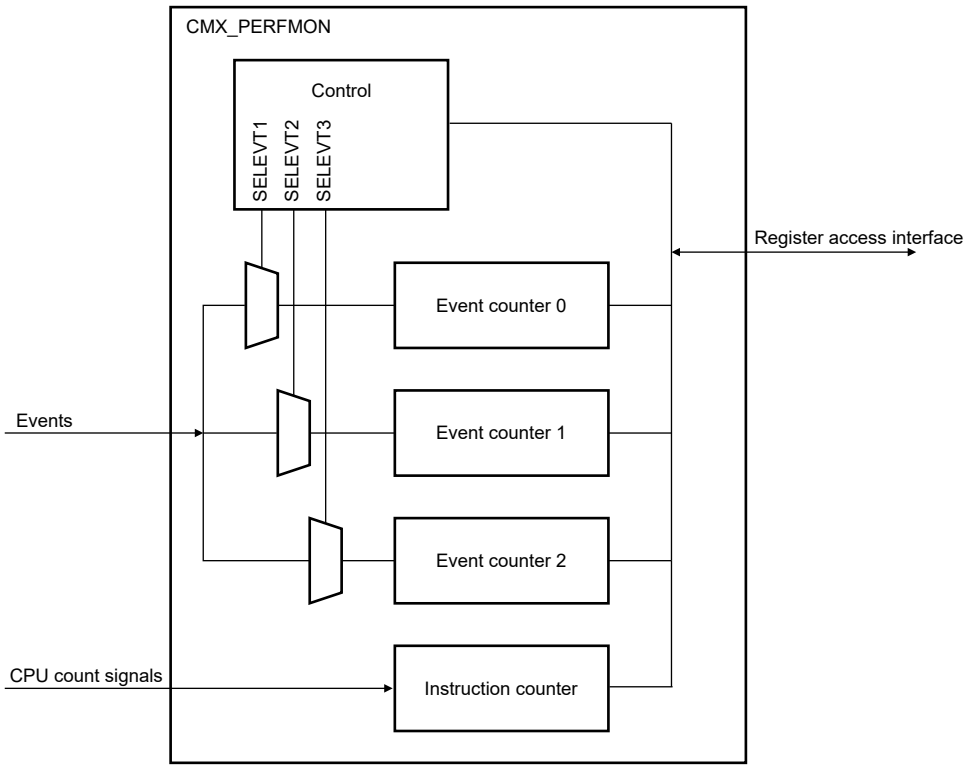


Figure 31. Block diagram

16.2.2 Features

- Three 40-bit event counters
- One 40-bit instruction counter
- Programmable event select
- Countable events that include cache, branch fetches, and stalls
- Memory mapped performance monitor that counts instructions, cache, branch, IPS, and TCM stats
- Count filter based on privilege level

16.3 Functional description

16.3.1 Operations

See the chip-specific CMX_PERFMON information for details about the countable events for the performance counters per CMX_PERFMON instance on this chip.

CMX_PERFMON has 3 programmable counters to count CPU events. The event to be counted is configured using the PMCR*m* register.

Each counter is a 40-bit register that tracks the number of occurrences of the selected event.

The counters can be programmed to take one of the 3 states:

- At idle state, the counters are in idle and do not perform any operation.

- At local start state, the counters are actively listening to CPU and cache and start incrementing when the selected events are observed.
- At local stop state, the counters stop listening to all components. The 3 event counter can be independently cleared by writing to PMCR*m*[RECTR*n*], where *n* denotes the counter instance.

16.3.2 Operating modes

The following table shows:

- The operating modes that you can specify for CMX_PERFMON.
- The PMCR*n*[CMODE] value that you must specify to count events in each mode.

Table 85. Operating modes

Mode	Description	CMODE value (binary)
Privileged	If supported, only counts events in Privilege mode.	11
User	If supported, only counts events in User mode.	10
Both Privileged and User	Counts events in both Privilege and User mode.	00

Use PMCR*n*[SSC] to start and stop the counters.

16.3.3 Clocking

Clock name	Direction	Description
CPU clock (hclk)	Input	Configures the counter logic.
Bus clock (ipg_clk)	Input	Controls the access to the registers.

16.3.4 Interrupts

This module has no interrupts.

16.4 External signals

CMX_PERFMON has no external signals.

16.5 Initialization

To initialize CMX_PERFMON:

1. Program PMCR*m*[SELEVT*n*] to select the specific CPU event to be counted by this counter, where *n* denotes the counter instance.
2. Set PMCR*n*[CMODE] to filter each counter for the selected event while the CPU is in privilege, user, or either mode.
3. Set PMCR*n*[SSC] to program the counter to be in one of idle, local start, or local stop states.
4. Set PMCR*m*[RECTR*n*] to clear the counters independently where *n* denotes the counter instance.

16.6 Memory map and register definition

16.6.1 CMX_PERFMON register descriptions

16.6.1.1 CMX_PERFMON memory map

M33_PCF1 base address: 443E_0000h

M33_PSF1 base address: 443F_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Performance Monitor Control (PMCR)	32	RW	0000_0000
10	Performance Monitor Instruction Counter (PMICTR_HI)	8	R	00
14	Performance Monitor Instruction Counter (PMICTR_LO)	32	R	0000_0000
18	Performance Monitor Event Counter (PMECTR1_HI)	8	R	00
1C	Performance Monitor Event Counter (PMECTR1_LO)	32	R	0000_0000
20	Performance Monitor Event Counter (PMECTR2_HI)	8	R	00
24	Performance Monitor Event Counter (PMECTR2_LO)	32	R	0000_0000
28	Performance Monitor Event Counter (PMECTR3_HI)	8	R	00
2C	Performance Monitor Event Counter (PMECTR3_LO)	32	R	0000_0000

16.6.1.2 Performance Monitor Control (PMCR)

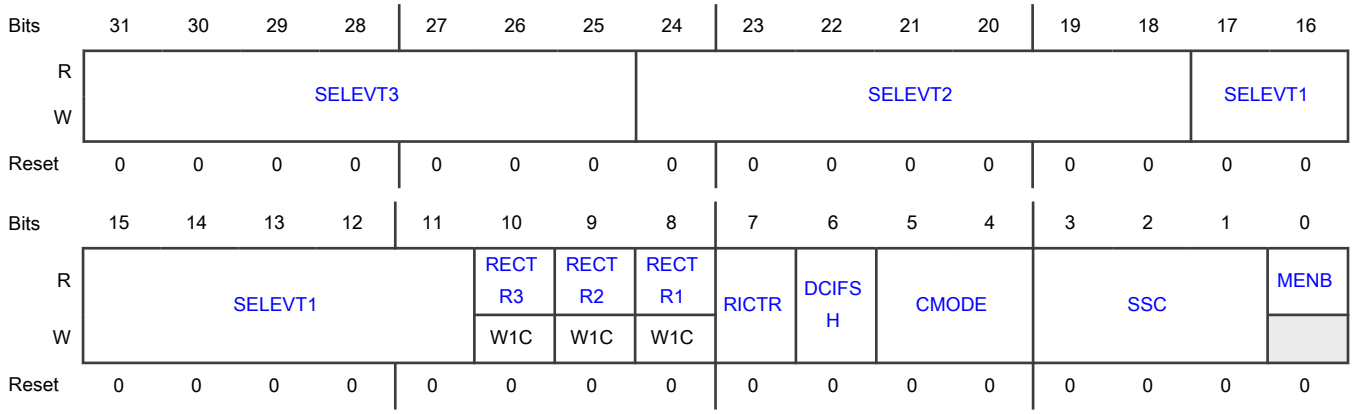
Specifies:

- The events that the PMECTR*n*HI and PMECTR*n*LO count.
- The count mode.
- The start and stop control.
- The enables for the counters.

Offset

Register	Offset
PMCR	0h

Diagram



Fields

Field	Description
31-25 SELEVT3	Select Event 3 Selects the event to be counted in PMECTR3. See the chip-specific CMX_PERFMON information to select which event PMECTR3 counts.
24-18 SELEVT2	Select Event 2 Selects the event to be counted in PMECTR2. See the chip-specific CMX_PERFMON information to select which event PMECTR2 counts.
17-11 SELEVT1	Select Event 1 Selects the event to be counted in PMECTR1. See the chip-specific CMX_PERFMON information to select which event PMECTR1 counts.
10 RECTR3	Reset Event Counter 3 Writing 1 to this field resets the event counter at the end of the cycle. 0 - Run normally 1 - Reset
9 RECTR2	Reset Event Counter 2 Writing 1 to this field resets the event counter at the end of the cycle. 0 - Run normally 1 - Reset
8 RECTR1	Reset Event Counter 1 Writing 1 to this field resets the event counter at the end of the cycle. 0 - Run normally 1 - Reset

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 RICTR	Reset Instruction Counter Determines whether events reset or clear the instruction counter. Write 1 to clear the instruction counter. This field must be 0 to restart the instruction counter. 0 - Do not reset 1 - Clear
6 DCIFSH	Disable Counters if Stopped or Halted Specifies whether counters continue counting or stop counting when the CPU is halted. 0 - Continue 1 - Stop
5-4 CMODE	Count Mode Determines whether events are counted for Privileged mode, User mode, or both Privileged and User modes. This setting affects the operation of all event counters. 00 - Counted in both User and Privileged modes 01 - Reserved 10 - Reserved 11 - Reserved
3-1 SSC	Start and Stop Control Provides a three-phase mechanism to start and stop the counters. It includes a prioritized scheme with the following relative priorities: local start > local stop The start or stop affects all performance monitor event counters concurrently, so that the counters are coherent. 000 - Idle or no-op 001 - Local stop 010,011 - Local start 100 - Reserved 101 - Reserved 110,111 - Reserved
0 MENB	Module Is Enabled Indicates whether CMX_PERFMON is enabled. Enabling CMX_PERFMON does not start the counters. To start the counters, write a nonzero value to PMCR*n*[SSC]. 0 - Disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enabled

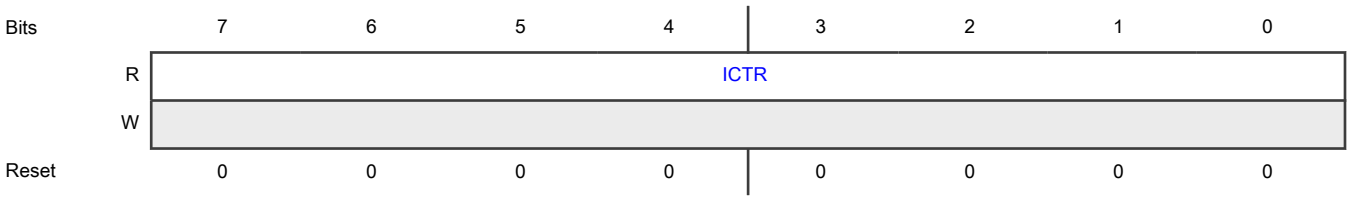
16.6.1.3 Performance Monitor Instruction Counter (PMICTR_HI)

Provides part of a 40-bit counter that counts executed instructions.

Offset

Register	Offset
PMICTR_HI	10h

Diagram



Fields

Field	Description
7-0 ICTR	Instruction Counter Specifies the upper 8 bits of the 40-bit instruction counter that reflects the number of instructions executed. This field is updated based on the instruction execution status, as indicated on the processor ETM interface. The ETM interface indicates when instructions are valid, folded, cancelled or stalled, from which the CMX_PERFMON deduces the overall instruction count.

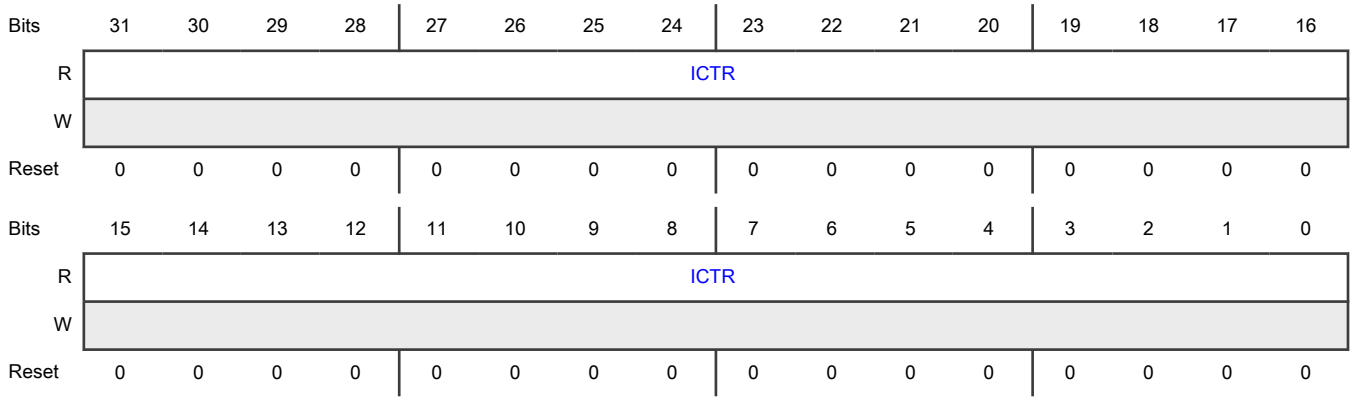
16.6.1.4 Performance Monitor Instruction Counter (PMICTR_LO)

Provides part of a 40-bit counter that counts executed instructions.

Offset

Register	Offset
PMICTR_LO	14h

Diagram



Fields

Field	Description
31-0 ICTR	Instruction Counter Specifies the lower 32 bits of the instruction counter that reflects the number of instructions executed. This field is updated based on the instruction execution status, as indicated on the processor ETM interface. The ETM interface indicates when instructions are valid, folded, cancelled or stalled, from which the CMX_PERFMON deduces the overall instruction count.

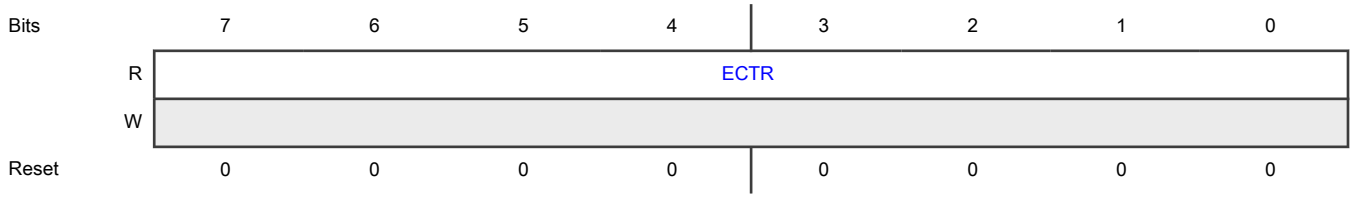
16.6.1.5 Performance Monitor Event Counter (PMECTR1_HI - PMECTR3_HI)

Provides part of a 40-bit event counter that you configure with PMCR*m*[SELEVT*n*].

Offset

Register	Offset
PMECTR1_HI	18h
PMECTR2_HI	20h
PMECTR3_HI	28h

Diagram



Fields

Field	Description
7-0 ECTR	Event Counter Specifies the upper 8 bits of the event[*n*] counter.

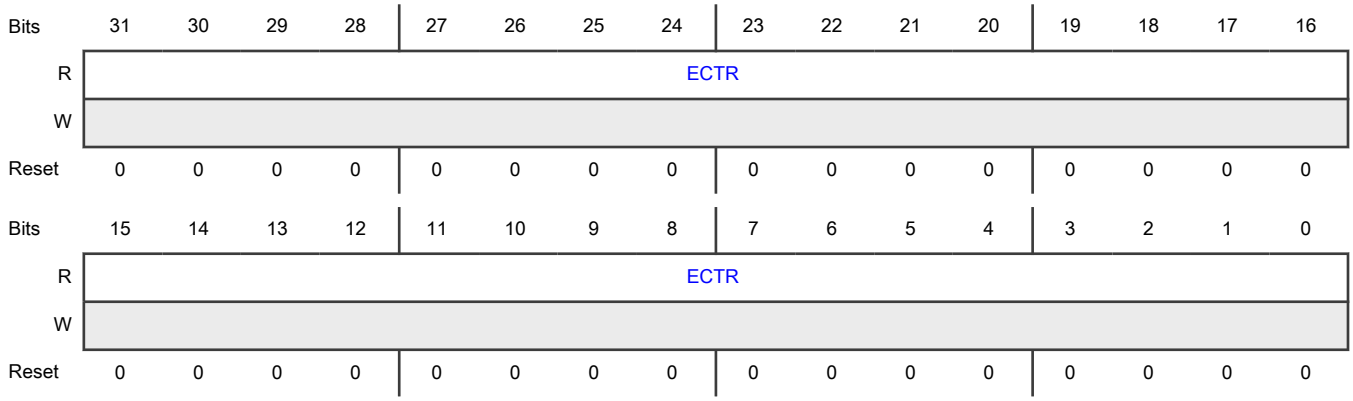
16.6.1.6 Performance Monitor Event Counter (PMECTR1_LO - PMECTR3_LO)

Provides part of a 40-bit event counter that you configure with PMCR*m*[SELEVT*n*].

Offset

Register	Offset
PMECTR1_LO	1Ch
PMECTR2_LO	24h
PMECTR3_LO	2Ch

Diagram



Fields

Field	Description
31-0 ECTR	Event Counter Specifies the lower 32 bits of the event[*n*] counter. The 40-bit counter value increments each time the event selected in PMCR*m*[SELEVT*n*] occurs.

Chapter 17
Neural Processing Unit (NPU)

17.1 Overview

The Neural Processing Unit (NPU) improves the inference performance of neural networks and is supported with an open-source driver and compiler. The NPU targets quantized Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

17.1.1 Block diagram

The following figure shows the main components of NPU.

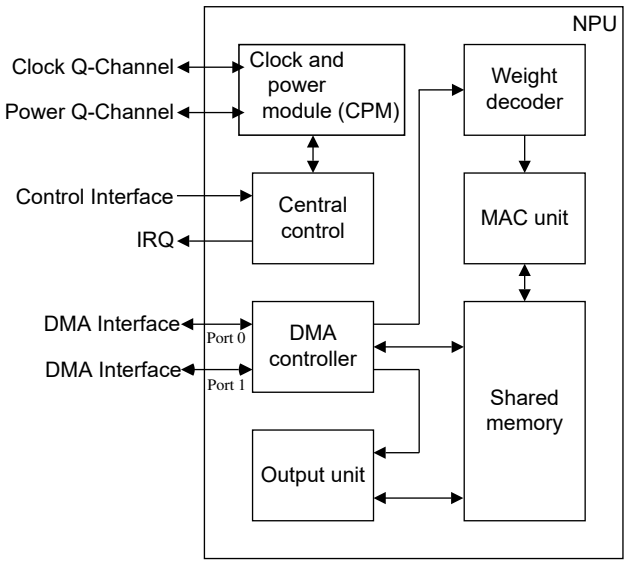


Figure 32. Functional block diagram

17.1.2 Features

- NPU performs 256 8x8 MAC's per cycle.
- NPU targets 8-bit and 16-bit integer RNN
- Handles 8-bit weights

17.2 Functional Description

A neural network must be compiled offline to produce a command stream. The application invokes the driver, which communicates with the NPU to tell it where the command stream is and initiates the network traversal. The command stream describes the steps necessary for the NPU to execute the operators compiled into the command stream autonomously. When complete, the NPU raises an IRQ to the driver. The driver programs the memory location of the command stream and other payloads into registers in the NPU. The Central Control (CC) processes the command stream. The NPU includes a Direct Memory Access (DMA) controller that can read and write to external memory. When the NPU performs inferences, the DMA controller reads the neural network description. This description contains:

- The command stream
- Network weights
- Bias information
- Scale information

The DMA controller also transfers the Input Feature Maps (IFMs) and the Output Feature Maps (OFMs) and NPU-private intermediate data that is also held in system memory.

The NPU consists of the Central Control (CC), a DMA controller, a MAC unit, an Output unit, and the interconnect fabric. The following are descriptions of the units of the NPU:

- The CC receives tasks from the external host application processor. The CC queues and dispatches units of work to the DMA and engines.
- The DMA controller uses its two DMA interfaces to move data between external memory and internal shared memory.
- The MAC unit has various internal units for reading IFMs, performing dot products and accumulations.

17.2.1 External interfaces

The NPU uses two DMA interfaces, a control interface with wake-up signaling, an interrupt interface, two Q-Channel interfaces, clock, and reset to enable access to and from external components.

- **Two DMA interfaces:** These interfaces enable read-access and write-access to external memory for the DMA controller. The NPU has two read/write ports, P0 and P1.
- **Control interface with wake-up signaling:** Enables the device driver that runs on the external host application processor to access the control registers of the NPU.
- **Interrupt interface:** Sends interrupt requests to the external host application processor, usually to signal a completed job.
- **Two Q-Channel interfaces:** These interfaces enable communication with an external clock controller and power controller. This communication enables the system to automatically disable the clock of the NPU or disable the power to it. The clock is otherwise free-running.

17.2.2 Weight Decoder

The Weight Decoder (WD) reads the weight stream from the DMA controller. The decoder decompresses and stores this stream in a double-buffered register, ready for the MAC unit to consume it.

17.2.3 MAC unit

The MAC unit performs multiply-accumulate operations that are required for convolution, depth-wise pooling, vector products, and the max operation required for max pooling.

- **IFM Unit:** The IFM unit inside the MAC unit reads the input feature maps from the shared SRAM and stores them in register slices. These slices are fed into the multipliers in the dot product units. The IFM unit also performs some extra services as part of other operations. The IFM unit handles zero-padding around the outside edge of feature maps and the upscaling that deconvolution requires. Deconvolution upscaling uses nearest neighbor or zero insertion.
- **Dot product units:** The MAC unit contains several dot product units. These dot product units perform the multiply-accumulate operations that are required for convolutions. The dot product units contain a max operator that they use for max pooling.
- **Adder array** The adder array reads a set of accumulators from the shared RAM buffer and updates them with partial accumulations from the dot product units. The adder array then writes the result back. Accuracy is maintained throughout this process. The internal accumulators retain precision so that the output is bit-exact to the software reference, in this case TFL. The compiler selects the accumulator format in the shared buffer. This format can be:
  - 32-bit two's complement
  - 40-bit two's complement

You can also configure the compiler to use 16-bit floating-point format, which improves performance but impacts accuracy.

These formats are only used internally.

17.2.4 Output unit

The Output unit reads finished accumulators from the shared RAM and converts them into output activations. This process includes performing scaling for each OFM, adding the bias to values, and applying the activation function to each point.

Every layer is written to external SRAM, but the activation function and scaling are normally fused. There is no forwarding path from output to input inside the NPU. Although layers can be split into horizontal stripes and run in "cascade" to minimize the SRAM footprint. This means that the external SRAM footprint can be smaller than the largest layer. The activation functions that the Output unit supports are:

- ReLU, ReLU1, ReLU6, and Leaky ReLU
- tanh
- sigmoid
- Configurable Lookup Table (LUT)
- None or bypass

The elementwise operations that the Output unit supports are:

- Elementwise ADD and SUB
- Elementwise Multiplication (MUL)
- Elementwise Min and Max
- Elementwise ABS
- Elementwise Shift Left (SHL) and Elementwise Shift Right (SHR)
- Elementwise Count-leading Zero (CLZ)

When the Output unit has computed output activations, it writes them back into the shared RAM. The output activations are buffered in the shared RAM where they wait for the DMA controller to send them to external memory.

- **Scaling unit:** The Scaling unit in the Output unit performs scaling in convolutions and division in average pooling. The number of scaling operations that are performed per clock depends on the configuration. The number of outputs per clock varies, depending on the operation.
- **ReLU and Leaky ReLU:** Rectified Linear Unit (ReLU) operations are typically performed after scaling and bias addition. The number of ReLU operations that are done in parallel is the same as the number of parallel operations that the Scaling unit performs.

  Leaky ReLU (LReLU) is a variant, a nonzero ReLU with a small positive gradient that targets negative values, unlike standard ReLU functions. Leaky ReLU implements Leaky ReLU as long as the input and output quantization scale are the same. The most recent TensorFlow Lite allows the quantization scale to differ. In that case, we recommend using the LUT for 8-bit activations and element wise operators for 16- bit activations
- **tanh, sigmoid, and LUT:** The Output unit supports tanh and sigmoid functions using a hardwired table combined with bilinear interpolation. The same table is used for both functions, because they are mathematically related. The Output unit can perform one tanh or sigmoid function per cycle.

There is also a Configurable Lookup Table (LUT) that can be used for any point-wise activation or function. For 8-bit activations, the LUT holds up to 256 8-bit values that are directly mapped from IFM to OFM. The LUT size increases to 512 for 16-bit values; however, the outputs are interpolated, bilinear values. The LUT can be configured by setting up a mem2mem transfer.

17.2.5 Clock and Power Module

The Clock and Power Module (CPM) handles hard and soft resets, contains registers for the current security settings, the main clock gate, and the QLPI interface.

- **QLPI for clock:** To enable high-level clock gating, the NPU exposes a Q-Channel slave port. This slave port enables the system to automatically disable the clock of the NPU, that is free-running except during reset. If the entire NPU is in stopped

state, it indicates when the clock can be turned off. You can configure the NPU registers using the control interface, so that it keeps requesting a clock in stopped state.

- **QLPI for power:** For high-level power gating, the NPU exposes a Q-Channel slave port. This slave port permits the system to automatically disable the power of the NPU. If the entire NPU is in stopped state, it indicates when power can be turned off. You can configure the NPU using the control interface, so that it keeps requesting power in stopped state.
- **Clock and power module clock gates:** The CPM contains one main clock gate. Other clock gating is performed inside each of the blocks, which the CPM can override. These clock gates are explicitly instantiated, with the CPM clock gate preceding the block level clock gates.

17.2.6 Clock signals

See SoC Clock Control Module (CCM) Chapter for information on clocking for NPU.

17.2.7 Reset

See SoC System Reset Controller (SRC) Chapter for information on Reset for NPU.

17.2.8 Interrupt signals

See Arm's General Interrupt Controller (GIC) documentation for NPU block interrupts.

17.2.9 Central control

The Central Control (CC) is the main control unit inside the NPU. The CC controls how the NPU processes neural networks, maintains synchronization, and handles data dependencies.

The CC receives tasks from the external host application processor. The CC queues and dispatches units of work to the DMA controller, weight decoder, MAC unit, and Output unit. The DMA controller and MAC unit send events to the CC to signal the completion of work.

The CC contains multiple sets of operation settings to increase efficiency. This enables the CC to set up the next piece of work while the current one is being processed.

After completing scheduling, dispatching, or processing work, the CC checks for any events that have been triggered. If there are no new events, the CC requests underclocking or powerdown, depending on the configuration.

The CC comprises a Traversal unit. The CC instructs this unit to handle commands that require traversal. This unit breaks commands down into smaller commands, performs synchronization as they execute, and implements the different data flows the NPU requires.

The CC comprises a Command unit. This unit receives commands and parses them. Traversal tasks are passed to the traversal unit. Data dependencies can be coded into the NPU command stream by the Offline Compiler, so that data dependencies between commands are not broken. Measuring the data dependency is an NPU internal process.

Other commands can:

- Trigger interrupts.
- Cause the NPU to wait for a data dependency to be cleared.
- Set up internal registers with information relating to the next execution step.

The CC implements a control interface. This interface enables the application processor to control the NPU. This interface also enables performance measurements.

17.2.10 DMA controller

The DMA controller manages all transactions that use the DMA interfaces. The channels that the DMA controller uses are:

- **Command channel:** The NPU uses this channel to read the command stream, normally from external flash. The NPU moves the commands into CC. The application processor activates the command channel when it sets up the location and size of the Command queue. It sets up the Command queue by using the registers that are mapped to the Control Interface.
- **IFM channel:** The NPU uses the IFM channel to read input feature maps and stores them in its shared RAM. Because the shared buffer must store activations from different x,y coordinates in different words, the DMA controller unpacks data which is stored in NHWC format. This might require extra internal buffering, but only for the initial layer of a job. Internal layers can use a more efficient format. The DMA controller considers the kernel stride, because this affects which bank or address the DMA controller requires to store activations. When the DMA controller is in vector-product mode, it supports fetching multiple batches. The IFM channel is triggered once per block for blocks that require input feature maps.
- **OFM channel:** The NPU uses the OFM channel to write output feature maps from shared RAM to external RAM. Because the output is double-buffered in the shared RAM, the DMA requires an interface to synchronize with the output module to notify the DMA which buffer is empty or full. For the last layer of a job, the output must be written out in NHWC format. This may require the DMA to pack the data, depending on the depth of the layer. Since this process reduces the bandwidth, this process is possible in a small register bank inside the DMA. The traversal unit triggers the OFM channel once per output block for blocks that require transfer to external memory.
- **Weight channel:** The weight channel transfers compressed weights from external memory to the weight decoder. The DMA controller uses a read buffer to hide bus latency from the weight decoder and to enable the DMA to handle data arriving out of order. The traversal unit triggers the weight channel for blocks that require the transfer of weights. The weight stream must be quantized to 8 bits or less by an offline tool. When passed through the offline compiler, weights are compressed losslessly and reordered into an NPU-specific weight stream. This process is effective, if the quantizer uses less than 8 bits or if it uses clustering and pruning techniques, it may also employ all three methods. Using lossless compression, an average of ~2 bits is possible in the final weight stream, especially if the weight stream has many zeros.
- **mem2mem channel:** The NPU uses this channel to stream general data from memory to memory. The main purpose of this channel is to read weights from slow, non-volatile memory and store them in the SRAM. This might be performed in preparation for a layer which reads the weights multiple times. Having the weights in SRAM saves power and improves performance compared to reads to nonvolatile memory. The traversal unit triggers mem2mem operations on specific API commands.
- **Bias and scale channel:** This channel streams data to the Output unit. The data that it transmits is the scale and bias necessary for the block that the NPU is processing. Layers that pass through the Output unit are written to the external SRAM. As the layers pass through the Output unit, activation functions can be fused.

NOTE

Only the mem2mem DMA channel is controllable directly by the command stream. The other channels are used to load or store data required by NPU operations. Write DMA channels must always use port 0. Read DMA channels can use port 0 or 1 according to which region is configured for the memory.

17.3 External signals

This module has no external signals.

17.4 Initialization

See Arm's Ethos-U65 NPU documentation for initialization.

17.5 Application information

See Arm's Ethos-U65 NPU documentation for application information.

17.6 Memory map and registers

See Arm's Ethos-U65 NPU documentation for NPU block register description.

Chapter 18
On-chip RAM (OCRAM)

18.1 Overview

The on-chip RAM is implemented as a follower module on the 64-bit system bus. Designed as a simple on-chip memory block, it has one bank of single port SRAM and supports one AXI port with memory banks. For the AXI port, the read and write transactions are handled by two independent modules. The memory banks are organized with the lower two bits of the address which is the AXI bus address and is 64 bits aligned interleaved. This allows a read access and a write access to be processed at the same time if they are targeted to different memory banks.

The OCRAM supports up to 640 Kbytes.

Various options are provided for adding a pipeline or wait-states in a read/write access, in order to ensure flexible timing control at both high and low frequencies.

18.1.1 Block diagram

The OCRAM block diagram is shown in the figure below.

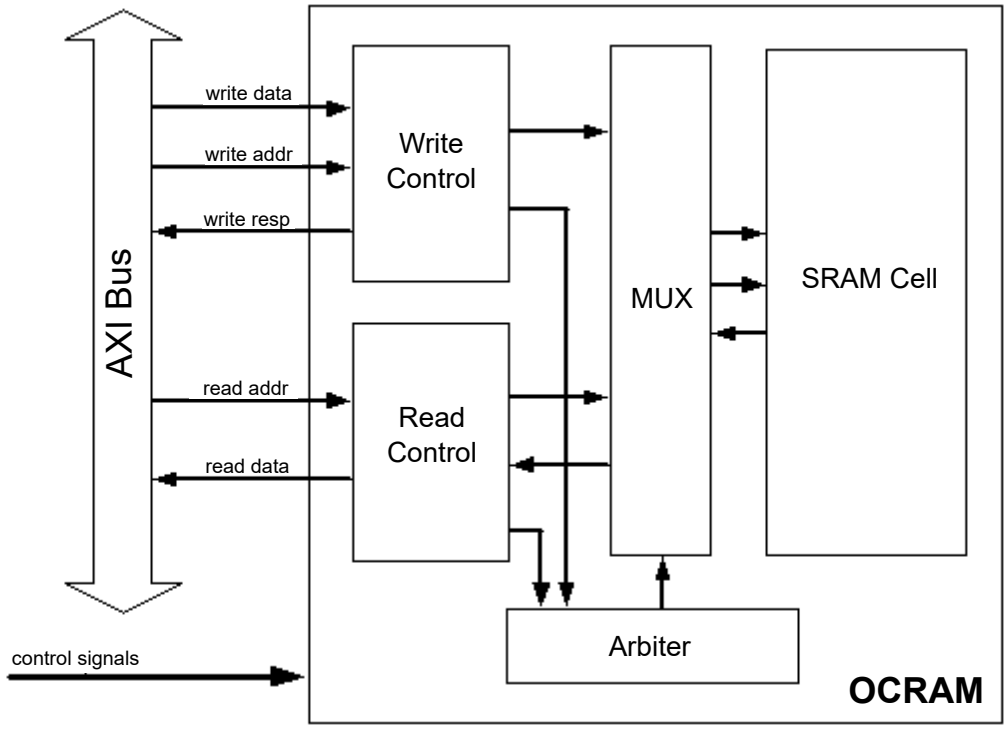


Figure 33. OCRAM block diagram

18.1.2 Features

The OCRAM module provides following features:

- Supports one AXI port with memory banks

- Supports Read/Write arbitration
- Supports TrustZone
- Supports read/write data/address pipeline
- Supports up to four Chip Select (CS)
- Supports up to 640 kBytes

18.2 Functional description

18.2.1 Mode

The following modes are available:

- Read data wait
- Read address pipeline
- Write data pipeline
- Write address pipeline

18.2.2 Memory map

The total on-chip RAM size for the chip is 640 kBytes, organized as 16K x 64 bits. Memory alias is also supported for on-chip RAM, any read/write operation to this area will be mapped back to the 640 kBytes area.

The 32-bit AXI address are arranged as shown in the table below.

Table 86. On-Chip RAM address bits

AXI ADDR bits	Usage	Description
19:7	Memory Address	Selects one of the 160K words in the memory (256-bit word)
6:5	Bank Address	Selects one of the four memory banks (256-bit word)

18.2.3 Read/Write arbitration

The arbiter is used to handle the read/write request from the read control module and write control module. It grants read/write access to the memory cell according to the internal arbitration state and the request/switch signal from the read control module and write control module. Generally, the arbitration is using the round-robin method.

The detailed rules used in arbitration are as follows:

- If there is no granted read or write in the last cycle, and there is only a read request or a write request, the request will be granted.
- If there is no granted read or write in the last cycle, and there are both read or write requests coming in at the same time, the read request will be granted first.
- If a granted read/write transaction has just finished, the write/read request will have the higher priority in the next cycle.
- If the first read/write access request in a transaction is granted, all the data transfer in this burst will be finished before the next arbitration begins, that is, the round-robin arbitration mechanism is based on AXI transaction, not data access.

18.2.4 TrustZone

The OCRAM supports TrustZone. When the SECURE_ENBL bit is set, the STARTADDR and ENDADDR bit-fields in this register establish the region of OCRAM that can only be accessed (both read and write according to the execution mode policy. If this bit is cleared to zero, the entire OCRAM can be accessed in either secure or non-secure mode. The TrustZone bits shows in Programmable registers.

NOTE

The ENDADDR is not configurable and its value is the last address of the OCRAM space. The STARTADDR granularity is 4KB.

18.2.5 Advanced features

This section describes some advanced features designed to avoid timing issues when the OCRAM is working at high frequency.

18.2.5.1 Read data wait state

When the wait state is enabled, it will take 2 cycles for each read access (each beat of a read burst). This can avoid the potential timing problem caused by the longer memory access time at higher frequency. When this feature is disabled, it only takes 1 clock cycle to finish a read transaction. That is, read data is available in the next cycle of read request becomes valid on the bus.

18.2.5.2 Read address pipeline

When this feature is enabled, the read address from the AXI leader is delayed 1 cycle before it can be accepted by the OCRAM. This can avoid setup time issues for the read access on the memory cell at high frequency. Enabling this feature can cost, at most, 1 more clock cycle for each AXI read transaction, that is, at most 1 more clock cycle for each read burst with multiple beats of data.

When this feature is disabled, the read address from the AXI leader can be accepted by the OCRAM without delay, and data can become ready for leader at next clock cycle (if no other access and no read data wait).

18.2.5.3 Write data pipeline

When this feature is enabled, the write data from the AXI leader would be delayed 1 cycle before it can be accepted by the OCRAM. This can avoid setup time issue for the write access on the memory cell at high frequency. Enabling this feature would cost at most 1 more clock cycle for each AXI write transaction, that is, at most 1 more clock cycle for each write burst with multiple beats of data.

When this feature is disabled, the write data from the AXI leader can be accepted by the OCRAM without delay, and data can be written to memory at this cycle (if no other access and write address is also ready at this cycle).

18.2.5.4 Write address pipeline

When this feature is enabled, the write address from the AXI leader would be delayed 1 cycle before it can be accepted by the OCRAM. This can avoid setup time issue for the write access on the memory cell at high frequency.

Enabling this feature would take at most 1 more clock cycle for each AXI write transaction, that is, at most 1 more clock cycle for each write burst with multiple beats of data. When this feature is disabled, the write address from the AXI leader can be accepted by the OCRAM without delay, and data can be written to memory at this cycle (if no other access and write data is also ready at this cycle).

18.2.6 Clocks

The following table describes the clock sources for OCRAM. See Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 87. Clocks

Clock name	Description
clk	Synchronized to AXI bus clock

18.2.7 Reset

There is a hardware reset for the OCRAM Controller, which reset all the logic inside it.

18.2.8 Interrupts

There is no interrupt output from the OCRAM controller.

18.3 External signals

The OCRAM has no external pins.

18.4 Initialization

To initialize the block:

- Ensure the clock for the OCRAM controller is enabled.
- Configure corresponding pin MUX for the OCRAM controller external signals.
- Configure registers of the OCRAM Controller properly, according to the applications.

18.5 Application information

- Configure OCRAM Control Registers with valid settings that matches RAM device.
- Write and read operations can be triggered by AXI command.

18.6 Programmable registers

There are no programmable registers in this block. However, OCRAM related programming should be defined in the chip GPRs/CSRs.

Chapter 19
Memory ECC Controller (MECC256)

19.1 Overview

The Memory ECC Controller (MECC256) module supports Single Error Correction and Double Error Detection (SEC-DED) ECC functions to On-Chip RAM (OCRAM) access. It can generate a 8-bit ECC code by Hsiao Hamming algorithm for 64-bit data. Corresponding OCRAM space is available to store ECC code.

19.1.1 Block diagram

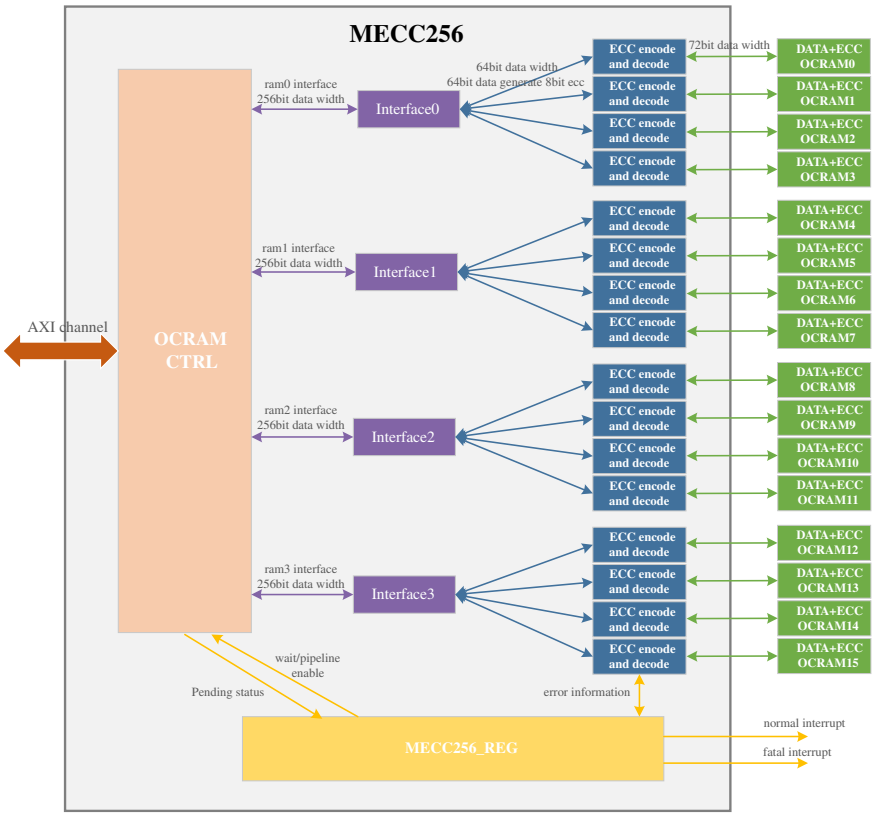


Figure 34. MECC256 block diagram

The following list shows detailed register/bitfield correspondance in the block diagram.

- ecc enable:
  - PIPE_ECC_EN[ECC_EN]
- error injection:
  - ERR_DATA_INJ_LOW/HIGH*n*
  - ERR_ECC_INJ*n*
- error information:

- SINGLE_ERR_ADDR_ECC*n*
- SINGLE_ERR_DATA_LOW/HIGH*n*
- SINGLE_ERR_POS_LOW/HIGH*n*
- MULTI_ERR_ADDR_ECC*n*
- MULTI_ERR_DATA_LOW/HIGH*n*

- pending status:
  - PENDING_STAT
- wait/pipeline enable:
  - PIPE_ECC_EN[READ_DATA_WAIT_EN]
  - PIPE_ECC_EN[READ_ADDR_PIPE_EN]
  - PIPE_ECC_EN[WRITE_DATA_PIPE_EN]
  - PIPE_ECC_EN[WRITE_ADDR_PIPE_EN]

19.1.2 Features

The MECC256 includes the following features:

- Generates a 8-bit ECC code for 64-bit write data. Write data and ECC code are written into seperate OCRAM banks.
- Supports SEC-DED function for 64-bit read data and 8-bit ECC code.
- Integrate 4 banks OCRAM controller and 16 banks memory for converting AXI transaction to RAM interface.
- Error injection on write data and ECC code for debug purpose.
- Two different interrupt types: Normal and Fatal.

19.2 Functional description

The MECC256 converts AXI interface to RAM interface signals, and provides ECC encode/decode function for OCRAM write/read. For each data in OCRAM bank, corresponding space in ECC OCRAM banks is available to store ECC code when ECC function is enabled. When ECC function is disabled, ECC OCRAM banks can also be used to store data.

The OCRAM bank address allocation is as below:

- Bank0: ocram_base_address+0x80*i
- Bank1: ocram_base_address+0x80*i+0x8
- Bank2: ocram_base_address+0x80*i+0x10
- Bank3: ocram_base_address+0x80*i+0x18
- Bank4: ocram_base_address+0x80*i+0x20
- Bank5: ocram_base_address+0x80*i+0x28
- Bank6: ocram_base_address+0x80*i+0x30
- Bank7: ocram_base_address+0x80*i+0x38
- Bank8: ocram_base_address+0x80*i+0x40
- Bank9: ocram_base_address+0x80*i+0x48
- Bank10: ocram_base_address+0x80*i+0x50
- Bank11: ocram_base_address+0x80*i+0x58
- Bank12: ocram_base_address+0x80*i+0x60

- Bank13: ocram_base_address+0x80*i+0x68
- Bank14: ocram_base_address+0x80*i+0x70
- Bank15: ocram_base_address+0x80*i+0x78

where i = 0,1,2,3,4,...

19.2.1 Interface Conversion

The MECC256 integrates 4 banks OCRAM controller which converts AXI master interface signals to 4 groups of OCRAM interface signals (see OCRAM chapter for details). The 4 OCRAM banks are organized with lower 2 bits of address interleaved. This allows a read access and a write access to be processed at the same time if they are targeted to different OCRAM banks. OCRAM controller can support to add pipeline or wait-state in read/write access by register configuration. Each OCRAM bank connects to 4 ECC caluation modules and memory. MECC256 has 16 ECC calculation modules in total. (see block diagram for details)

19.2.2 ECC encode

When ECC is enabled, the 4 groups of OCRAM write access will enter into 16 seperate ECC encode channels. In a ECC encode channel, it will generate 8 bits ECC code for 64 bits OCRAM write data. Then 64 bits OCRAM write access will split into one 64 bits data write and one 8 bits ECC code write. The 64 bits data and 8 bits ecc will be written into one of the OCRAM memory banks according to address.

19.2.3 Error injection

The MECC256 can inject error in write data and ECC code by configuring ERR_DATA_INJ* and ERR_ECC_INJ* registers. The 64 bits write data and 8 bits ECC code can be inverted if corresponding bits in ERR_DATA_INJ* and ERR_ECC_INJ* registers are set. Each OCRAM bank has separate error injection registers. This feature is mainly used for debug.

19.2.4 ECC decode

When ECC is enabled, the 64 bits OCRAM read access will get 64bit read data and 8 bit ECC code from the corresponding OCRAM memory, ECC decode channel can report single bit error or multiple bits error if ECC check is failed. At the same time, the error information will be saved in the module's error registers for debug. When single bit error is detected on read data, the error bit in read data will be corrected. In this case, AXI master will get read data correctly. When multiple bits error is detected on read data, error read data will not be corrected. In this case, master will get read data in error. When single bit error or multiple bits error happens, interrupt will be asserted if the corresponding enable bits are set. Each OCRAM bank has separate error and interrupt registers.

19.2.5 Interrupts

The MECC256 can generate two kinds of interrupts:

- Normal interrupt can be generated for the following cases:
  - Single bit error is corrected for read;
  - AXI access address exceeds OCRAM address range;
  - AXI write access to OCRAM bank is not 64bit/128bit/256bit aligned.
- Fatal interrupt can be generated when 'multiple bits error' is detected for read.

19.2.6 Clocks

There are two clocks in MECC256:

Clock	Description
clk	Main clock
ips_clk	This is for IPS bus, used for register write/read.

There are no special requirements for these two clocks.

19.3 Signals

There are no external signals for MECC.

19.4 Initialization

The steps for initialization process is as below:

1. Enable MECC function (set PIPE_ECC_EN[ECC_EN] = 1).
2. Enable memory initialization (set MEM_INIT[MEM_INIT_ENABLE] = 1).
3. Enable interrupts (set ERR_STAT_EN = 0xFFFF, ERR_SIG_EN = 0xFFFF).

NOTE

- When the MECC function is enabled, the OCRAM memory region can be initialized with 0x00 or 0xFF (initialization data for the first ECC encode), to generate ECC code within OCRAM ECC code region.
  - Initialization is required because the MECC64 needs a complete OCRAM and OCRAM ECC data. When initialized, it generates ECC code depending on initialization data, store data and its ECC code in OCRAM and OCRAM ECC memory.
  - Do such initialization with 64 bits access, with STRD or STMIA instruction. Do not use memset() API, which is STR instruction, in software coding level.
- OCRAM memory with ECC should be non-cacheable, or cacheable with write through, to assure data in OCRAM memory at the write time. It may have different methods to achieve this goal, such as Memory Protection Unit (MPU) or disable cache.

19.5 Application information

There are no application pseudocode examples for MECC256.

19.6 Memory Map and register definition

This section includes the MECC256 module memory map and detailed descriptions of all registers.

19.6.1 MECC256 register descriptions

19.6.1.1 MECC256 memory map

OCRAM_MECC1 base address: 490A_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Error Interrupt Status Register (SINGLE_ERR_STATUS)	32	RW	0000_0000
4	MULTI Error Interrupt Status Register (MULTI_ERR_STATUS)	32	RW	0000_0000
8	STRB Error Interrupt Status Register (STRB_ERR_STATUS)	32	RW	0000_0000
C	ADDR Error Interrupt Status Register (ADDR_ERR_STATUS)	32	RW	0000_0000
10	Single Error Interrupt Status Enable Register (SINGLE_ERR_STAT_EN)	32	RW	0000_0000
14	MULTI Error Interrupt Status Enable Register (MULTI_ERR_STAT_EN)	32	RW	0000_0000
18	STRB Error Interrupt Status Enable Register (STRB_ERR_STAT_EN)	32	RW	0000_0000
1C	ADDR Error Interrupt Status Enable Register (ADDR_ERR_STAT_EN)	32	RW	0000_0000
20	Single Error Interrupt Enable Register (SINGLE_ERR_SIG_EN)	32	RW	0000_0000
24	MULTI Error Interrupt Enable Register (MULTI_ERR_SIG_EN)	32	RW	0000_0000
28	STRB Error Interrupt Enable Register (STRB_ERR_SIG_EN)	32	RW	0000_0000
2C	ADDR Error Interrupt Enable Register (ADDR_ERR_SIG_EN)	32	RW	0000_0000
30	Error Injection On Low 32 bits Of Ocram Bank0 Write Data (ERR_DATA_INJ_LOW0)	32	RW	0000_0000
34	Error Injection On High 32 bits Of Ocram Bank0 Write Data (ERR_DATA_INJ_HIGH0)	32	RW	0000_0000
38	Error Injection On 8 bits ECC code Of Ocram Bank0 Write Data (ERR_ECC_INJ0)	32	RW	0000_0000
3C	Error Injection On Low 32 bits Of Ocram Bank1 Write Data (ERR_DATA_INJ_LOW1)	32	RW	0000_0000
40	Error Injection On High 32 bits Of Ocram Bank1 Write Data (ERR_DATA_INJ_HIGH1)	32	RW	0000_0000
44	Error Injection On 8 bits ECC code Of Ocram Bank1 Write Data (ERR_ECC_INJ1)	32	RW	0000_0000
48	Error Injection On Low 32 bits Of Ocram Bank2 Write Data (ERR_DATA_INJ_LOW2)	32	RW	0000_0000
4C	Error Injection On High 32 bits Of Ocram Bank2 Write Data (ERR_DATA_INJ_HIGH2)	32	RW	0000_0000
50	Error Injection On 8 bits ECC code Of Ocram Bank2 Write Data (ERR_ECC_INJ2)	32	RW	0000_0000
54	Error Injection On Low 32 bits Of Ocram Bank3 Write Data (ERR_DATA_INJ_LOW3)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
58	Error Injection On High 32 bits Of Ocram Bank3 Write Data (ERR_DATA_INJ_HIGH3)	32	RW	0000_0000
5C	Error Injection On 8 bits ECC code Of Ocram Bank3 Write Data (ERR_ECC_INJ3)	32	RW	0000_0000
60	Error Injection On Low 32 bits Of Ocram Bank4 Write Data (ERR_DATA_INJ_LOW4)	32	RW	0000_0000
64	Error Injection On High 32 bits Of Ocram Bank4 Write Data (ERR_DATA_INJ_HIGH4)	32	RW	0000_0000
68	Error Injection On 8 bits ECC code Of Ocram Bank4 Write Data (ERR_ECC_INJ4)	32	RW	0000_0000
6C	Error Injection On Low 32 bits Of Ocram Bank5 Write Data (ERR_DATA_INJ_LOW5)	32	RW	0000_0000
70	Error Injection On High 32 bits Of Ocram Bank5 Write Data (ERR_DATA_INJ_HIGH5)	32	RW	0000_0000
74	Error Injection On 8 bits ECC code Of Ocram Bank5 Write Data (ERR_ECC_INJ5)	32	RW	0000_0000
78	Error Injection On Low 32 bits Of Ocram Bank6 Write Data (ERR_DATA_INJ_LOW6)	32	RW	0000_0000
7C	Error Injection On High 32 bits Of Ocram Bank6 Write Data (ERR_DATA_INJ_HIGH6)	32	RW	0000_0000
80	Error Injection On 8 bits ECC code Of Ocram Bank6 Write Data (ERR_ECC_INJ6)	32	RW	0000_0000
84	Error Injection On Low 32 bits Of Ocram Bank7 Write Data (ERR_DATA_INJ_LOW7)	32	RW	0000_0000
88	Error Injection On High 32 bits Of Ocram Bank7 Write Data (ERR_DATA_INJ_HIGH7)	32	RW	0000_0000
8C	Error Injection On 8 bits ECC code Of Ocram Bank7 Write Data (ERR_ECC_INJ7)	32	RW	0000_0000
90	Error Injection On Low 32 bits Of Ocram Bank8 Write Data (ERR_DATA_INJ_LOW8)	32	RW	0000_0000
94	Error Injection On High 32 bits Of Ocram Bank8 Write Data (ERR_DATA_INJ_HIGH8)	32	RW	0000_0000
98	Error Injection On 8 bits ECC code Of Ocram Bank8 Write Data (ERR_ECC_INJ8)	32	RW	0000_0000
9C	Error Injection On Low 32 bits Of Ocram Bank9 Write Data (ERR_DATA_INJ_LOW9)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
A0	Error Injection On High 32 bits Of Ocram Bank9 Write Data (ERR_DATA_INJ_HIGH9)	32	RW	0000_0000
A4	Error Injection On 8 bits ECC code Of Ocram Bank9 Write Data (ERR_ECC_INJ9)	32	RW	0000_0000
A8	Error Injection On Low 32 bits Of Ocram Bank10 Write Data (ERR_DATA_INJ_LOW10)	32	RW	0000_0000
AC	Error Injection On High 32 bits Of Ocram Bank10 Write Data (ERR_DATA_INJ_HIGH10)	32	RW	0000_0000
B0	Error Injection On 8 bits ECC code Of Ocram Bank10 Write Data (ERR_ECC_INJ10)	32	RW	0000_0000
B4	Error Injection On Low 32 bits Of Ocram Bank11 Write Data (ERR_DATA_INJ_LOW11)	32	RW	0000_0000
B8	Error Injection On High 32 bits Of Ocram Bank11 Write Data (ERR_DATA_INJ_HIGH11)	32	RW	0000_0000
BC	Error Injection On 8 bits ECC code Of Ocram Bank11 Write Data (ERR_ECC_INJ11)	32	RW	0000_0000
C0	Error Injection On Low 32 bits Of Ocram Bank12 Write Data (ERR_DATA_INJ_LOW12)	32	RW	0000_0000
C4	Error Injection On High 32 bits Of Ocram Bank12 Write Data (ERR_DATA_INJ_HIGH12)	32	RW	0000_0000
C8	Error Injection On 8 bits ECC code Of Ocram Bank12 Write Data (ERR_ECC_INJ12)	32	RW	0000_0000
CC	Error Injection On Low 32 bits Of Ocram Bank13 Write Data (ERR_DATA_INJ_LOW13)	32	RW	0000_0000
D0	Error Injection On High 32 bits Of Ocram Bank13 Write Data (ERR_DATA_INJ_HIGH13)	32	RW	0000_0000
D4	Error Injection On 8 bits ECC code Of Ocram Bank13 Write Data (ERR_ECC_INJ13)	32	RW	0000_0000
D8	Error Injection On Low 32 bits Of Ocram Bank14 Write Data (ERR_DATA_INJ_LOW14)	32	RW	0000_0000
DC	Error Injection On High 32 bits Of Ocram Bank14 Write Data (ERR_DATA_INJ_HIGH14)	32	RW	0000_0000
E0	Error Injection On 8 bits ECC code Of Ocram Bank14 Write Data (ERR_ECC_INJ14)	32	RW	0000_0000
E4	Error Injection On Low 32 bits Of Ocram Bank15 Write Data (ERR_DATA_INJ_LOW15)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
E8	Error Injection On High 32 bits Of Ocram Bank15 Write Data (ERR_DATA_INJ_HIGH15)	32	RW	0000_0000
EC	Error Injection On 8 bits ECC code Of Ocram Bank15 Write Data (ERR_ECC_INJ15)	32	RW	0000_0000
F0	Single Error Address And ECC Code On Ocram Bank0 (SINGLE_ERR_ADDR_ECC0)	32	R	0000_0000
F4	Low 32 Bits Single Error Read Data On Ocram Bank0 (SINGLE_ERR_DATA_LOW0)	32	R	0000_0000
F8	High 32 Bits Single Error Read Data On Ocram Bank0 (SINGLE_ERR_DATA_HIGH0)	32	R	0000_0000
FC	Low Single Error Bit Position On Ocram Bank0 (SINGLE_ERR_POS_LOW0)	32	R	0000_0000
100	High Single Error Bit Position On Ocram Bank0 (SINGLE_ERR_POS_HIGH0)	32	R	0000_0000
104	Single Error Address And ECC Code On Ocram Bank1 (SINGLE_ERR_ADDR_ECC1)	32	R	0000_0000
108	Low 32 Bits Single Error Read Data On Ocram Bank1 (SINGLE_ERR_DATA_LOW1)	32	R	0000_0000
10C	High 32 Bits Single Error Read Data On Ocram Bank1 (SINGLE_ERR_DATA_HIGH1)	32	R	0000_0000
110	Low Single Error Bit Position On Ocram Bank1 (SINGLE_ERR_POS_LOW1)	32	R	0000_0000
114	High Single Error Bit Position On Ocram Bank1 (SINGLE_ERR_POS_HIGH1)	32	R	0000_0000
118	Single Error Address And ECC Code On Ocram Bank2 (SINGLE_ERR_ADDR_ECC2)	32	R	0000_0000
11C	Low 32 Bits Single Error Read Data On Ocram Bank2 (SINGLE_ERR_DATA_LOW2)	32	R	0000_0000
120	High 32 Bits Single Error Read Data On Ocram Bank2 (SINGLE_ERR_DATA_HIGH2)	32	R	0000_0000
124	Low Single Error Bit Position On Ocram Bank2 (SINGLE_ERR_POS_LOW2)	32	R	0000_0000
128	High Single Error Bit Position On Ocram Bank2 (SINGLE_ERR_POS_HIGH2)	32	R	0000_0000
12C	Single Error Address And ECC Code On Ocram Bank3 (SINGLE_ERR_ADDR_ECC3)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
130	Low 32 Bits Single Error Read Data On Ocram Bank3 (SINGLE_ERR_DATA_LOW3)	32	R	0000_0000
134	High 32 Bits Single Error Read Data On Ocram Bank3 (SINGLE_ERR_DATA_HIGH3)	32	R	0000_0000
138	Low Single Error Bit Position On Ocram Bank3 (SINGLE_ERR_POS_LOW3)	32	R	0000_0000
13C	High Single Error Bit Position On Ocram Bank3 (SINGLE_ERR_POS_HIGH3)	32	R	0000_0000
140	Single Error Address And ECC Code On Ocram Bank4 (SINGLE_ERR_ADDR_ECC4)	32	R	0000_0000
144	Low 32 Bits Single Error Read Data On Ocram Bank4 (SINGLE_ERR_DATA_LOW4)	32	R	0000_0000
148	High 32 Bits Single Error Read Data On Ocram Bank4 (SINGLE_ERR_DATA_HIGH4)	32	R	0000_0000
14C	Low Single Error Bit Position On Ocram Bank4 (SINGLE_ERR_POS_LOW4)	32	R	0000_0000
150	High Single Error Bit Position On Ocram Bank4 (SINGLE_ERR_POS_HIGH4)	32	R	0000_0000
154	Single Error Address And ECC Code On Ocram Bank5 (SINGLE_ERR_ADDR_ECC5)	32	R	0000_0000
158	Low 32 Bits Single Error Read Data On Ocram Bank5 (SINGLE_ERR_DATA_LOW5)	32	R	0000_0000
15C	High 32 Bits Single Error Read Data On Ocram Bank5 (SINGLE_ERR_DATA_HIGH5)	32	R	0000_0000
160	Low Single Error Bit Position On Ocram Bank5 (SINGLE_ERR_POS_LOW5)	32	R	0000_0000
164	High Single Error Bit Position On Ocram Bank5 (SINGLE_ERR_POS_HIGH5)	32	R	0000_0000
168	Single Error Address And ECC Code On Ocram Bank6 (SINGLE_ERR_ADDR_ECC6)	32	R	0000_0000
16C	Low 32 Bits Single Error Read Data On Ocram Bank6 (SINGLE_ERR_DATA_LOW6)	32	R	0000_0000
170	High 32 Bits Single Error Read Data On Ocram Bank6 (SINGLE_ERR_DATA_HIGH6)	32	R	0000_0000
174	Low Single Error Bit Position On Ocram Bank6 (SINGLE_ERR_POS_LOW6)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
178	High Single Error Bit Position On Ocram Bank6 (SINGLE_ERR_POS_HIGH6)	32	R	0000_0000
17C	Single Error Address And ECC Code On Ocram Bank7 (SINGLE_ERR_ADDR_ECC7)	32	R	0000_0000
180	Low 32 Bits Single Error Read Data On Ocram Bank7 (SINGLE_ERR_DATA_LOW7)	32	R	0000_0000
184	High 32 Bits Single Error Read Data On Ocram Bank7 (SINGLE_ERR_DATA_HIGH7)	32	R	0000_0000
188	Low Single Error Bit Position On Ocram Bank7 (SINGLE_ERR_POS_LOW7)	32	R	0000_0000
18C	High Single Error Bit Position On Ocram Bank7 (SINGLE_ERR_POS_HIGH7)	32	R	0000_0000
190	Single Error Address And ECC Code On Ocram Bank8 (SINGLE_ERR_ADDR_ECC8)	32	R	0000_0000
194	Low 32 Bits Single Error Read Data On Ocram Bank8 (SINGLE_ERR_DATA_LOW8)	32	R	0000_0000
198	High 32 Bits Single Error Read Data On Ocram Bank8 (SINGLE_ERR_DATA_HIGH8)	32	R	0000_0000
19C	Low Single Error Bit Position On Ocram Bank8 (SINGLE_ERR_POS_LOW8)	32	R	0000_0000
1A0	High Single Error Bit Position On Ocram Bank8 (SINGLE_ERR_POS_HIGH8)	32	R	0000_0000
1A4	Single Error Address And ECC Code On Ocram Bank9 (SINGLE_ERR_ADDR_ECC9)	32	R	0000_0000
1A8	Low 32 Bits Single Error Read Data On Ocram Bank9 (SINGLE_ERR_DATA_LOW9)	32	R	0000_0000
1AC	High 32 Bits Single Error Read Data On Ocram Bank9 (SINGLE_ERR_DATA_HIGH9)	32	R	0000_0000
1B0	Low Single Error Bit Position On Ocram Bank9 (SINGLE_ERR_POS_LOW9)	32	R	0000_0000
1B4	High Single Error Bit Position On Ocram Bank9 (SINGLE_ERR_POS_HIGH9)	32	R	0000_0000
1B8	Single Error Address And ECC Code On Ocram Bank10 (SINGLE_ERR_ADDR_ECC10)	32	R	0000_0000
1BC	Low 32 Bits Single Error Read Data On Ocram Bank10 (SINGLE_ERR_DATA_LOW10)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1C0	High 32 Bits Single Error Read Data On Ocram Bank10 (SINGLE_ERR_DATA_HIGH10)	32	R	0000_0000
1C4	Low Single Error Bit Position On Ocram Bank10 (SINGLE_ERR_POS_LOW10)	32	R	0000_0000
1C8	High Single Error Bit Position On Ocram Bank10 (SINGLE_ERR_POS_HIGH10)	32	R	0000_0000
1CC	Single Error Address And ECC Code On Ocram Bank11 (SINGLE_ERR_ADDR_ECC11)	32	R	0000_0000
1D0	Low 32 Bits Single Error Read Data On Ocram Bank11 (SINGLE_ERR_DATA_LOW11)	32	R	0000_0000
1D4	High 32 Bits Single Error Read Data On Ocram Bank11 (SINGLE_ERR_DATA_HIGH11)	32	R	0000_0000
1D8	Low Single Error Bit Position On Ocram Bank11 (SINGLE_ERR_POS_LOW11)	32	R	0000_0000
1DC	High Single Error Bit Position On Ocram Bank11 (SINGLE_ERR_POS_HIGH11)	32	R	0000_0000
1E0	Single Error Address And ECC Code On Ocram Bank12 (SINGLE_ERR_ADDR_ECC12)	32	R	0000_0000
1E4	Low 32 Bits Single Error Read Data On Ocram Bank12 (SINGLE_ERR_DATA_LOW12)	32	R	0000_0000
1E8	High 32 Bits Single Error Read Data On Ocram Bank12 (SINGLE_ERR_DATA_HIGH12)	32	R	0000_0000
1EC	Low Single Error Bit Position On Ocram Bank12 (SINGLE_ERR_POS_LOW12)	32	R	0000_0000
1F0	High Single Error Bit Position On Ocram Bank12 (SINGLE_ERR_POS_HIGH12)	32	R	0000_0000
1F4	Single Error Address And ECC Code On Ocram Bank13 (SINGLE_ERR_ADDR_ECC13)	32	R	0000_0000
1F8	Low 32 Bits Single Error Read Data On Ocram Bank13 (SINGLE_ERR_DATA_LOW13)	32	R	0000_0000
1FC	High 32 Bits Single Error Read Data On Ocram Bank13 (SINGLE_ERR_DATA_HIGH13)	32	R	0000_0000
200	Low Single Error Bit Position On Ocram Bank13 (SINGLE_ERR_POS_LOW13)	32	R	0000_0000
204	High Single Error Bit Position On Ocram Bank13 (SINGLE_ERR_POS_HIGH13)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
208	Single Error Address And ECC Code On Ocram Bank14 (SINGLE_ERR_ADDR_ECC14)	32	R	0000_0000
20C	Low 32 Bits Single Error Read Data On Ocram Bank14 (SINGLE_ERR_DATA_LOW14)	32	R	0000_0000
210	High 32 Bits Single Error Read Data On Ocram Bank14 (SINGLE_ERR_DATA_HIGH14)	32	R	0000_0000
214	Low Single Error Bit Position On Ocram Bank14 (SINGLE_ERR_POS_LOW14)	32	R	0000_0000
218	High Single Error Bit Position On Ocram Bank14 (SINGLE_ERR_POS_HIGH14)	32	R	0000_0000
21C	Single Error Address And ECC Code On Ocram Bank15 (SINGLE_ERR_ADDR_ECC15)	32	R	0000_0000
220	Low 32 Bits Single Error Read Data On Ocram Bank15 (SINGLE_ERR_DATA_LOW15)	32	R	0000_0000
224	High 32 Bits Single Error Read Data On Ocram Bank15 (SINGLE_ERR_DATA_HIGH15)	32	R	0000_0000
228	Low Single Error Bit Position On Ocram Bank15 (SINGLE_ERR_POS_LOW15)	32	R	0000_0000
22C	High Single Error Bit Position On Ocram Bank15 (SINGLE_ERR_POS_HIGH15)	32	R	0000_0000
230	Multiple Error Address And ECC Code On Ocram Bank0 (MULTI_ERR_ADDR_ECC0)	32	R	0000_0000
234	Low 32 Bits Multiple Error Read Data On Ocram Bank0 (MULTI_ERR_DATA_LOW0)	32	R	0000_0000
238	High 32 Bits Multiple Error Read Data On Ocram Bank0 (MULTI_ERR_DATA_HIGH0)	32	R	0000_0000
23C	Multiple Error Address And ECC Code On Ocram Bank1 (MULTI_ERR_ADDR_ECC1)	32	R	0000_0000
240	Low 32 Bits Multiple Error Read Data On Ocram Bank1 (MULTI_ERR_DATA_LOW1)	32	R	0000_0000
244	High 32 Bits Multiple Error Read Data On Ocram Bank1 (MULTI_ERR_DATA_HIGH1)	32	R	0000_0000
248	Multiple Error Address And ECC Code On Ocram Bank2 (MULTI_ERR_ADDR_ECC2)	32	R	0000_0000
24C	Low 32 Bits Multiple Error Read Data On Ocram Bank2 (MULTI_ERR_DATA_LOW2)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
250	High 32 Bits Multiple Error Read Data On Ocram Bank2 (MULTI_ERR_DATA_HIGH2)	32	R	0000_0000
254	Multiple Error Address And ECC Code On Ocram Bank3 (MULTI_ERR_ADDR_ECC3)	32	R	0000_0000
258	Low 32 Bits Multiple Error Read Data On Ocram Bank3 (MULTI_ERR_DATA_LOW3)	32	R	0000_0000
25C	High 32 Bits Multiple Error Read Data On Ocram Bank3 (MULTI_ERR_DATA_HIGH3)	32	R	0000_0000
260	Multiple Error Address And ECC Code On Ocram Bank4 (MULTI_ERR_ADDR_ECC4)	32	R	0000_0000
264	Low 32 Bits Multiple Error Read Data On Ocram Bank4 (MULTI_ERR_DATA_LOW4)	32	R	0000_0000
268	High 32 Bits Multiple Error Read Data On Ocram Bank4 (MULTI_ERR_DATA_HIGH4)	32	R	0000_0000
26C	Multiple Error Address And ECC Code On Ocram Bank5 (MULTI_ERR_ADDR_ECC5)	32	R	0000_0000
270	Low 32 Bits Multiple Error Read Data On Ocram Bank5 (MULTI_ERR_DATA_LOW5)	32	R	0000_0000
274	High 32 Bits Multiple Error Read Data On Ocram Bank5 (MULTI_ERR_DATA_HIGH5)	32	R	0000_0000
278	Multiple Error Address And ECC Code On Ocram Bank6 (MULTI_ERR_ADDR_ECC6)	32	R	0000_0000
27C	Low 32 Bits Multiple Error Read Data On Ocram Bank6 (MULTI_ERR_DATA_LOW6)	32	R	0000_0000
280	High 32 Bits Multiple Error Read Data On Ocram Bank6 (MULTI_ERR_DATA_HIGH6)	32	R	0000_0000
284	Multiple Error Address And ECC Code On Ocram Bank7 (MULTI_ERR_ADDR_ECC7)	32	R	0000_0000
288	Low 32 Bits Multiple Error Read Data On Ocram Bank7 (MULTI_ERR_DATA_LOW7)	32	R	0000_0000
28C	High 32 Bits Multiple Error Read Data On Ocram Bank7 (MULTI_ERR_DATA_HIGH7)	32	R	0000_0000
290	Multiple Error Address And ECC Code On Ocram Bank8 (MULTI_ERR_ADDR_ECC8)	32	R	0000_0000
294	Low 32 Bits Multiple Error Read Data On Ocram Bank8 (MULTI_ERR_DATA_LOW8)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
298	High 32 Bits Multiple Error Read Data On Ocram Bank8 (MULTI_ERR_DATA_HIGH8)	32	R	0000_0000
29C	Multiple Error Address And ECC Code On Ocram Bank9 (MULTI_ERR_ADDR_ECC9)	32	R	0000_0000
2A0	Low 32 Bits Multiple Error Read Data On Ocram Bank9 (MULTI_ERR_DATA_LOW9)	32	R	0000_0000
2A4	High 32 Bits Multiple Error Read Data On Ocram Bank9 (MULTI_ERR_DATA_HIGH9)	32	R	0000_0000
2A8	Multiple Error Address And ECC Code On Ocram Bank10 (MULTI_ERR_ADDR_ECC10)	32	R	0000_0000
2AC	Low 32 Bits Multiple Error Read Data On Ocram Bank10 (MULTI_ERR_DATA_LOW10)	32	R	0000_0000
2B0	High 32 Bits Multiple Error Read Data On Ocram Bank10 (MULTI_ERR_DATA_HIGH10)	32	R	0000_0000
2B4	Multiple Error Address And ECC Code On Ocram Bank11 (MULTI_ERR_ADDR_ECC11)	32	R	0000_0000
2B8	Low 32 Bits Multiple Error Read Data On Ocram Bank11 (MULTI_ERR_DATA_LOW11)	32	R	0000_0000
2BC	High 32 Bits Multiple Error Read Data On Ocram Bank11 (MULTI_ERR_DATA_HIGH11)	32	R	0000_0000
2C0	Multiple Error Address And ECC Code On Ocram Bank12 (MULTI_ERR_ADDR_ECC12)	32	R	0000_0000
2C4	Low 32 Bits Multiple Error Read Data On Ocram Bank12 (MULTI_ERR_DATA_LOW12)	32	R	0000_0000
2C8	High 32 Bits Multiple Error Read Data On Ocram Bank12 (MULTI_ERR_DATA_HIGH12)	32	R	0000_0000
2CC	Multiple Error Address And ECC Code On Ocram Bank13 (MULTI_ERR_ADDR_ECC13)	32	R	0000_0000
2D0	Low 32 Bits Multiple Error Read Data On Ocram Bank13 (MULTI_ERR_DATA_LOW13)	32	R	0000_0000
2D4	High 32 Bits Multiple Error Read Data On Ocram Bank13 (MULTI_ERR_DATA_HIGH13)	32	R	0000_0000
2D8	Multiple Error Address And ECC Code On Ocram Bank14 (MULTI_ERR_ADDR_ECC14)	32	R	0000_0000
2DC	Low 32 Bits Multiple Error Read Data On Ocram Bank14 (MULTI_ERR_DATA_LOW14)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2E0	High 32 Bits Multiple Error Read Data On Ocram Bank14 (MULTI_ERR_DATA_HIGH14)	32	R	0000_0000
2E4	Multiple Error Address And ECC Code On Ocram Bank15 (MULTI_ERR_ADDR_ECC15)	32	R	0000_0000
2E8	Low 32 Bits Multiple Error Read Data On Ocram Bank15 (MULTI_ERR_DATA_LOW15)	32	R	0000_0000
2EC	High 32 Bits Multiple Error Read Data On Ocram Bank15 (MULTI_ERR_DATA_HIGH15)	32	R	0000_0000
2F0	Ocram Pipeline And ECC Enable (PIPE_ECC_EN)	32	RW	0000_0010
2F4	Pending Status (PENDING_STAT)	32	R	0000_0000
2F8	Memory initialization (MEM_INIT)	32	RW	0000_0000

19.6.1.2 Error Interrupt Status Register (SINGLE_ERR_STATUS)

Error Interrupt Status Register.

Offset

Register	Offset
SINGLE_ERR_STATUS	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
15 SINGLE_ERR15	Ocram Access Error On Bank3 When OCRAM access unallocated address on bank3, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Ocram access error does not happen on ocram bank3. 1 - Ocram access error happens on ocram bank3.
14 SINGLE_ERR14	Ocram Access Error On Bank2 When OCRAM access unallocated address on bank2, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Ocram access error does not happen on ocram bank2. 1 - Ocram access error happens on ocram bank2.
13 SINGLE_ERR13	Ocram Access Error On Bank1 When OCRAM access unallocated address on bank1, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Ocram access error does not happen on ocram bank1. 1 - Ocram access error happens on ocram bank1.
12 SINGLE_ERR12	Ocram Access Error On Bank0 When OCRAM access unallocated address on bank0, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Ocram access error does not happen on ocram bank0. 1 - Ocram access error happens on ocram bank0.
11 SINGLE_ERR11	AXI Strobe Error On Ocram Bank3 When AXI write access to ocram bank3 is not full size(wstrb=0xff), this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - AXI strobe error does not happen on ocram bank3. 1 - AXI strobe error happens on ocram bank3.
10 SINGLE_ERR10	AXI Strobe Error On Ocram Bank2 When AXI write access to ocram bank2 is not full size(wstrb=0xff), this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - AXI strobe error does not happen on ocram bank2. 1 - AXI strobe error happens on ocram bank2.
9 SINGLE_ERR9	AXI Strobe Error On Ocram Bank1

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When AXI write access to ocram bank1 is not full size(wstrb=0xff), this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - AXI strobe error does not happen on ocram bank1. 1 - AXI strobe error happens on ocram bank1.
8 SINGLE_ERR8	AXI Strobe Error On Ocram Bank0 When AXI write access to ocram bank0 is not full size(wstrb=0xff), this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - AXI strobe error does not happen on ocram bank0. 1 - AXI strobe error happens on ocram bank0.
7 SINGLE_ERR7	Multiple Bits Error On Ocram Bank3 When detect multiple bits error on read data from ocram bank3, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Multiple bits error does not happen on ocram bank3. 1 - Multiple bits error happens on ocram bank3.
6 SINGLE_ERR6	Multiple Bits Error On Ocram Bank2 When detect multiple bits error on read data from ocram bank2, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Multiple bits error does not happen on ocram bank2. 1 - Multiple bits error happens on ocram bank2.
5 SINGLE_ERR5	Multiple Bits Error On Ocram Bank1 When detect multiple bits error on read data from ocram bank1, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Multiple bits error does not happen on ocram bank1. 1 - Multiple bits error happens on ocram bank1.
4 SINGLE_ERR4	Multiple Bits Error On Ocram Bank0 When detect multiple bits error on read data from ocram bank0, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Multiple bits error does not happen on ocram bank0. 1 - Multiple bits error happens on ocram bank0.
3 SINGLE_ERR3	Single Bit Error On Ocram Bank3 When detect single bit error on read data from ocram bank3, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Single bit error does not happen on ocram bank3. 1 - Single bit error happens on ocram bank3.

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 SINGLE_ERR2	Single Bit Error On Ocram Bank2 When detect single bit error on read data from ocram bank2, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Single bit error does not happen on ocram bank2. 1 - Single bit error happens on ocram bank2.
1 SINGLE_ERR1	Single Bit Error On Ocram Bank1 When detect single bit error on read data from ocram bank1, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Single bit error does not happen on ocram bank1. 1 - Single bit error happens on ocram bank1.
0 SINGLE_ERR0	Single Bit Error On Ocram Bank0 When detect single bit error on read data from ocram bank0, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Single bit error does not happen on ocram bank0. 1 - Single bit error happens on ocram bank0.

19.6.1.3 MULTI Error Interrupt Status Register (MULTI_ERR_STATUS)

Offset

Register	Offset
MULTI_ERR_STATUS	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...	MULTI_E...
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 MULTI_ERR15	
14 MULTI_ERR14	
13 MULTI_ERR13	
12 MULTI_ERR12	
11 MULTI_ERR11	
10 MULTI_ERR10	
9 MULTI_ERR9	
8 MULTI_ERR8	
7 MULTI_ERR7	
6 MULTI_ERR6	
5 MULTI_ERR5	
4 MULTI_ERR4	
3 MULTI_ERR3	
2	

Table continues on the next page...

Table continued from the previous page...

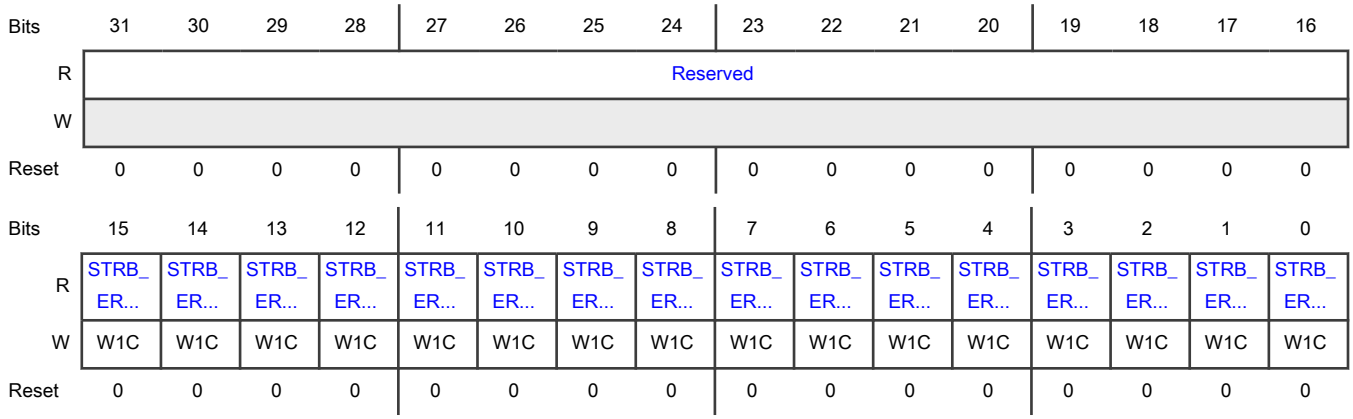
Field	Description
MULTI_ERR2	
1 MULTI_ERR1	
0 MULTI_ERR0	

19.6.1.4 STRB Error Interrupt Status Register (STRB_ERR_STATUS)

Offset

Register	Offset
STRB_ERR_STATUS	8h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 STRB_ERR15	
14 STRB_ERR14	

Table continues on the next page...

Table continued from the previous page...

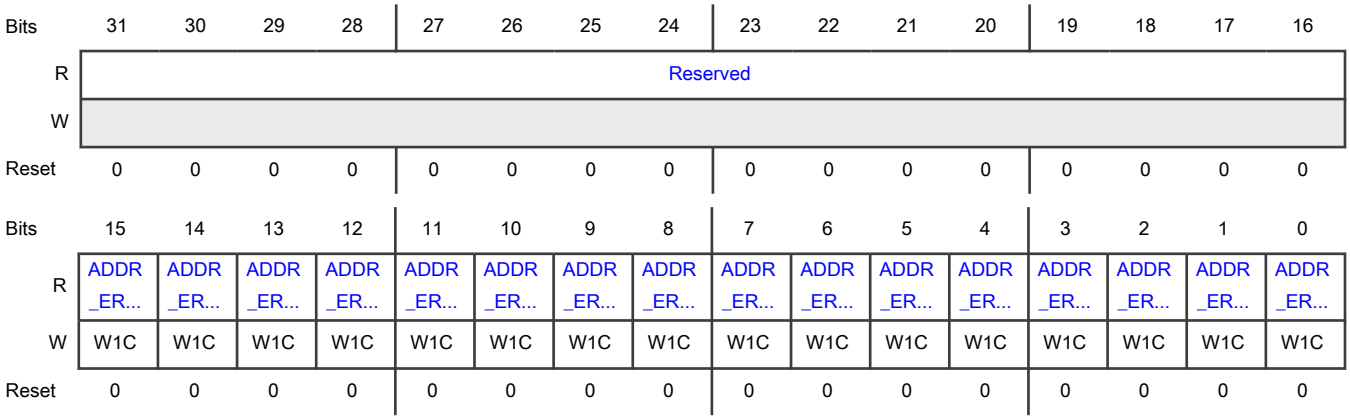
Field	Description
13 STRB_ERR13	
12 STRB_ERR12	
11 STRB_ERR11	
10 STRB_ERR10	
9 STRB_ERR9	
8 STRB_ERR8	
7 STRB_ERR7	
6 STRB_ERR6	
5 STRB_ERR5	
4 STRB_ERR4	
3 STRB_ERR3	
2 STRB_ERR2	
1 STRB_ERR1	
0 STRB_ERR0	

19.6.1.5 ADDR Error Interrupt Status Register (ADDR_ERR_STATUS)

Offset

Register	Offset
ADDR_ERR_STATUS	Ch

Diagram



Fields

Field	Description
31-16 —	Reserved
15 ADDR_ERR15	
14 ADDR_ERR14	
13 ADDR_ERR13	
12 ADDR_ERR12	
11 ADDR_ERR11	
10 ADDR_ERR10	
9 ADDR_ERR9	

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 ADDR_ERR8	
7 ADDR_ERR7	
6 ADDR_ERR6	
5 ADDR_ERR5	
4 ADDR_ERR4	
3 ADDR_ERR3	
2 ADDR_ERR2	
1 ADDR_ERR1	
0 ADDR_ERR0	

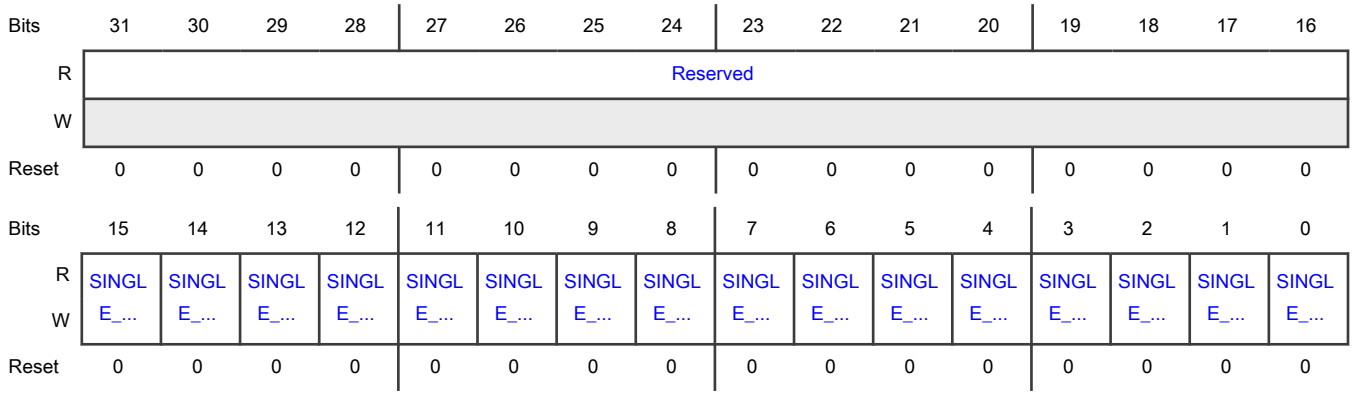
19.6.1.6 Single Error Interrupt Status Enable Register (SINGLE_ERR_STAT_EN)

Setting the bits in this register to 1 enables the corresponding Error Interrupt Status to be set by the specified event.

Offset

Register	Offset
SINGLE_ERR_STAT_EN	10h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 SINGLE_ERR15_STAT_EN	Ocram Access Error Status Enable On Bank3 0 - Masked 1 - Enabled
14 SINGLE_ERR14_STAT_EN	Ocram Access Error Status Enable On Bank2 0 - Masked 1 - Enabled
13 SINGLE_ERR13_STAT_EN	Ocram Access Error Status Enable On Bank1 0 - Masked 1 - Enabled
12 SINGLE_ERR12_STAT_EN	Ocram Access Error Status Enable On Bank0 0 - Masked 1 - Enabled
11 SINGLE_ERR11_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank3 0 - Masked 1 - Enabled
10 SINGLE_ERR10_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank2 0 - Masked 1 - Enabled
9	AXI Strobe Error Status Enable On Ocram Bank1 0 - Masked

Table continues on the next page...

Table continued from the previous page...

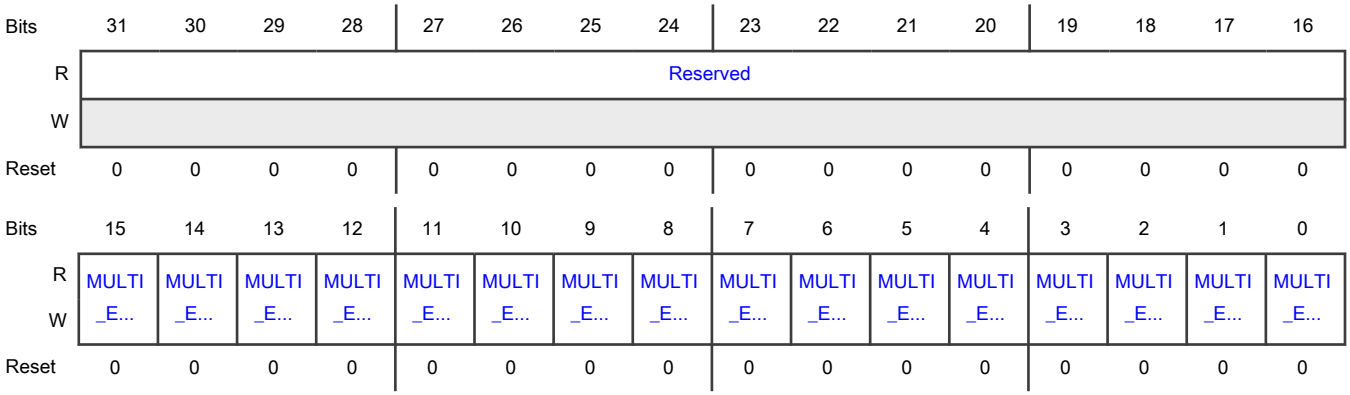
Field	Description
SINGLE_ERR9_STAT_EN	1 - Enabled
8 SINGLE_ERR8_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank0 0 - Masked 1 - Enabled
7 SINGLE_ERR7_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank3 0 - Masked 1 - Enabled
6 SINGLE_ERR6_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank2 0 - Masked 1 - Enabled
5 SINGLE_ERR5_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank1 0 - Masked 1 - Enabled
4 SINGLE_ERR4_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank0 0 - Masked 1 - Enabled
3 SINGLE_ERR3_STAT_EN	Single Bit Error Status Enable On Ocram Bank3 0 - Masked 1 - Enabled
2 SINGLE_ERR2_STAT_EN	Single Bit Error Status Enable On Ocram Bank2 0 - Masked 1 - Enabled
1 SINGLE_ERR1_STAT_EN	Single Bit Error Status Enable On Ocram Bank1 0 - Masked 1 - Enabled
0 SINGLE_ERR0_STAT_EN	Single Bit Error Status Enable On Ocram Bank0 0 - Masked 1 - Enabled

19.6.1.7 MULTI Error Interrupt Status Enable Register (MULTI_ERR_STAT_EN)

Offset

Register	Offset
MULTI_ERR_STAT_EN	14h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 MULTI_ERR15_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank3 0 - Masked 1 - Enabled
14 MULTI_ERR14_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank2 0 - Masked 1 - Enabled
13 MULTI_ERR13_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank1 0 - Masked 1 - Enabled
12 MULTI_ERR12_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank0 0 - Masked 1 - Enabled
11 MULTI_ERR11_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank3 0 - Masked 1 - Enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
10 MULTI_ERR10_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank2 0 - Masked 1 - Enabled
9 MULTI_ERR9_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank1 0 - Masked 1 - Enabled
8 MULTI_ERR8_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank0 0 - Masked 1 - Enabled
7 MULTI_ERR7_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank3 0 - Masked 1 - Enabled
6 MULTI_ERR6_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank2 0 - Masked 1 - Enabled
5 MULTI_ERR5_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank1 0 - Masked 1 - Enabled
4 MULTI_ERR4_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank0 0 - Masked 1 - Enabled
3 MULTI_ERR3_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank3 0 - Masked 1 - Enabled
2 MULTI_ERR2_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank2 0 - Masked 1 - Enabled
1 MULTI_ERR1_STAT_EN	Multiple Bits Error Status Enable On Ocram Bank1 0 - Masked 1 - Enabled
0	Multiple Bits Error Status Enable On Ocram Bank0 0 - Masked

Table continues on the next page...

Table continued from the previous page...

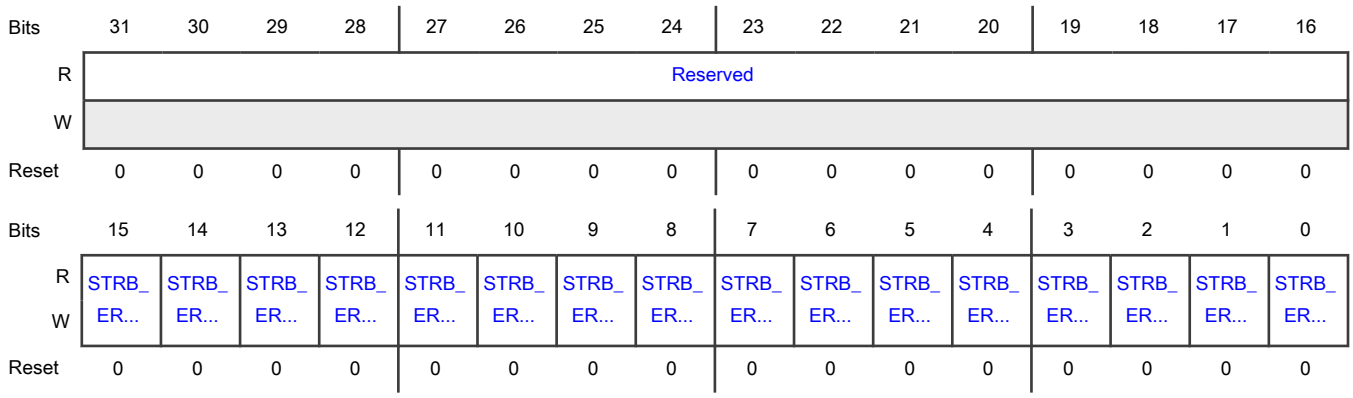
Field	Description
MULTI_ERR0_STAT_EN	1 - Enabled

19.6.1.8 STRB Error Interrupt Status Enable Register (STRB_ERR_STAT_EN)

Offset

Register	Offset
STRB_ERR_STAT_EN	18h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 STRB_ERR15_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank3 0 - Masked 1 - Enabled
14 STRB_ERR14_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank2 0 - Masked 1 - Enabled
13	AXI Strobe Error Status Enable On Ocram Bank1 0 - Masked

Table continues on the next page...

Table continued from the previous page...

Field	Description
STRB_ERR13_STAT_EN	1 - Enabled
12 STRB_ERR12_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank0 0 - Masked 1 - Enabled
11 STRB_ERR11_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank3 0 - Masked 1 - Enabled
10 STRB_ERR10_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank2 0 - Masked 1 - Enabled
9 STRB_ERR9_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank1 0 - Masked 1 - Enabled
8 STRB_ERR8_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank0 0 - Masked 1 - Enabled
7 STRB_ERR7_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank3 0 - Masked 1 - Enabled
6 STRB_ERR6_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank2 0 - Masked 1 - Enabled
5 STRB_ERR5_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank1 0 - Masked 1 - Enabled
4 STRB_ERR4_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank0 0 - Masked 1 - Enabled
3 STRB_ERR3_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank3 0 - Masked 1 - Enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 STRB_ERR2_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank2 0 - Masked 1 - Enabled
1 STRB_ERR1_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank1 0 - Masked 1 - Enabled
0 STRB_ERR0_STAT_EN	AXI Strobe Error Status Enable On Ocram Bank0 0 - Masked 1 - Enabled

19.6.1.9 ADDR Error Interrupt Status Enable Register (ADDR_ERR_STAT_EN)

Offset

Register	Offset
ADDR_ERR_STAT_EN	1Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...	ADDR_ER...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15	Ocram Access Error Status Enable On Bank3

Table continues on the next page...

Table continued from the previous page...

Field	Description
ADDR_ERR15_STAT_EN	0 - Masked 1 - Enabled
14 ADDR_ERR14_STAT_EN	Ocram Access Error Status Enable On Bank2 0 - Masked 1 - Enabled
13 ADDR_ERR13_STAT_EN	Ocram Access Error Status Enable On Bank1 0 - Masked 1 - Enabled
12 ADDR_ERR12_STAT_EN	Ocram Access Error Status Enable On Bank0 0 - Masked 1 - Enabled
11 ADDR_ERR11_STAT_EN	Ocram Access Error Status Enable On Bank3 0 - Masked 1 - Enabled
10 ADDR_ERR10_STAT_EN	Ocram Access Error Status Enable On Bank2 0 - Masked 1 - Enabled
9 ADDR_ERR9_STAT_EN	Ocram Access Error Status Enable On Bank1 0 - Masked 1 - Enabled
8 ADDR_ERR8_STAT_EN	Ocram Access Error Status Enable On Bank0 0 - Masked 1 - Enabled
7 ADDR_ERR7_STAT_EN	Ocram Access Error Status Enable On Bank3 0 - Masked 1 - Enabled
6 ADDR_ERR6_STAT_EN	Ocram Access Error Status Enable On Bank2 0 - Masked 1 - Enabled
5 ADDR_ERR5_STAT_EN	Ocram Access Error Status Enable On Bank1 0 - Masked 1 - Enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 ADDR_ERR4_STAT_EN	Ocram Access Error Status Enable On Bank0 0 - Masked 1 - Enabled
3 ADDR_ERR3_STAT_EN	Ocram Access Error Status Enable On Bank3 0 - Masked 1 - Enabled
2 ADDR_ERR2_STAT_EN	Ocram Access Error Status Enable On Bank2 0 - Masked 1 - Enabled
1 ADDR_ERR1_STAT_EN	Ocram Access Error Status Enable On Bank1 0 - Masked 1 - Enabled
0 ADDR_ERR0_STAT_EN	Ocram Access Error Status Enable On Bank0 0 - Masked 1 - Enabled

19.6.1.10 Single Error Interrupt Enable Register (SINGLE_ERR_SIG_EN)

This register is used to select which interrupt status is indicated to the Host System as the interrupt. Setting any of these bits to 1 enables interrupt generation. The corresponding Status register bit will generate an interrupt when the corresponding interrupt signal enable bit is set.

Offset

Register	Offset
SINGLE_ERR_SIG_EN	20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...	SINGLE_...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 SINGLE_ERR15_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank3 0 - Masked 1 - Enabled
14 SINGLE_ERR14_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank2 0 - Masked 1 - Enabled
13 SINGLE_ERR13_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank1 0 - Masked 1 - Enabled
12 SINGLE_ERR12_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank0 0 - Masked 1 - Enabled
11 SINGLE_ERR11_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank3 0 - Masked 1 - Enabled
10 SINGLE_ERR10_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank2 0 - Masked 1 - Enabled
9 SINGLE_ERR9_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank3 0 - Masked 1 - Enabled
8 SINGLE_ERR8_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank2 0 - Masked 1 - Enabled
7 SINGLE_ERR7_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank1 0 - Masked 1 - Enabled
6 SINGLE_ERR6_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank0 0 - Masked 1 - Enabled

Table continues on the next page...

Table continued from the previous page...

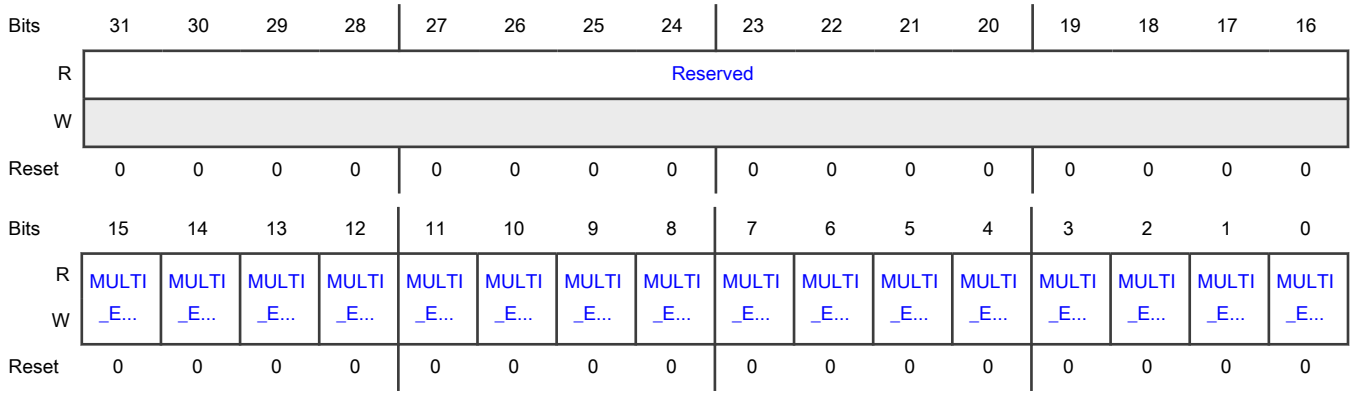
Field	Description
5 SINGLE_ERR5_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank3 0 - Masked 1 - Enabled
4 SINGLE_ERR4_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank2 0 - Masked 1 - Enabled
3 SINGLE_ERR3_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank1 0 - Masked 1 - Enabled
2 SINGLE_ERR2_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank0 0 - Masked 1 - Enabled
1 SINGLE_ERR1_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank1 0 - Masked 1 - Enabled
0 SINGLE_ERR0_SIG_EN	Single Bit Error Interrupt Enable On Ocram Bank0 0 - Masked 1 - Enabled

19.6.1.11 MULTI Error Interrupt Enable Register (MULTI_ERR_SIG_EN)

Offset

Register	Offset
MULTI_ERR_SIG_EN	24h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 MULTI_ERR15_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank3 0 - Masked 1 - Enabled
14 MULTI_ERR14_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank2 0 - Masked 1 - Enabled
13 MULTI_ERR13_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank1 0 - Masked 1 - Enabled
12 MULTI_ERR12_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank0 0 - Masked 1 - Enabled
11 MULTI_ERR11_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank3 0 - Masked 1 - Enabled
10 MULTI_ERR10_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank2 0 - Masked 1 - Enabled
9	Multiple Bits Error Interrupt Enable On Ocram Bank1 0 - Masked

Table continues on the next page...

Table continued from the previous page...

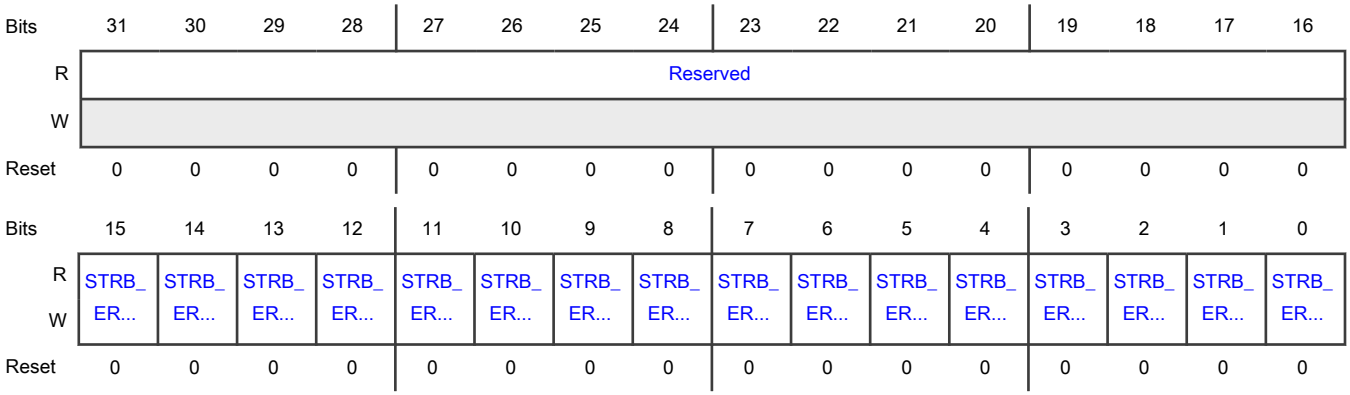
Field	Description
MULTI_ERR9_SIG_EN	1 - Enabled
8 MULTI_ERR8_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank0 0 - Masked 1 - Enabled
7 MULTI_ERR7_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank3 0 - Masked 1 - Enabled
6 MULTI_ERR6_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank2 0 - Masked 1 - Enabled
5 MULTI_ERR5_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank1 0 - Masked 1 - Enabled
4 MULTI_ERR4_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank0 0 - Masked 1 - Enabled
3 MULTI_ERR3_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank3 0 - Masked 1 - Enabled
2 MULTI_ERR2_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank2 0 - Masked 1 - Enabled
1 MULTI_ERR1_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank1 0 - Masked 1 - Enabled
0 MULTI_ERR0_SIG_EN	Multiple Bits Error Interrupt Enable On Ocram Bank0 0 - Masked 1 - Enabled

19.6.1.12 STRB Error Interrupt Enable Register (STRB_ERR_SIG_EN)

Offset

Register	Offset
STRB_ERR_SIG_EN	28h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 STRB_ERR15_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank3 0 - Masked 1 - Enabled
14 STRB_ERR14_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank2 0 - Masked 1 - Enabled
13 STRB_ERR13_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank1 0 - Masked 1 - Enabled
12 STRB_ERR12_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank0 0 - Masked 1 - Enabled
11 STRB_ERR11_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank3 0 - Masked 1 - Enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
10 STRB_ERR10_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank2 0 - Masked 1 - Enabled
9 STRB_ERR9_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank1 0 - Masked 1 - Enabled
8 STRB_ERR8_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank0 0 - Masked 1 - Enabled
7 STRB_ERR7_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank3 0 - Masked 1 - Enabled
6 STRB_ERR6_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank2 0 - Masked 1 - Enabled
5 STRB_ERR5_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank1 0 - Masked 1 - Enabled
4 STRB_ERR4_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank0 0 - Masked 1 - Enabled
3 STRB_ERR3_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank3 0 - Masked 1 - Enabled
2 STRB_ERR2_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank2 0 - Masked 1 - Enabled
1 STRB_ERR1_SIG_EN	AXI Strobe Error Interrupt Enable On Ocram Bank1 0 - Masked 1 - Enabled
0	AXI Strobe Error Interrupt Enable On Ocram Bank0 0 - Masked

Table continues on the next page...

Table continued from the previous page...

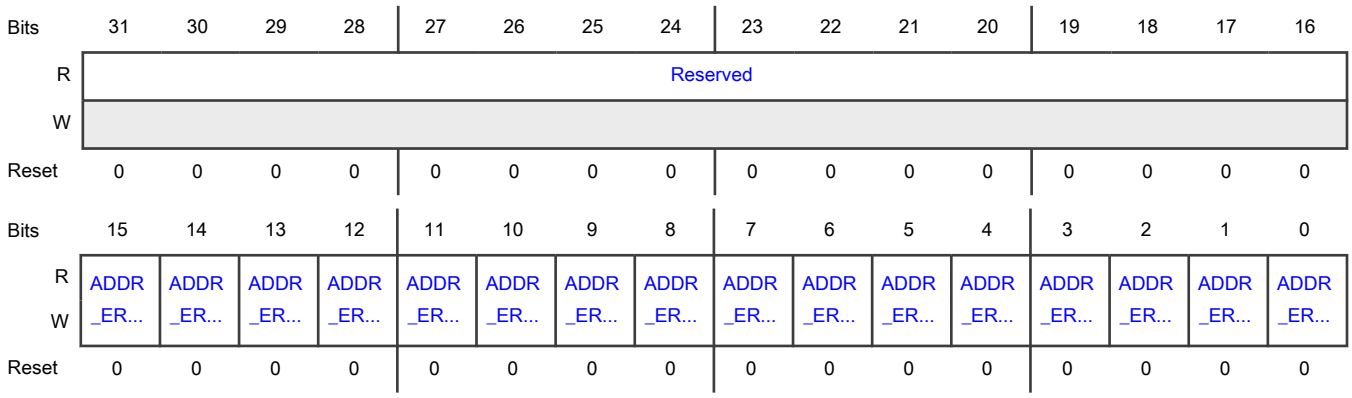
Field	Description
STRB_ERR0_SIG_EN	1 - Enabled

19.6.1.13 ADDR Error Interrupt Enable Register (ADDR_ERR_SIG_EN)

Offset

Register	Offset
ADDR_ERR_SIG_EN	2Ch

Diagram



Fields

Field	Description
31-16 —	Reserved
15 ADDR_ERR15_SIG_EN	Ocram Access Error Interrupt Enable On Bank3 0 - Masked 1 - Enabled
14 ADDR_ERR14_SIG_EN	Ocram Access Error Interrupt Enable On Bank2 0 - Masked 1 - Enabled
13	Ocram Access Error Interrupt Enable On Bank1 0 - Masked

Table continues on the next page...

Table continued from the previous page...

Field	Description
ADDR_ERR13_SIG_EN	1 - Enabled
12 ADDR_ERR12_SIG_EN	Ocram Access Error Interrupt Enable On Bank0 0 - Masked 1 - Enabled
11 ADDR_ERR11_SIG_EN	Ocram Access Error Interrupt Enable On Bank3 0 - Masked 1 - Enabled
10 ADDR_ERR10_SIG_EN	Ocram Access Error Interrupt Enable On Bank2 0 - Masked 1 - Enabled
9 ADDR_ERR9_SIG_EN	Ocram Access Error Interrupt Enable On Bank1 0 - Masked 1 - Enabled
8 ADDR_ERR8_SIG_EN	Ocram Access Error Interrupt Enable On Bank0 0 - Masked 1 - Enabled
7 ADDR_ERR7_SIG_EN	Ocram Access Error Interrupt Enable On Bank3 0 - Masked 1 - Enabled
6 ADDR_ERR6_SIG_EN	Ocram Access Error Interrupt Enable On Bank2 0 - Masked 1 - Enabled
5 ADDR_ERR5_SIG_EN	Ocram Access Error Interrupt Enable On Bank1 0 - Masked 1 - Enabled
4 ADDR_ERR4_SIG_EN	Ocram Access Error Interrupt Enable On Bank0 0 - Masked 1 - Enabled
3 ADDR_ERR3_SIG_EN	Ocram Access Error Interrupt Enable On Bank3 0 - Masked 1 - Enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 ADDR_ERR2_SIG_EN	Ocram Access Error Interrupt Enable On Bank2 0 - Masked 1 - Enabled
1 ADDR_ERR1_SIG_EN	Ocram Access Error Interrupt Enable On Bank1 0 - Masked 1 - Enabled
0 ADDR_ERR0_SIG_EN	Ocram Access Error Interrupt Enable On Bank0 0 - Masked 1 - Enabled

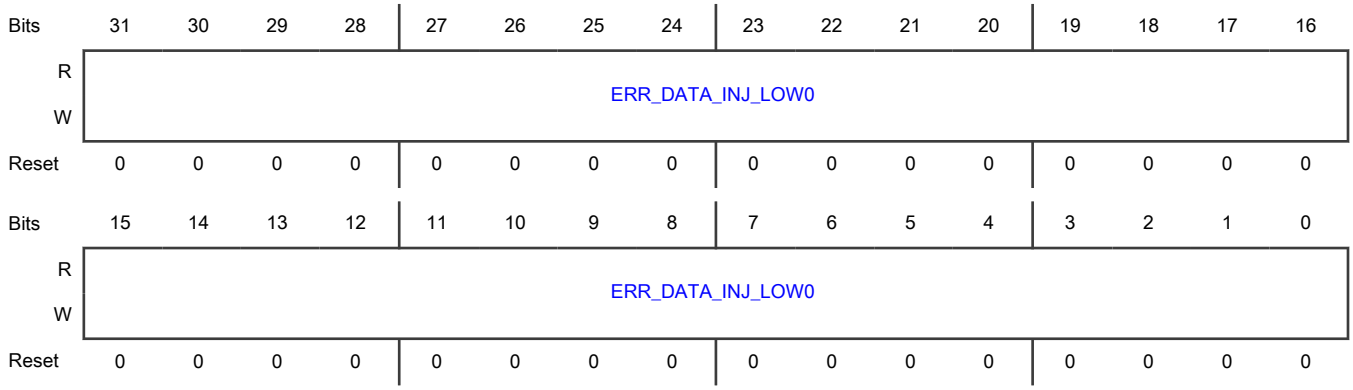
19.6.1.14 Error Injection On Low 32 bits Of Ocram Bank0 Write Data (ERR_DATA_INJ_LOW0)

Low 32 bits of ocram bank0 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW0	30h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_LOW0	Error Injection On Low 32 bits Of Ocram Bank0 Write Data Low 32 bits of ocram bank0 write data will be reversed if corresponding bits in this register are set.

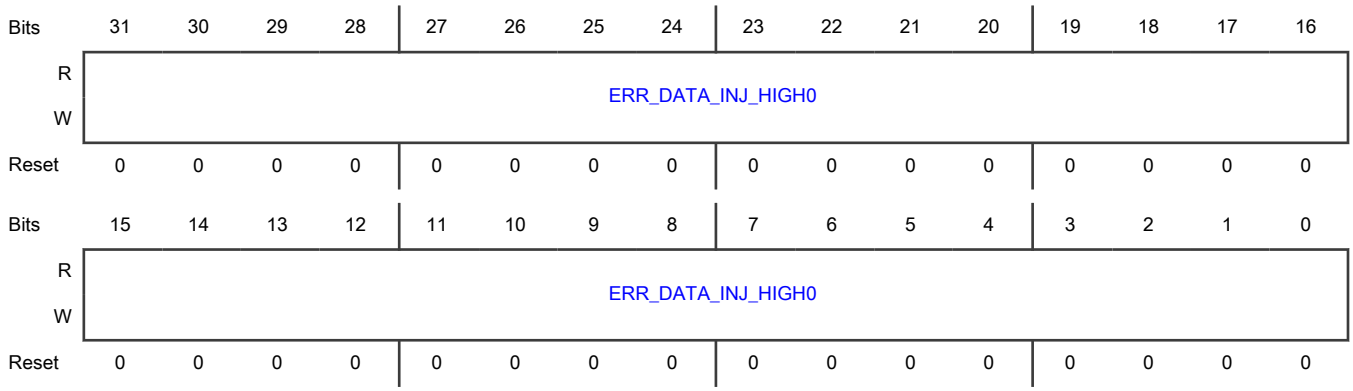
19.6.1.15 Error Injection On High 32 bits Of Ocram Bank0 Write Data (ERR_DATA_INJ_HIGH0)

High 32 bits of ocram bank0 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH0	34h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH0	Error Injection On High 32 bits Of Ocram Bank0 Write Data High 32 bits of ocram bank0 write data will be reversed if corresponding bits in this register are set.

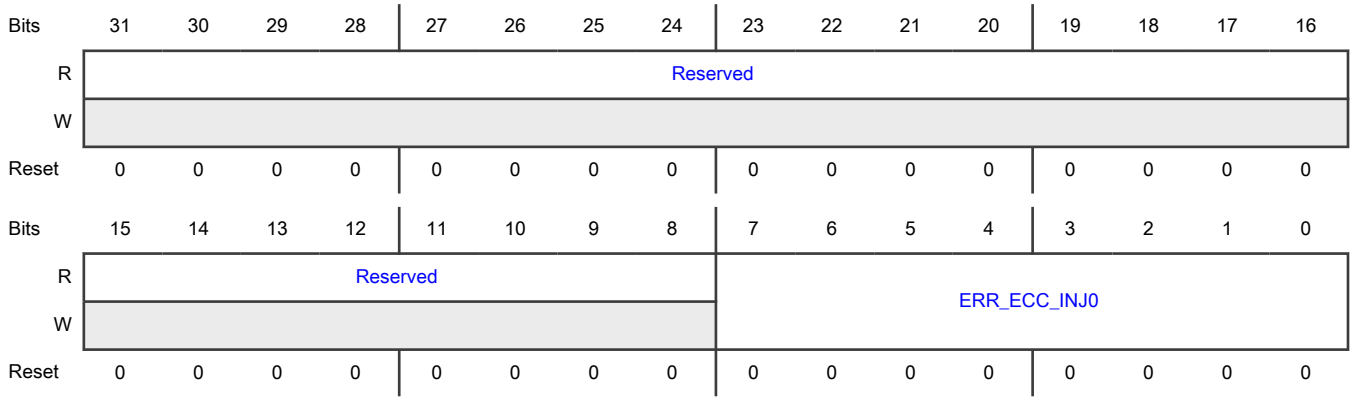
19.6.1.16 Error Injection On 8 bits ECC code Of Ocram Bank0 Write Data (ERR_ECC_INJ0)

8 bits ECC code of ocram bank0 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ0	38h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ0	Error Injection On 8 bits ECC code Of Ocram Bank0 Write Data 8 bits ECC code of ocram bank0 write data will be reversed if corresponding bits in this register are set.

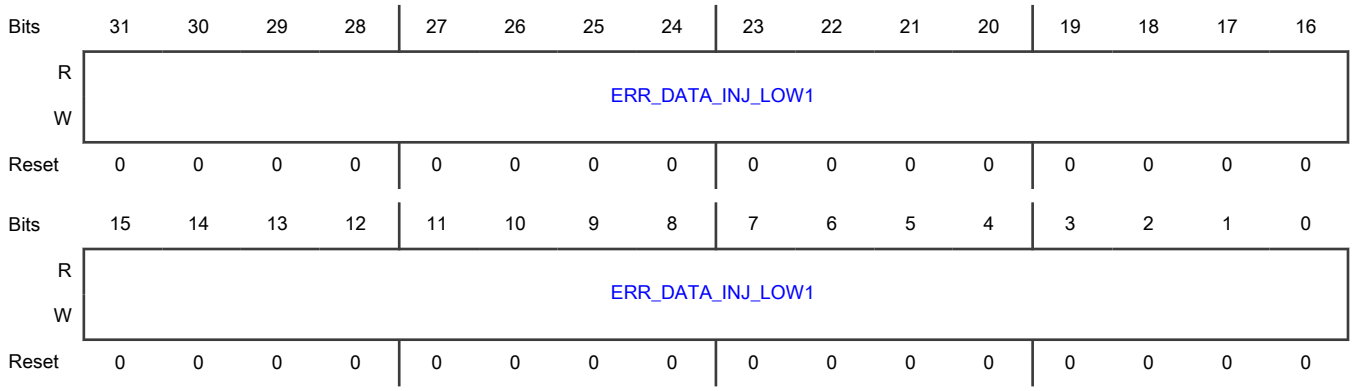
19.6.1.17 Error Injection On Low 32 bits Of Ocram Bank1 Write Data (ERR_DATA_INJ_LOW1)

Low 32 bits of ocram bank1 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW1	3Ch

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_LOW1	Error Injection On Low 32 bits Of Ocram Bank1 Write Data Low 32 bits of ocram bank1 write data will be reversed if corresponding bits in this register are set.

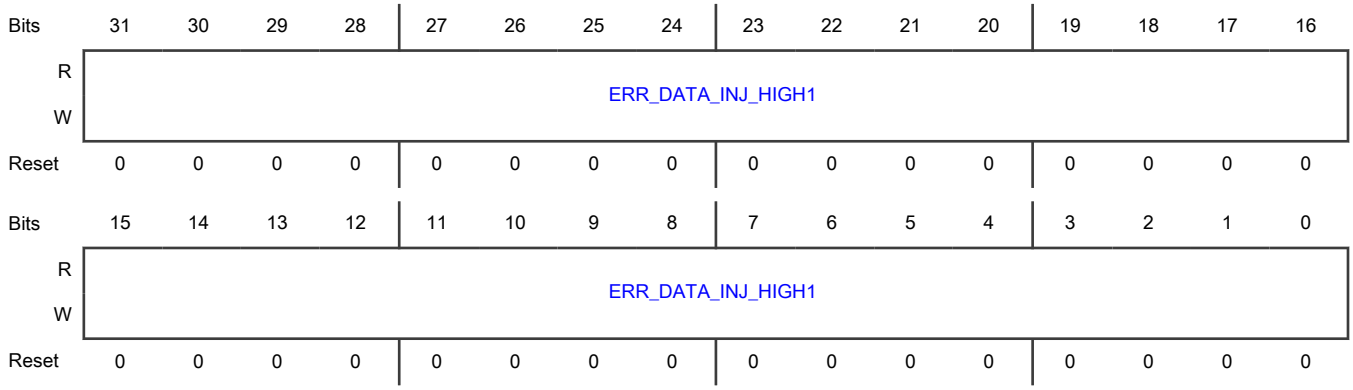
19.6.1.18 Error Injection On High 32 bits Of Ocram Bank1 Write Data (ERR_DATA_INJ_HIGH1)

High 32 bits of ocram bank1 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH1	40h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH1	Error Injection On High 32 bits Of Ocram Bank1 Write Data High 32 bits of ocram bank1 write data will be reversed if corresponding bits in this register are set.

19.6.1.19 Error Injection On 8 bits ECC code Of Ocram Bank1 Write Data (ERR_ECC_INJ1)

8 bits ECC code of ocram bank1 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ1	44h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								ERR_ECC_INJ1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ 1	Error Injection On 8 bits ECC code Of Ocram Bank1 Write Data 8 bits ECC code of ocram bank1 write data will be reversed if corresponding bits in this register are set.

19.6.1.20 Error Injection On Low 32 bits Of Ocram Bank2 Write Data (ERR_DATA_INJ_LOW2)

Low 32 bits of ocram bank2 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW2	48h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR_DATA_INJ_LOW2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_DATA_INJ_LOW2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ERR_DATA_INJ_LOW2	Error Injection On Low 32 bits Of Ocram Bank2 Write Data Low 32 bits of ocram bank2 write data will be reversed if corresponding bits in this register are set.

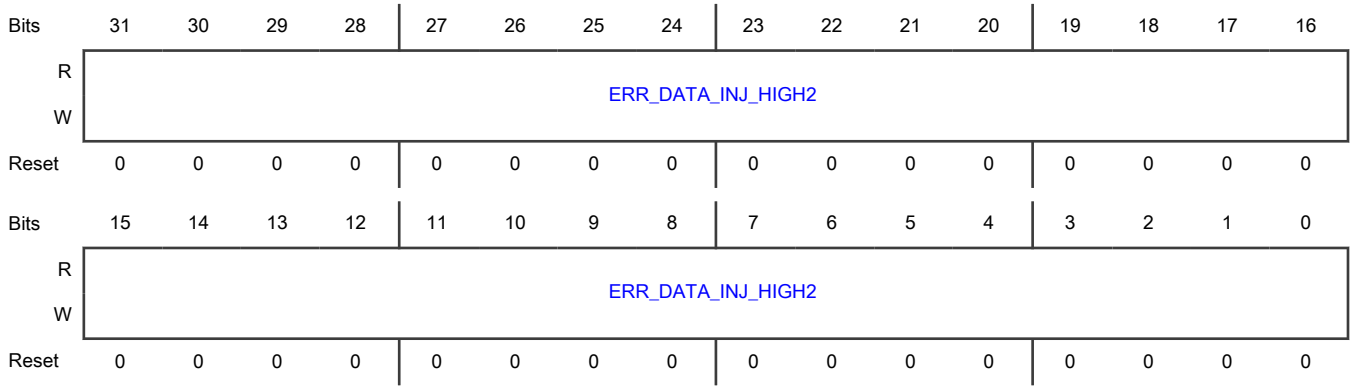
19.6.1.21 Error Injection On High 32 bits Of Ocram Bank2 Write Data (ERR_DATA_INJ_HIGH2)

High 32 bits of ocram bank2 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH2	4Ch

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH2	Error Injection On High 32 bits Of Ocram Bank2 Write Data High 32 bits of ocram bank2 write data will be reversed if corresponding bits in this register are set.

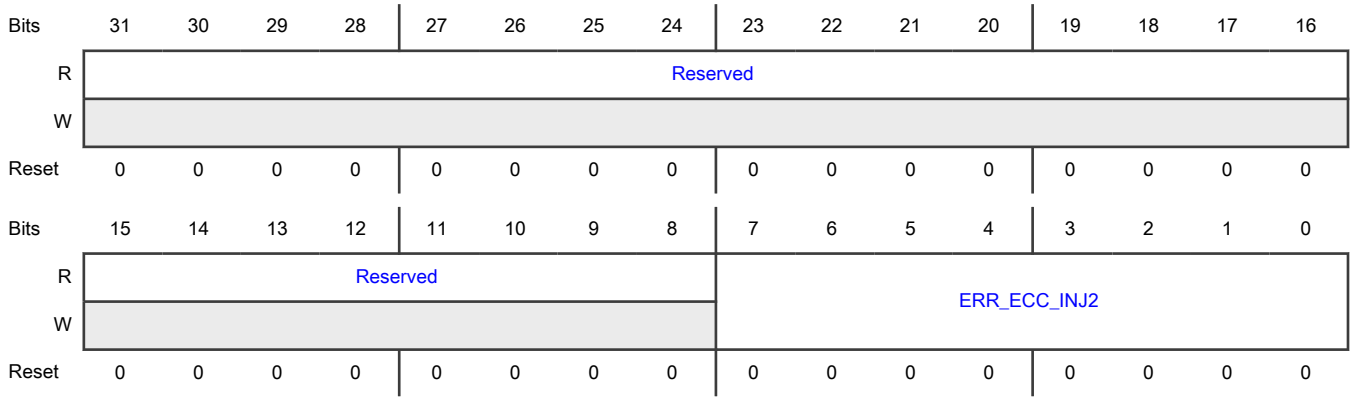
19.6.1.22 Error Injection On 8 bits ECC code Of Ocram Bank2 Write Data (ERR_ECC_INJ2)

8 bits ECC code of ocram bank2 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ2	50h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ2	Error Injection On 8 bits ECC code Of Ocram Bank2 Write Data 8 bits ECC code of ocram bank2 write data will be reversed if corresponding bits in this register are set.

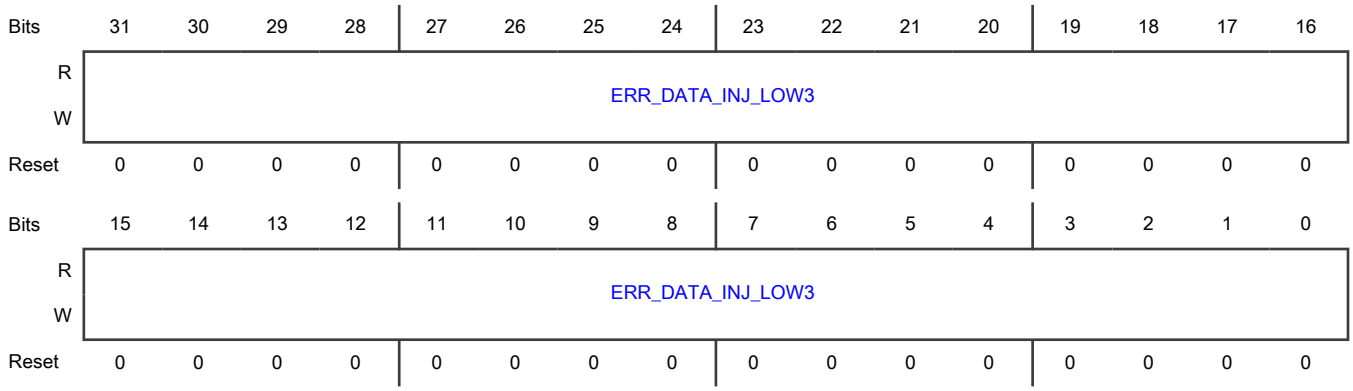
19.6.1.23 Error Injection On Low 32 bits Of Ocram Bank3 Write Data (ERR_DATA_INJ_LOW3)

Low 32 bits of ocram bank3 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW3	54h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_LOW3	Error Injection On Low 32 bits Of Ocram Bank3 Write Data Low 32 bits of ocram bank3 write data will be reversed if corresponding bits in this register are set.

19.6.1.24 Error Injection On High 32 bits Of Ocram Bank3 Write Data (ERR_DATA_INJ_HIGH3)

High 32 bits of ocram bank3 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH3	58h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERR_DATA_INJ_HIGH3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERR_DATA_INJ_HIGH3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH3	Error Injection On High 32 bits Of Ocram Bank2 Write Data High 32 bits of ocram bank2 write data will be reversed if corresponding bits in this register are set.

19.6.1.25 Error Injection On 8 bits ECC code Of Ocram Bank3 Write Data (ERR_ECC_INJ3)

8 bits ECC code of ocram bank3 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ3	5Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								ERR_ECC_INJ3							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ3	Error Injection On 8 bits ECC code Of Ocram Bank3 Write Data 8 bits ECC code of ocram bank3 write data will be reversed if corresponding bits in this register are set.

19.6.1.26 Error Injection On Low 32 bits Of Ocram Bank4 Write Data (ERR_DATA_INJ_LOW4)

Low 32 bits of ocram bank4 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW4	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR_DATA_INJ_LOW4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_DATA_INJ_LOW4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ERR_DATA_INJ_LOW4	Error Injection On Low 32 bits Of Ocram Bank4 Write Data Low 32 bits of ocram bank4 write data will be reversed if corresponding bits in this register are set.

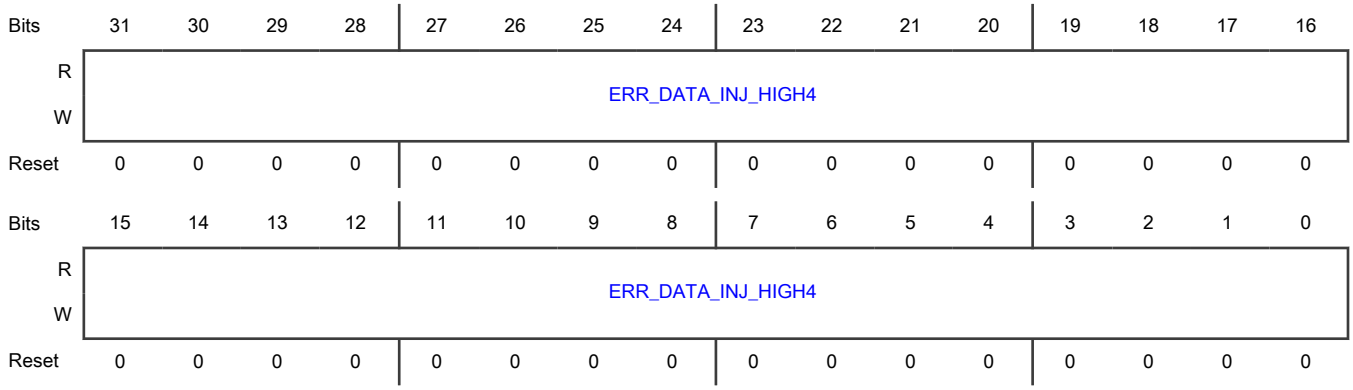
19.6.1.27 Error Injection On High 32 bits Of Ocram Bank4 Write Data (ERR_DATA_INJ_HIGH4)

High 32 bits of ocram bank4 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH4	64h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH4	Error Injection On High 32 bits Of Ocram Bank4 Write Data High 32 bits of ocram bank4 write data will be reversed if corresponding bits in this register are set.

19.6.1.28 Error Injection On 8 bits ECC code Of Ocram Bank4 Write Data (ERR_ECC_INJ4)

8 bits ECC code of ocram bank4 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ4	68h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								ERR_ECC_INJ4							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ4	Error Injection On 8 bits ECC code Of Ocram Bank4 Write Data 8 bits ECC code of ocram bank4 write data will be reversed if corresponding bits in this register are set.

19.6.1.29 Error Injection On Low 32 bits Of Ocram Bank5 Write Data (ERR_DATA_INJ_LOW5)

Low 32 bits of ocram bank5 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW5	6Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR_DATA_INJ_LOW5															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_DATA_INJ_LOW5															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ERR_DATA_INJ_LOW5	Error Injection On Low 32 bits Of Ocram Bank5 Write Data Low 32 bits of ocram bank5 write data will be reversed if corresponding bits in this register are set.

19.6.1.30 Error Injection On High 32 bits Of Ocram Bank5 Write Data (ERR_DATA_INJ_HIGH5)

High 32 bits of ocram bank5 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH5	70h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERR_DATA_INJ_HIGH5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERR_DATA_INJ_HIGH5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH5	Error Injection On High 32 bits Of Ocram Bank5 Write Data High 32 bits of ocram bank5 write data will be reversed if corresponding bits in this register are set.

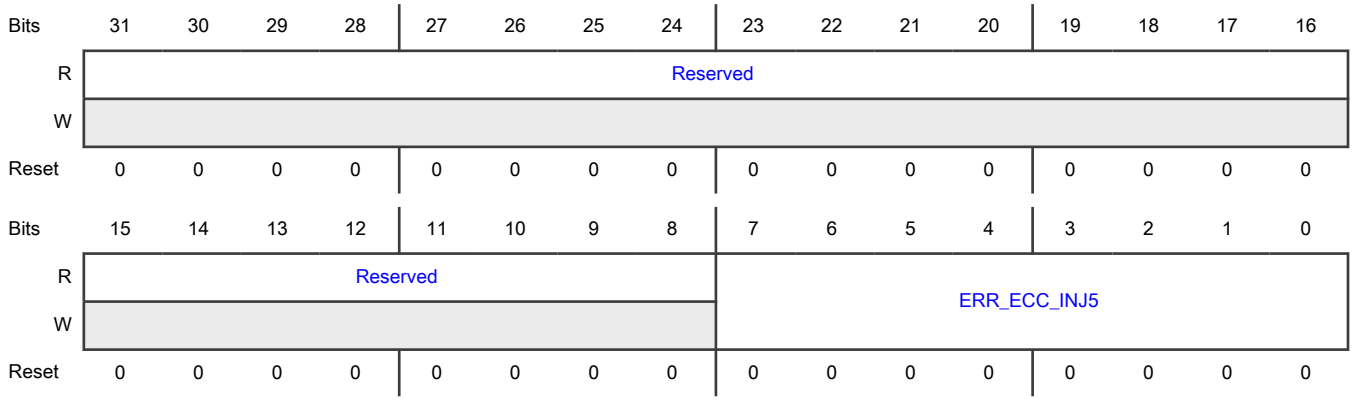
19.6.1.31 Error Injection On 8 bits ECC code Of Ocram Bank5 Write Data (ERR_ECC_INJ5)

8 bits ECC code of ocram bank5 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ5	74h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ5	Error Injection On 8 bits ECC code Of Ocram Bank5 Write Data 8 bits ECC code of ocram bank5 write data will be reversed if corresponding bits in this register are set.

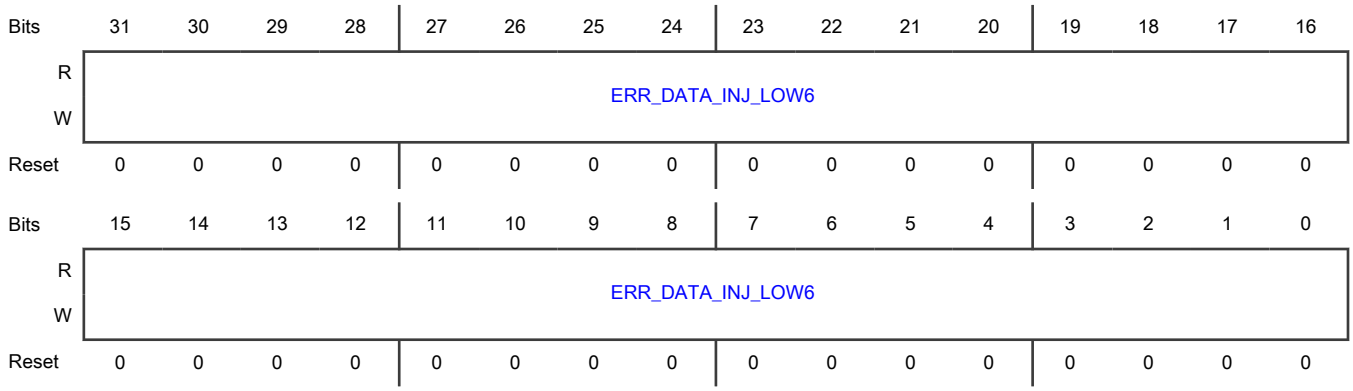
19.6.1.32 Error Injection On Low 32 bits Of Ocram Bank6 Write Data (ERR_DATA_INJ_LOW6)

Low 32 bits of ocram bank6 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW6	78h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_LOW6	Error Injection On Low 32 bits Of Ocram Bank6 Write Data Low 32 bits of ocram bank6 write data will be reversed if corresponding bits in this register are set.

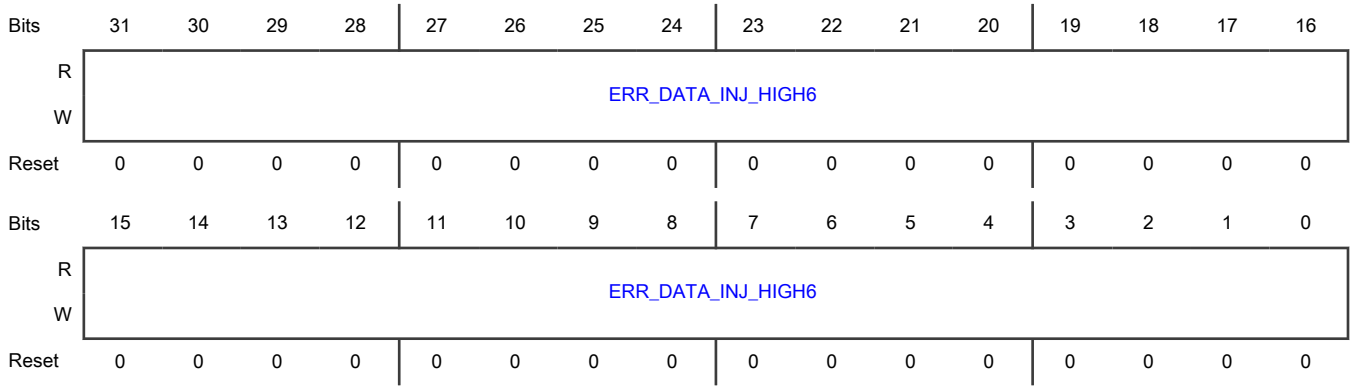
19.6.1.33 Error Injection On High 32 bits Of Ocram Bank6 Write Data (ERR_DATA_INJ_HIGH6)

High 32 bits of ocram bank6 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH6	7Ch

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH6	Error Injection On High 32 bits Of Ocram Bank6 Write Data High 32 bits of ocram bank6 write data will be reversed if corresponding bits in this register are set.

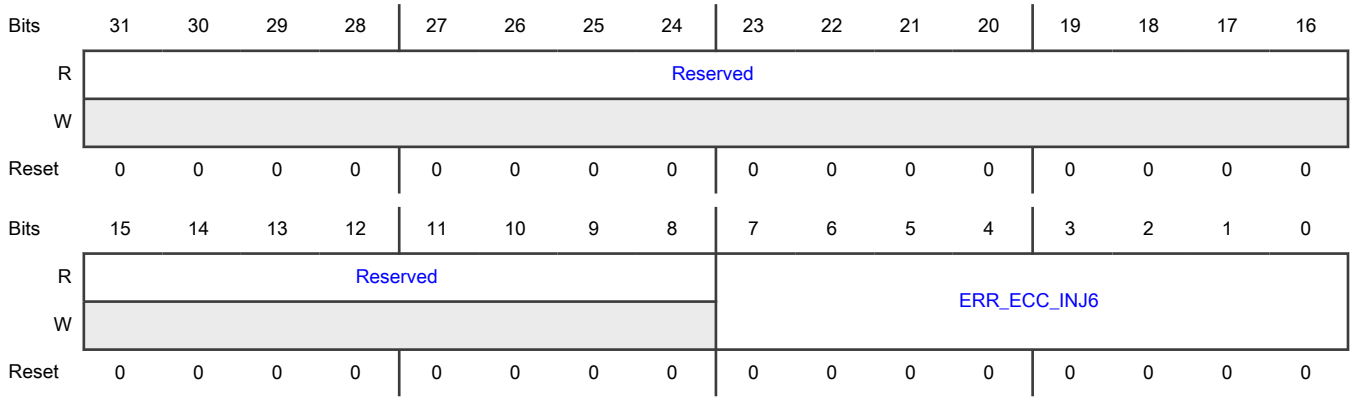
19.6.1.34 Error Injection On 8 bits ECC code Of Ocram Bank6 Write Data (ERR_ECC_INJ6)

8 bits ECC code of ocram bank6 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ6	80h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ6	Error Injection On 8 bits ECC code Of Ocram Bank6 Write Data 8 bits ECC code of ocram bank6 write data will be reversed if corresponding bits in this register are set.

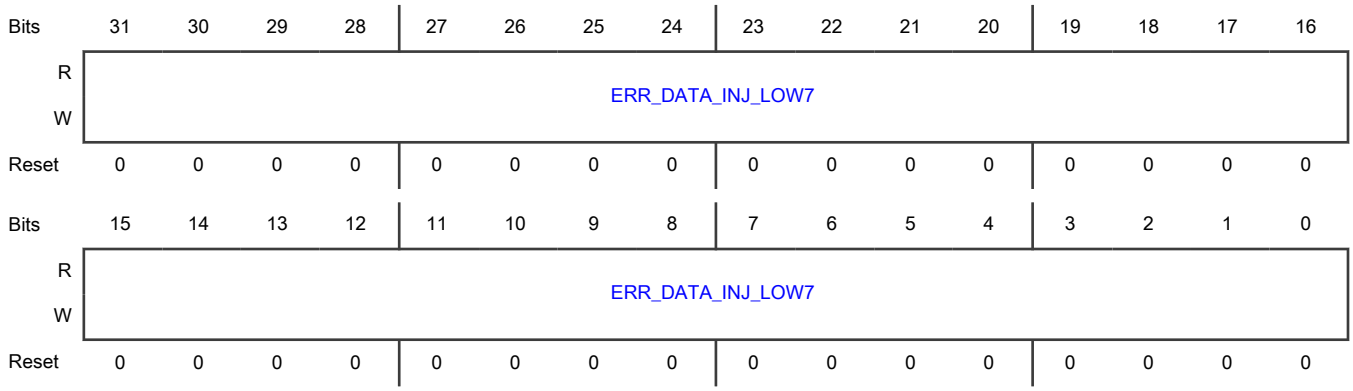
19.6.1.35 Error Injection On Low 32 bits Of Ocram Bank7 Write Data (ERR_DATA_INJ_LOW7)

Low 32 bits of ocram bank7 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW7	84h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_LOW7	Error Injection On Low 32 bits Of Ocram Bank7 Write Data Low 32 bits of ocram bank7 write data will be reversed if corresponding bits in this register are set.

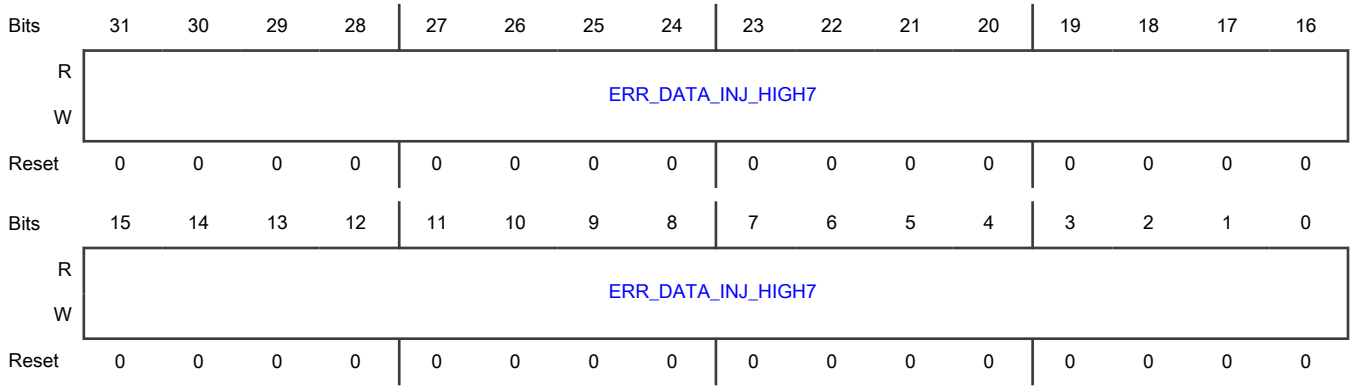
19.6.1.36 Error Injection On High 32 bits Of Ocram Bank7 Write Data (ERR_DATA_INJ_HIGH7)

High 32 bits of ocram bank7 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH7	88h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH7	Error Injection On High 32 bits Of Ocram Bank7 Write Data High 32 bits of ocram bank7 write data will be reversed if corresponding bits in this register are set.

19.6.1.37 Error Injection On 8 bits ECC code Of Ocram Bank7 Write Data (ERR_ECC_INJ7)

8 bits ECC code of ocram bank7 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ7	8Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								ERR_ECC_INJ7							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ7	Error Injection On 8 bits ECC code Of Ocram Bank7 Write Data 8 bits ECC code of ocram bank7 write data will be reversed if corresponding bits in this register are set.

19.6.1.38 Error Injection On Low 32 bits Of Ocram Bank8 Write Data (ERR_DATA_INJ_LOW8)

Low 32 bits of ocram bank8 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW8	90h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR_DATA_INJ_LOW8															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_DATA_INJ_LOW8															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ERR_DATA_INJ_LOW8	Error Injection On Low 32 bits Of Ocram Bank8 Write Data Low 32 bits of ocram bank8 write data will be reversed if corresponding bits in this register are set.

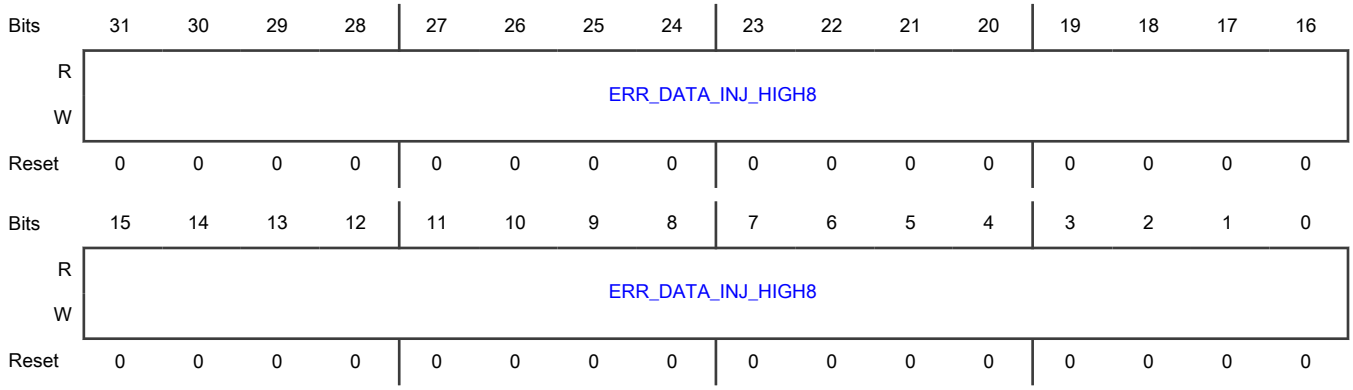
19.6.1.39 Error Injection On High 32 bits Of Ocram Bank8 Write Data (ERR_DATA_INJ_HIGH8)

High 32 bits of ocram bank8 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH8	94h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH8	Error Injection On High 32 bits Of Ocram Bank8 Write Data High 32 bits of ocram bank8 write data will be reversed if corresponding bits in this register are set.

19.6.1.40 Error Injection On 8 bits ECC code Of Ocram Bank8 Write Data (ERR_ECC_INJ8)

8 bits ECC code of ocram bank8 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ8	98h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								ERR_ECC_INJ8							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ8	Error Injection On 8 bits ECC code Of Ocram Bank8 Write Data 8 bits ECC code of ocram bank8 write data will be reversed if corresponding bits in this register are set.

19.6.1.41 Error Injection On Low 32 bits Of Ocram Bank9 Write Data (ERR_DATA_INJ_LOW9)

Low 32 bits of ocram bank9 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW9	9Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR_DATA_INJ_LOW9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_DATA_INJ_LOW9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ERR_DATA_INJ_LOW9	Error Injection On Low 32 bits Of Ocram Bank9 Write Data Low 32 bits of ocram bank9 write data will be reversed if corresponding bits in this register are set.

19.6.1.42 Error Injection On High 32 bits Of Ocram Bank9 Write Data (ERR_DATA_INJ_HIGH9)

High 32 bits of ocram bank9 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH9	A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERR_DATA_INJ_HIGH9															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERR_DATA_INJ_HIGH9															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH9	Error Injection On High 32 bits Of Ocram Bank9 Write Data High 32 bits of ocram bank9 write data will be reversed if corresponding bits in this register are set.

19.6.1.43 Error Injection On 8 bits ECC code Of Ocram Bank9 Write Data (ERR_ECC_INJ9)

8 bits ECC code of ocram bank9 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ9	A4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								ERR_ECC_INJ9							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ9	Error Injection On 8 bits ECC code Of Ocram Bank9 Write Data 8 bits ECC code of ocram bank9 write data will be reversed if corresponding bits in this register are set.

19.6.1.44 Error Injection On Low 32 bits Of Ocram Bank10 Write Data (ERR_DATA_INJ_LOW10)

Low 32 bits of ocram bank10 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW10	A8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR_DATA_INJ_LOW10															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_DATA_INJ_LOW10															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ERR_DATA_INJ_LOW10	Error Injection On Low 32 bits Of Ocram Bank10 Write Data Low 32 bits of ocram bank10 write data will be reversed if corresponding bits in this register are set.

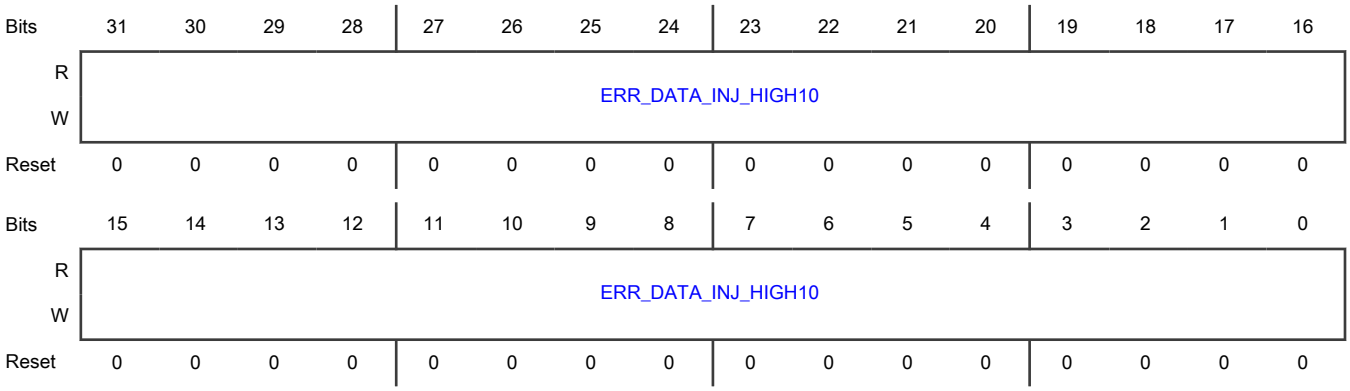
19.6.1.45 Error Injection On High 32 bits Of Ocram Bank10 Write Data (ERR_DATA_INJ_HIGH10)

High 32 bits of ocram bank10 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH10	ACh

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH10	Error Injection On High 32 bits Of Ocram Bank10 Write Data High 32 bits of ocram bank10 write data will be reversed if corresponding bits in this register are set.

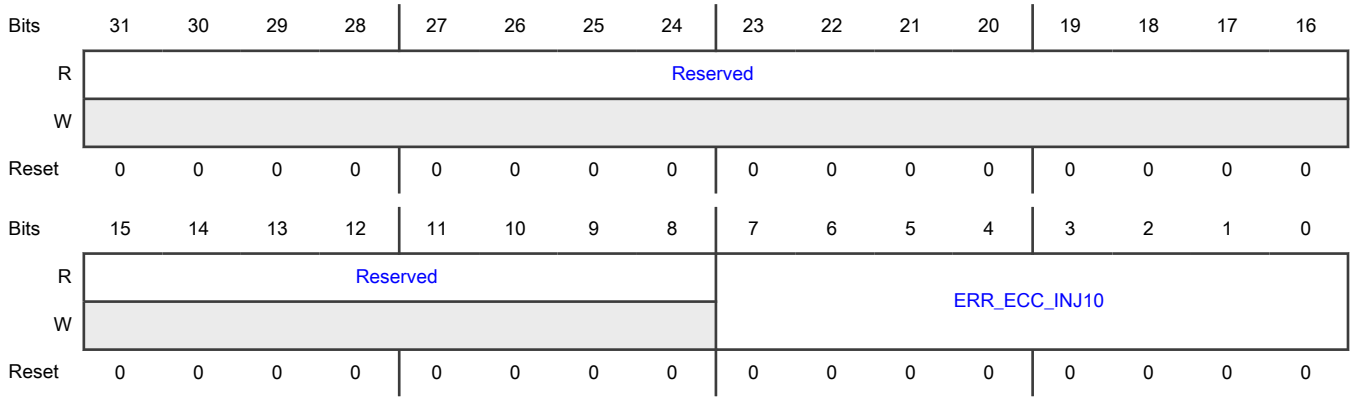
19.6.1.46 Error Injection On 8 bits ECC code Of Ocram Bank10 Write Data (ERR_ECC_INJ10)

8 bits ECC code of ocram bank10 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ10	B0h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ10	Error Injection On 8 bits ECC code Of Ocram Bank10 Write Data 8 bits ECC code of ocram bank10 write data will be reversed if corresponding bits in this register are set.

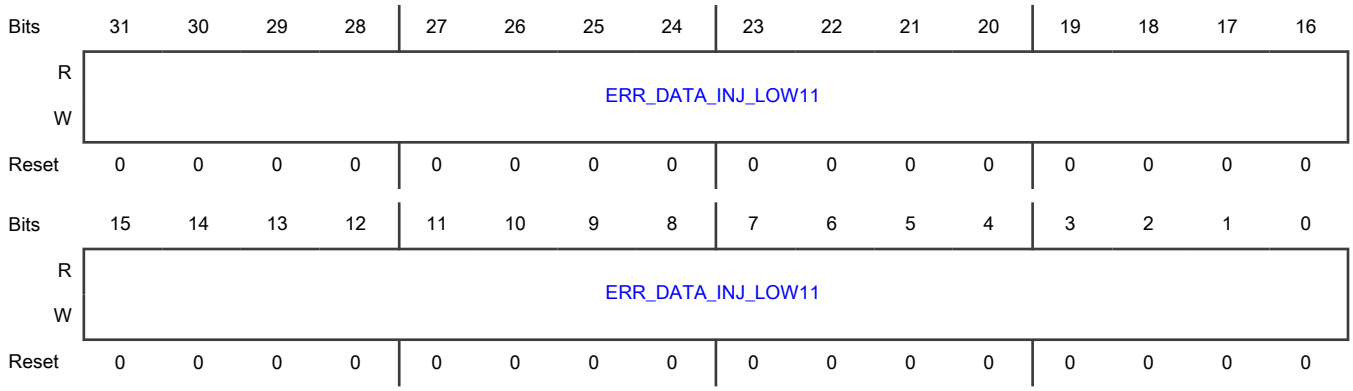
19.6.1.47 Error Injection On Low 32 bits Of Ocram Bank11 Write Data (ERR_DATA_INJ_LOW11)

Low 32 bits of ocram bank11 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW11	B4h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_LOW11	Error Injection On Low 32 bits Of Ocram Bank11 Write Data Low 32 bits of ocram bank11 write data will be reversed if corresponding bits in this register are set.

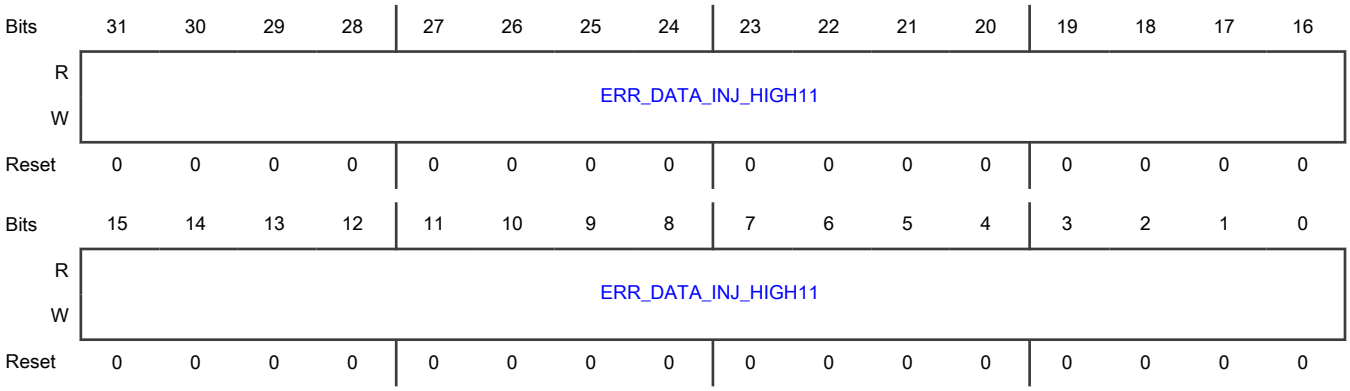
19.6.1.48 Error Injection On High 32 bits Of Ocram Bank11 Write Data (ERR_DATA_INJ_HIGH11)

High 32 bits of ocram bank11 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH11	B8h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH11	Error Injection On High 32 bits Of Ocram Bank11 Write Data High 32 bits of ocram bank11 write data will be reversed if corresponding bits in this register are set.

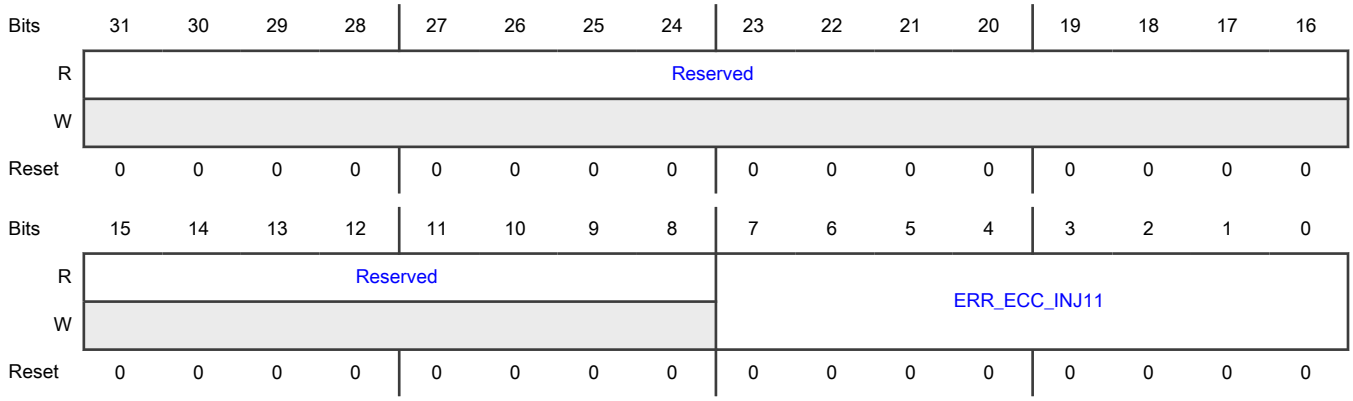
19.6.1.49 Error Injection On 8 bits ECC code Of Ocram Bank11 Write Data (ERR_ECC_INJ11)

8 bits ECC code of ocram bank11 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ11	BCh

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ11	Error Injection On 8 bits ECC code Of Ocram Bank11 Write Data 8 bits ECC code of ocram bank11 write data will be reversed if corresponding bits in this register are set.

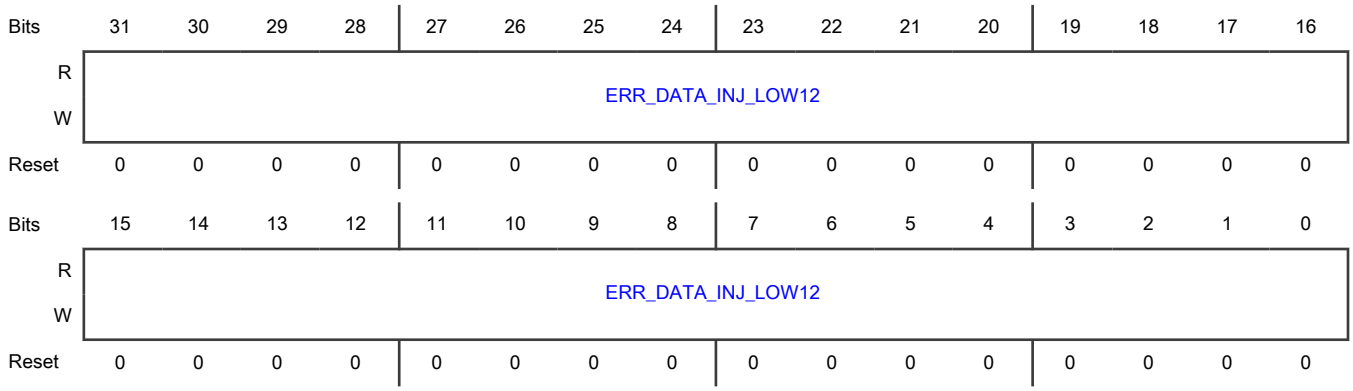
19.6.1.50 Error Injection On Low 32 bits Of Ocram Bank12 Write Data (ERR_DATA_INJ_LOW12)

Low 32 bits of ocram bank12 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW12	C0h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_LOW12	Error Injection On Low 32 bits Of Ocram Bank12 Write Data Low 32 bits of ocram bank12 write data will be reversed if corresponding bits in this register are set.

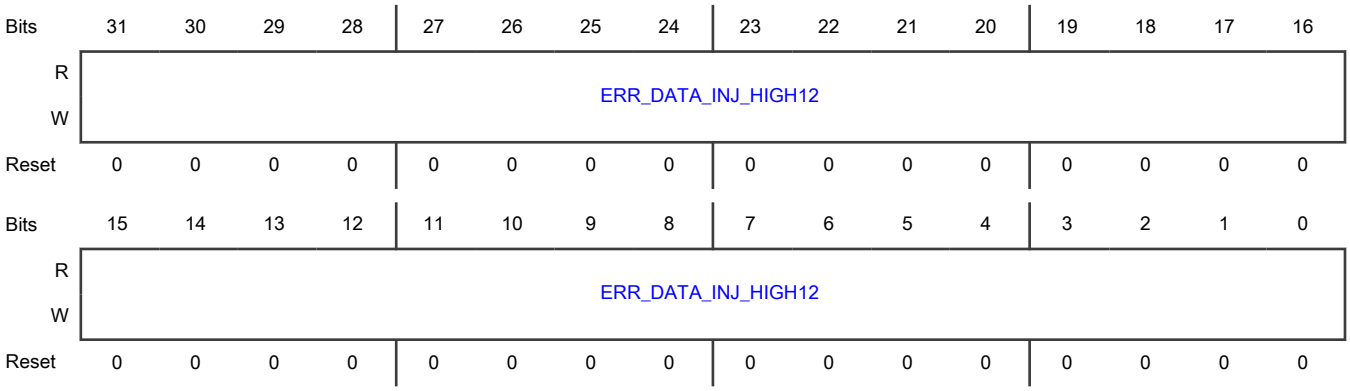
19.6.1.51 Error Injection On High 32 bits Of Ocram Bank12 Write Data (ERR_DATA_INJ_HIGH12)

High 32 bits of ocram bank12 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH12	C4h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH12	Error Injection On High 32 bits Of Ocram Bank12 Write Data High 32 bits of ocram bank12 write data will be reversed if corresponding bits in this register are set.

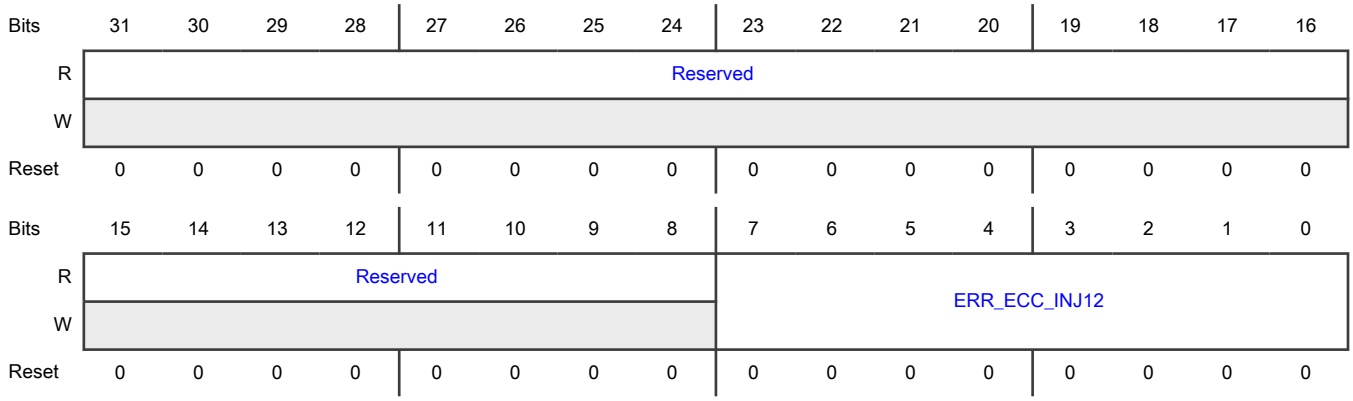
19.6.1.52 Error Injection On 8 bits ECC code Of Ocram Bank12 Write Data (ERR_ECC_INJ12)

8 bits ECC code of ocram bank12 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ12	C8h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ12	Error Injection On 8 bits ECC code Of Ocram Bank12 Write Data 8 bits ECC code of ocram bank12 write data will be reversed if corresponding bits in this register are set.

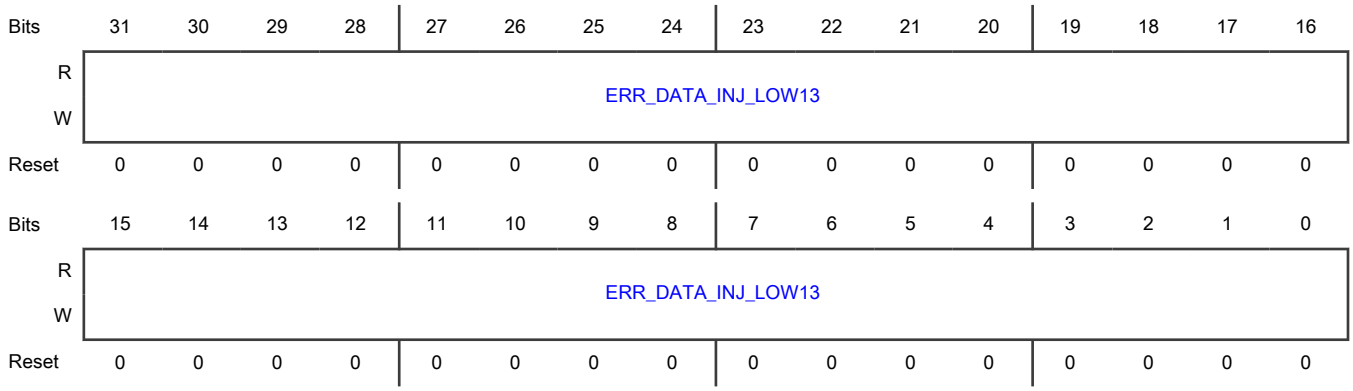
19.6.1.53 Error Injection On Low 32 bits Of Ocram Bank13 Write Data (ERR_DATA_INJ_LOW13)

Low 32 bits of ocram bank13 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW13	CCh

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_LOW13	Error Injection On Low 32 bits Of Ocram Bank13 Write Data Low 32 bits of ocram bank13 write data will be reversed if corresponding bits in this register are set.

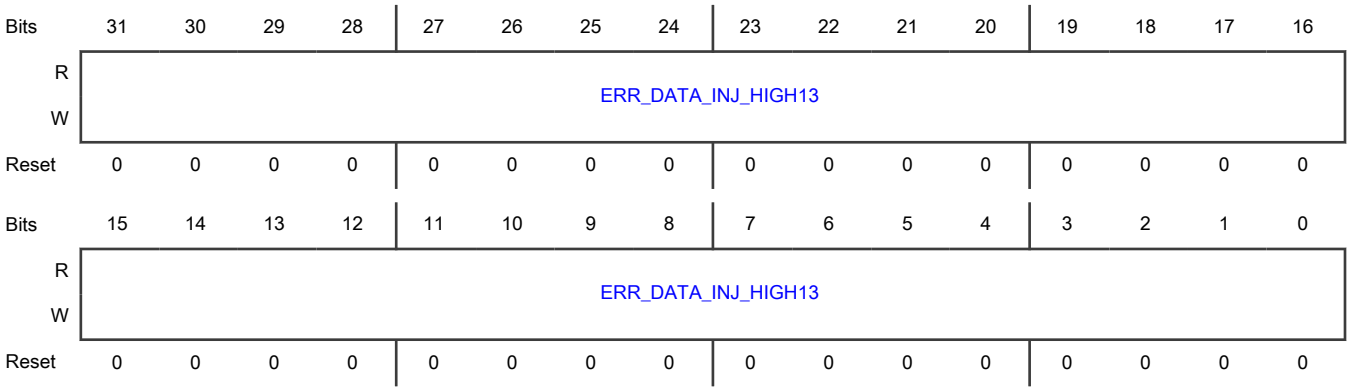
19.6.1.54 Error Injection On High 32 bits Of Ocram Bank13 Write Data (ERR_DATA_INJ_HIGH13)

High 32 bits of ocram bank13 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH13	D0h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH13	Error Injection On High 32 bits Of Ocram Bank13 Write Data High 32 bits of ocram bank13 write data will be reversed if corresponding bits in this register are set.

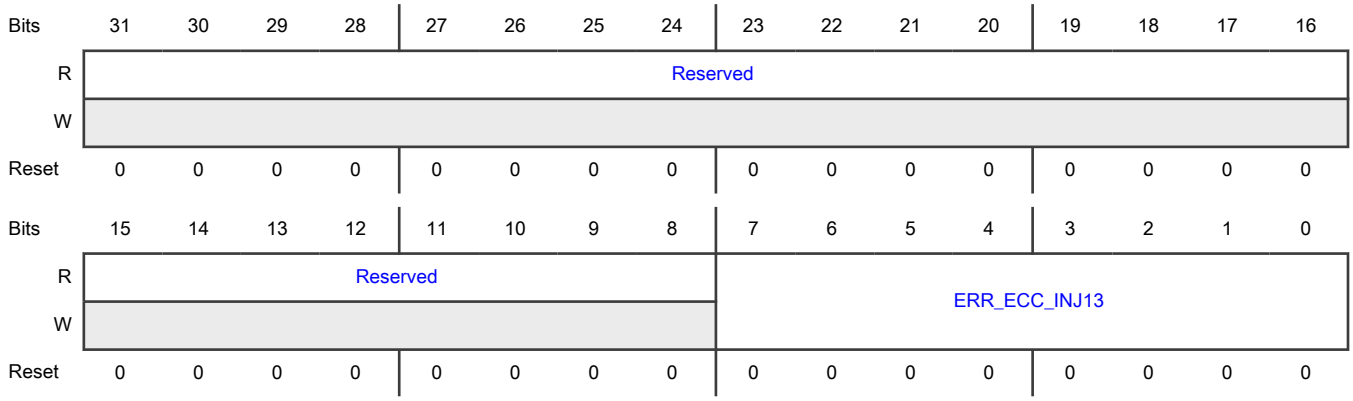
19.6.1.55 Error Injection On 8 bits ECC code Of Ocram Bank13 Write Data (ERR_ECC_INJ13)

8 bits ECC code of ocram bank13 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ13	D4h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ13	Error Injection On 8 bits ECC code Of Ocram Bank13 Write Data 8 bits ECC code of ocram bank13 write data will be reversed if corresponding bits in this register are set.

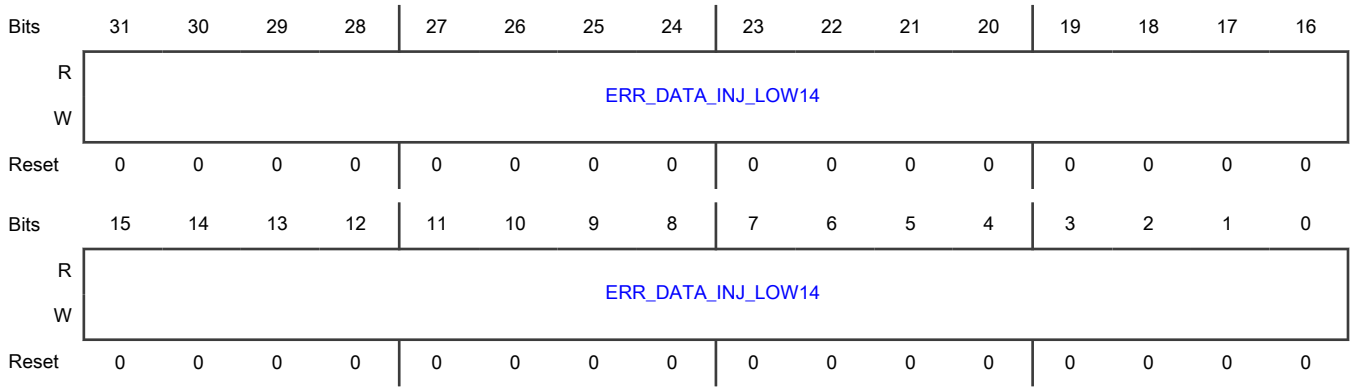
19.6.1.56 Error Injection On Low 32 bits Of Ocram Bank14 Write Data (ERR_DATA_INJ_LOW14)

Low 32 bits of ocram bank14 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW14	D8h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_LOW14	Error Injection On Low 32 bits Of Ocram Bank14 Write Data Low 32 bits of ocram bank14 write data will be reversed if corresponding bits in this register are set.

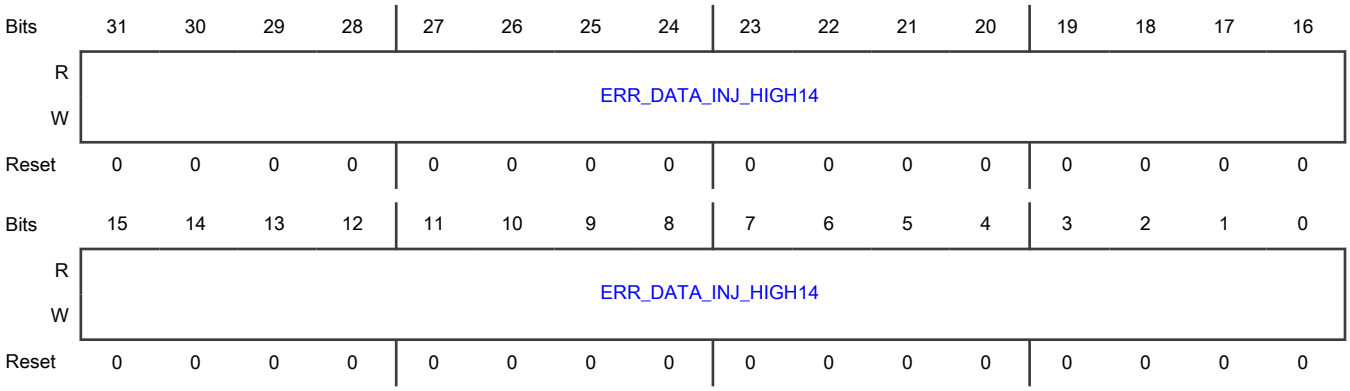
19.6.1.57 Error Injection On High 32 bits Of Ocram Bank14 Write Data (ERR_DATA_INJ_HIGH14)

High 32 bits of ocram bank14 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH14	DCh

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH14	Error Injection On High 32 bits Of Ocram Bank14 Write Data High 32 bits of ocram bank14 write data will be reversed if corresponding bits in this register are set.

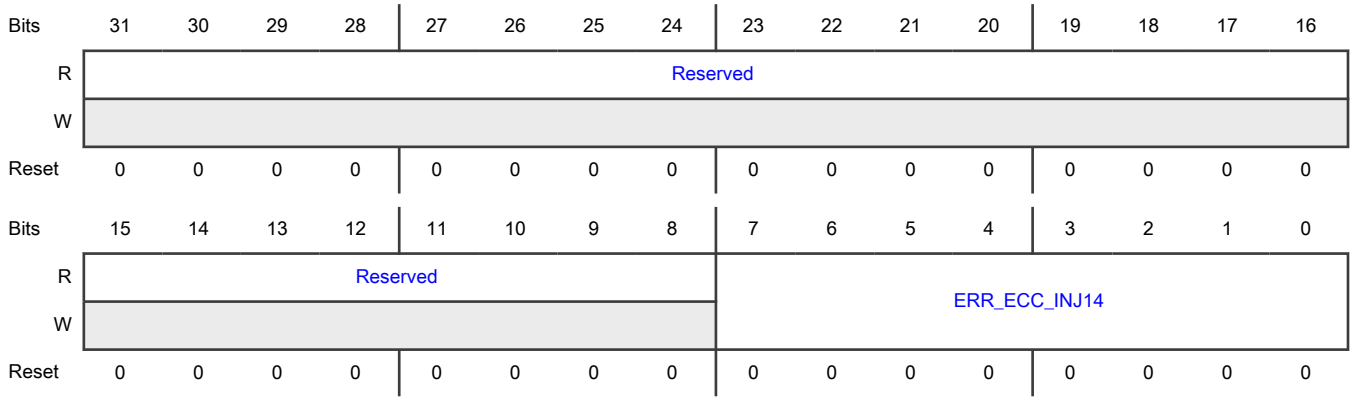
19.6.1.58 Error Injection On 8 bits ECC code Of Ocram Bank14 Write Data (ERR_ECC_INJ14)

8 bits ECC code of ocram bank14 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ14	E0h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ14	Error Injection On 8 bits ECC code Of Ocram Bank14 Write Data 8 bits ECC code of ocram bank14 write data will be reversed if corresponding bits in this register are set.

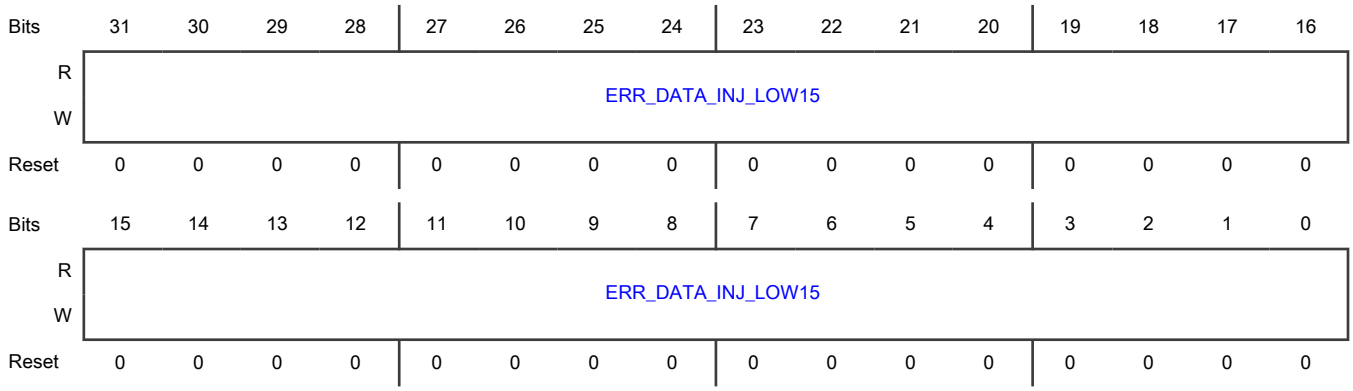
19.6.1.59 Error Injection On Low 32 bits Of Ocram Bank15 Write Data (ERR_DATA_INJ_LOW15)

Low 32 bits of ocram bank15 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_LOW15	E4h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_LOW15	Error Injection On Low 32 bits Of Ocram Bank15 Write Data Low 32 bits of ocram bank15 write data will be reversed if corresponding bits in this register are set.

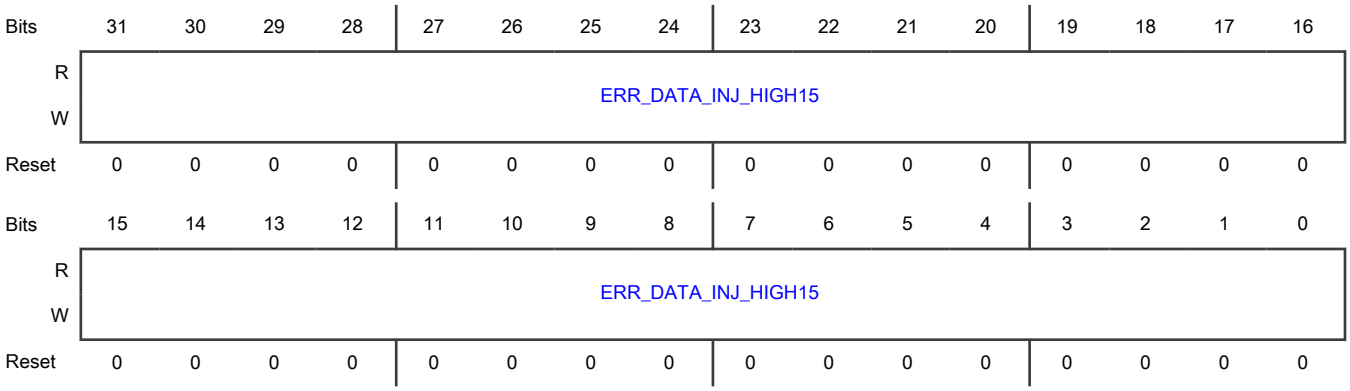
19.6.1.60 Error Injection On High 32 bits Of Ocram Bank15 Write Data (ERR_DATA_INJ_HIGH15)

High 32 bits of ocram bank15 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_DATA_INJ_HIGH15	E8h

Diagram



Fields

Field	Description
31-0 ERR_DATA_INJ_HIGH15	Error Injection On High 32 bits Of Ocram Bank15 Write Data High 32 bits of ocram bank15 write data will be reversed if corresponding bits in this register are set.

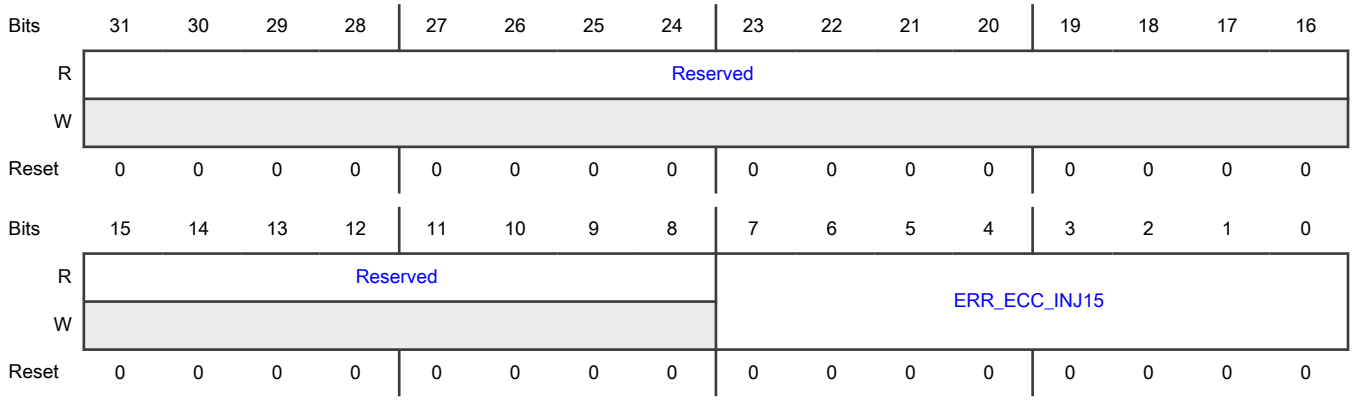
19.6.1.61 Error Injection On 8 bits ECC code Of Ocram Bank15 Write Data (ERR_ECC_INJ15)

8 bits ECC code of ocram bank15 write data will be reversed if corresponding bits in this register are set.

Offset

Register	Offset
ERR_ECC_INJ15	ECh

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ15	Error Injection On 8 bits ECC code Of Ocram Bank15 Write Data 8 bits ECC code of ocram bank15 write data will be reversed if corresponding bits in this register are set.

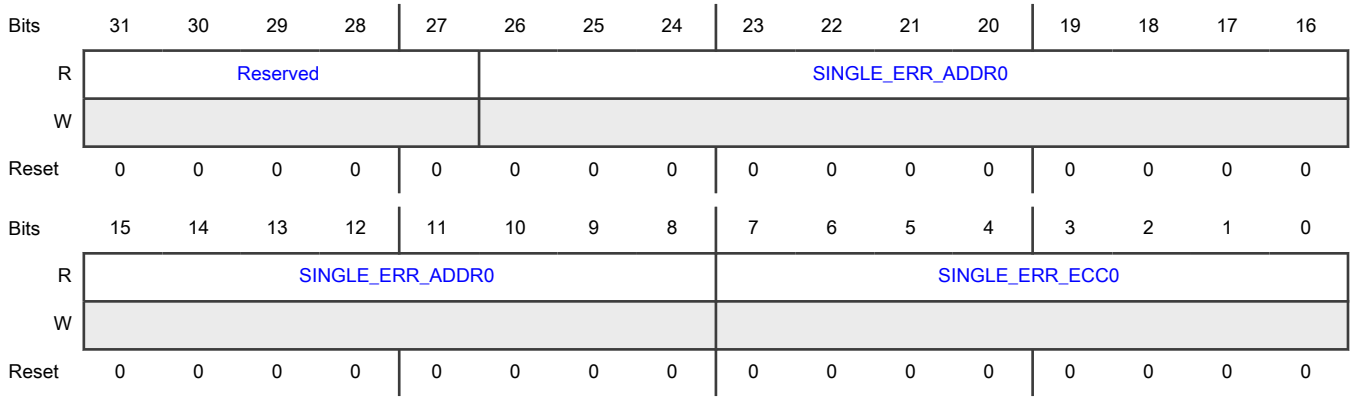
19.6.1.62 Single Error Address And ECC Code On Ocram Bank0 (SINGLE_ERR_ADDR_ECC0)

When single bit error is corrected in ocram bank0 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC0	F0h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR0	Single Error Address On Ocram Bank0 When single bit error is corrected in ocram bank0 read access, store ocram address.
7-0 SINGLE_ERR_ECC0	Single Error ECC Code On Ocram Bank0 When single bit error is corrected in ocram bank0 read access, store ecc code.

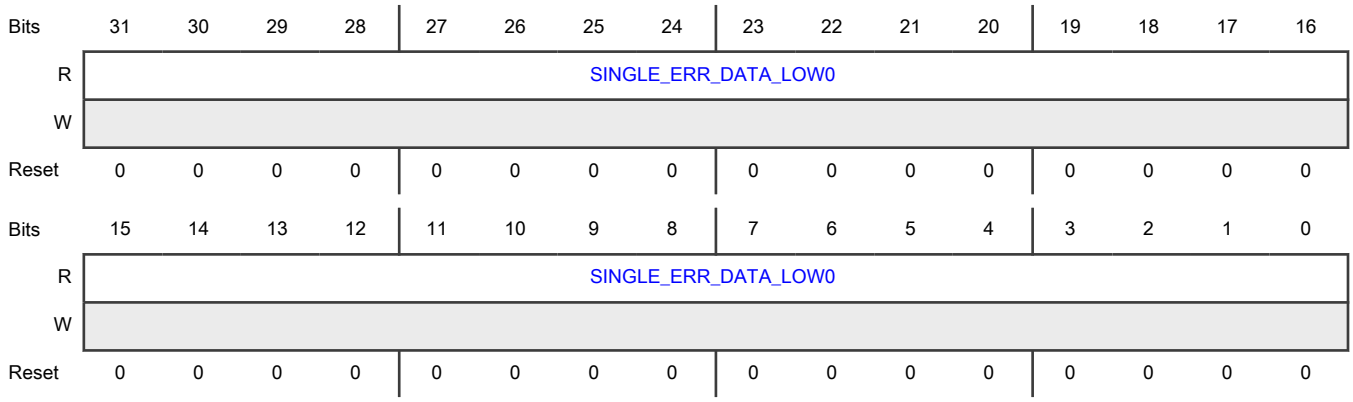
19.6.1.63 Low 32 Bits Single Error Read Data On Ocram Bank0 (SINGLE_ERR_DATA_LOW0)

When single bit error is corrected in ocram bank0 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW0	F4h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW0	Low 32 Bits Single Error Read Data On Ocram Bank0 When single bit error is corrected in ocram bank0 read access, store low 32 bits uncorrected read data.

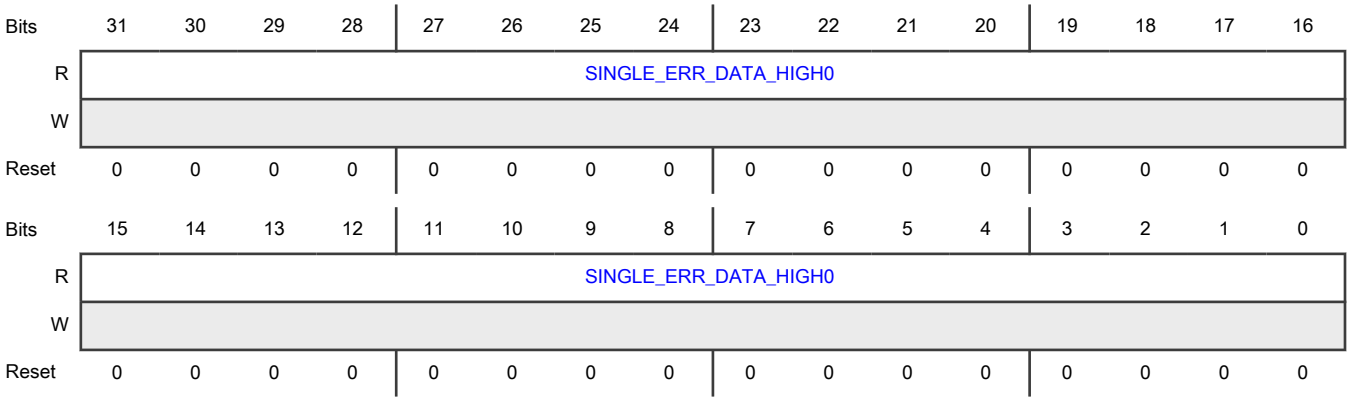
19.6.1.64 High 32 Bits Single Error Read Data On Ocram Bank0 (SINGLE_ERR_DATA_HIGH0)

When single bit error is corrected in ocram bank0 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH0	F8h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH0	High 32 Bits Single Error Read Data On Ocram Bank0 When single bit error is corrected in ocram bank0 read access, store high 32 bits uncorrected read data.

19.6.1.65 Low Single Error Bit Position On Ocram Bank0 (SINGLE_ERR_POS_LOW0)

When single bit error is corrected in ocram bank0 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW0	FCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS_LOW0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS_LOW0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW0	Low Single Error Bit Position On Ocram Bank0 When single bit error is corrected in ocram bank0 read access, indicates which bit is corrected in low 32 bits of read data.

19.6.1.66 High Single Error Bit Position On Ocram Bank0 (SINGLE_ERR_POS_HIGH0)

When single bit error is corrected in ocram bank0 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH0	100h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS_HIGH0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS_HIGH0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH0	High Single Error Bit Position On Ocram Bank0 When single bit error is corrected in ocram bank0 read access, indicates which bit is corrected in high 32 bits of read data.

19.6.1.67 Single Error Address And ECC Code On Ocram Bank1 (SINGLE_ERR_ADDR_ECC1)

When single bit error is corrected in ocram bank1 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC1	104h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				SINGLE_ERR_ADDR1											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_ADDR1								SINGLE_ERR_ECC1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR1	Single Error Address On Ocram Bank1 When single bit error is corrected in ocram bank1 read access, store ocram address.
7-0 SINGLE_ERR_ECC1	Single Error ECC Code On Ocram Bank1 When single bit error is corrected in ocram bank1 read access, store ecc code.

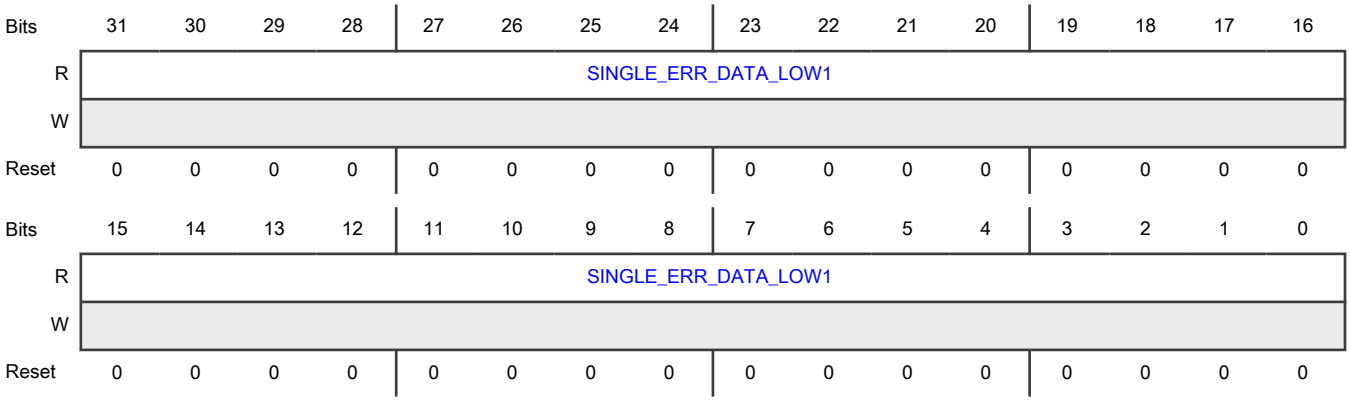
19.6.1.68 Low 32 Bits Single Error Read Data On Ocram Bank1 (SINGLE_ERR_DATA_LOW1)

When single bit error is corrected in ocram bank1 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW1	108h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW1	Low 32 Bits Single Error Read Data On Ocram Bank1 When single bit error is corrected in ocram bank1 read access, store low 32 bits uncorrected read data.

19.6.1.69 High 32 Bits Single Error Read Data On Ocram Bank1 (SINGLE_ERR_DATA_HIGH1)

When single bit error is corrected in ocram bank1 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH1	10Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_HIGH1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_HIGH1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH1	High 32 Bits Single Error Read Data On Ocram Bank1 When single bit error is corrected in ocram bank1 read access, store high 32 bits uncorrected read data.

19.6.1.70 Low Single Error Bit Position On Ocram Bank1 (SINGLE_ERR_POS_LOW1)

When single bit error is corrected in ocram bank1 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW1	110h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS_LOW1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS_LOW1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW1	Low Single Error Bit Position On Ocram Bank1 When single bit error is corrected in ocram bank1 read access, indicates which bit is corrected in low 32 bits of read data.

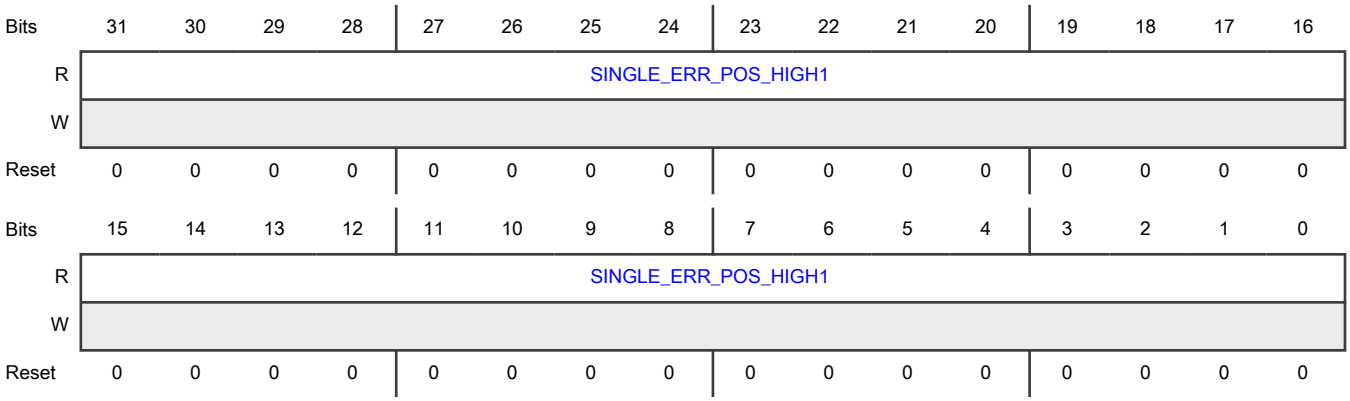
19.6.1.71 High Single Error Bit Position On Ocram Bank1 (SINGLE_ERR_POS_HIGH1)

When single bit error is corrected in ocram bank1 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH1	114h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH1	High Single Error Bit Position On Ocram Bank1 When single bit error is corrected in ocram bank1 read access, indicates which bit is corrected in high 32 bits of read data.

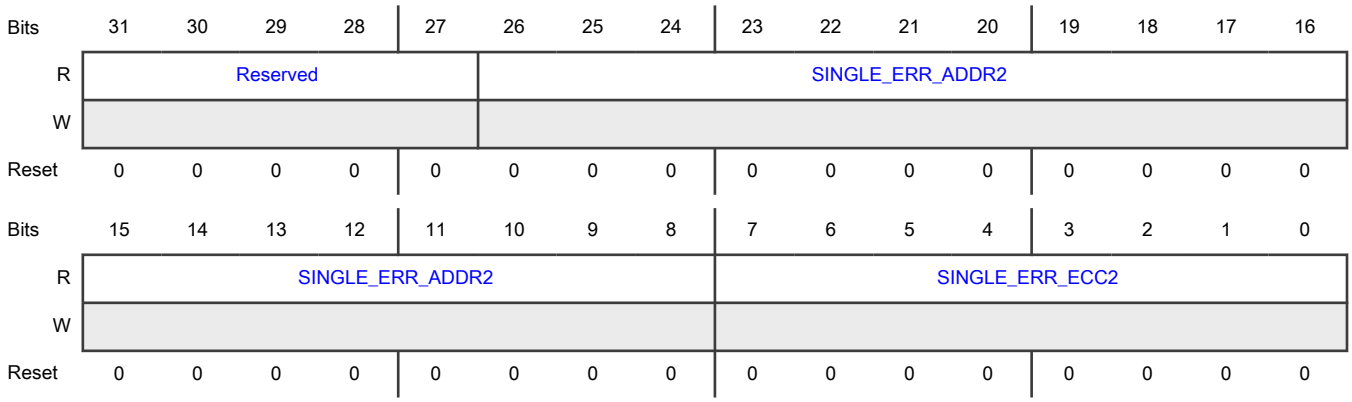
19.6.1.72 Single Error Address And ECC Code On Ocram Bank2 (SINGLE_ERR_ADDR_ECC2)

When single bit error is corrected in ocram bank2 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC2	118h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR2	Single Error Address On Ocram Bank2 When single bit error is corrected in ocram bank2 read access, store ocram address.
7-0 SINGLE_ERR_ECC2	Single Error ECC Code On Ocram Bank2 When single bit error is corrected in ocram bank2 read access, store ecc code.

19.6.1.73 Low 32 Bits Single Error Read Data On Ocram Bank2 (SINGLE_ERR_DATA_LOW2)

When single bit error is corrected in ocram bank2 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW2	11Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_LOW2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_LOW2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW2	Low 32 Bits Single Error Read Data On Ocram Bank2 When single bit error is corrected in ocram bank2 read access, store low 32 bits uncorrected read data.

19.6.1.74 High 32 Bits Single Error Read Data On Ocram Bank2 (SINGLE_ERR_DATA_HIGH2)

When single bit error is corrected in ocram bank2 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH2	120h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_HIGH2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_HIGH2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH2	High 32 Bits Single Error Read Data On Ocram Bank2 When single bit error is corrected in ocram bank2 read access, store high 32 bits uncorrected read data.

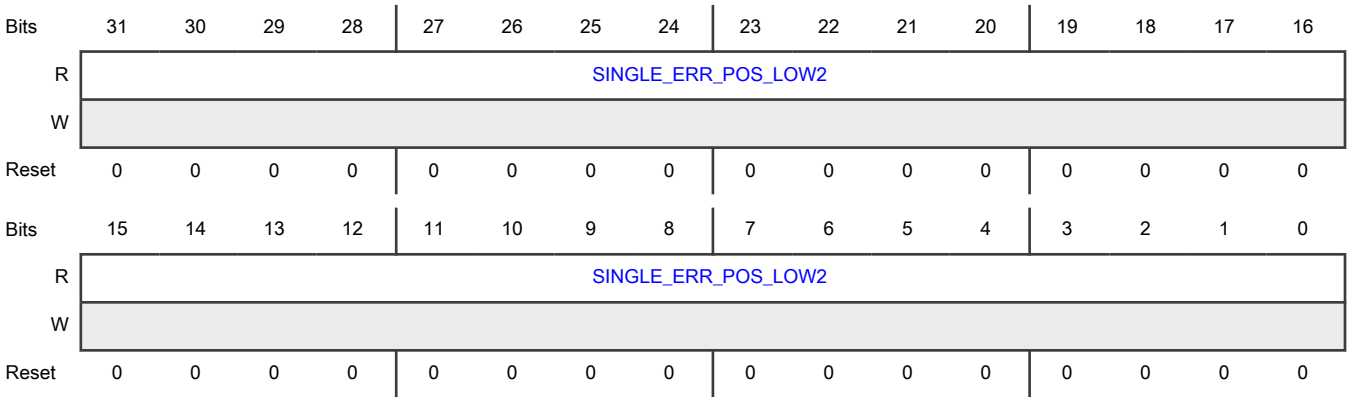
19.6.1.75 Low Single Error Bit Position On Ocram Bank2 (SINGLE_ERR_POS_LOW2)

When single bit error is corrected in ocram bank2 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW2	124h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW2	Low Single Error Bit Position On Ocram Bank2 When single bit error is corrected in ocram bank2 read access, indicates which bit is corrected in low 32 bits of read data.

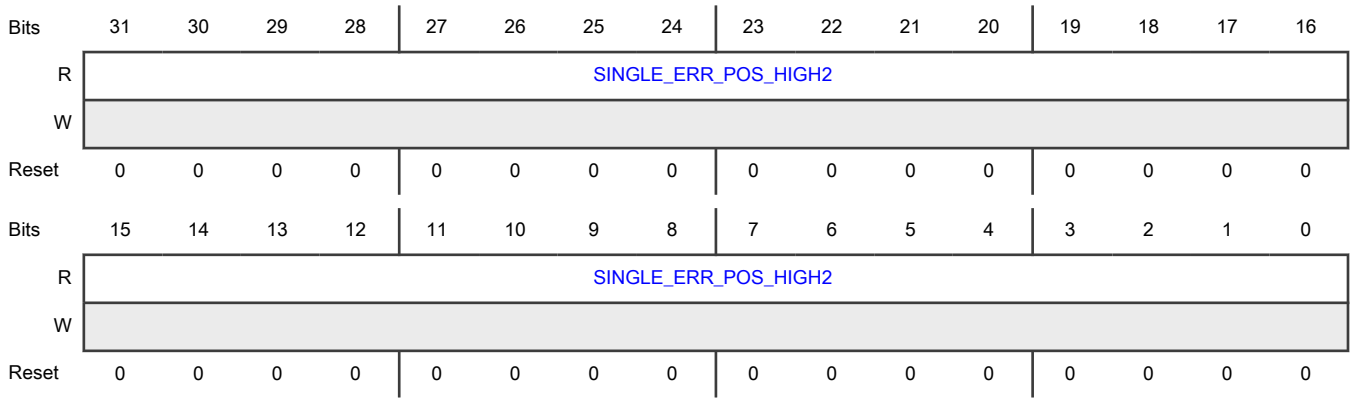
19.6.1.76 High Single Error Bit Position On Ocram Bank2 (SINGLE_ERR_POS_HIGH2)

When single bit error is corrected in ocram bank2 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH2	128h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH2	High Single Error Bit Position On Ocram Bank2 When single bit error is corrected in ocram bank2 read access, indicates which bit is corrected in high 32 bits of read data.

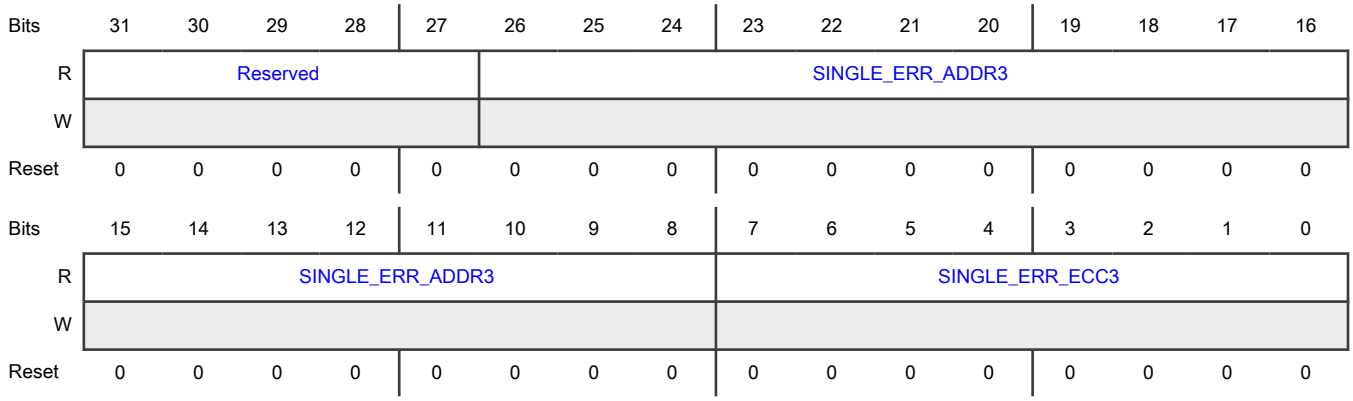
19.6.1.77 Single Error Address And ECC Code On Ocram Bank3 (SINGLE_ERR_ADDR_ECC3)

When single bit error is corrected in ocram bank3 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC3	12Ch

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR3	Single Error Address On Ocram Bank3 When single bit error is corrected in ocram bank3 read access, store ocram address.
7-0 SINGLE_ERR_ECC3	Single Error ECC Code On Ocram Bank3 When single bit error is corrected in ocram bank3 read access, store ecc code.

19.6.1.78 Low 32 Bits Single Error Read Data On Ocram Bank3 (SINGLE_ERR_DATA_LOW3)

When single bit error is corrected in ocram bank3 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW3	130h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_LOW3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_LOW3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW3	Low 32 Bits Single Error Read Data On Ocram Bank3 When single bit error is corrected in ocram bank3 read access, store low 32 bits uncorrected read data.

19.6.1.79 High 32 Bits Single Error Read Data On Ocram Bank3 (SINGLE_ERR_DATA_HIGH3)

When single bit error is corrected in ocram bank3 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH3	134h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_HIGH3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_HIGH3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH3	High 32 Bits Single Error Read Data On Ocram Bank3 When single bit error is corrected in ocram bank3 read access, store high 32 bits uncorrected read data.

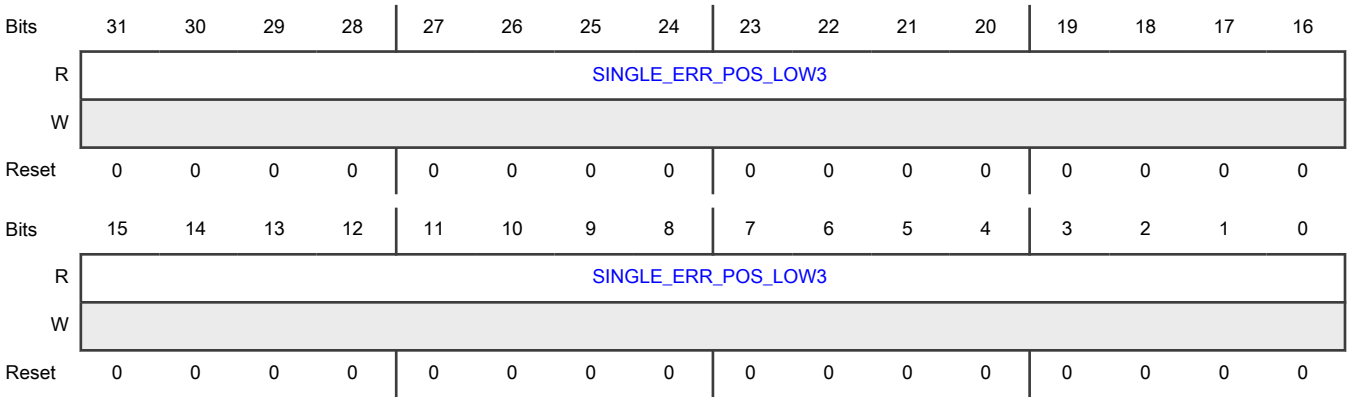
19.6.1.80 Low Single Error Bit Position On Ocram Bank3 (SINGLE_ERR_POS_LOW3)

When single bit error is corrected in ocram bank3 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW3	138h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW3	Low Single Error Bit Position On Ocram Bank3 When single bit error is corrected in ocram bank3 read access, indicates which bit is corrected in low 32 bits of read data.

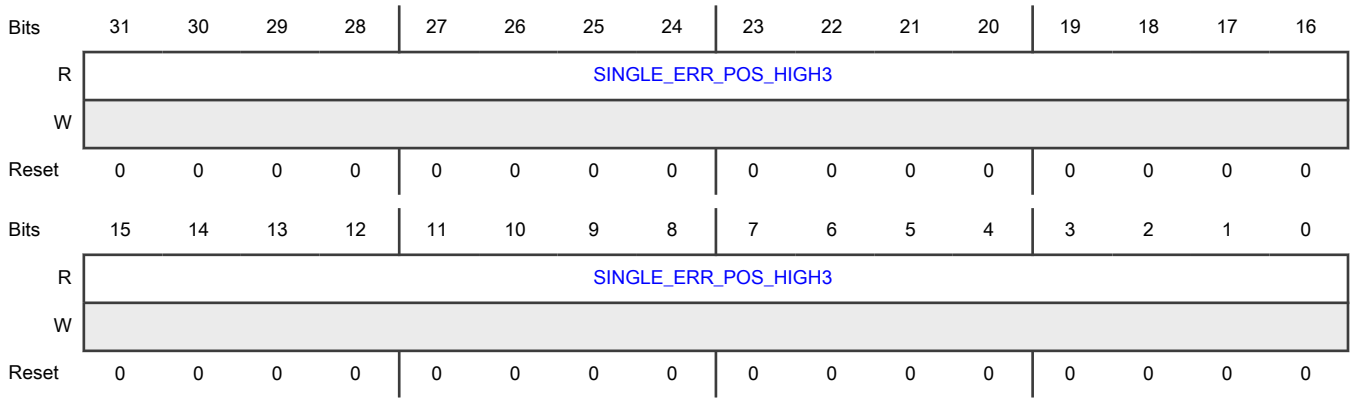
19.6.1.81 High Single Error Bit Position On Ocram Bank3 (SINGLE_ERR_POS_HIGH3)

When single bit error is corrected in ocram bank3 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH3	13Ch

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH3	High Single Error Bit Position On Ocram Bank3 When single bit error is corrected in ocram bank3 read access, indicates which bit is corrected in high 32 bits of read data.

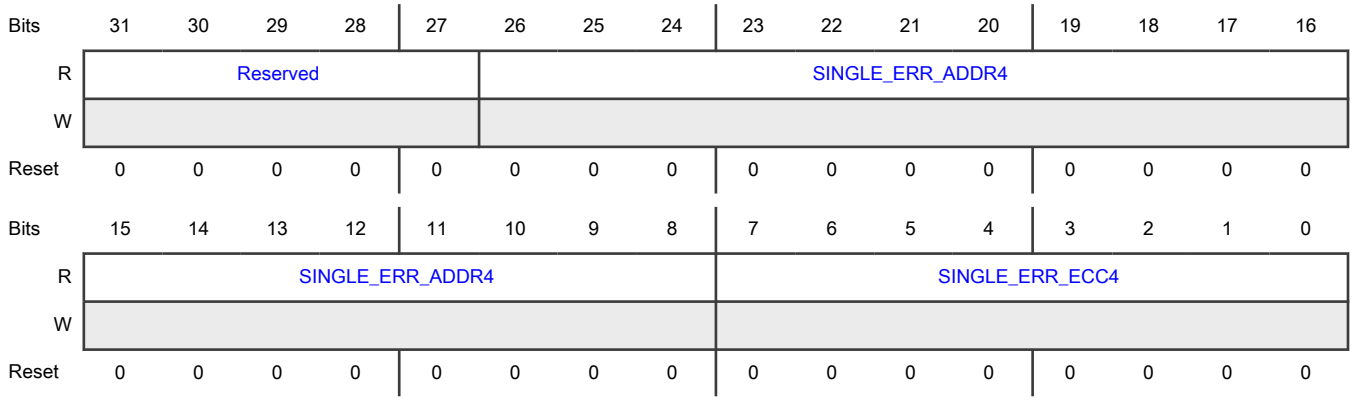
19.6.1.82 Single Error Address And ECC Code On Ocram Bank4 (SINGLE_ERR_ADDR_ECC4)

When single bit error is corrected in ocram bank4 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC4	140h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR4	Single Error Address On Ocram Bank4 When single bit error is corrected in ocram bank4 read access, store ocram address.
7-0 SINGLE_ERR_ECC4	Single Error ECC Code On Ocram Bank4 When single bit error is corrected in ocram bank4 read access, store ecc code.

19.6.1.83 Low 32 Bits Single Error Read Data On Ocram Bank4 (SINGLE_ERR_DATA_LOW4)

When single bit error is corrected in ocram bank4 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW4	144h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_LOW4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_LOW4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW4	Low 32 Bits Single Error Read Data On Ocram Bank4 When single bit error is corrected in ocram bank4 read access, store low 32 bits uncorrected read data.

19.6.1.84 High 32 Bits Single Error Read Data On Ocram Bank4 (SINGLE_ERR_DATA_HIGH4)

When single bit error is corrected in ocram bank4 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH4	148h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_HIGH4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_HIGH4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH4	High 32 Bits Single Error Read Data On Ocram Bank4 When single bit error is corrected in ocram bank4 read access, store high 32 bits uncorrected read data.

19.6.1.85 Low Single Error Bit Position On Ocram Bank4 (SINGLE_ERR_POS_LOW4)

When single bit error is corrected in ocram bank4 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW4	14Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS_LOW4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS_LOW4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW4	Low Single Error Bit Position On Ocram Bank4 When single bit error is corrected in ocram bank4 read access, indicates which bit is corrected in low 32 bits of read data.

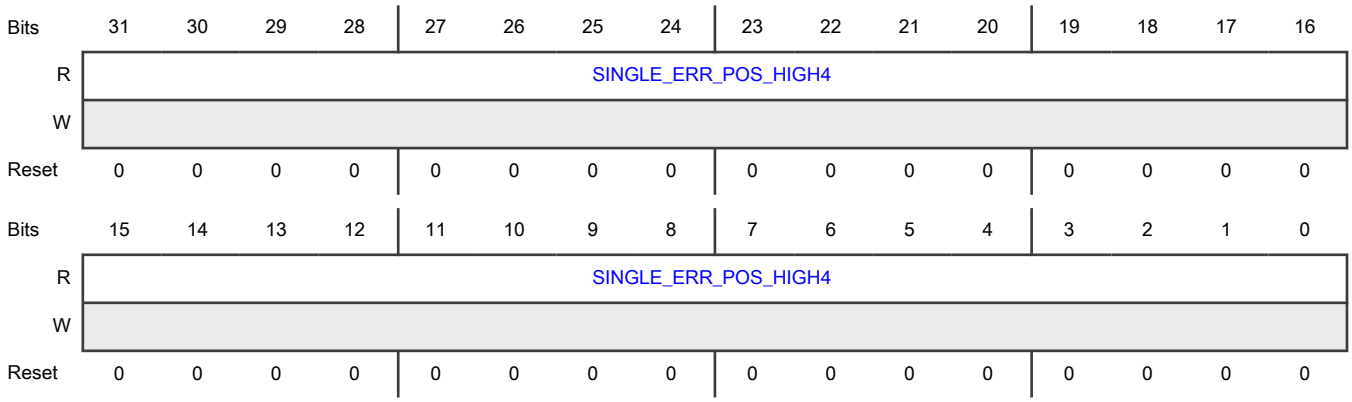
19.6.1.86 High Single Error Bit Position On Ocram Bank4 (SINGLE_ERR_POS_HIGH4)

When single bit error is corrected in ocram bank4 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH4	150h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH4	High Single Error Bit Position On Ocram Bank4 When single bit error is corrected in ocram bank4 read access, indicates which bit is corrected in high 32 bits of read data.

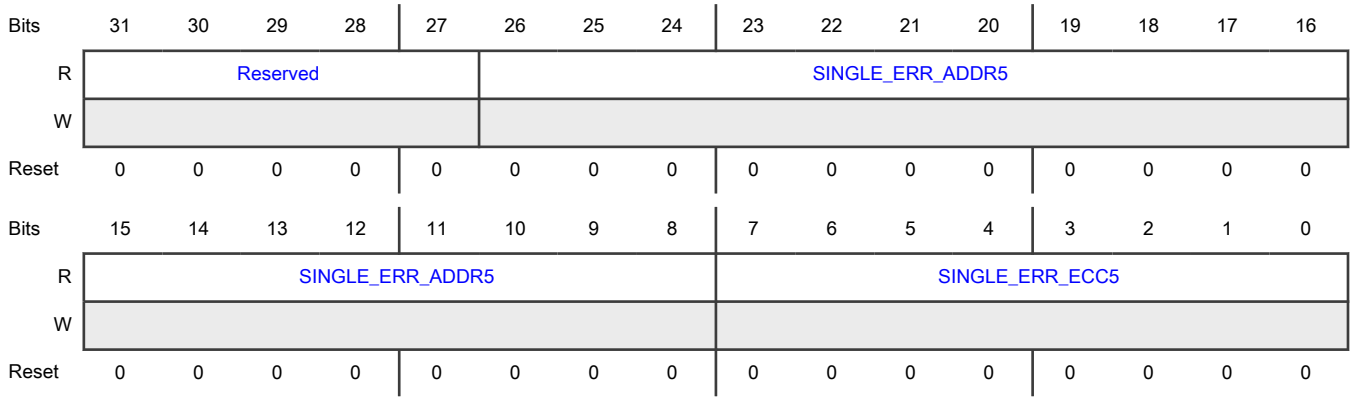
19.6.1.87 Single Error Address And ECC Code On Ocram Bank5 (SINGLE_ERR_ADDR_ECC5)

When single bit error is corrected in ocram bank5 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC5	154h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR5	Single Error Address On Ocram Bank5 When single bit error is corrected in ocram bank5 read access, store ocram address.
7-0 SINGLE_ERR_ECC5	Single Error ECC Code On Ocram Bank5 When single bit error is corrected in ocram bank5 read access, store ecc code.

19.6.1.88 Low 32 Bits Single Error Read Data On Ocram Bank5 (SINGLE_ERR_DATA_LOW5)

When single bit error is corrected in ocram bank5 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW5	158h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_LOW5															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_LOW5															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW5	Low 32 Bits Single Error Read Data On Ocram Bank5 When single bit error is corrected in ocram bank5 read access, store low 32 bits uncorrected read data.

19.6.1.89 High 32 Bits Single Error Read Data On Ocram Bank5 (SINGLE_ERR_DATA_HIGH5)

When single bit error is corrected in ocram bank5 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH5	15Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_HIGH5															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_HIGH5															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH5	High 32 Bits Single Error Read Data On Ocram Bank5 When single bit error is corrected in ocram bank5 read access, store high 32 bits uncorrected read data.

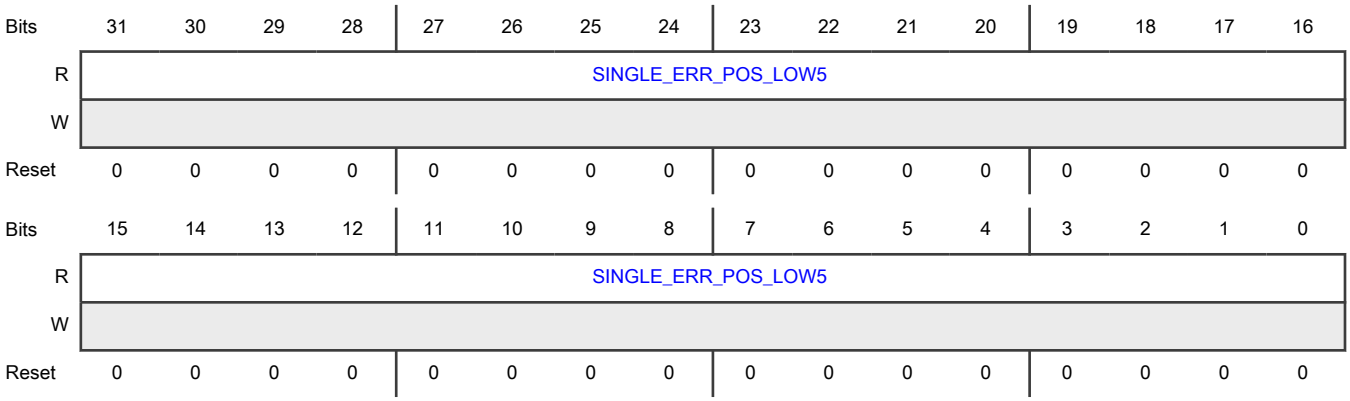
19.6.1.90 Low Single Error Bit Position On Ocram Bank5 (SINGLE_ERR_POS_LOW5)

When single bit error is corrected in ocram bank5 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW5	160h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW5	Low Single Error Bit Position On Ocram Bank5 When single bit error is corrected in ocram bank5 read access, indicates which bit is corrected in low 32 bits of read data.

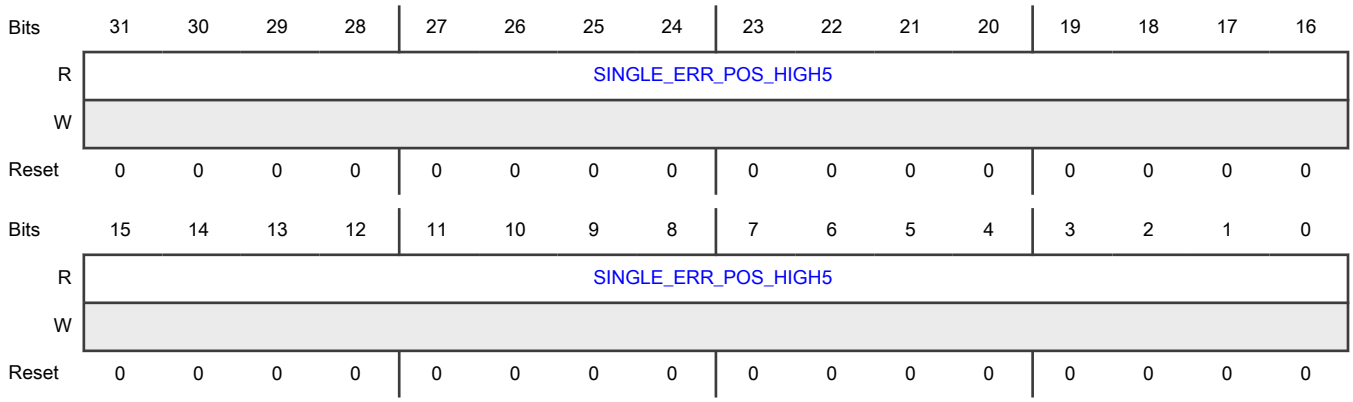
19.6.1.91 High Single Error Bit Position On Ocram Bank5 (SINGLE_ERR_POS_HIGH5)

When single bit error is corrected in ocram bank5 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH5	164h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH5	High Single Error Bit Position On Ocram Bank5 When single bit error is corrected in ocram bank5 read access, indicates which bit is corrected in high 32 bits of read data.

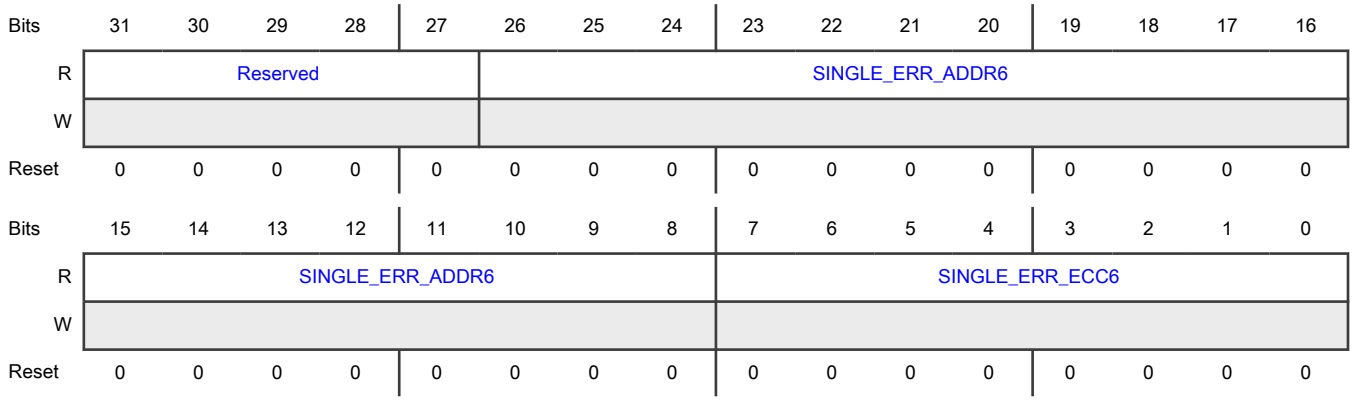
19.6.1.92 Single Error Address And ECC Code On Ocram Bank6 (SINGLE_ERR_ADDR_ECC6)

When single bit error is corrected in ocram bank6 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC6	168h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR6	Single Error Address On Ocram Bank6 When single bit error is corrected in ocram bank6 read access, store ocram address.
7-0 SINGLE_ERR_ECC6	Single Error ECC Code On Ocram Bank6 When single bit error is corrected in ocram bank6 read access, store ecc code.

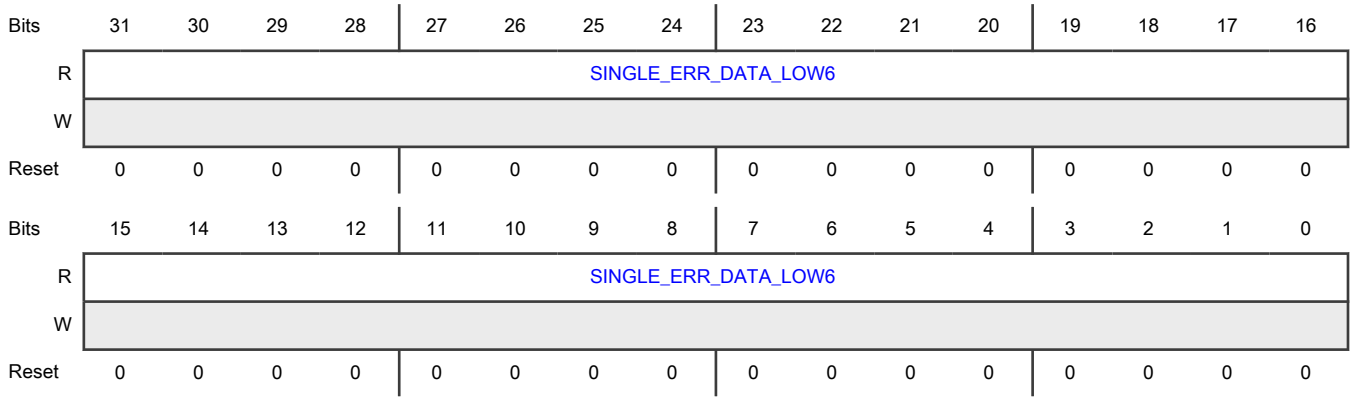
19.6.1.93 Low 32 Bits Single Error Read Data On Ocram Bank6 (SINGLE_ERR_DATA_LOW6)

When single bit error is corrected in ocram bank6 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW6	16Ch

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW6	Low 32 Bits Single Error Read Data On Ocram Bank6 When single bit error is corrected in ocram bank6 read access, store low 32 bits uncorrected read data.

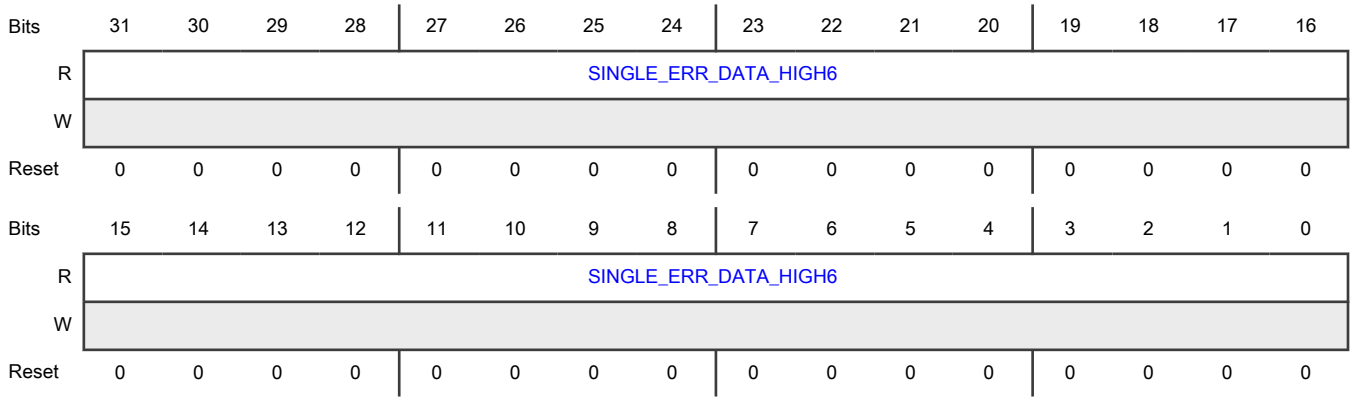
19.6.1.94 High 32 Bits Single Error Read Data On Ocram Bank6 (SINGLE_ERR_DATA_HIGH6)

When single bit error is corrected in ocram bank6 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH6	170h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH6	High 32 Bits Single Error Read Data On Ocram Bank6 When single bit error is corrected in ocram bank6 read access, store high 32 bits uncorrected read data.

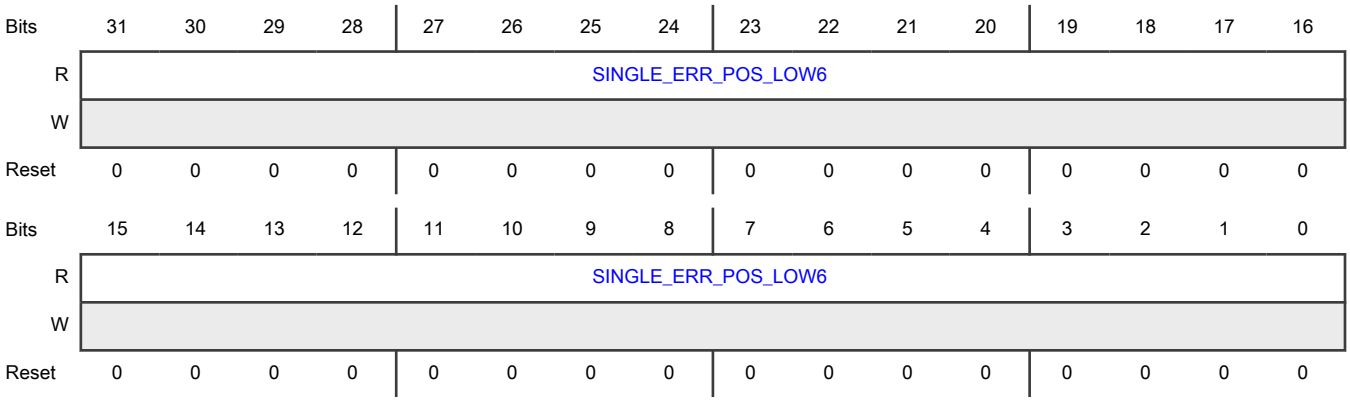
19.6.1.95 Low Single Error Bit Position On Ocram Bank6 (SINGLE_ERR_POS_LOW6)

When single bit error is corrected in ocram bank6 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW6	174h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW6	Low Single Error Bit Position On Ocram Bank6 When single bit error is corrected in ocram bank6 read access, indicates which bit is corrected in low 32 bits of read data.

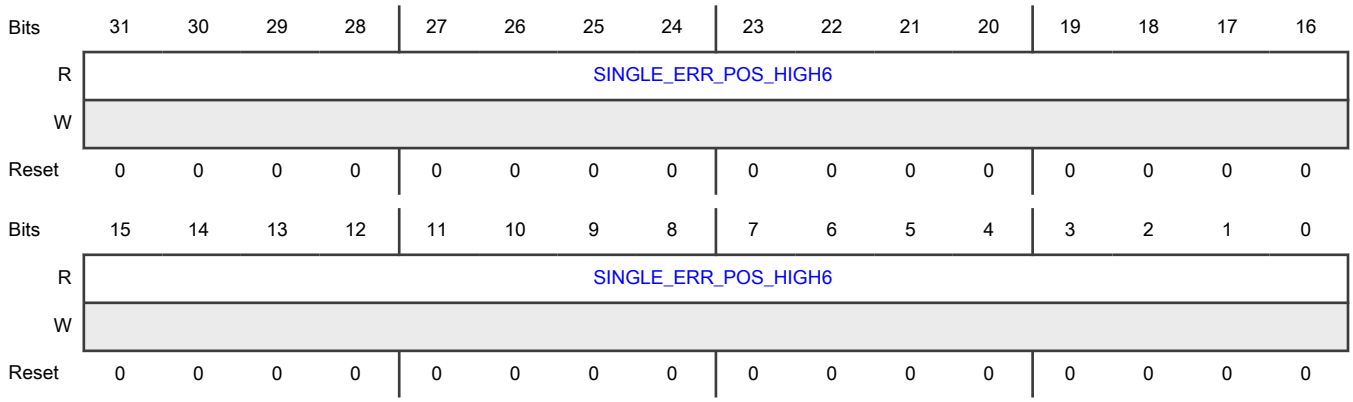
19.6.1.96 High Single Error Bit Position On Ocram Bank6 (SINGLE_ERR_POS_HIGH6)

When single bit error is corrected in ocram bank6 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH6	178h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH6	High Single Error Bit Position On Ocram Bank6 When single bit error is corrected in ocram bank6 read access, indicates which bit is corrected in high 32 bits of read data.

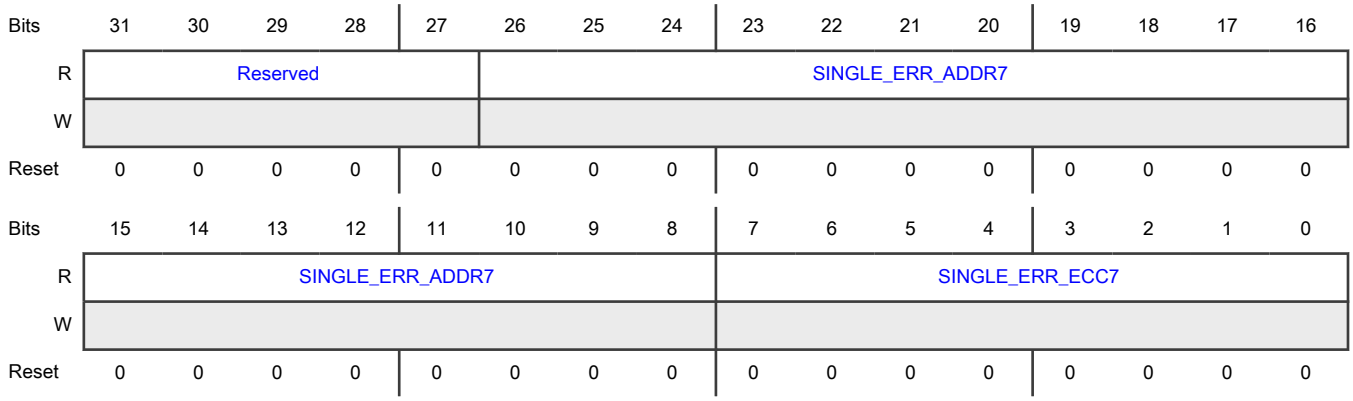
19.6.1.97 Single Error Address And ECC Code On Ocram Bank7 (SINGLE_ERR_ADDR_ECC7)

When single bit error is corrected in ocram bank7 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC7	17Ch

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR7	Single Error Address On Ocram Bank7 When single bit error is corrected in ocram bank7 read access, store ocram address.
7-0 SINGLE_ERR_ECC7	Single Error ECC Code On Ocram Bank7 When single bit error is corrected in ocram bank7 read access, store ecc code.

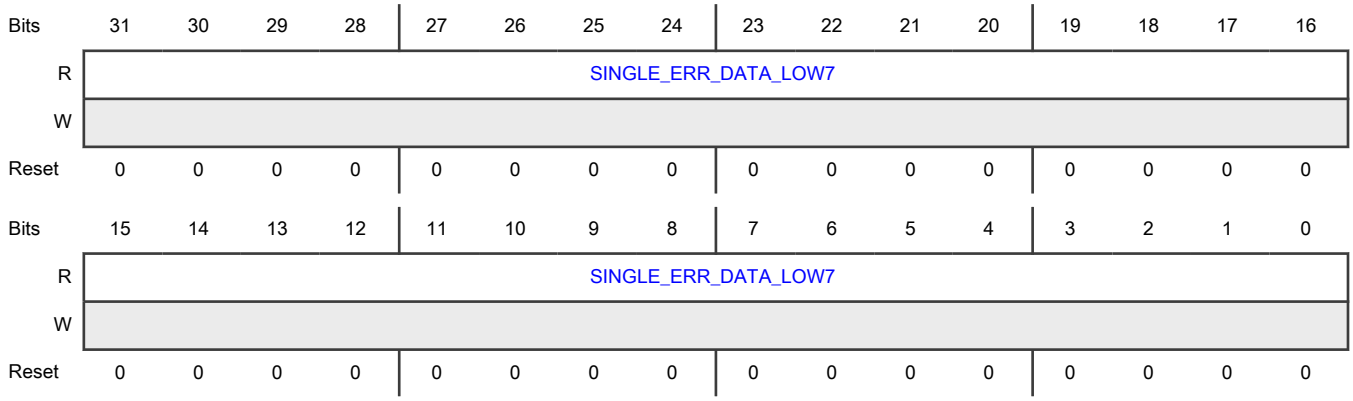
19.6.1.98 Low 32 Bits Single Error Read Data On Ocram Bank7 (SINGLE_ERR_DATA_LOW7)

When single bit error is corrected in ocram bank7 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW7	180h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW7	Low 32 Bits Single Error Read Data On Ocram Bank7 When single bit error is corrected in ocram bank7 read access, store low 32 bits uncorrected read data.

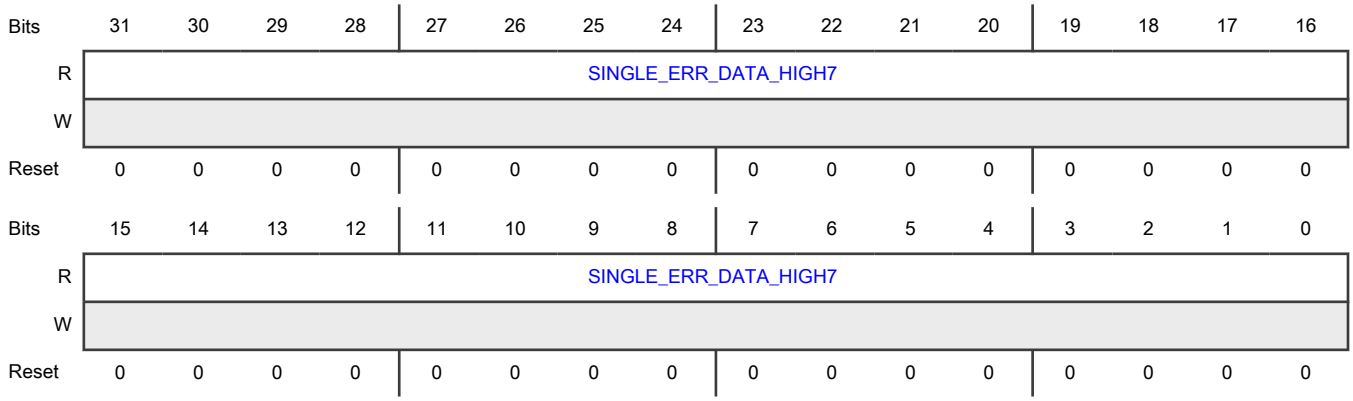
19.6.1.99 High 32 Bits Single Error Read Data On Ocram Bank7 (SINGLE_ERR_DATA_HIGH7)

When single bit error is corrected in ocram bank7 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH7	184h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH7	High 32 Bits Single Error Read Data On Ocram Bank7 When single bit error is corrected in ocram bank7 read access, store high 32 bits uncorrected read data.

19.6.1.100 Low Single Error Bit Position On Ocram Bank7 (SINGLE_ERR_POS_LOW7)

When single bit error is corrected in ocram bank7 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW7	188h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS_LOW7															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS_LOW7															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW7	Low Single Error Bit Position On Ocram Bank7 When single bit error is corrected in ocram bank7 read access, indicates which bit is corrected in low 32 bits of read data.

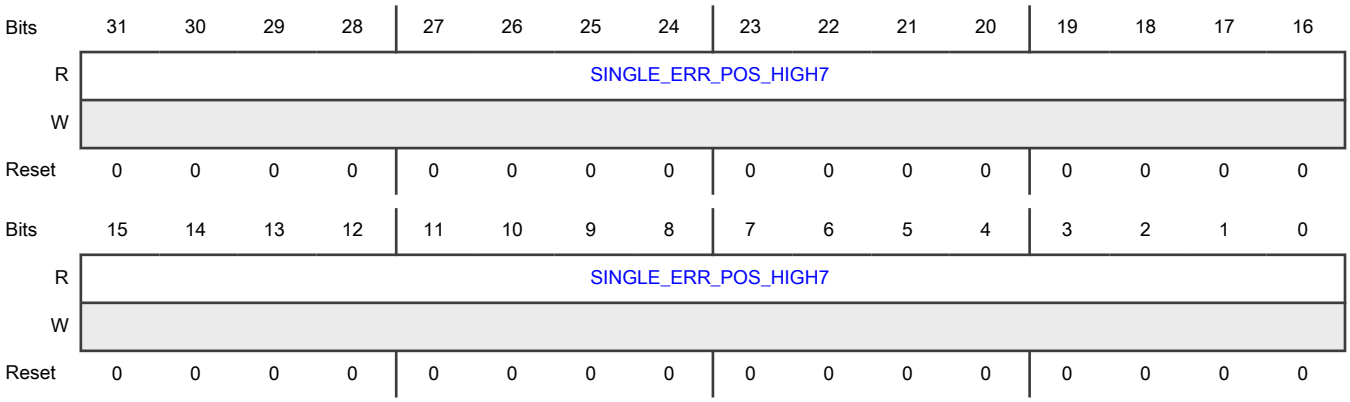
19.6.1.101 High Single Error Bit Position On Ocram Bank7 (SINGLE_ERR_POS_HIGH7)

When single bit error is corrected in ocram bank7 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH7	18Ch

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH7	High Single Error Bit Position On Ocram Bank7 When single bit error is corrected in ocram bank7 read access, indicates which bit is corrected in high 32 bits of read data.

19.6.1.102 Single Error Address And ECC Code On Ocram Bank8 (SINGLE_ERR_ADDR_ECC8)

When single bit error is corrected in ocram bank8 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC8	190h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved					SINGLE_ERR_ADDR8										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_ADDR8								SINGLE_ERR_ECC8							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR8	Single Error Address On Ocram Bank8 When single bit error is corrected in ocram bank8 read access, store ocram address.
7-0 SINGLE_ERR_ECC8	Single Error ECC Code On Ocram Bank8 When single bit error is corrected in ocram bank8 read access, store ecc code.

19.6.1.103 Low 32 Bits Single Error Read Data On Ocram Bank8 (SINGLE_ERR_DATA_LOW8)

When single bit error is corrected in ocram bank8 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW8	194h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_LOW8															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_LOW8															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW8	Low 32 Bits Single Error Read Data On Ocram Bank8 When single bit error is corrected in ocram bank8 read access, store low 32 bits uncorrected read data.

19.6.1.104 High 32 Bits Single Error Read Data On Ocram Bank8 (SINGLE_ERR_DATA_HIGH8)

When single bit error is corrected in ocram bank8 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH8	198h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_HIGH8															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_HIGH8															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH8	High 32 Bits Single Error Read Data On Ocram Bank8 When single bit error is corrected in ocram bank8 read access, store high 32 bits uncorrected read data.

19.6.1.105 Low Single Error Bit Position On Ocram Bank8 (SINGLE_ERR_POS_LOW8)

When single bit error is corrected in ocram bank8 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW8	19Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS_LOW8															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS_LOW8															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW8	Low Single Error Bit Position On Ocram Bank8 When single bit error is corrected in ocram bank8 read access, indicates which bit is corrected in low 32 bits of read data.

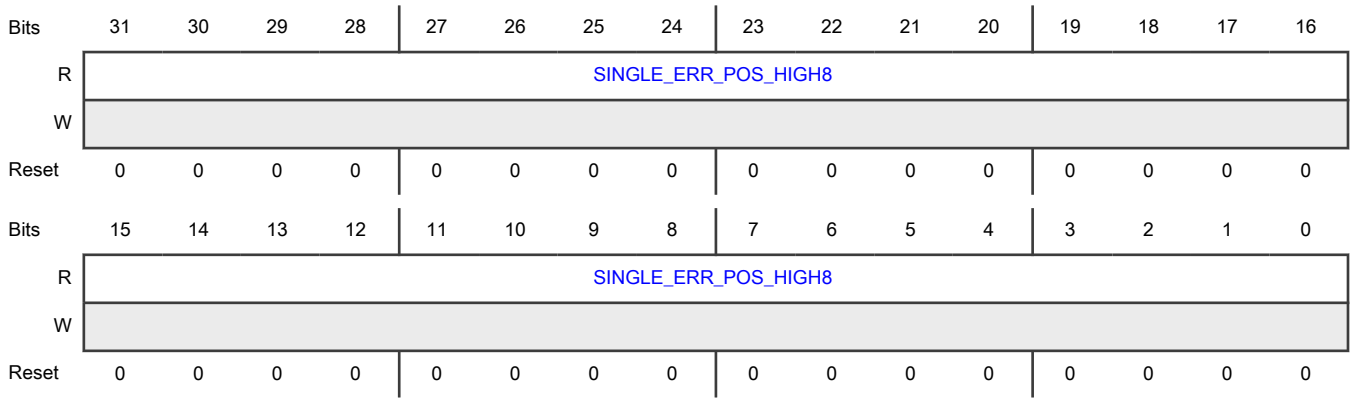
19.6.1.106 High Single Error Bit Position On Ocram Bank8 (SINGLE_ERR_POS_HIGH8)

When single bit error is corrected in ocram bank8 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH8	1A0h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH8	High Single Error Bit Position On Ocram Bank8 When single bit error is corrected in ocram bank8 read access, indicates which bit is corrected in high 32 bits of read data.

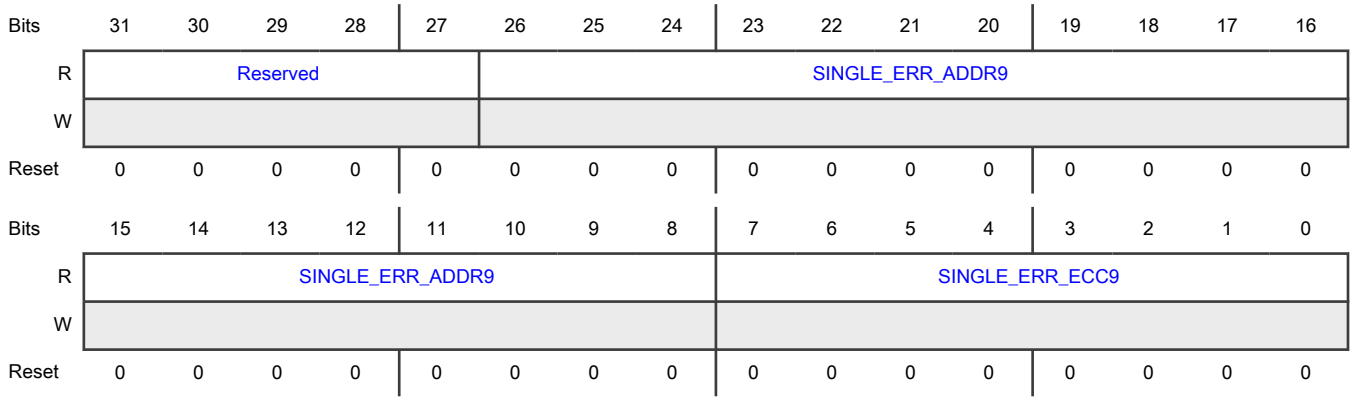
19.6.1.107 Single Error Address And ECC Code On Ocram Bank9 (SINGLE_ERR_ADDR_ECC9)

When single bit error is corrected in ocram bank9 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC9	1A4h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR9	Single Error Address On Ocram Bank9 When single bit error is corrected in ocram bank9 read access, store ocram address.
7-0 SINGLE_ERR_ECC9	Single Error ECC Code On Ocram Bank9 When single bit error is corrected in ocram bank9 read access, store ecc code.

19.6.1.108 Low 32 Bits Single Error Read Data On Ocram Bank9 (SINGLE_ERR_DATA_LOW9)

When single bit error is corrected in ocram bank9 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW9	1A8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_LOW9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_LOW9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW9	Low 32 Bits Single Error Read Data On Ocram Bank9 When single bit error is corrected in ocram bank9 read access, store low 32 bits uncorrected read data.

19.6.1.109 High 32 Bits Single Error Read Data On Ocram Bank9 (SINGLE_ERR_DATA_HIGH9)

When single bit error is corrected in ocram bank9 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH9	1ACh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_HIGH9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_HIGH9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH9	High 32 Bits Single Error Read Data On Ocram Bank9 When single bit error is corrected in ocram bank9 read access, store high 32 bits uncorrected read data.

19.6.1.110 Low Single Error Bit Position On Ocram Bank9 (SINGLE_ERR_POS_LOW9)

When single bit error is corrected in ocram bank9 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW9	1B0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS_LOW9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS_LOW9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW9	Low Single Error Bit Position On Ocram Bank9 When single bit error is corrected in ocram bank9 read access, indicates which bit is corrected in low 32 bits of read data.

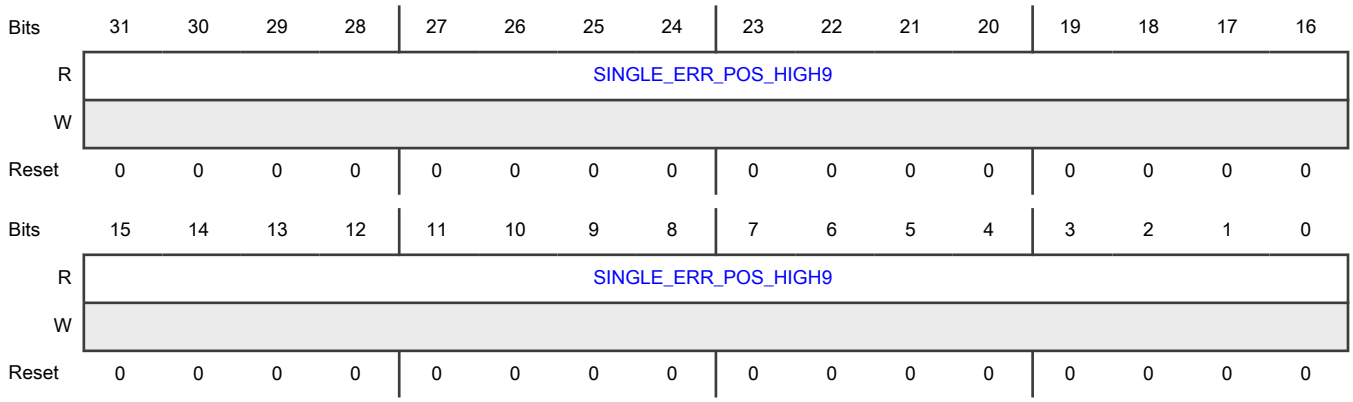
19.6.1.111 High Single Error Bit Position On Ocram Bank9 (SINGLE_ERR_POS_HIGH9)

When single bit error is corrected in ocram bank9 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH9	1B4h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH9	High Single Error Bit Position On Ocram Bank9 When single bit error is corrected in ocram bank9 read access, indicates which bit is corrected in high 32 bits of read data.

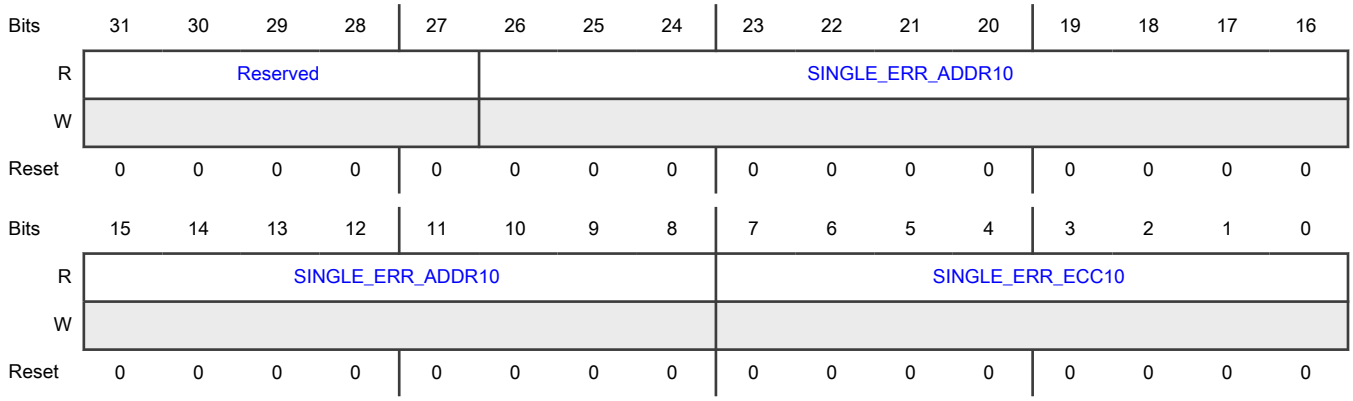
19.6.1.112 Single Error Address And ECC Code On Ocram Bank10 (SINGLE_ERR_ADDR_ECC10)

When single bit error is corrected in ocram bank10 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC10	1B8h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR10	Single Error Address On Ocram Bank10 When single bit error is corrected in ocram bank10 read access, store ocram address.
7-0 SINGLE_ERR_ECC10	Single Error ECC Code On Ocram Bank2 When single bit error is corrected in ocram bank10 read access, store ecc code.

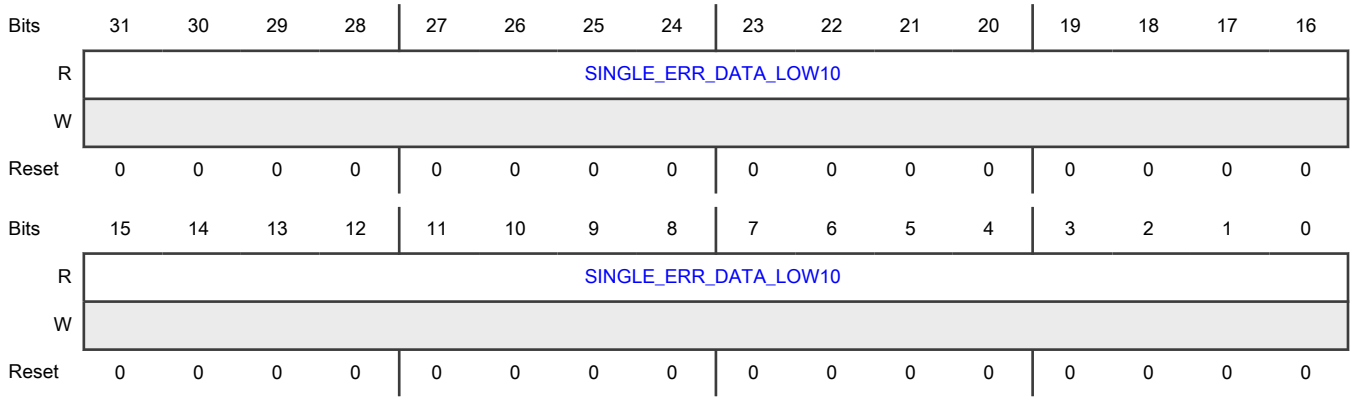
19.6.1.113 Low 32 Bits Single Error Read Data On Ocram Bank10 (SINGLE_ERR_DATA_LOW10)

When single bit error is corrected in ocram bank10 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW10	1BCh

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW10	Low 32 Bits Single Error Read Data On Ocram Bank10 When single bit error is corrected in ocram bank10 read access, store low 32 bits uncorrected read data.

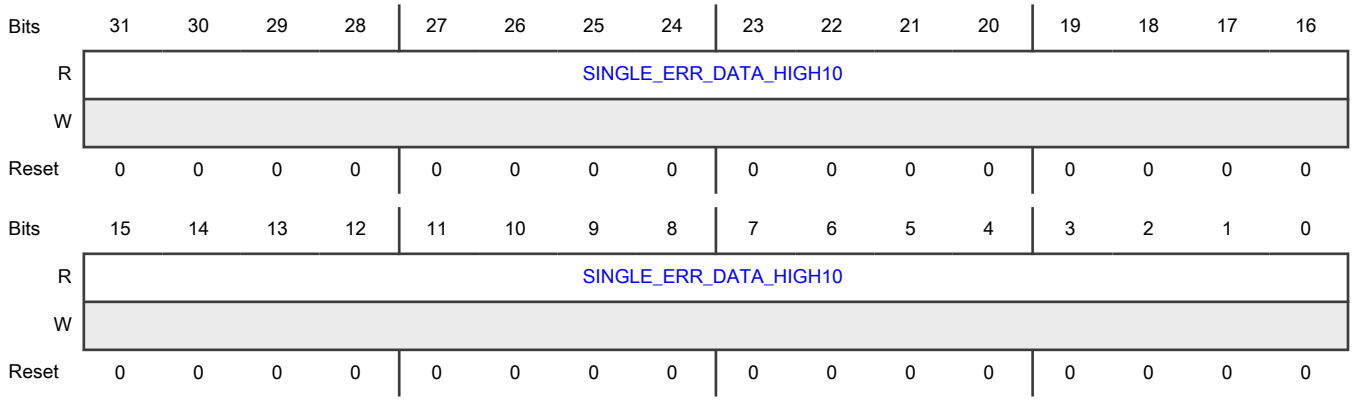
19.6.1.114 High 32 Bits Single Error Read Data On Ocram Bank10 (SINGLE_ERR_DATA_HIGH10)

When single bit error is corrected in ocram bank10 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH10	1C0h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH10	High 32 Bits Single Error Read Data On Ocram Bank10 When single bit error is corrected in ocram bank10 read access, store high 32 bits uncorrected read data.

19.6.1.115 Low Single Error Bit Position On Ocram Bank10 (SINGLE_ERR_POS_LOW10)

When single bit error is corrected in ocram bank10 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW10	1C4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS_LOW10															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS_LOW10															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW10	Low Single Error Bit Position On Ocram Bank10 When single bit error is corrected in ocram bank10 read access, indicates which bit is corrected in low 32 bits of read data.

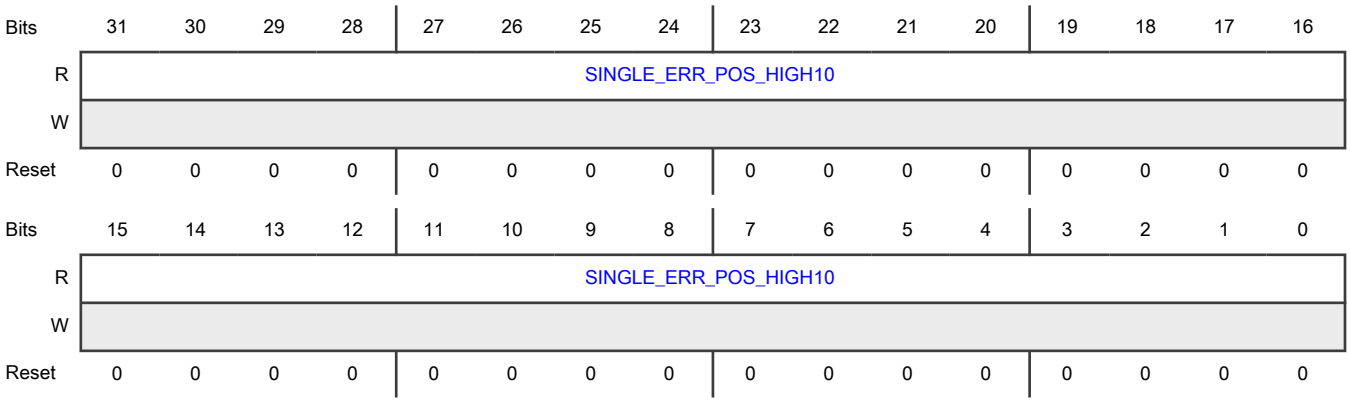
19.6.1.116 High Single Error Bit Position On Ocram Bank10 (SINGLE_ERR_POS_HIGH10)

When single bit error is corrected in ocram bank10 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH10	1C8h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH10	High Single Error Bit Position On Ocram Bank10 When single bit error is corrected in ocram bank10 read access, indicates which bit is corrected in high 32 bits of read data.

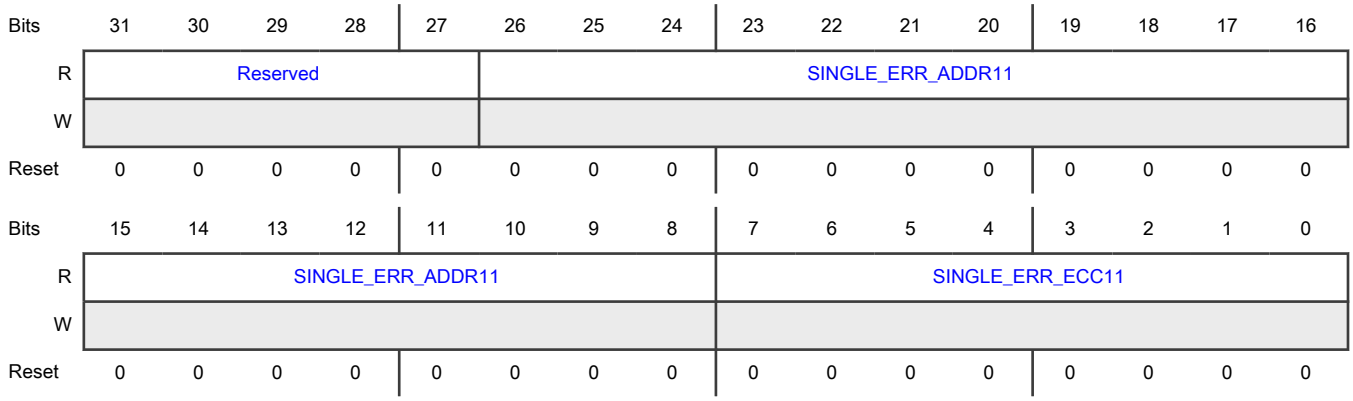
19.6.1.117 Single Error Address And ECC Code On Ocram Bank11 (SINGLE_ERR_ADDR_ECC11)

When single bit error is corrected in ocram bank11 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC11	1CCh

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR11	Single Error Address On Ocram Bank11 When single bit error is corrected in ocram bank11 read access, store ocram address.
7-0 SINGLE_ERR_ECC11	Single Error ECC Code On Ocram Bank11 When single bit error is corrected in ocram bank11 read access, store ecc code.

19.6.1.118 Low 32 Bits Single Error Read Data On Ocram Bank11 (SINGLE_ERR_DATA_LOW11)

When single bit error is corrected in ocram bank11 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW11	1D0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_LOW11															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_LOW11															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW11	Low 32 Bits Single Error Read Data On Ocram Bank11 When single bit error is corrected in ocram bank11 read access, store low 32 bits uncorrected read data.

19.6.1.119 High 32 Bits Single Error Read Data On Ocram Bank11 (SINGLE_ERR_DATA_HIGH11)

When single bit error is corrected in ocram bank11 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH11	1D4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_HIGH11															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_HIGH11															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH11	High 32 Bits Single Error Read Data On Ocram Bank11 When single bit error is corrected in ocram bank11 read access, store high 32 bits uncorrected read data.

19.6.1.120 Low Single Error Bit Position On Ocram Bank11 (SINGLE_ERR_POS_LOW11)

When single bit error is corrected in ocram bank11 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW11	1D8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS_LOW11															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS_LOW11															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW11	Low Single Error Bit Position On Ocram Bank11 When single bit error is corrected in ocram bank11 read access, indicates which bit is corrected in low 32 bits of read data.

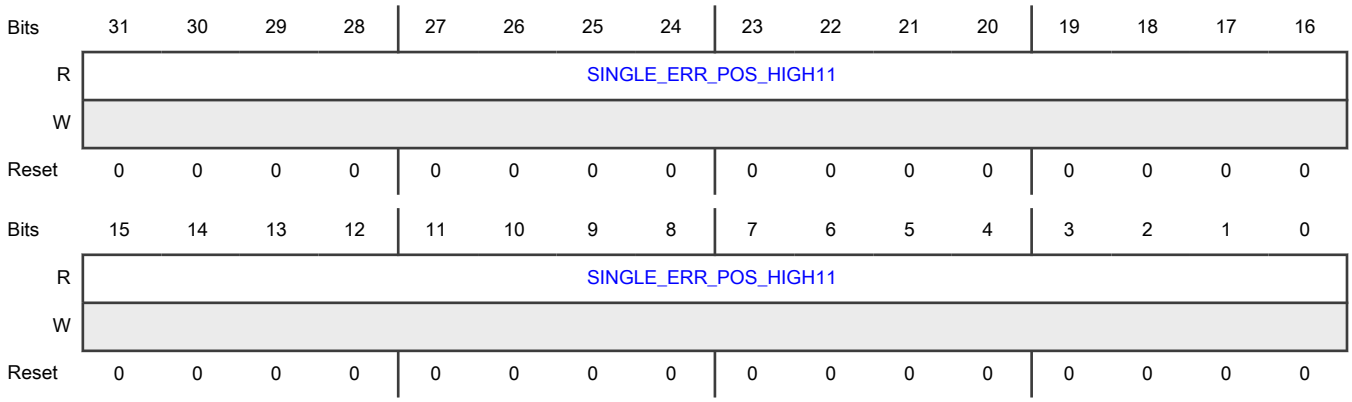
19.6.1.121 High Single Error Bit Position On Ocram Bank11 (SINGLE_ERR_POS_HIGH11)

When single bit error is corrected in ocram bank11 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH11	1DCh

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH11	High Single Error Bit Position On Ocram Bank11 When single bit error is corrected in ocram bank11 read access, indicates which bit is corrected in high 32 bits of read data.

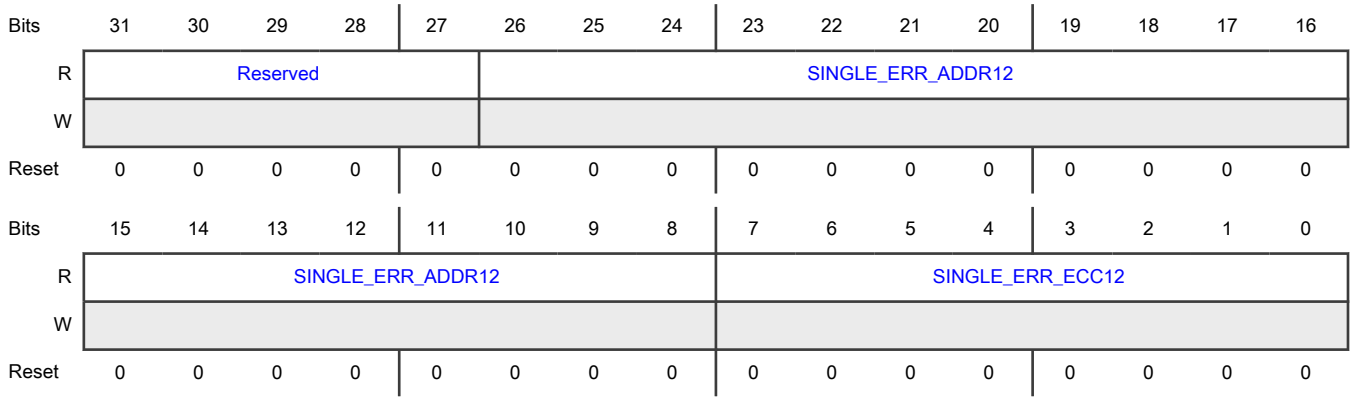
19.6.1.122 Single Error Address And ECC Code On Ocram Bank12 (SINGLE_ERR_ADDR_ECC12)

When single bit error is corrected in ocram bank12 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC12	1E0h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR12	Single Error Address On Ocram Bank12 When single bit error is corrected in ocram bank12 read access, store ocram address.
7-0 SINGLE_ERR_ECC12	Single Error ECC Code On Ocram Bank12 When single bit error is corrected in ocram bank12 read access, store ecc code.

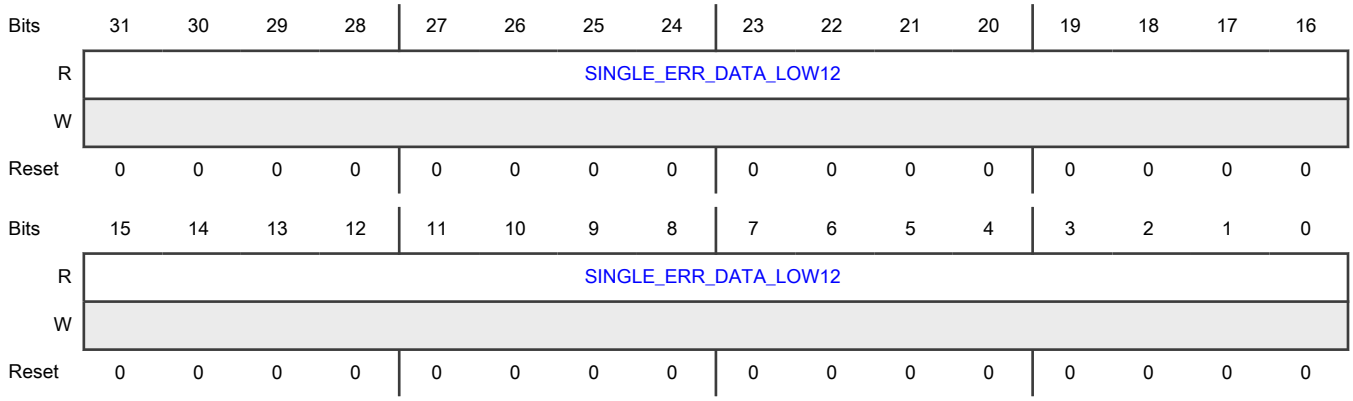
19.6.1.123 Low 32 Bits Single Error Read Data On Ocram Bank12 (SINGLE_ERR_DATA_LOW12)

When single bit error is corrected in ocram bank12 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW12	1E4h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW12	Low 32 Bits Single Error Read Data On Ocram Bank12 When single bit error is corrected in ocram bank12 read access, store low 32 bits uncorrected read data.

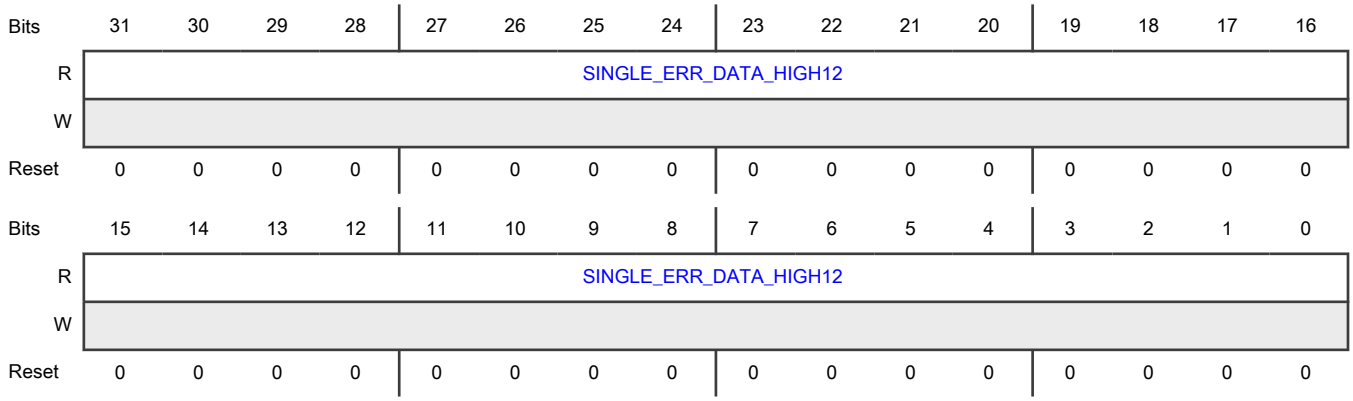
19.6.1.124 High 32 Bits Single Error Read Data On Ocram Bank12 (SINGLE_ERR_DATA_HIGH12)

When single bit error is corrected in ocram bank12 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH12	1E8h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH12	High 32 Bits Single Error Read Data On Ocram Bank12 When single bit error is corrected in ocram bank12 read access, store high 32 bits uncorrected read data.

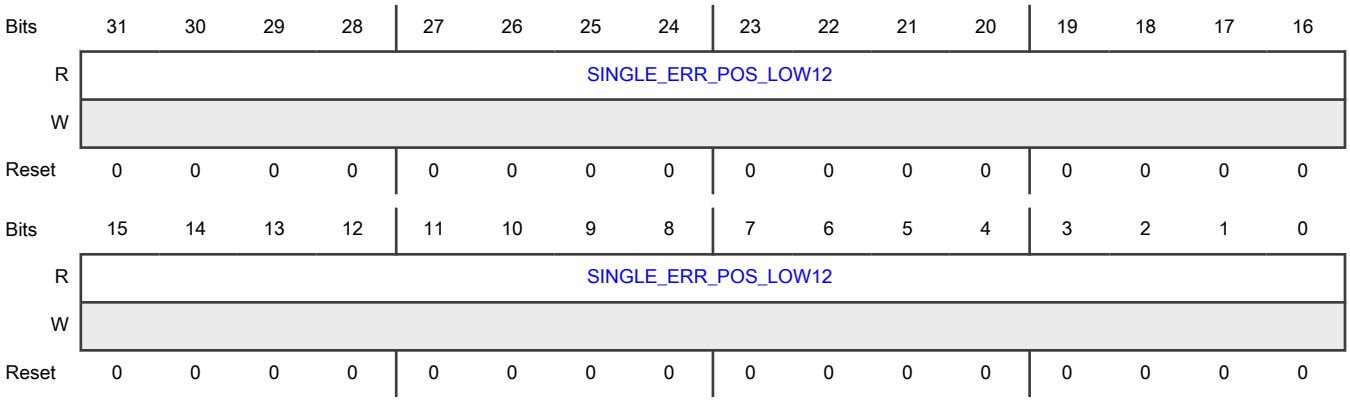
19.6.1.125 Low Single Error Bit Position On Ocram Bank12 (SINGLE_ERR_POS_LOW12)

When single bit error is corrected in ocram bank12 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW12	1ECh

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW12	Low Single Error Bit Position On Ocram Bank12 When single bit error is corrected in ocram bank12 read access, indicates which bit is corrected in low 32 bits of read data.

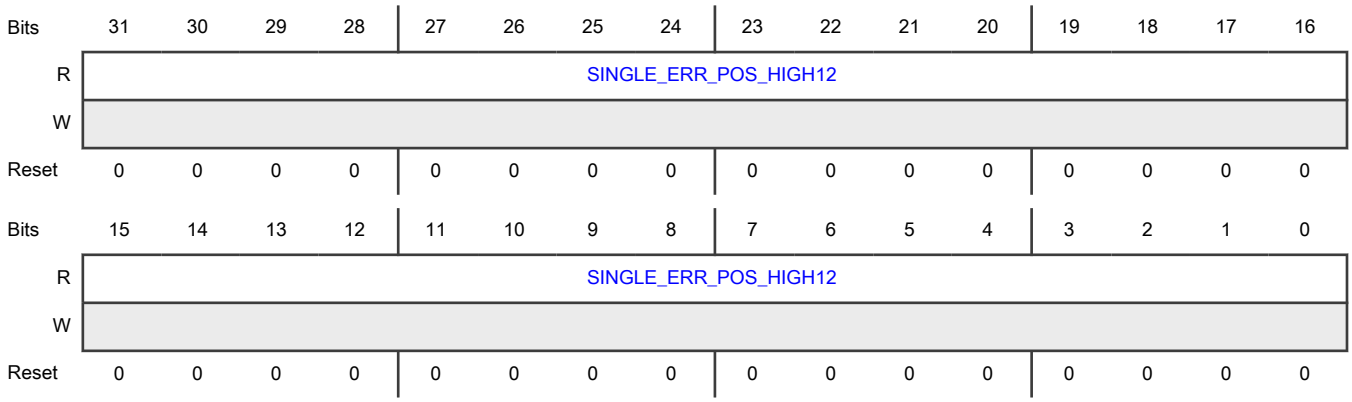
19.6.1.126 High Single Error Bit Position On Ocram Bank12 (SINGLE_ERR_POS_HIGH12)

When single bit error is corrected in ocram bank12 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH12	1F0h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH12	High Single Error Bit Position On Ocram Bank12 When single bit error is corrected in ocram bank12 read access, indicates which bit is corrected in high 32 bits of read data.

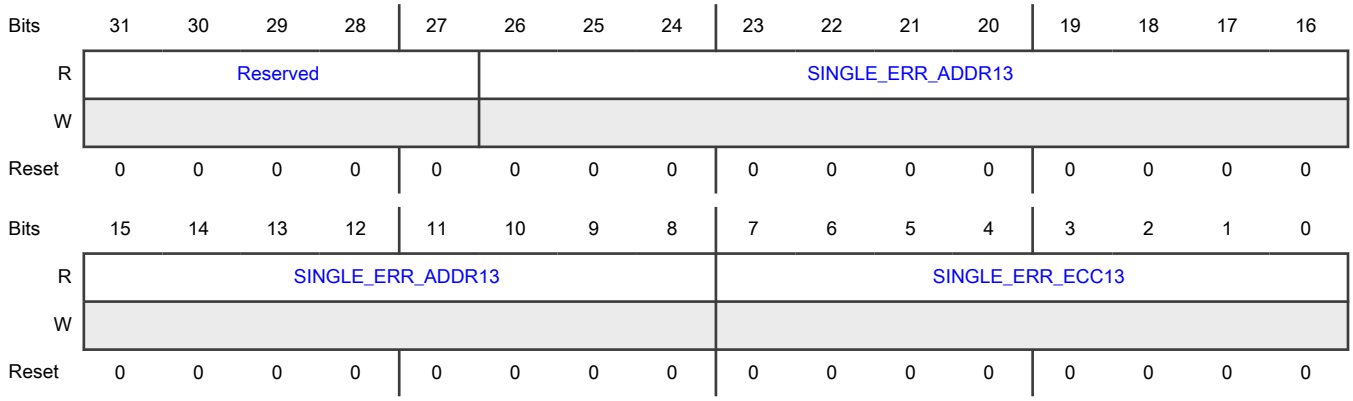
19.6.1.127 Single Error Address And ECC Code On Ocram Bank13 (SINGLE_ERR_ADDR_ECC13)

When single bit error is corrected in ocram bank13 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC13	1F4h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR13	Single Error Address On Ocram Bank13 When single bit error is corrected in ocram bank13 read access, store ocram address.
7-0 SINGLE_ERR_ECC13	Single Error ECC Code On Ocram Bank13 When single bit error is corrected in ocram bank13 read access, store ecc code.

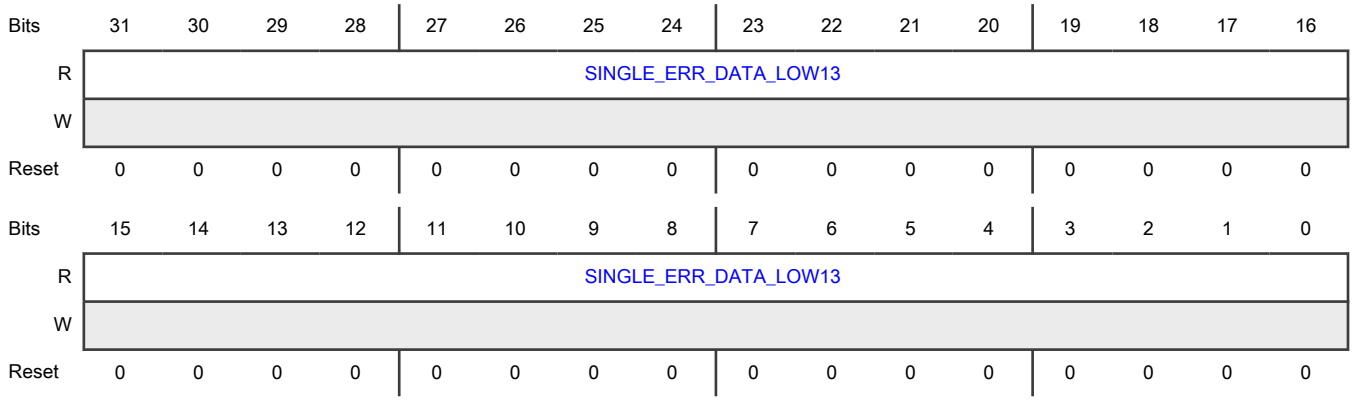
19.6.1.128 Low 32 Bits Single Error Read Data On Ocram Bank13 (SINGLE_ERR_DATA_LOW13)

When single bit error is corrected in ocram bank13 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW13	1F8h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW13	Low 32 Bits Single Error Read Data On Ocram Bank13 When single bit error is corrected in ocram bank13 read access, store low 32 bits uncorrected read data.

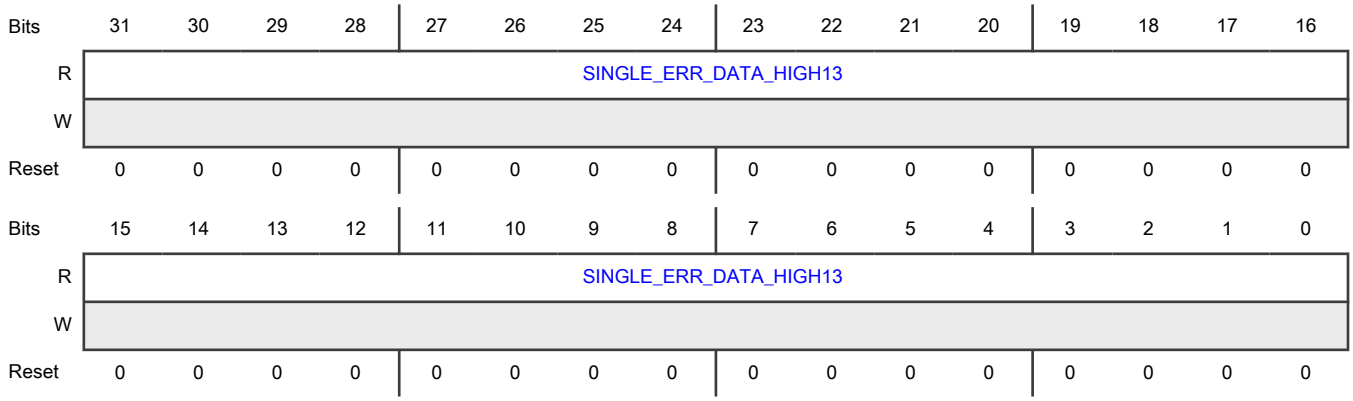
19.6.1.129 High 32 Bits Single Error Read Data On Ocram Bank13 (SINGLE_ERR_DATA_HIGH13)

When single bit error is corrected in ocram bank13 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH13	1FCh

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH13	High 32 Bits Single Error Read Data On Ocram Bank13 When single bit error is corrected in ocram bank13 read access, store high 32 bits uncorrected read data.

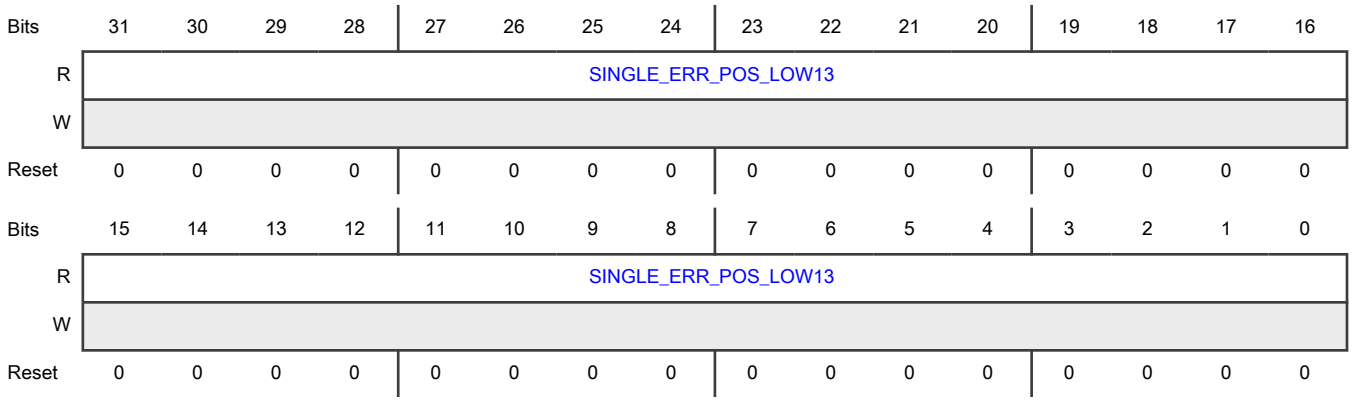
19.6.1.130 Low Single Error Bit Position On Ocram Bank13 (SINGLE_ERR_POS_LOW13)

When single bit error is corrected in ocram bank13 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW13	200h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW13	Low Single Error Bit Position On Ocram Bank13 When single bit error is corrected in ocram bank13 read access, indicates which bit is corrected in low 32 bits of read data.

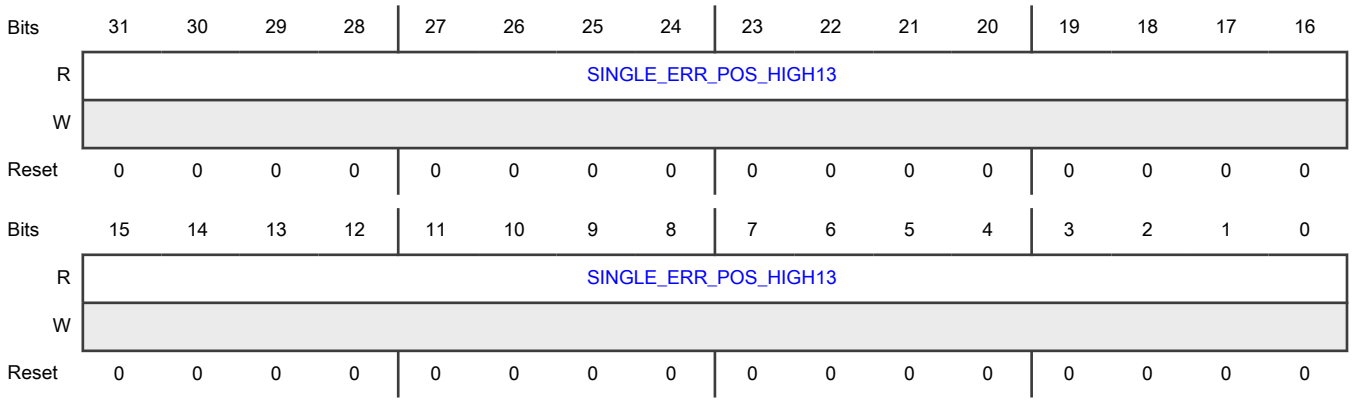
19.6.1.131 High Single Error Bit Position On Ocram Bank13 (SINGLE_ERR_POS_HIGH13)

When single bit error is corrected in ocram bank13 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH13	204h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH13	High Single Error Bit Position On Ocram Bank13 When single bit error is corrected in ocram bank13 read access, indicates which bit is corrected in high 32 bits of read data.

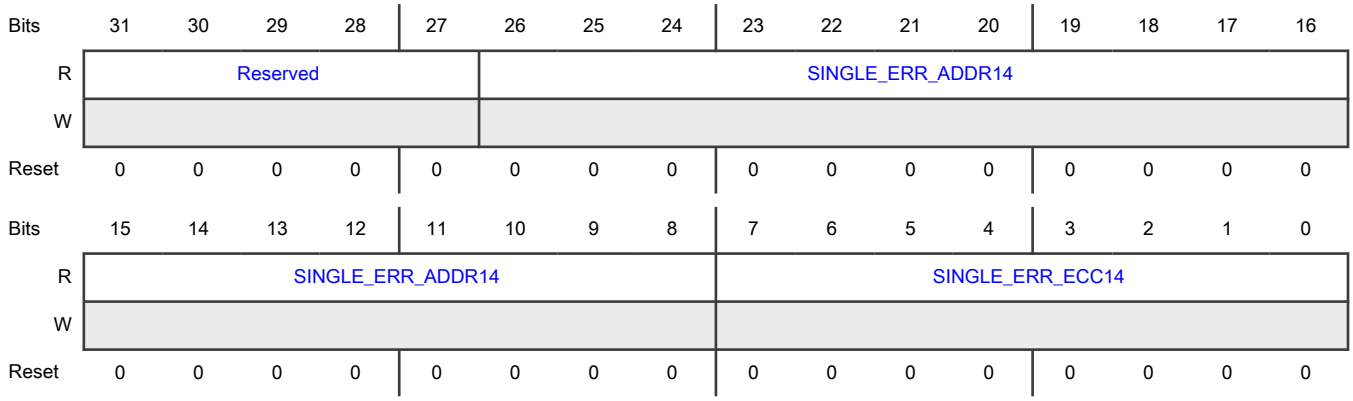
19.6.1.132 Single Error Address And ECC Code On Ocram Bank14 (SINGLE_ERR_ADDR_ECC14)

When single bit error is corrected in ocram bank14 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC14	208h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR14	Single Error Address On Ocram Bank14 When single bit error is corrected in ocram bank14 read access, store ocram address.
7-0 SINGLE_ERR_ECC14	Single Error ECC Code On Ocram Bank14 When single bit error is corrected in ocram bank14 read access, store ecc code.

19.6.1.133 Low 32 Bits Single Error Read Data On Ocram Bank14 (SINGLE_ERR_DATA_LOW14)

When single bit error is corrected in ocram bank14 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW14	20Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_LOW14															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_LOW14															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW14	Low 32 Bits Single Error Read Data On Ocram Bank14 When single bit error is corrected in ocram bank14 read access, store low 32 bits uncorrected read data.

19.6.1.134 High 32 Bits Single Error Read Data On Ocram Bank14 (SINGLE_ERR_DATA_HIGH14)

When single bit error is corrected in ocram bank14 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH14	210h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA_HIGH14															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA_HIGH14															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH14	High 32 Bits Single Error Read Data On Ocram Bank14 When single bit error is corrected in ocram bank14 read access, store high 32 bits uncorrected read data.

19.6.1.135 Low Single Error Bit Position On Ocram Bank14 (SINGLE_ERR_POS_LOW14)

When single bit error is corrected in ocram bank14 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW14	214h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS_LOW14															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS_LOW14															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW14	Low Single Error Bit Position On Ocram Bank14 When single bit error is corrected in ocram bank14 read access, indicates which bit is corrected in low 32 bits of read data.

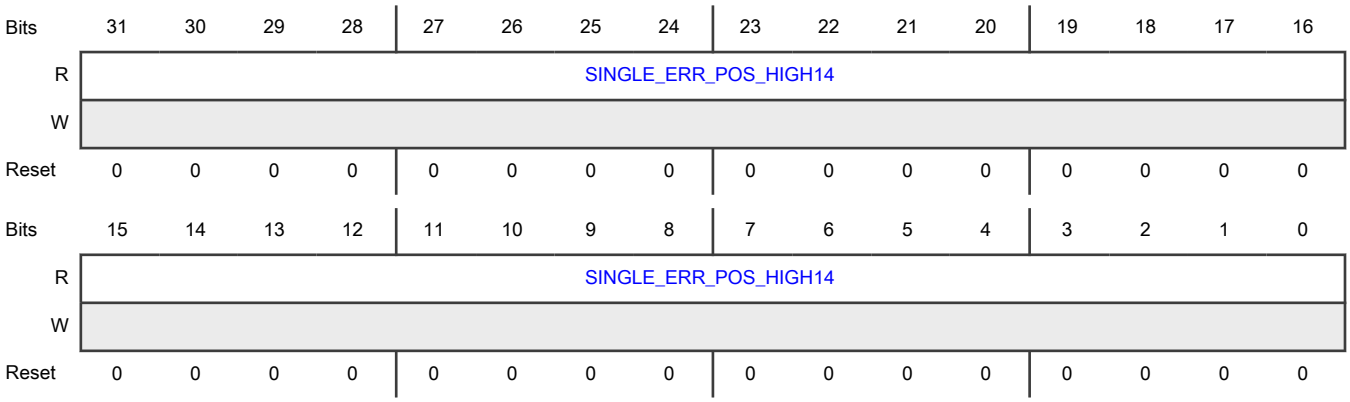
19.6.1.136 High Single Error Bit Position On Ocram Bank14 (SINGLE_ERR_POS_HIGH14)

When single bit error is corrected in ocram bank14 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH14	218h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH14	High Single Error Bit Position On Ocram Bank14 When single bit error is corrected in ocram bank14 read access, indicates which bit is corrected in high 32 bits of read data.

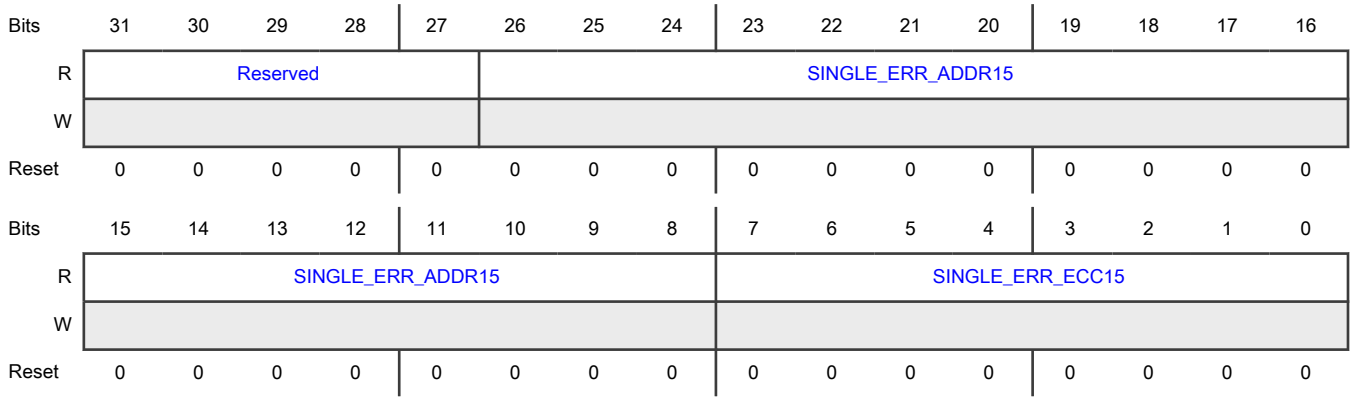
19.6.1.137 Single Error Address And ECC Code On Ocram Bank15 (SINGLE_ERR_ADDR_ECC15)

When single bit error is corrected in ocram bank15 read access, store ocram address and ecc code.

Offset

Register	Offset
SINGLE_ERR_ADDR_ECC15	21Ch

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR15	Single Error Address On Ocram Bank15 When single bit error is corrected in ocram bank15 read access, store ocram address.
7-0 SINGLE_ERR_ECC15	Single Error ECC Code On Ocram Bank15 When single bit error is corrected in ocram bank15 read access, store ecc code.

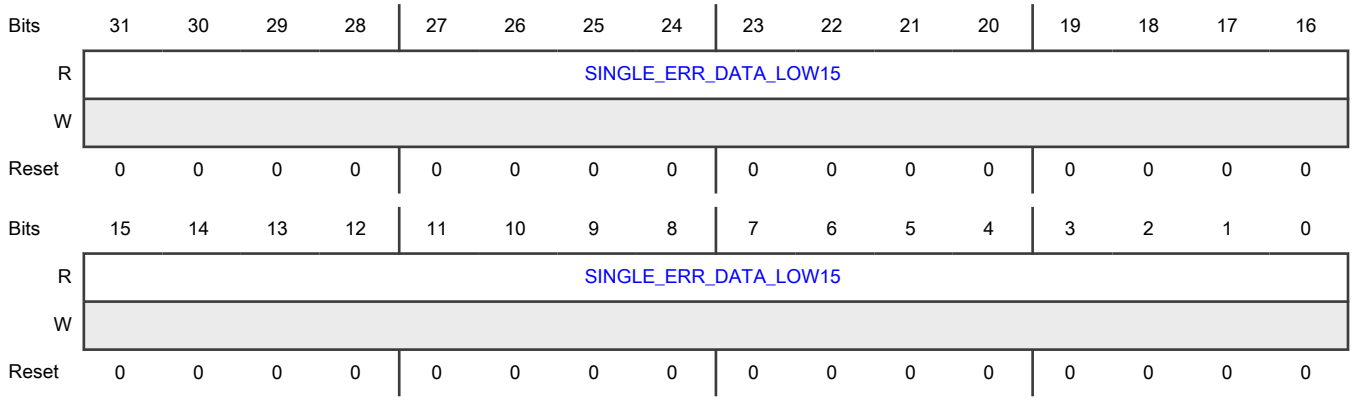
19.6.1.138 Low 32 Bits Single Error Read Data On Ocram Bank15 (SINGLE_ERR_DATA_LOW15)

When single bit error is corrected in ocram bank15 read access, store low 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_LOW15	220h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_LOW15	Low 32 Bits Single Error Read Data On Ocram Bank15 When single bit error is corrected in ocram bank15 read access, store low 32 bits uncorrected read data.

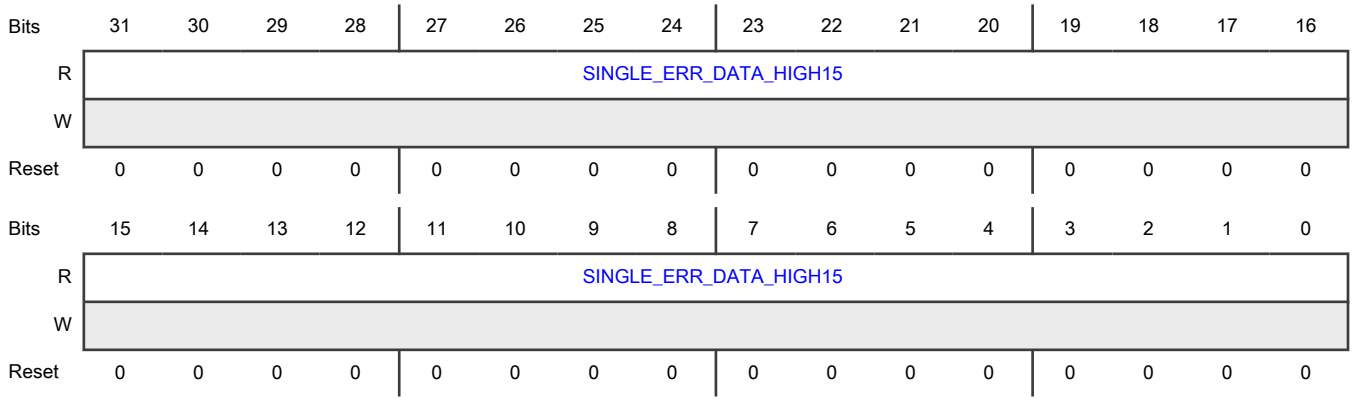
19.6.1.139 High 32 Bits Single Error Read Data On Ocram Bank15 (SINGLE_ERR_DATA_HIGH15)

When single bit error is corrected in ocram bank15 read access, store high 32 bits uncorrected read data.

Offset

Register	Offset
SINGLE_ERR_DATA_HIGH15	224h

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_DATA_HIGH15	High 32 Bits Single Error Read Data On Ocram Bank15 When single bit error is corrected in ocram bank15 read access, store high 32 bits uncorrected read data.

19.6.1.140 Low Single Error Bit Position On Ocram Bank15 (SINGLE_ERR_POS_LOW15)

When single bit error is corrected in ocram bank15 read access, indicates which bit is corrected in low 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_LOW15	228h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS_LOW15															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS_LOW15															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SINGLE_ERR_POS_LOW15	Low Single Error Bit Position On Ocram Bank15 When single bit error is corrected in ocram bank15 read access, indicates which bit is corrected in low 32 bits of read data.

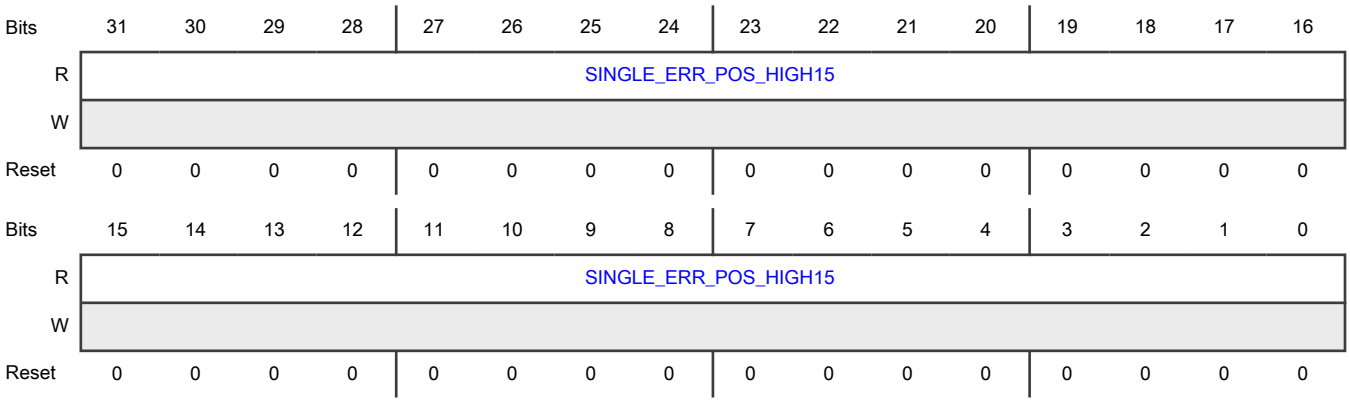
19.6.1.141 High Single Error Bit Position On Ocram Bank15 (SINGLE_ERR_POS_HIGH15)

When single bit error is corrected in ocram bank15 read access, indicates which bit is corrected in high 32 bits of read data.

Offset

Register	Offset
SINGLE_ERR_POS_HIGH15	22Ch

Diagram



Fields

Field	Description
31-0 SINGLE_ERR_POS_HIGH15	High Single Error Bit Position On Ocram Bank15 When single bit error is corrected in ocram bank15 read access, indicates which bit is corrected in high 32 bits of read data.

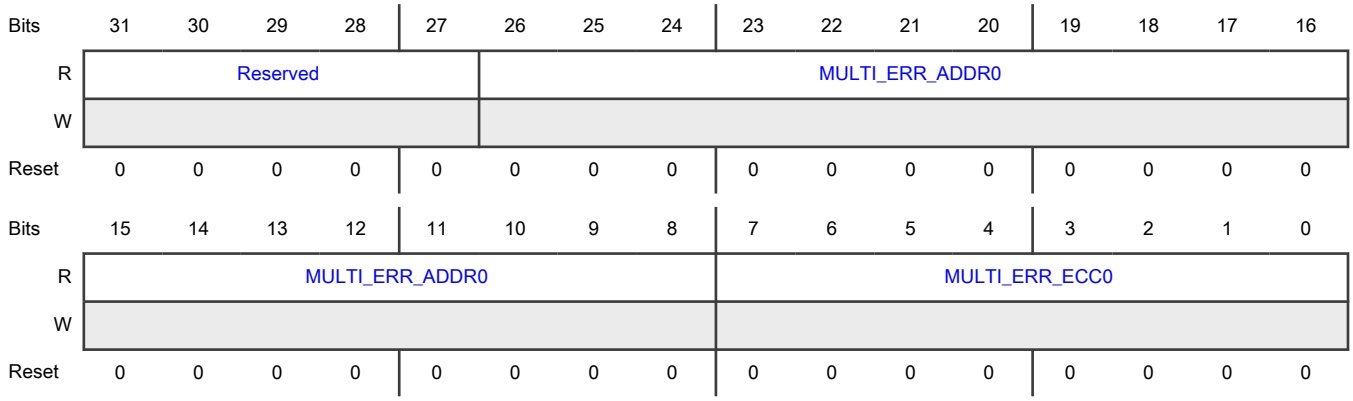
19.6.1.142 Multiple Error Address And ECC Code On Ocram Bank0 (MULTI_ERR_ADDR_ECC0)

When multiple bits error are detected in ocram bank0 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC0	230h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR0	Multiple Error Address On Ocram Bank0 When multiple bits error are detected in ocram bank0 read access, store ocram address.
7-0 MULTI_ERR_ECC0	Multiple Error ECC Code On Ocram Bank0 When multiple bits error are detected in ocram bank0 read access, store ecc code.

19.6.1.143 Low 32 Bits Multiple Error Read Data On Ocram Bank0 (MULTI_ERR_DATA_LOW0)

When multiple bits error are detected in ocram bank0 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW0	234h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MULTI_ERR_DATA_LOW0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_ERR_DATA_LOW0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW0	Low 32 Bits Multiple Error Read Data On Ocram Bank0 When multiple bits error are detected in ocram bank0 read access, store low 32 bits read data.

19.6.1.144 High 32 Bits Multiple Error Read Data On Ocram Bank0 (MULTI_ERR_DATA_HIGH0)

When multiple bits error are detected in ocram bank0 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH0	238h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MULTI_ERR_DATA_HIGH0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_ERR_DATA_HIGH0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH0	High 32 Bits Multiple Error Read Data On Ocram Bank0 When multiple bits error are detected in ocram bank0 read access, store high 32 bits read data.

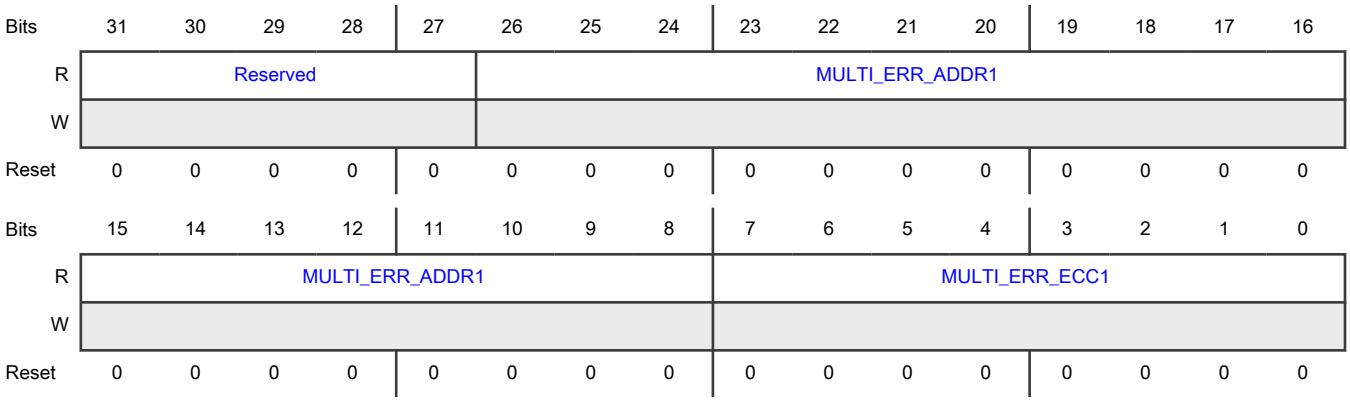
19.6.1.145 Multiple Error Address And ECC Code On Ocram Bank1 (MULTI_ERR_ADDR_ECC1)

When multiple bits error are detected in ocram bank1 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC1	23Ch

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR1	Multiple Error Address On Ocram Bank1 When multiple bits error are detected in ocram bank1 read access, store ocram address.
7-0 MULTI_ERR_ECC1	Multiple Error ECC Code On Ocram Bank1 When multiple bits error are detected in ocram bank1 read access, store ecc code.

19.6.1.146 Low 32 Bits Multiple Error Read Data On Ocram Bank1 (MULTI_ERR_DATA_LOW1)

When multiple bits error are detected in ocram bank1 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW1	240h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MULTI_ERR_DATA_LOW1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_ERR_DATA_LOW1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW1	Low 32 Bits Multiple Error Read Data On Ocram Bank1 When multiple bits error are detected in ocram bank1 read access, store low 32 bits read data.

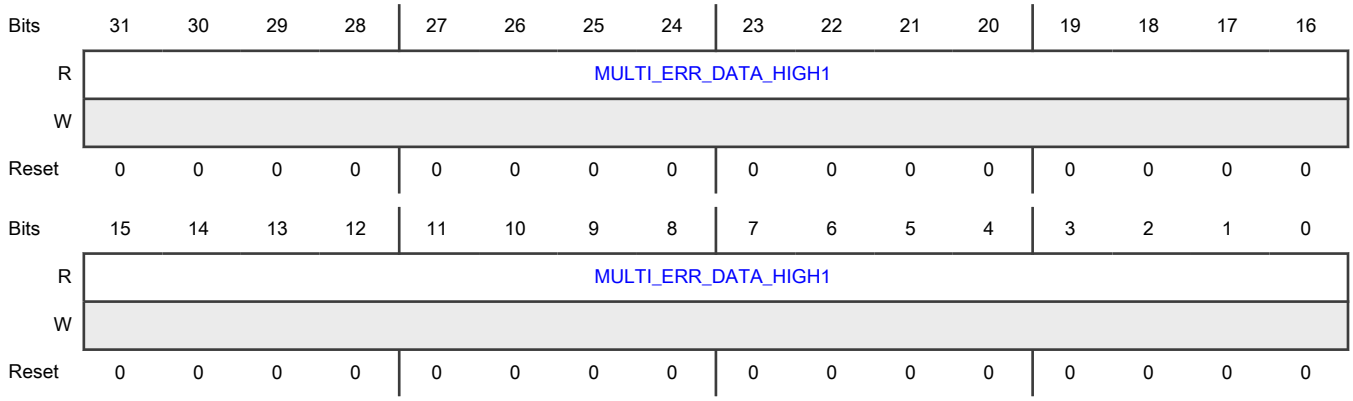
19.6.1.147 High 32 Bits Multiple Error Read Data On Ocram Bank1 (MULTI_ERR_DATA_HIGH1)

When multiple bits error are detected in ocram bank1 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH1	244h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH1	High 32 Bits Multiple Error Read Data On Ocram Bank1 When multiple bits error are detected in ocram bank1 read access, store high 32 bits read data.

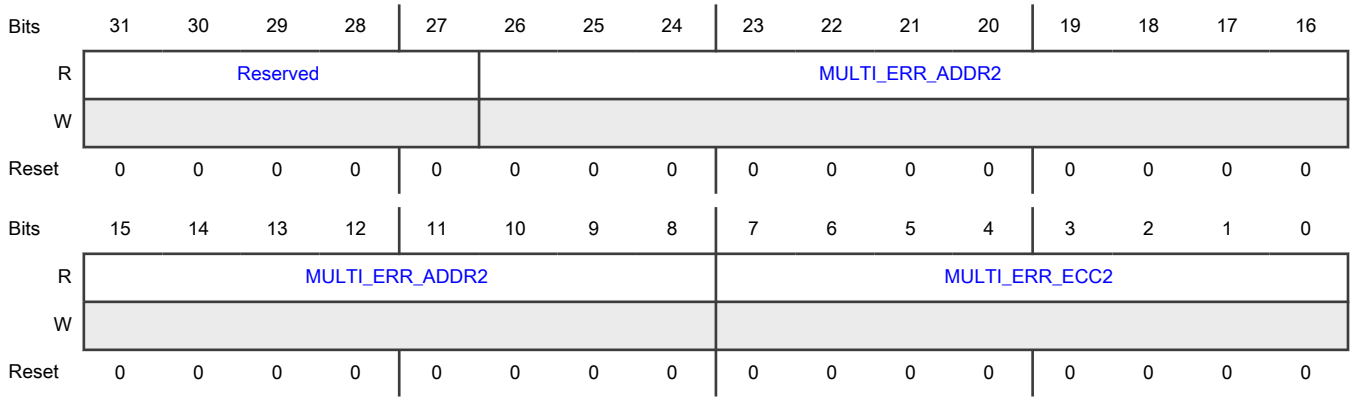
19.6.1.148 Multiple Error Address And ECC Code On Ocram Bank2 (MULTI_ERR_ADDR_ECC2)

When multiple bits error are detected in ocram bank2 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC2	248h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR2	Multiple Error Address On Ocram Bank2 When multiple bits error are detected in ocram bank2 read access, store ocram address.
7-0 MULTI_ERR_ECC2	Multiple Error ECC Code On Ocram Bank2 When multiple bits error are detected in ocram bank2 read access, store ecc code.

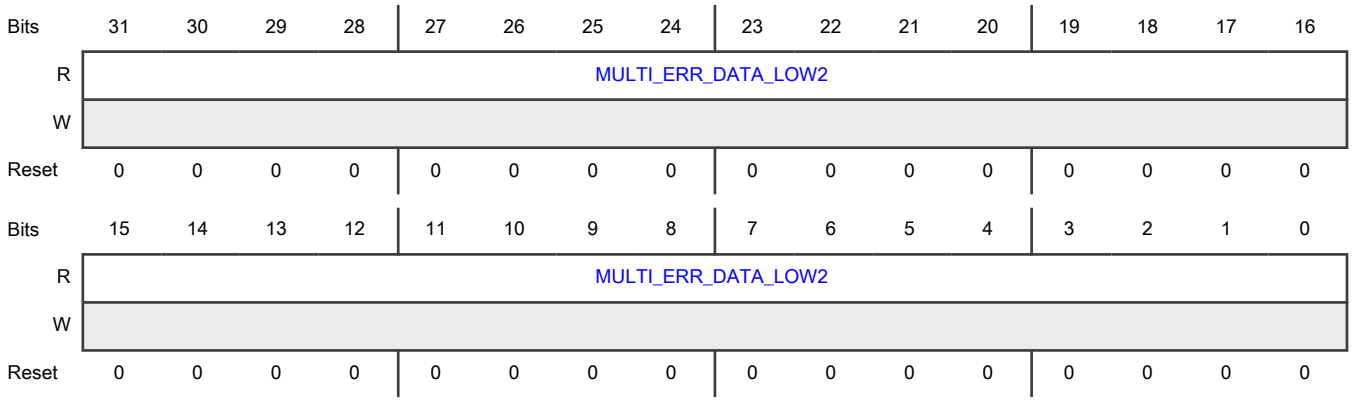
19.6.1.149 Low 32 Bits Multiple Error Read Data On Ocram Bank2 (MULTI_ERR_DATA_LOW2)

When multiple bits error are detected in ocram bank2 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW2	24Ch

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW2	Low 32 Bits Multiple Error Read Data On Ocram Bank2 When multiple bits error are detected in ocram bank2 read access, store low 32 bits read data.

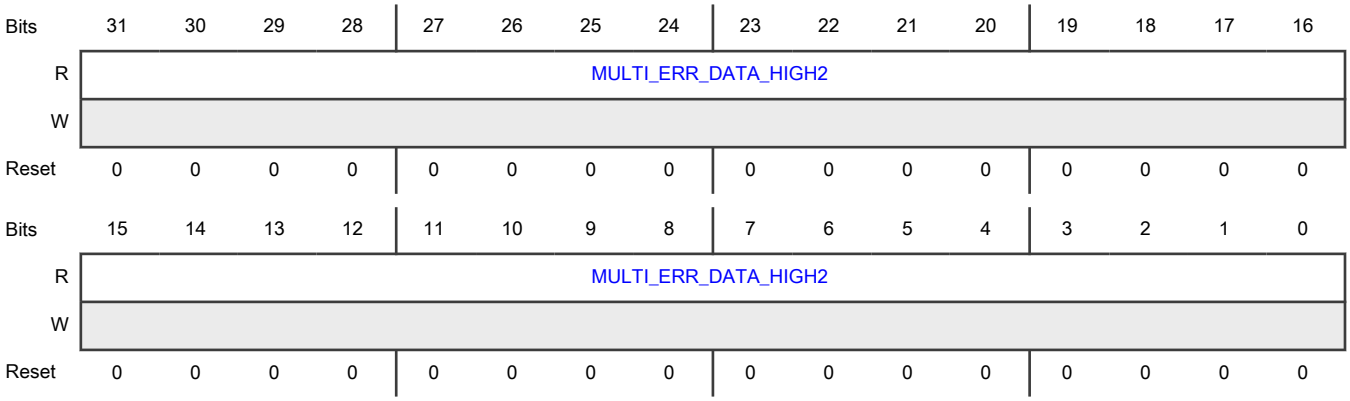
19.6.1.150 High 32 Bits Multiple Error Read Data On Ocram Bank2 (MULTI_ERR_DATA_HIGH2)

When multiple bits error are detected in ocram bank2 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH2	250h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH2	High 32 Bits Multiple Error Read Data On Ocram Bank2 When multiple bits error are detected in ocram bank2 read access, store high 32 bits read data.

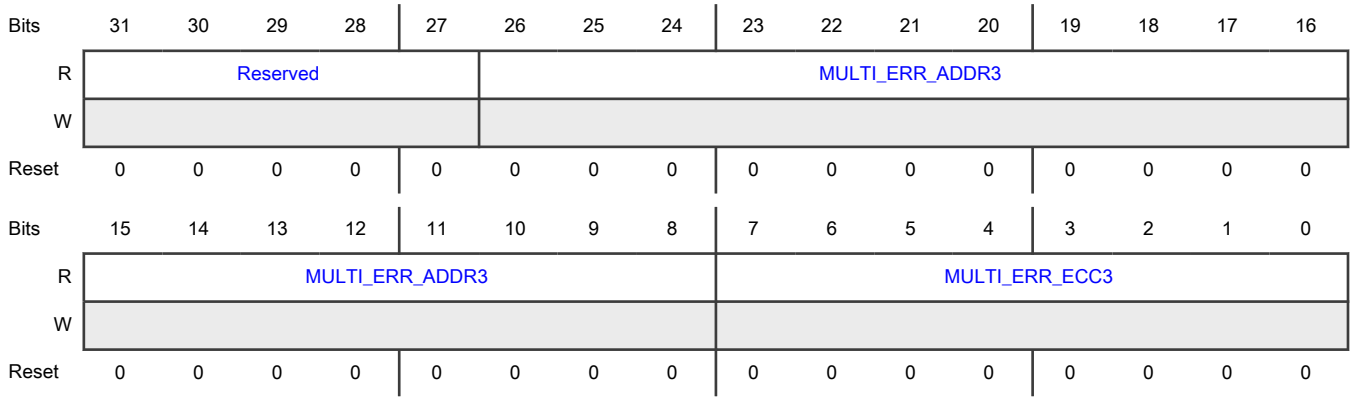
19.6.1.151 Multiple Error Address And ECC Code On Ocram Bank3 (MULTI_ERR_ADDR_ECC3)

When multiple bits error are detected in ocram bank3 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC3	254h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR3	Multiple Error Address On Ocram Bank3 When multiple bits error are detected in ocram bank3 read access, store ocram address.
7-0 MULTI_ERR_ECC3	Multiple Error ECC Code On Ocram Bank3 When multiple bits error are detected in ocram bank3 read access, store ecc code.

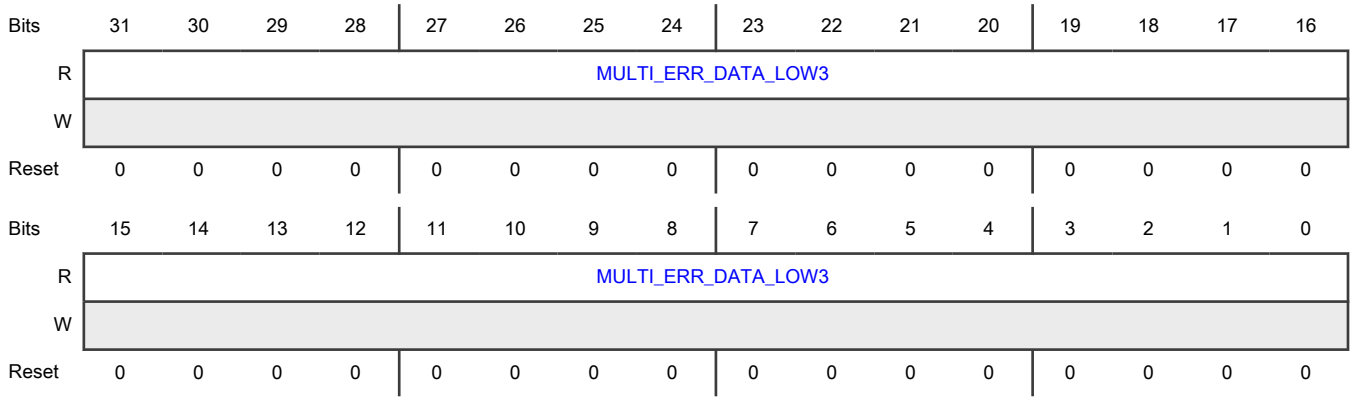
19.6.1.152 Low 32 Bits Multiple Error Read Data On Ocram Bank3 (MULTI_ERR_DATA_LOW3)

When multiple bits error are detected in ocram bank3 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW3	258h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW3	Low 32 Bits Multiple Error Read Data On Ocram Bank3 When multiple bits error are detected in ocram bank3 read access, store low 32 bits read data.

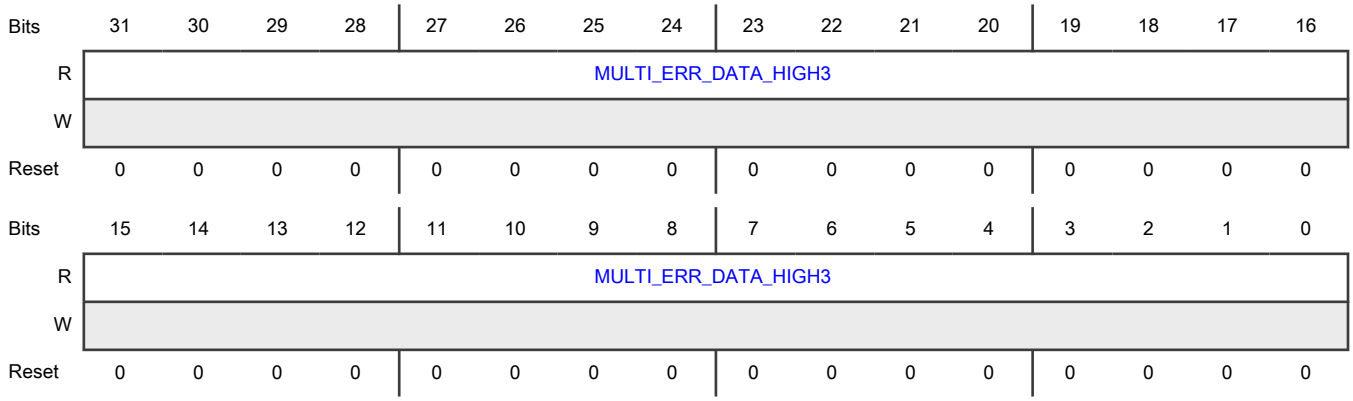
19.6.1.153 High 32 Bits Multiple Error Read Data On Ocram Bank3 (MULTI_ERR_DATA_HIGH3)

When multiple bits error are detected in ocram bank3 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH3	25Ch

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH3	High 32 Bits Multiple Error Read Data On Ocram Bank3 When multiple bits error are detected in ocram bank3 read access, store high 32 bits read data.

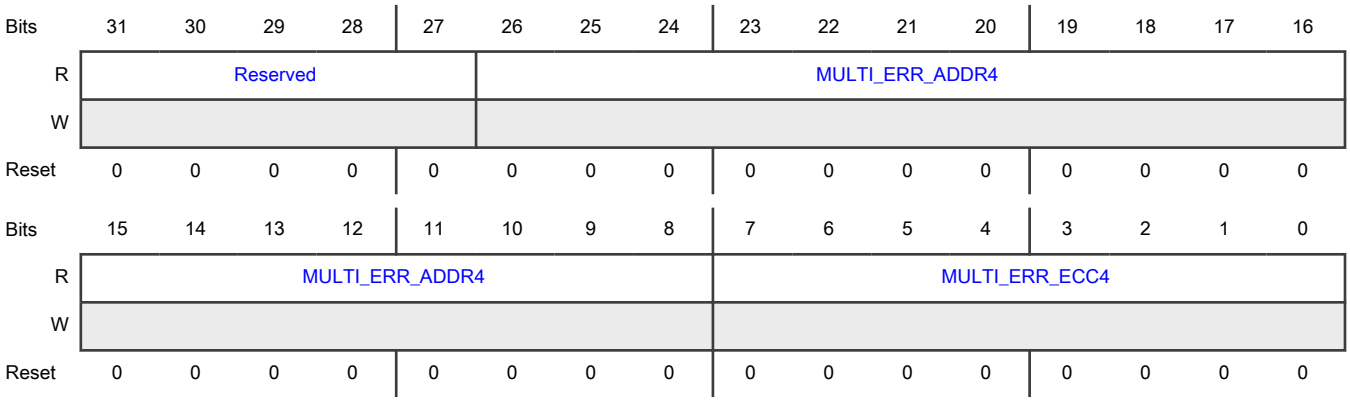
19.6.1.154 Multiple Error Address And ECC Code On Ocram Bank4 (MULTI_ERR_ADDR_ECC4)

When multiple bits error are detected in ocram bank4 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC4	260h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR4	Multiple Error Address On Ocram Bank4 When multiple bits error are detected in ocram bank4 read access, store ocram address.
7-0 MULTI_ERR_ECC4	Multiple Error ECC Code On Ocram Bank4 When multiple bits error are detected in ocram bank4 read access, store ecc code.

19.6.1.155 Low 32 Bits Multiple Error Read Data On Ocram Bank4 (MULTI_ERR_DATA_LOW4)

When multiple bits error are detected in ocram bank4 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW4	264h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MULTI_ERR_DATA_LOW4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_ERR_DATA_LOW4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW4	Low 32 Bits Multiple Error Read Data On Ocram Bank4 When multiple bits error are detected in ocram bank4 read access, store low 32 bits read data.

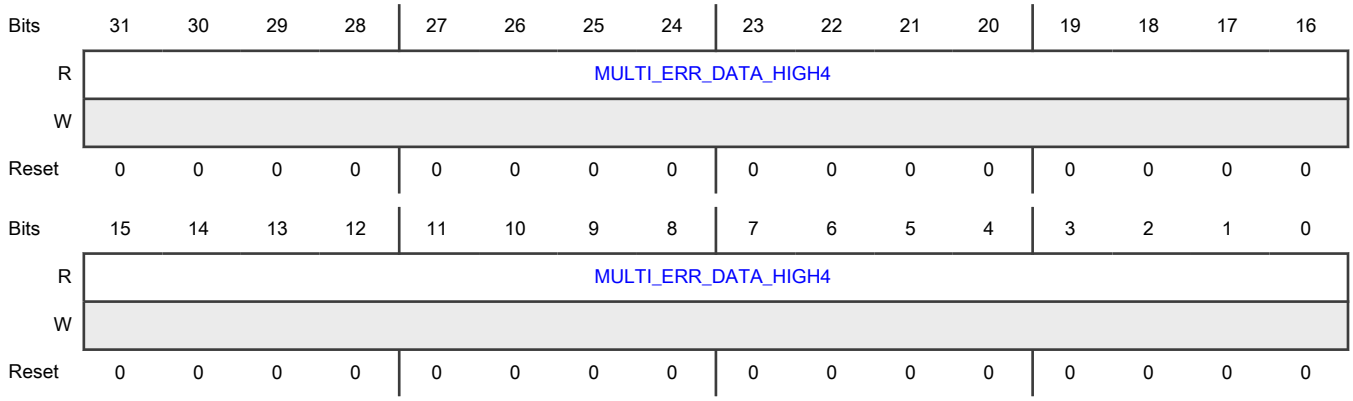
19.6.1.156 High 32 Bits Multiple Error Read Data On Ocram Bank4 (MULTI_ERR_DATA_HIGH4)

When multiple bits error are detected in ocram bank4 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH4	268h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH4	High 32 Bits Multiple Error Read Data On Ocram Bank4 When multiple bits error are detected in ocram bank4 read access, store high 32 bits read data.

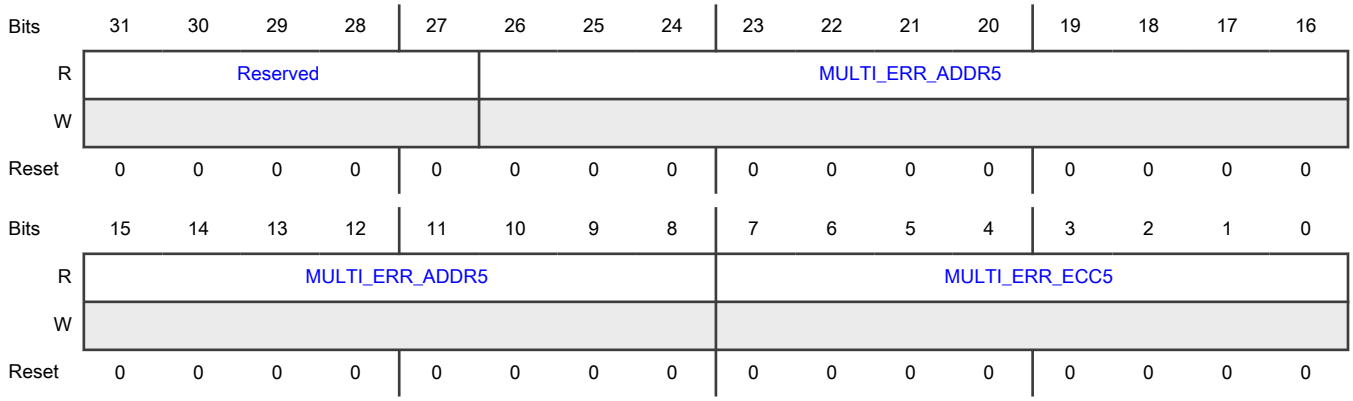
19.6.1.157 Multiple Error Address And ECC Code On Ocram Bank5 (MULTI_ERR_ADDR_ECC5)

When multiple bits error are detected in ocram bank5 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC5	26Ch

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR5	Multiple Error Address On Ocram Bank5 When multiple bits error are detected in ocram bank5 read access, store ocram address.
7-0 MULTI_ERR_ECC5	Multiple Error ECC Code On Ocram Bank5 When multiple bits error are detected in ocram bank5 read access, store ecc code.

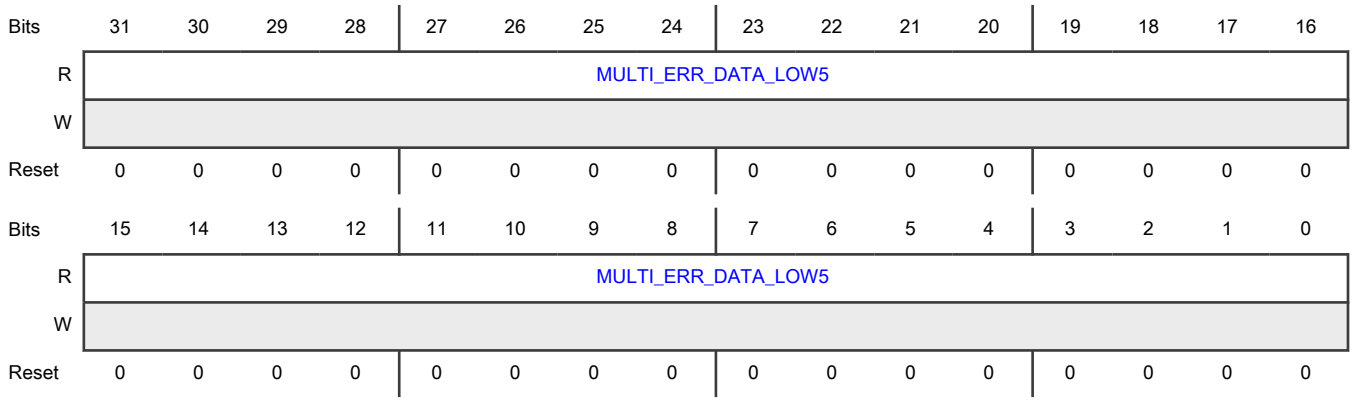
19.6.1.158 Low 32 Bits Multiple Error Read Data On Ocram Bank5 (MULTI_ERR_DATA_LOW5)

When multiple bits error are detected in ocram bank5 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW5	270h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW5	Low 32 Bits Multiple Error Read Data On Ocram Bank5 When multiple bits error are detected in ocram bank5 read access, store low 32 bits read data.

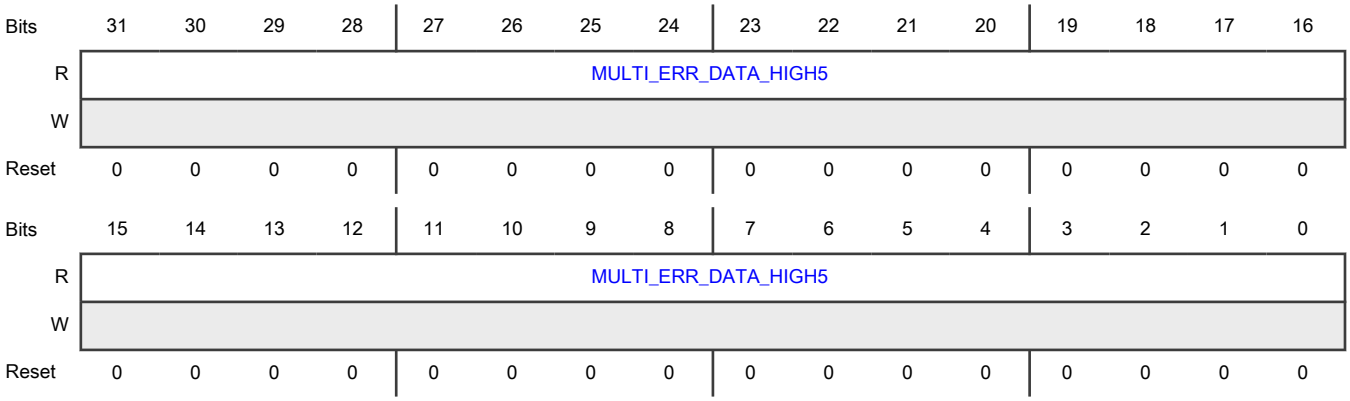
19.6.1.159 High 32 Bits Multiple Error Read Data On Ocram Bank5 (MULTI_ERR_DATA_HIGH5)

When multiple bits error are detected in ocram bank5 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH5	274h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH5	High 32 Bits Multiple Error Read Data On Ocram Bank5 When multiple bits error are detected in ocram bank5 read access, store high 32 bits read data.

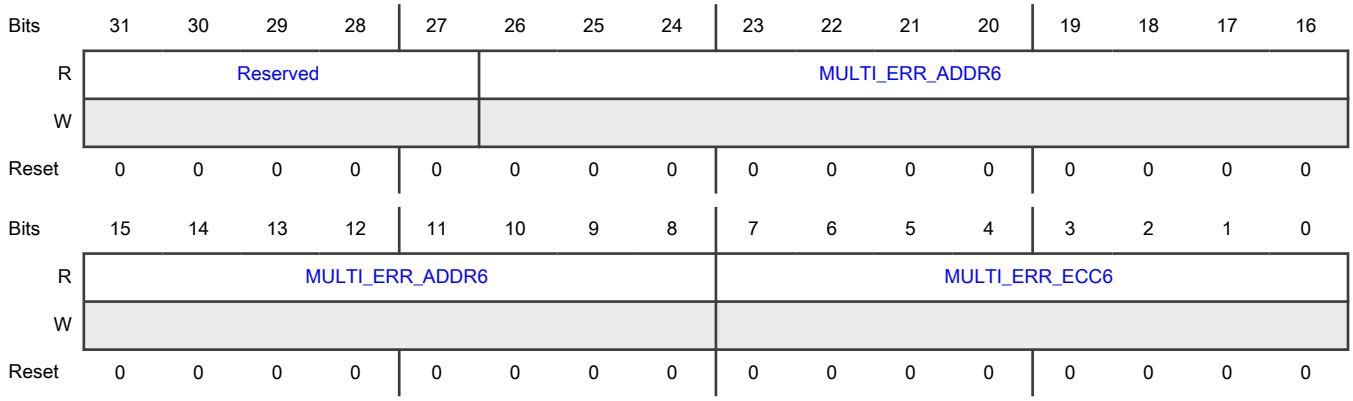
19.6.1.160 Multiple Error Address And ECC Code On Ocram Bank6 (MULTI_ERR_ADDR_ECC6)

When multiple bits error are detected in ocram bank6 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC6	278h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR6	Multiple Error Address On Ocram Bank6 When multiple bits error are detected in ocram bank6 read access, store ocram address.
7-0 MULTI_ERR_ECC6	Multiple Error ECC Code On Ocram Bank6 When multiple bits error are detected in ocram bank6 read access, store ecc code.

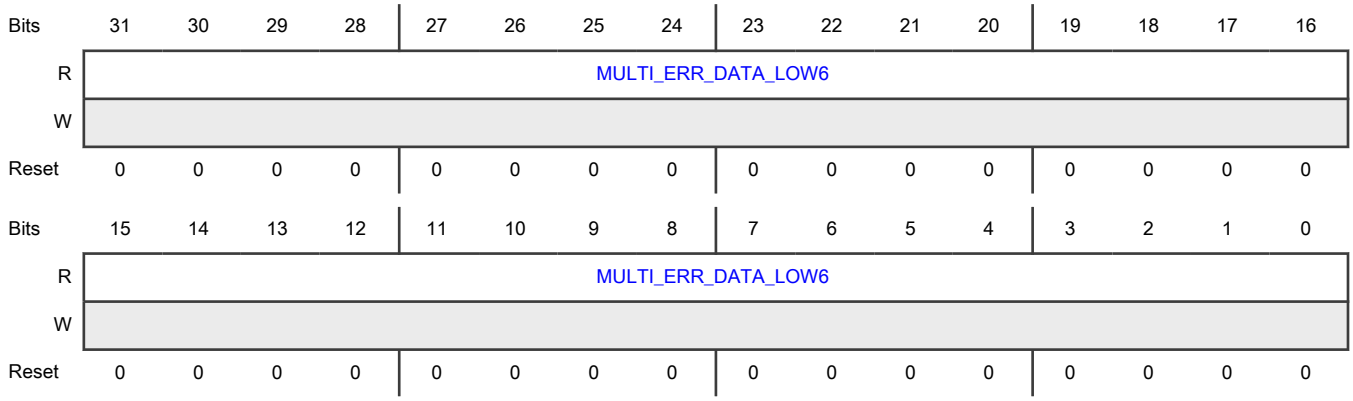
19.6.1.161 Low 32 Bits Multiple Error Read Data On Ocram Bank6 (MULTI_ERR_DATA_LOW6)

When multiple bits error are detected in ocram bank6 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW6	27Ch

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW6	Low 32 Bits Multiple Error Read Data On Ocram Bank6 When multiple bits error are detected in ocram bank6 read access, store low 32 bits read data.

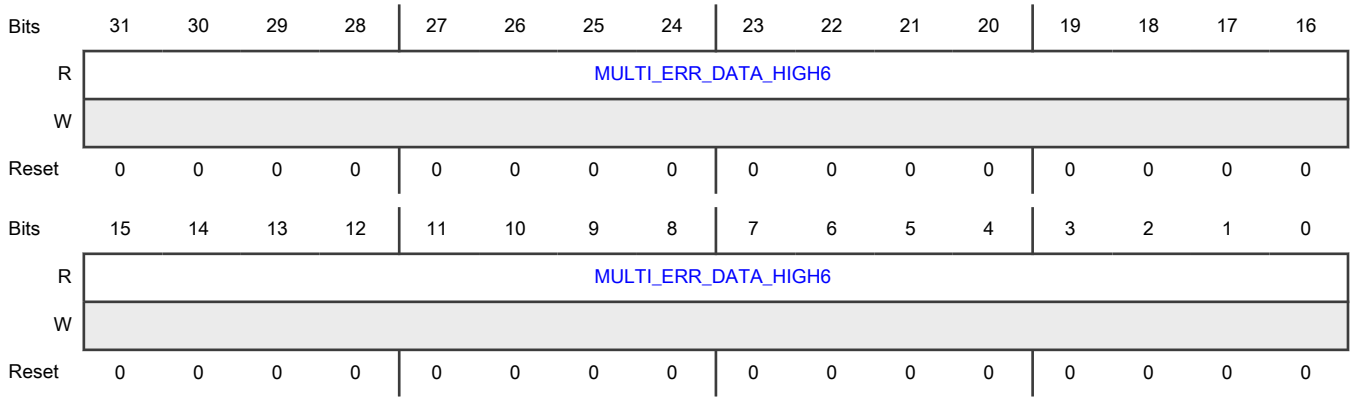
19.6.1.162 High 32 Bits Multiple Error Read Data On Ocram Bank6 (MULTI_ERR_DATA_HIGH6)

When multiple bits error are detected in ocram bank6 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH6	280h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH6	High 32 Bits Multiple Error Read Data On Ocram Bank6 When multiple bits error are detected in ocram bank6 read access, store high 32 bits read data.

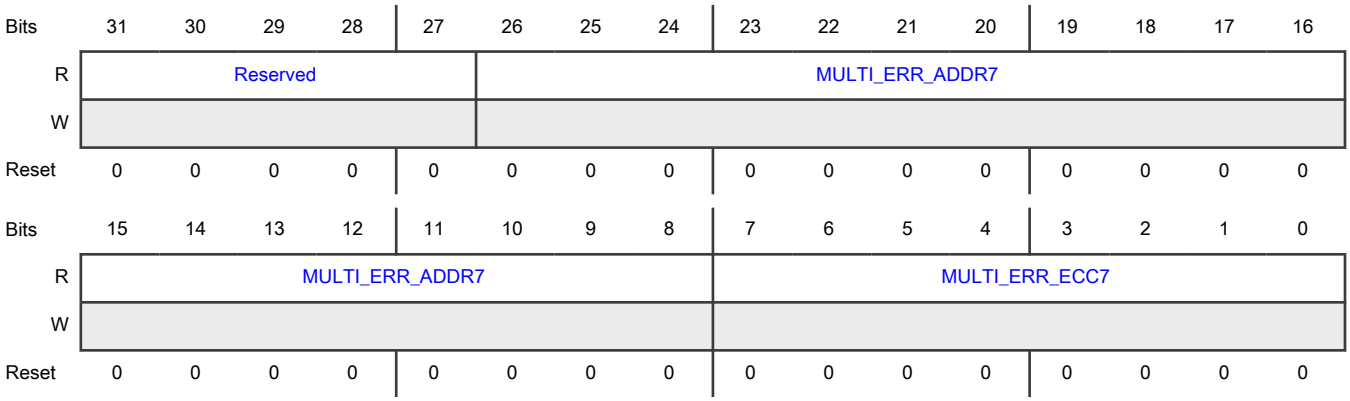
19.6.1.163 Multiple Error Address And ECC Code On Ocram Bank7 (MULTI_ERR_ADDR_ECC7)

When multiple bits error are detected in ocram bank7 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC7	284h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR7	Multiple Error Address On Ocram Bank7 When multiple bits error are detected in ocram bank7 read access, store ocram address.
7-0 MULTI_ERR_ECC7	Multiple Error ECC Code On Ocram Bank7 When multiple bits error are detected in ocram bank7 read access, store ecc code.

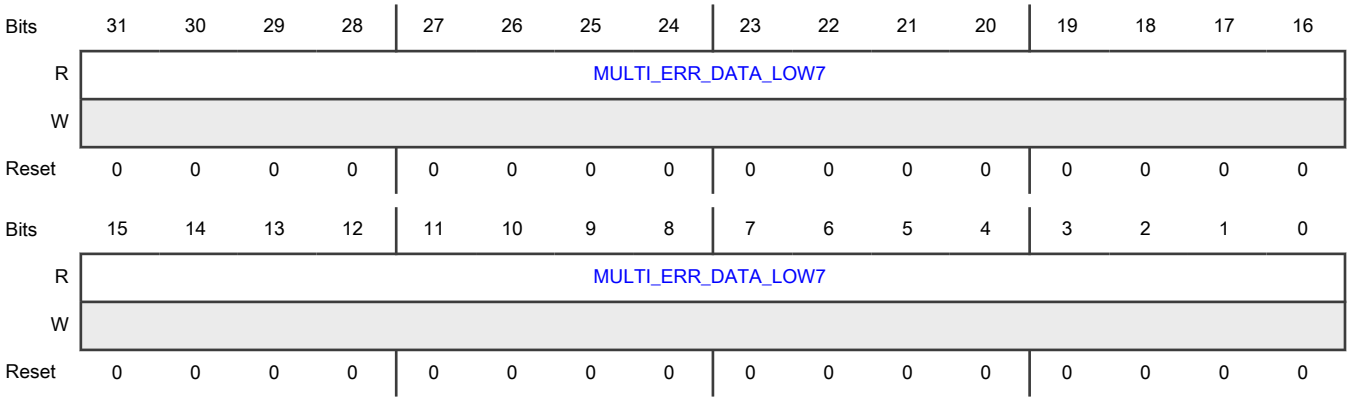
19.6.1.164 Low 32 Bits Multiple Error Read Data On Ocram Bank7 (MULTI_ERR_DATA_LOW7)

When multiple bits error are detected in ocram bank7 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW7	288h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW7	Low 32 Bits Multiple Error Read Data On Ocram Bank7 When multiple bits error are detected in ocram bank7 read access, store low 32 bits read data.

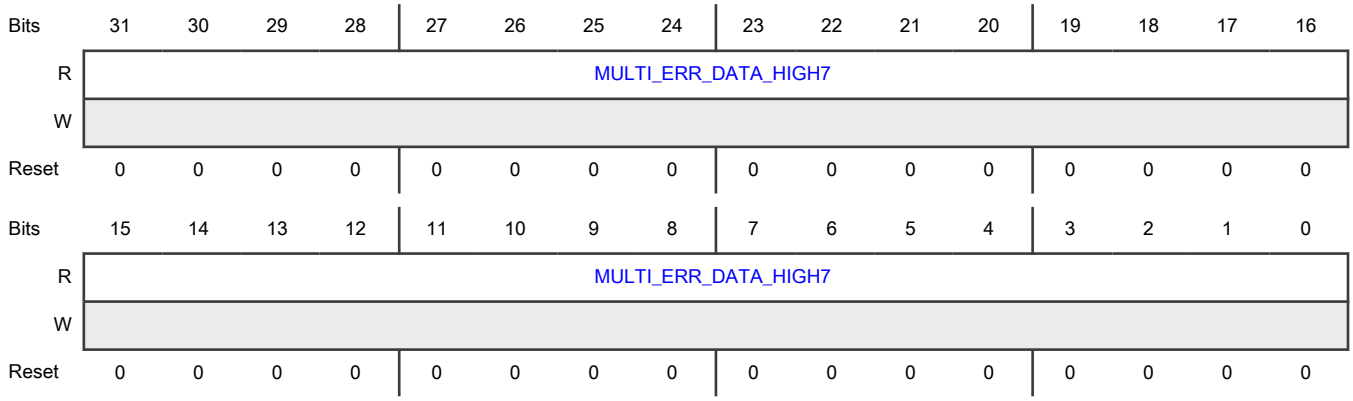
19.6.1.165 High 32 Bits Multiple Error Read Data On Ocram Bank7 (MULTI_ERR_DATA_HIGH7)

When multiple bits error are detected in ocram bank7 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH7	28Ch

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH7	High 32 Bits Multiple Error Read Data On Ocram Bank7 When multiple bits error are detected in ocram bank7 read access, store high 32 bits read data.

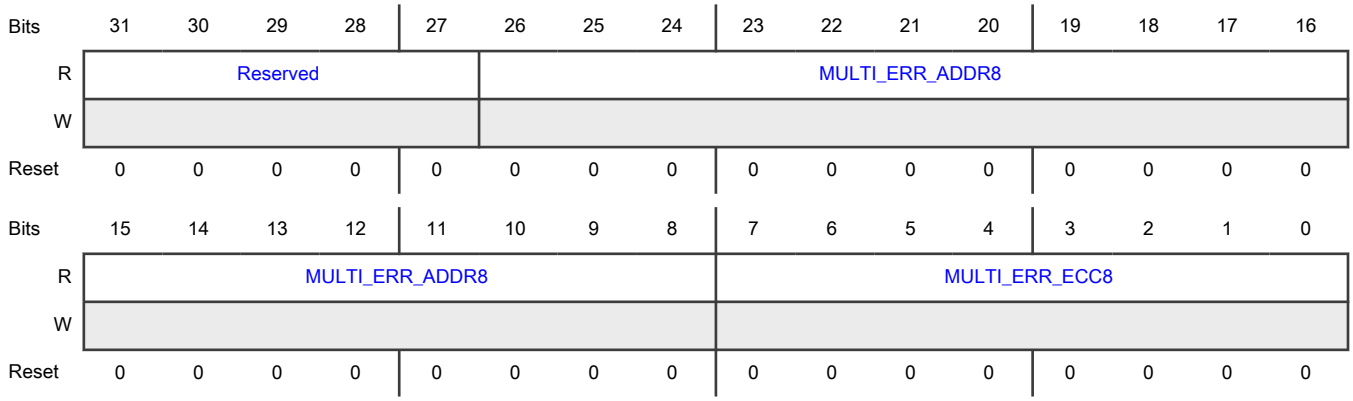
19.6.1.166 Multiple Error Address And ECC Code On Ocram Bank8 (MULTI_ERR_ADDR_ECC8)

When multiple bits error are detected in ocram bank8 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC8	290h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR8	Multiple Error Address On Ocram Bank8 When multiple bits error are detected in ocram bank8 read access, store ocram address.
7-0 MULTI_ERR_ECC8	Multiple Error ECC Code On Ocram Bank8 When multiple bits error are detected in ocram bank8 read access, store ecc code.

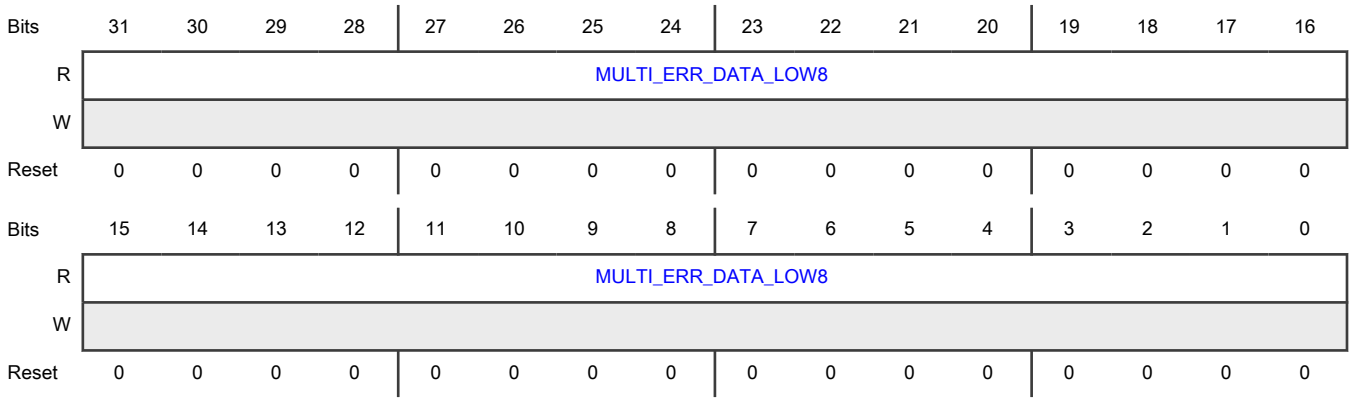
19.6.1.167 Low 32 Bits Multiple Error Read Data On Ocram Bank8 (MULTI_ERR_DATA_LOW8)

When multiple bits error are detected in ocram bank8 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW8	294h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW8	Low 32 Bits Multiple Error Read Data On Ocram Bank8 When multiple bits error are detected in ocram bank8 read access, store low 32 bits read data.

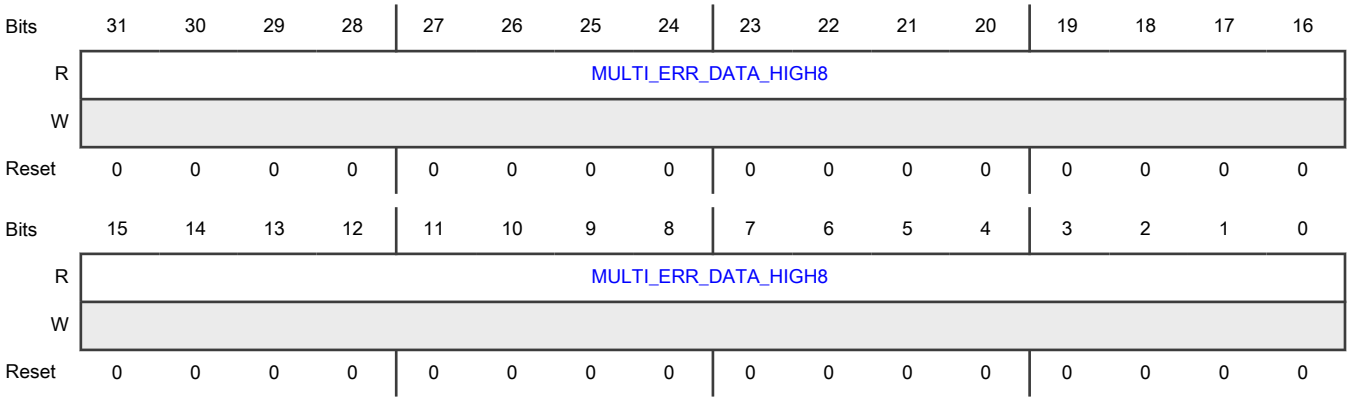
19.6.1.168 High 32 Bits Multiple Error Read Data On Ocram Bank8 (MULTI_ERR_DATA_HIGH8)

When multiple bits error are detected in ocram bank8 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH8	298h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH8	High 32 Bits Multiple Error Read Data On Ocram Bank8 When multiple bits error are detected in ocram bank8 read access, store high 32 bits read data.

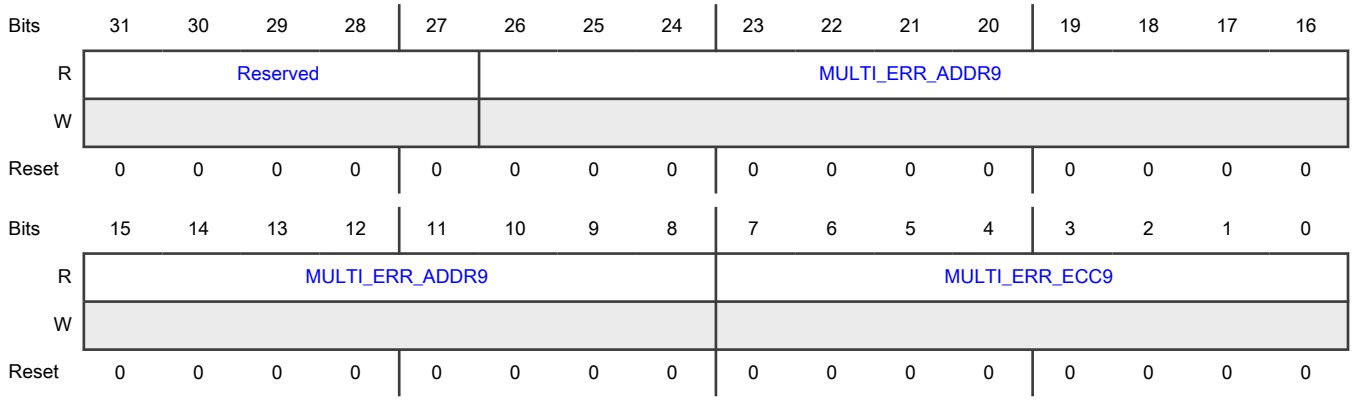
19.6.1.169 Multiple Error Address And ECC Code On Ocram Bank9 (MULTI_ERR_ADDR_ECC9)

When multiple bits error are detected in ocram bank9 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC9	29Ch

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR9	Multiple Error Address On Ocram Bank9 When multiple bits error are detected in ocram bank9 read access, store ocram address.
7-0 MULTI_ERR_ECC9	Multiple Error ECC Code On Ocram Bank9 When multiple bits error are detected in ocram bank9 read access, store ecc code.

19.6.1.170 Low 32 Bits Multiple Error Read Data On Ocram Bank9 (MULTI_ERR_DATA_LOW9)

When multiple bits error are detected in ocram bank9 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW9	2A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MULTI_ERR_DATA_LOW9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_ERR_DATA_LOW9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW9	Low 32 Bits Multiple Error Read Data On Ocram Bank9 When multiple bits error are detected in ocram bank9 read access, store low 32 bits read data.

19.6.1.171 High 32 Bits Multiple Error Read Data On Ocram Bank9 (MULTI_ERR_DATA_HIGH9)

When multiple bits error are detected in ocram bank9 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH9	2A4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MULTI_ERR_DATA_HIGH9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_ERR_DATA_HIGH9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH9	High 32 Bits Multiple Error Read Data On Ocram Bank9 When multiple bits error are detected in ocram bank9 read access, store high 32 bits read data.

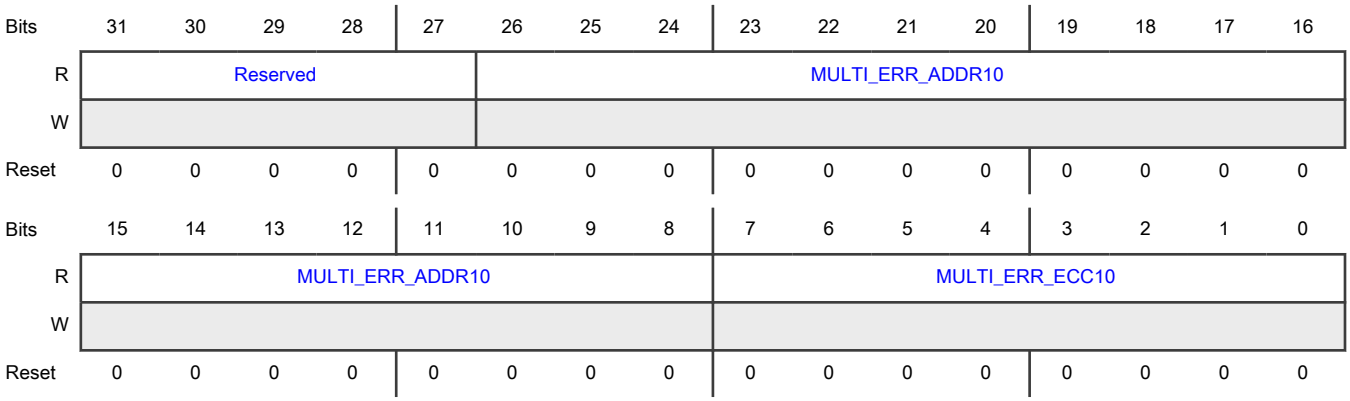
19.6.1.172 Multiple Error Address And ECC Code On Ocram Bank10 (MULTI_ERR_ADDR_ECC10)

When multiple bits error are detected in ocram bank10 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC10	2A8h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR10	Multiple Error Address On Ocram Bank10 When multiple bits error are detected in ocram bank10 read access, store ocram address.
7-0 MULTI_ERR_ECC10	Multiple Error ECC Code On Ocram Bank10 When multiple bits error are detected in ocram bank10 read access, store ecc code.

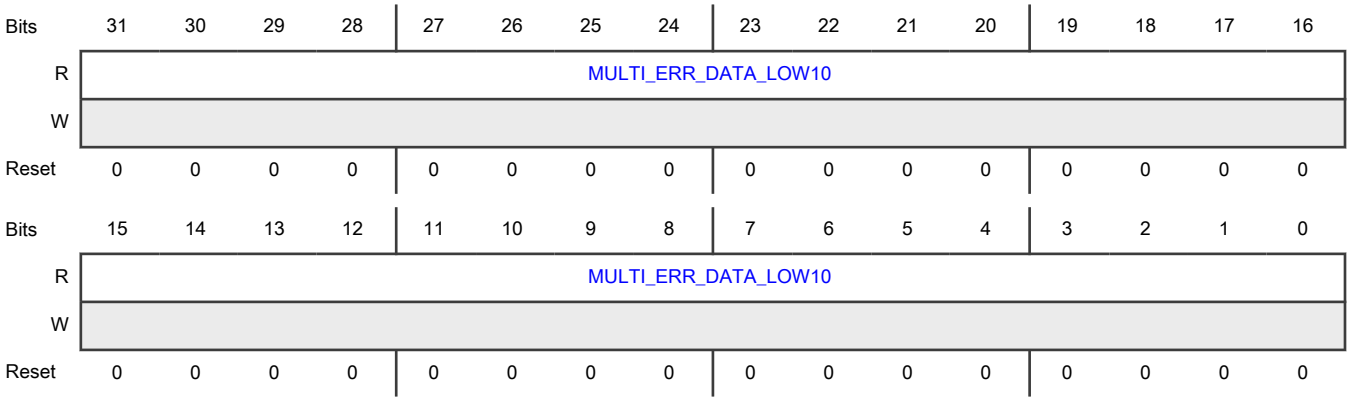
19.6.1.173 Low 32 Bits Multiple Error Read Data On Ocram Bank10 (MULTI_ERR_DATA_LOW10)

When multiple bits error are detected in ocram bank10 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW10	2ACh

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW10	Low 32 Bits Multiple Error Read Data On Ocram Bank10 When multiple bits error are detected in ocram bank10 read access, store low 32 bits read data.

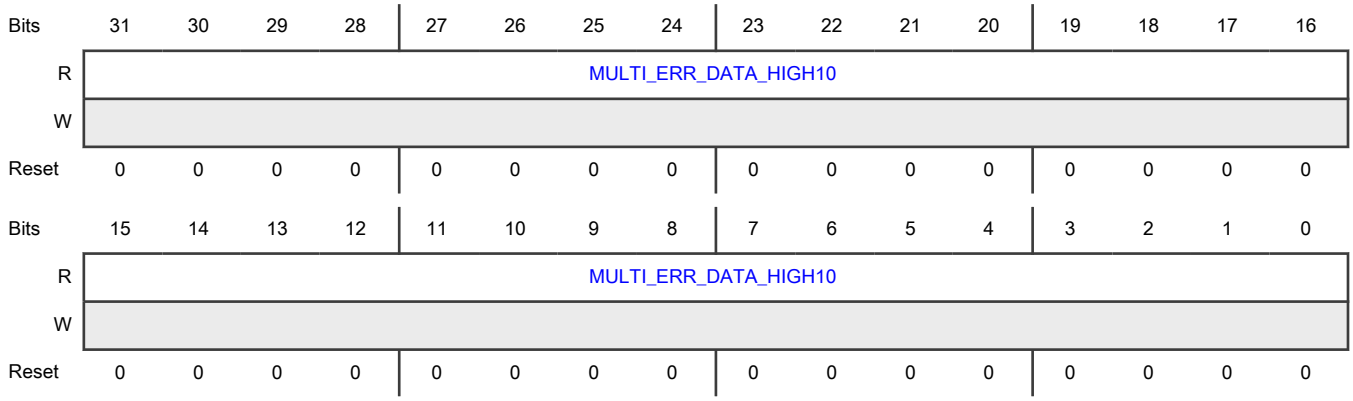
19.6.1.174 High 32 Bits Multiple Error Read Data On Ocram Bank10 (MULTI_ERR_DATA_HIGH10)

When multiple bits error are detected in ocram bank10 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH10	2B0h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH10	High 32 Bits Multiple Error Read Data On Ocram Bank10 When multiple bits error are detected in ocram bank10 read access, store high 32 bits read data.

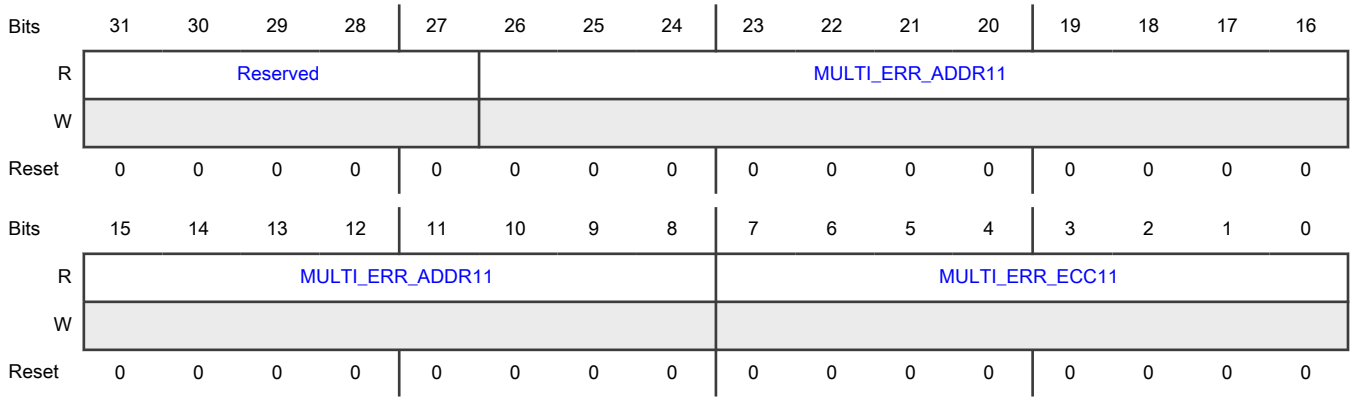
19.6.1.175 Multiple Error Address And ECC Code On Ocram Bank11 (MULTI_ERR_ADDR_ECC11)

When multiple bits error are detected in ocram bank11 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC11	2B4h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR11	Multiple Error Address On Ocram Bank11 When multiple bits error are detected in ocram bank11 read access, store ocram address.
7-0 MULTI_ERR_ECC11	Multiple Error ECC Code On Ocram Bank11 When multiple bits error are detected in ocram bank11 read access, store ecc code.

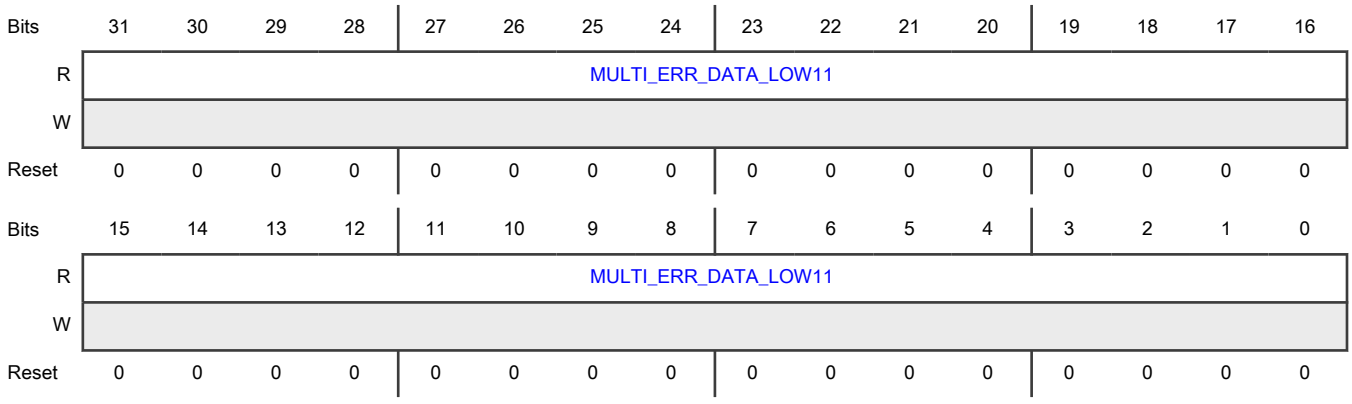
19.6.1.176 Low 32 Bits Multiple Error Read Data On Ocram Bank11 (MULTI_ERR_DATA_LOW11)

When multiple bits error are detected in ocram bank11 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW11	2B8h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW11	Low 32 Bits Multiple Error Read Data On Ocram Bank11 When multiple bits error are detected in ocram bank11 read access, store low 32 bits read data.

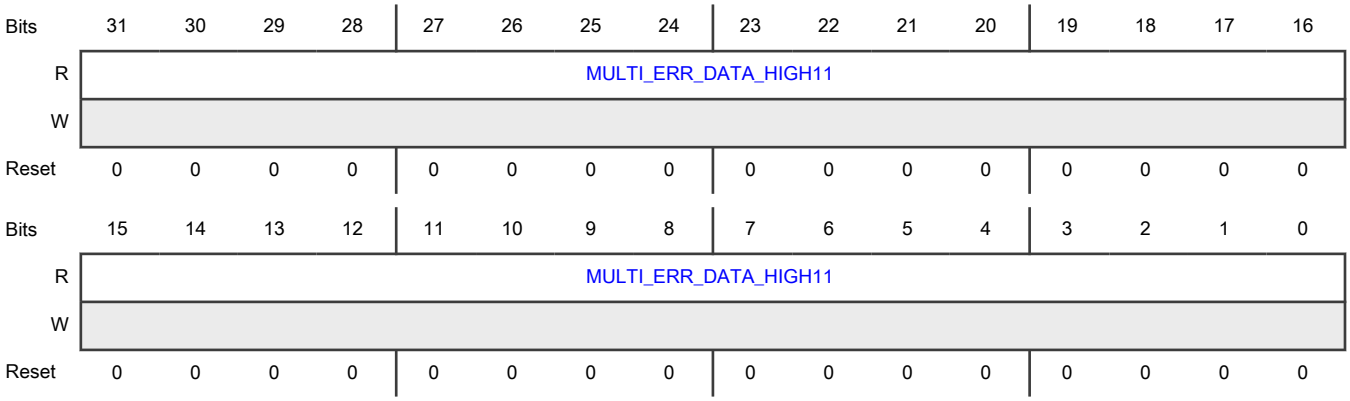
19.6.1.177 High 32 Bits Multiple Error Read Data On Ocram Bank11 (MULTI_ERR_DATA_HIGH11)

When multiple bits error are detected in ocram bank11 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH11	2BCh

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH11	High 32 Bits Multiple Error Read Data On Ocram Bank11 When multiple bits error are detected in ocram bank11 read access, store high 32 bits read data.

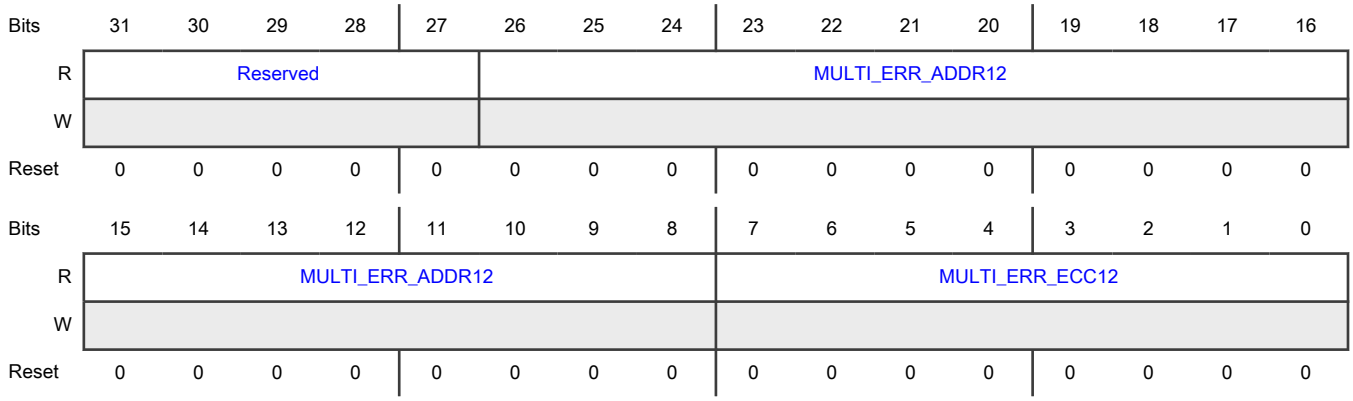
19.6.1.178 Multiple Error Address And ECC Code On Ocram Bank12 (MULTI_ERR_ADDR_ECC12)

When multiple bits error are detected in ocram bank12 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC12	2C0h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR12	Multiple Error Address On Ocram Bank12 When multiple bits error are detected in ocram bank12 read access, store ocram address.
7-0 MULTI_ERR_ECC12	Multiple Error ECC Code On Ocram Bank12 When multiple bits error are detected in ocram bank12 read access, store ecc code.

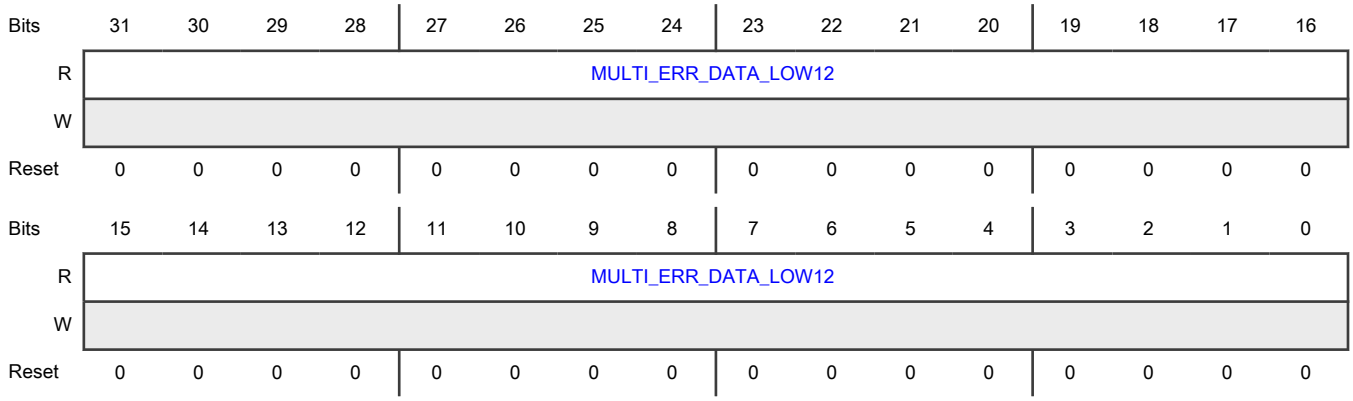
19.6.1.179 Low 32 Bits Multiple Error Read Data On Ocram Bank12 (MULTI_ERR_DATA_LOW12)

When multiple bits error are detected in ocram bank12 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW12	2C4h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW12	Low 32 Bits Multiple Error Read Data On Ocram Bank12 When multiple bits error are detected in ocram bank12 read access, store low 32 bits read data.

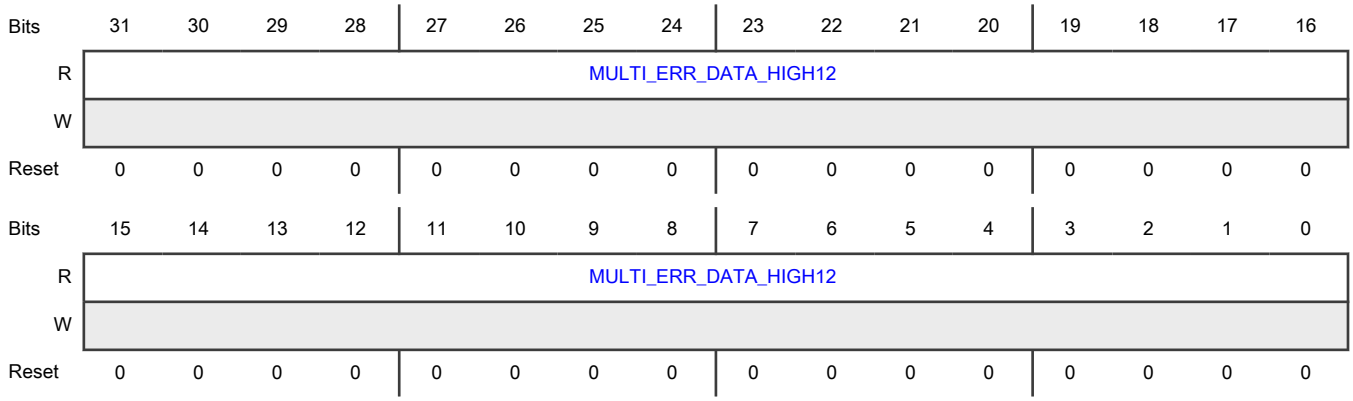
19.6.1.180 High 32 Bits Multiple Error Read Data On Ocram Bank12 (MULTI_ERR_DATA_HIGH12)

When multiple bits error are detected in ocram bank12 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH12	2C8h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH12	High 32 Bits Multiple Error Read Data On Ocram Bank12 When multiple bits error are detected in ocram bank12 read access, store high 32 bits read data.

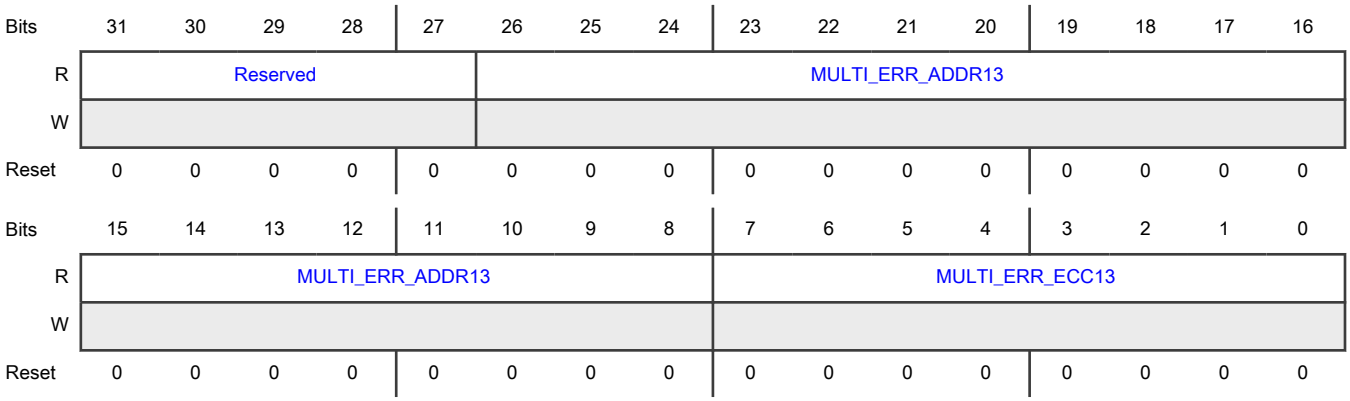
19.6.1.181 Multiple Error Address And ECC Code On Ocram Bank13 (MULTI_ERR_ADDR_ECC13)

When multiple bits error are detected in ocram bank13 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC13	2CCh

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR13	Multiple Error Address On Ocram Bank13 When multiple bits error are detected in ocram bank13 read access, store ocram address.
7-0 MULTI_ERR_ECC13	Multiple Error ECC Code On Ocram Bank13 When multiple bits error are detected in ocram bank13 read access, store ecc code.

19.6.1.182 Low 32 Bits Multiple Error Read Data On Ocram Bank13 (MULTI_ERR_DATA_LOW13)

When multiple bits error are detected in ocram bank13 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW13	2D0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MULTI_ERR_DATA_LOW13															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_ERR_DATA_LOW13															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW13	Low 32 Bits Multiple Error Read Data On Ocram Bank13 When multiple bits error are detected in ocram bank13 read access, store low 32 bits read data.

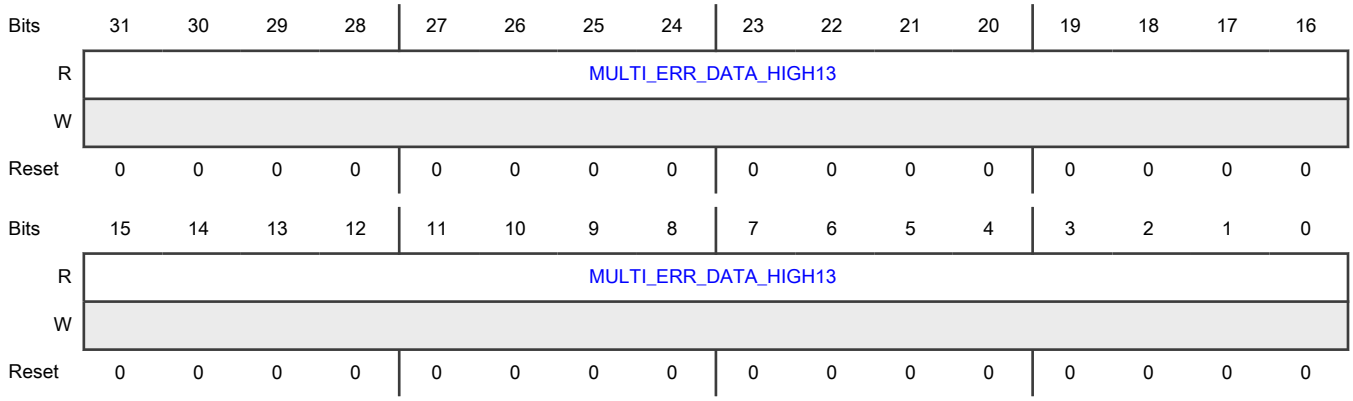
19.6.1.183 High 32 Bits Multiple Error Read Data On Ocram Bank13 (MULTI_ERR_DATA_HIGH13)

When multiple bits error are detected in ocram bank13 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH13	2D4h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH13	High 32 Bits Multiple Error Read Data On Ocram Bank13 When multiple bits error are detected in ocram bank13 read access, store high 32 bits read data.

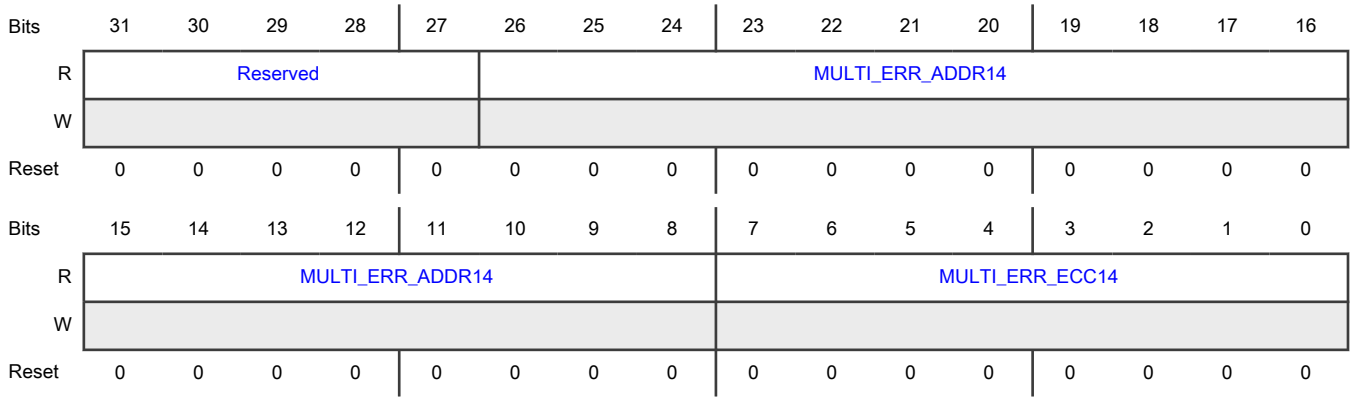
19.6.1.184 Multiple Error Address And ECC Code On Ocram Bank14 (MULTI_ERR_ADDR_ECC14)

When multiple bits error are detected in ocram bank14 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC14	2D8h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR14	Multiple Error Address On Ocram Bank14 When multiple bits error are detected in ocram bank14 read access, store ocram address.
7-0 MULTI_ERR_ECC14	Multiple Error ECC Code On Ocram Bank14 When multiple bits error are detected in ocram bank14 read access, store ecc code.

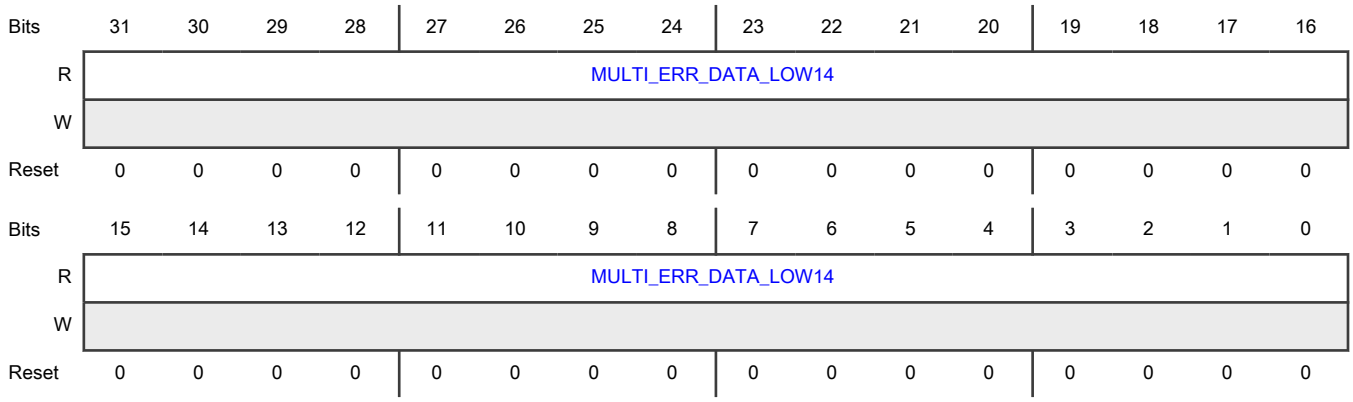
19.6.1.185 Low 32 Bits Multiple Error Read Data On Ocram Bank14 (MULTI_ERR_DATA_LOW14)

When multiple bits error are detected in ocram bank14 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW14	2DCh

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW14	Low 32 Bits Multiple Error Read Data On Ocram Bank14 When multiple bits error are detected in ocram bank14 read access, store low 32 bits read data.

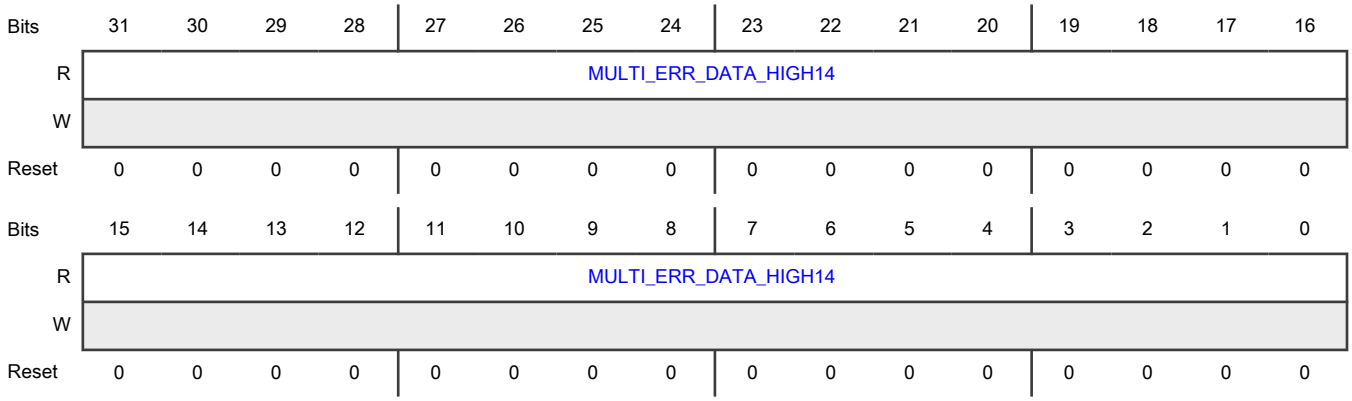
19.6.1.186 High 32 Bits Multiple Error Read Data On Ocram Bank14 (MULTI_ERR_DATA_HIGH14)

When multiple bits error are detected in ocram bank14 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH14	2E0h

Diagram



Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH14	High 32 Bits Multiple Error Read Data On Ocram Bank14 When multiple bits error are detected in ocram bank14 read access, store high 32 bits read data.

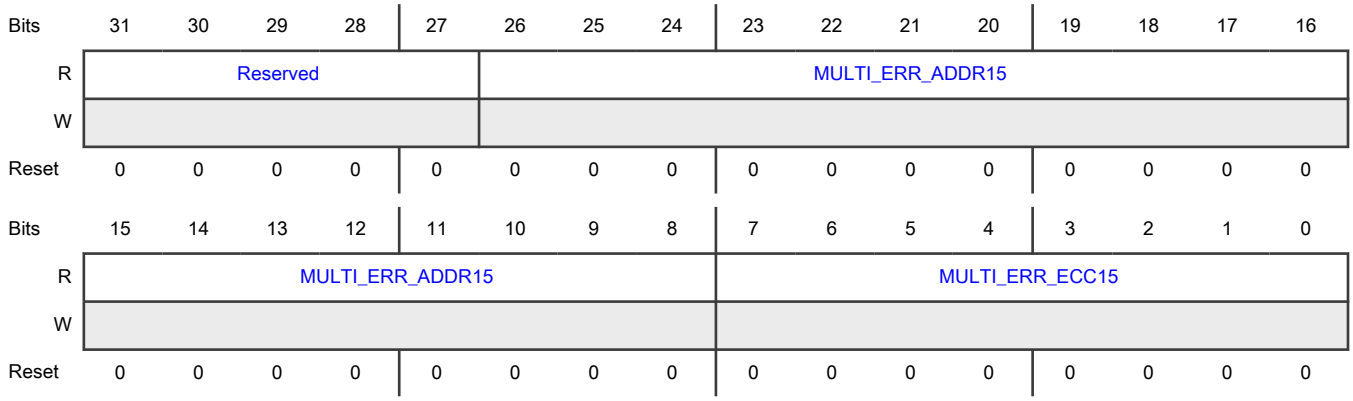
19.6.1.187 Multiple Error Address And ECC Code On Ocram Bank15 (MULTI_ERR_ADDR_ECC15)

When multiple bits error are detected in ocram bank15 read access, store ocram address and ecc code.

Offset

Register	Offset
MULTI_ERR_ADDR_ECC15	2E4h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR15	Multiple Error Address On Ocram Bank15 When multiple bits error are detected in ocram bank15 read access, store ocram address.
7-0 MULTI_ERR_ECC15	Multiple Error ECC Code On Ocram Bank15 When multiple bits error are detected in ocram bank15 read access, store ecc code.

19.6.1.188 Low 32 Bits Multiple Error Read Data On Ocram Bank15 (MULTI_ERR_DATA_LOW15)

When multiple bits error are detected in ocram bank15 read access, store low 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_LOW15	2E8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MULTI_ERR_DATA_LOW15															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_ERR_DATA_LOW15															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MULTI_ERR_DATA_LOW15	Low 32 Bits Multiple Error Read Data On Ocram Bank15 When multiple bits error are detected in ocram bank15 read access, store low 32 bits read data.

19.6.1.189 High 32 Bits Multiple Error Read Data On Ocram Bank15 (MULTI_ERR_DATA_HIGH15)

When multiple bits error are detected in ocram bank15 read access, store high 32 bits read data.

Offset

Register	Offset
MULTI_ERR_DATA_HIGH15	2ECh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MULTI_ERR_DATA_HIGH15															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_ERR_DATA_HIGH15															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MULTI_ERR_DATA_HIGH15	High 32 Bits Multiple Error Read Data On Ocram Bank15 When multiple bits error are detected in ocram bank15 read access, store high 32 bits read data.

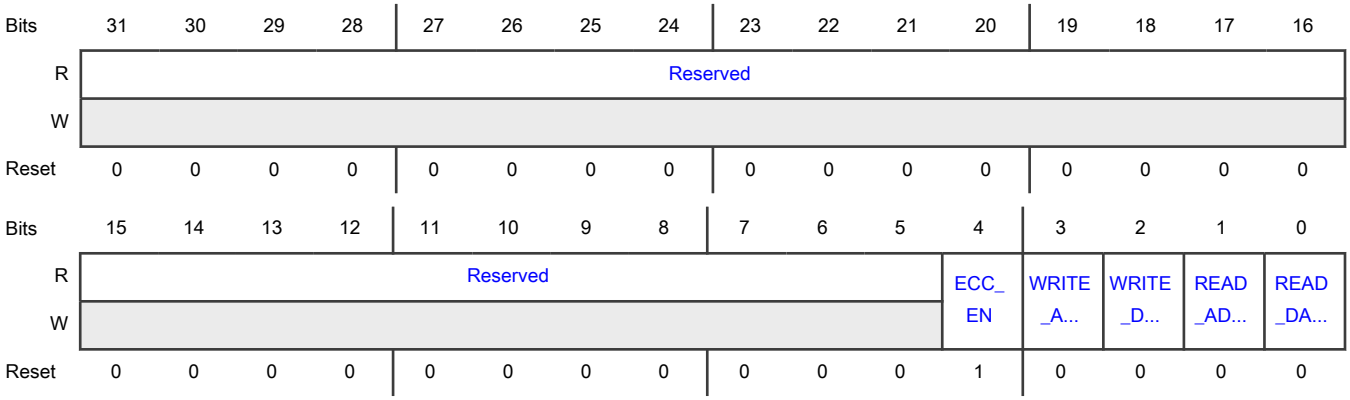
19.6.1.190 Ocram Pipeline And ECC Enable (PIPE_ECC_EN)

Ocram pipeline and ECC enable.

Offset

Register	Offset
PIPE_ECC_EN	2F0h

Diagram



Fields

Field	Description
31-5 —	Reserved
4 ECC_EN	ECC Function Enable Enable ECC64 function for ocram access. This register bit reset value comes from IOMUX GPR. 0 - Disable. 1 - Enable.
3 WRITE_ADDR_PIPE_EN	Write Address Pipeline Enable Pipeline enable signal for write address path, when this signal is set to 1, the write address will be registered before can be applied to memory cell.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Disable. 1 - Enable.
2 WRITE_DATA_PIPE_EN	Write Data Pipeline Enable Pipeline enable signal for write data path, when this signal is set to 1, the write data will be registered before can be applied to memory cell. 0 - Disable. 1 - Enable.
1 READ_ADDR_PIPE_EN	Read Address Pipeline Enable Pipeline enable signal for read address path, when this signal is set to 1, the read address will be registered before can be applied to memory cell. 0 - Disable. 1 - Enable.
0 READ_DATA_WAIT_EN	Read Data Wait Enable 1-cycle wait state in read access enable signal, when this signal is set to 1, 1-cycle wait state will be inserted into each read access. 0 - Disable. 1 - Enable.

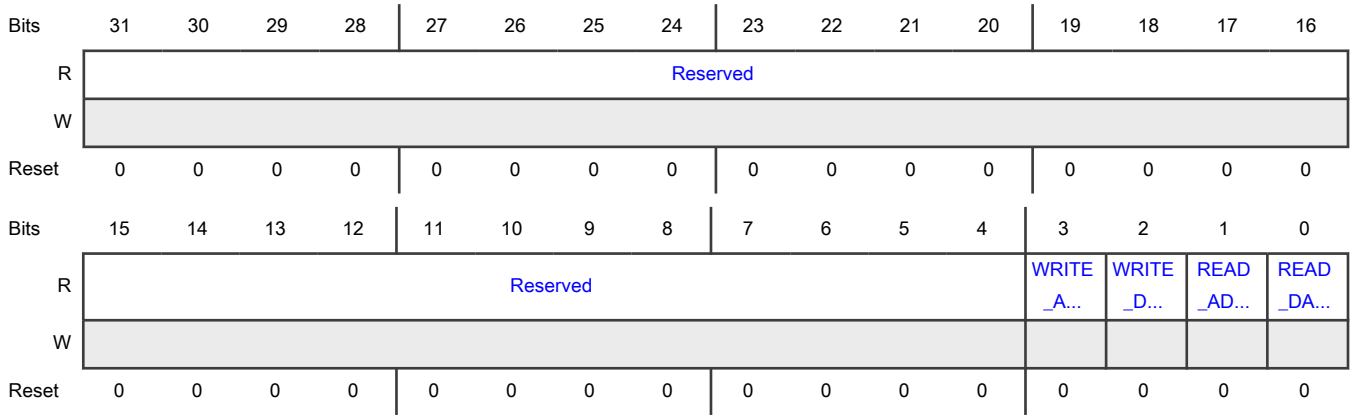
19.6.1.191 Pending Status (PENDING_STAT)

Update pending signal for ocram wait and pipeline enable.

Offset

Register	Offset
PENDING_STAT	2F4h

Diagram



Fields

Field	Description
31-4 —	Reserved
3 WRITE_ADDR_PIPE_PENDING	Write Address Pipeline Pending Update pending status for WRITE_ADDR_PIPE_EN. When WRITE_ADDR_PIPE_EN registe bit is changed, this register bit will be set until the new setup becomes valid in the controller.
2 WRITE_DATA_PIPE_PENDING	Write Data Pipeline Pending Update pending status for WRITE_DATA_PIPE_EN. When WRITE_DATA_PIPE_EN registe bit is changed, this register bit will be set until the new setup becomes valid in the controller.
1 READ_ADDR_PIPE_PENDING	Read Address Pipeline Pending Update pending status for READ_ADDR_PIPE_EN. When READ_ADDR_PIPE_EN registe bit is changed, this register bit will be set until the new setup becomes valid in the controller.
0 READ_DATA_WAIT_PENDING	Read Data Wait Pending Update pending status for READ_DATA_WAIT_EN. When READ_DATA_WAIT_EN registe bit is changed, this register bit will be set until the new setup becomes valid in the controller.

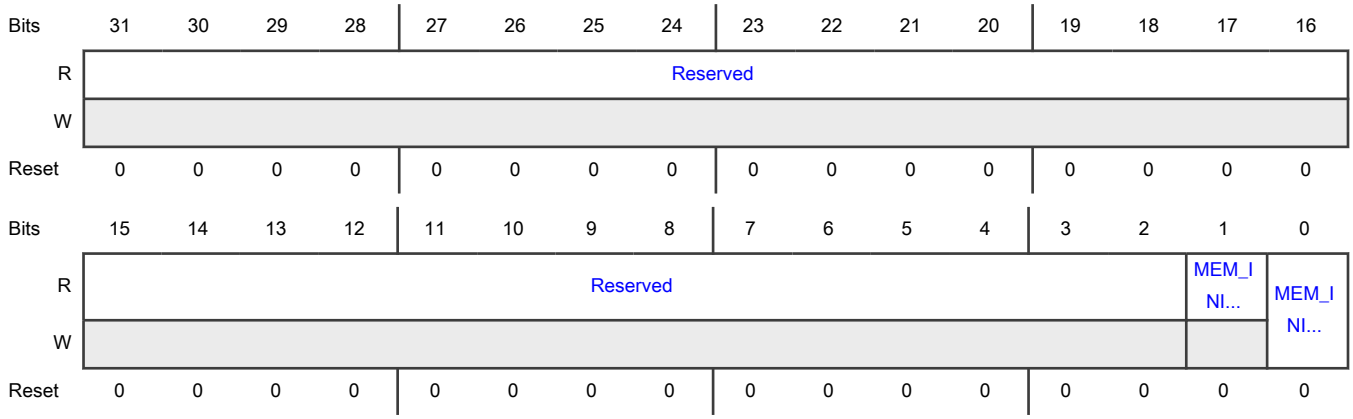
19.6.1.192 Memory initialization (MEM_INIT)

Memory initialization register.

Offset

Register	Offset
MEM_INIT	2F8h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 MEM_INIT_DONE	Memory initialization done. 0 - Memory initialization is not done. 1 - Memory initialization is done.
0 MEM_INIT_ENABLE	Memory initialization enable. 0 - Disable. 1 - Enable.

Chapter 20
Messaging Unit (MU)

20.1 Overview

MU enables two processors on a chip to communicate and coordinate by passing messages (for example, data, status, and control) through the MU interface. MU also allows one processor to signal the other processor using interrupts.

MU must synchronize the accesses from one side to the other because MU can manage messaging between processors using different clocks. MU accomplishes synchronization using two sets of matching registers: processor A-facing and processor B-facing.

20.1.1 Block diagram

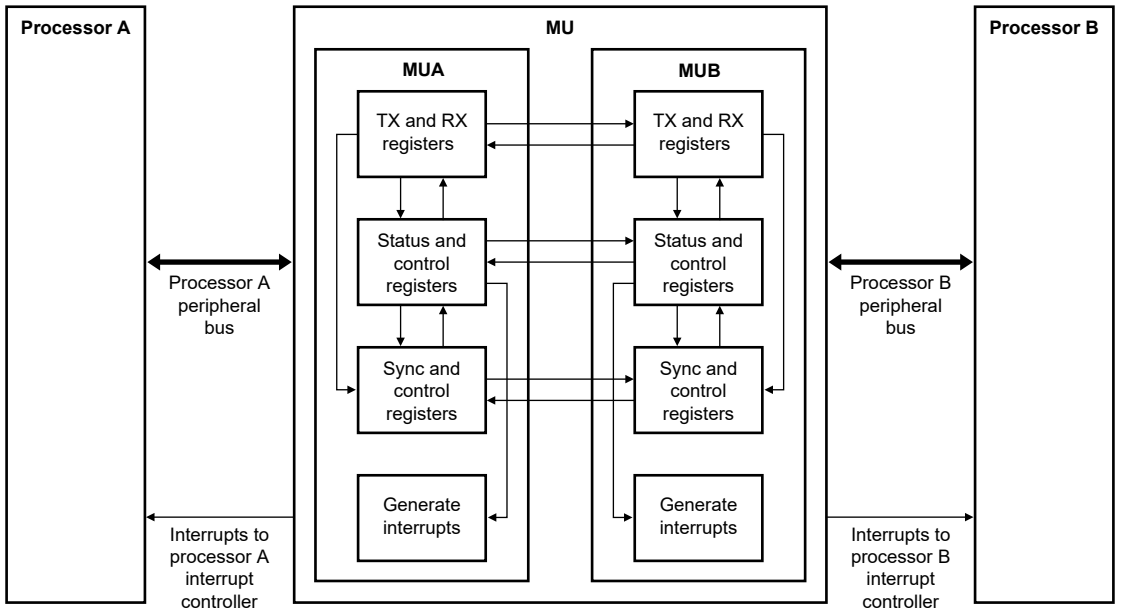


Figure 35. Block diagram

20.1.2 Features

- Memory-mapped registers (MU is connected as a peripheral under the peripheral bus on the processor A-side and processor B-side)
- Synchronized message transfers between cores
  - To send data or messages from one side to the other, MUA provides 4 transmit registers and 4 receive registers. MUB provides 4 transmit registers and 4 receive registers.
  - Transmit empty flags (TSR[TEn]) and receive full flags (RSR[RFn]) facilitate the transfer of data or messages between cores on both sides of MU.
  - A synchronization mechanism updates the transmit and the receive flags. There is an inherent latency between updating the flag on one side and reflecting its status on the other side. See Event update timing.
  - MU has a 3-bit flag data register, which you can use to send flag data between the two MU sides.
- Interprocessor interrupts

  - MU has 12 asynchronous interrupt sources on each side (processor A-side, processor B-side) for signaling the other processor. You can use the interrupts for notification of receive and transmit events and for general-purpose signaling between processors. There are 4 general-purpose interrupt requests available and 8 receive and transmit interrupt sources.
  - Processor A can initiate a non-maskable interrupt (NMI) to processor B.
  - Processor B can initiate a non-maskable interrupt (NMI) to processor A.
- Flexible processor configuration
  - A processor can issue a programmed hardware reset to the other processor by writing 1 to CCR0[HR] .
  - You can configure the boot source for the processor via CCR0[BOOT].
  - You can hold the processor in the reset state by configuring CCR0[RSTH].
  - You can enable the processor platform clock to continue running when the processor enters Stop mode, until the other processor also enters Stop mode. This feature allows the processor to continue accessing peripherals on the bus of the other processor, even when it has entered Stop mode.
- Reset (Each processor can issue a reset to MU via CR[MUR], which is a self-clearing field).

20.2 Functional description

MU enables two cores (processor A and processor B) to communicate with each other:

- By sharing messages and data.
- By enabling one core to wake the other core by using interrupts.

The messaging, control, and status registers of the two cores are mapped to processor A memory and processor B memory as a regular peripheral. The peripheral data bus is 32 bits wide inside MU.

Messaging logic is used with shared memory. Various messaging methods can implement a messaging protocol. For example, a message could mean one of the following:

- A message of *n* words has been written starting at offset *x* in the memory.
- The previous data block that was sent has been read.

The ability to keep messaging logic independent of the shared memory is not restricted to a predefined hardware protocol. The software required to manage the messaging is short and straightforward.

Most of the messaging mechanisms are symmetric. They are duplicated and are available on both the processor A side and processor B side. The messaging mechanisms are:

- 4 32-bit transmit registers, which are each reflected in 4 read-only receive registers on the side of the other processor. These registers can transfer 32-bit word messages or the frame information of the messages written to the shared memory. For example, they can transfer the number of words, initial address, and message type code.
- Writing to a transmitter-side transmit register (TR*n*) clears the Transmitter Empty flag in the transmitter-side Status register (TS*n*[TE*n*]), and sets the Receiver Full flag in the receiver-side Status register (RS*n*[RF*n*]). Setting the flag on the receiver side can trigger an interrupt on the receiver side (a maskable receive interrupt).
- Reading a receiver-side receive register (RR*n*) clears the Receiver Full flag in the receiver-side Status register (RS*n*[RF*n*]), and sets the Transmitter Empty flag in the transmitter-side Status register (TS*n*[TE*n*]). Setting the Transmitter Empty flag can trigger an interrupt on the transmitter side (a maskable transmit interrupt).
- 4 general-purpose interrupt request flags are reflected in General-purpose Status (GSR) on the receiver side.
- 3-bit flag data is transmitted from Flag Control (FCR) to Flag Status (FSR) on the side receiving the flag data. SR[FUP] sets when the flag data is transmitted and clears when the receiving side acknowledges the flag data (the flag is updated).

Writing to a transmit register signals to the receiver side that data is ready for retrieval.

Do not write to the transmit register again without verifying that the data is retrieved. The transmitter side cannot determine the exact time that the receiver attempts to retrieve the data. Before attempting to write to the transmit register again, the transmitter side must wait for the Transmitter Empty interrupt or it must poll TSR[TE*n*].

Reading a receive register signals to the transmitter side that data can be written to that register.

Do not read the receive register again without verifying that the data is written. The receiver side cannot determine the exact time that the transmitter attempts to write the data. Before attempting to read the receive register again, the receiver side must wait for the Receiver Full interrupt or it must poll RSR[RF*n*].

20.2.1 Submodules

20.2.1.1 MUA side

MUA receives its register configuration via the processor A peripheral bus. It sends or receives messages to or from MUB. Processor A can receive messages by reading MUA registers, and MUA can send interrupts to processor A when interrupts are enabled.

NOTE

Processor B should not access MUA registers. Processor C, if exist, should also not access MUA registers. TRDC might be needed to prevent such illegal access of Processor B or C to MUA registers.

20.2.1.2 MUB side

MUB receives its register configuration via the processor B peripheral bus. It sends or receives messages to or from MUA. Processor B can receive messages by reading MUB registers, and MUB can send interrupts to processor B when interrupts are enabled.

NOTE

Processor A should not access MUB registers. Processor C, if exist, should also not access MUB registers. TRDC might be needed to prevent such illegal access of Processor A or C to MUB registers.

20.2.2 Event update timing

The messaging side of each processor has a hardware mechanism to send event update requests to the other processor. An event occurs when the status register of the receiving processor must reflect any information change. The event update latency is the delay between the event being ready at one processor and the resulting update at the status register of the other processor:

- The minimum event latency is (1 clock cycle of the sending side) + (2.5 clock cycles of the receiving side). This minimum case happens when no event is pending when the new event occurs.
- The maximum event latency is (6 clock cycles of the sending side) + (6.5 clock cycles of the receiving side). The maximum case happens when the event occurs just after a previous event is sent to the other side.

The event update latency varies depending on the time at which the subsequent event is triggered.

20.2.3 Clocking

Table 88. MU clocks

Clock name	Description
Bus clock MUA	Used only for bus accesses to MUA control and configuration registers
Bus clock MUB	Used only for bus accesses to MUB control and configuration registers

20.2.4 Resets

The following sections list the reset sources included in MU. Each reset performs a different function for the MU module compared to the function it performs for the system.

20.2.4.1 Asynchronous system reset

When the asynchronous system reset on either side of MU is asserted, SR[MURS] becomes 1 until the asynchronous system reset sequence on both the MUA and MUB sides ends. Verify that SR[MURS] becomes 0 before accessing MU.

The asynchronous system reset on one side of MU resets the other side of MU. The reset forces all control and status registers to return to their default values and clears all internal states. The following table shows the exceptions to this behavior.

Table 89. Exceptions to asynchronous system reset

MUA-side exceptions	MUB-side exceptions
MUB_CCR0[HR]	MUA_CCR0[HR]
MUB_CCR0[HRM]	MUA_CCR0[HRM]
MUB_CCR0[RSTH]	MUA_CCR0[RSTH]
MUB_CCR0[BOOT]	MUA_CCR0[BOOT]
MUB_CR[MURIE]	MUA_CR[MURIE]
MUB_SR[MURIP]	MUA_SR[MURIP]
MUB_SR[MURS]	MUA_SR[MURS]
MUB_CSSR0	MUA_CSSR0

20.2.4.2 Processor hardware reset

Writing 1 to CCR0[HR] generates one hardware reset request to the other processor when CCR0[HRM] on the other MU side is cleared.

Only software asserts and clears CCR0[HR]. Set the required assertion length above the data sheet minimum.

If CCR0[HR] is not masked, MU sends a hardware reset request to the other processor domain reset logic. If CCR0[HR] is masked, MU does not issue the hardware reset request.

20.2.4.3 MU software reset

Writing 1 to CR[MUR] causes most control and status registers to return to their default values and clears all internal states.

The instruction immediately following the assertion of CR[MUR] must not write to the MU registers. The reset sequence may overwrite such a write, with the register retaining the reset value. To know the end of the reset sequence for both processors, monitor the value of SR[MURS]. After the reset sequence on both processors has ended, a write to the MU registers can be attempted.

NOTE

The process of CR[MUR] becoming 1 is delicate because it asynchronously affects the registers on the other side. CR[MUR] becoming 1 may cause unpredictable behavior if, for example, the other processor is concurrently testing an MU register field (TSR[TE*n*] in the other processor). Before writing 1 to CR[MUR], verify that the other processor is not engaged in an MU signaling activity.

20.2.5 Interrupts

MU controls interrupt requests that one processor makes to the other processor. This section describes all the interrupts that the module generates.

MU can generate these interrupt sources individually to send to the processors:

- 4 receive interrupts (asserted when the Receive Full flags are set in Receive Status (RSR) and enabled in Receive Control (RCR)) for each receive register. These interrupts are Ored into a single interrupt for each MUA or MUB, please consult the interrupt map file attached to this document.
- 4 transmit interrupts (asserted when the Transmit Empty flags are set in Transmit Status (TSR) and enabled in Transmit Control (TCR)) for each transmit register. These interrupts are Ored into a single interrupt for each MUA or MUB, please consult the interrupt map file attached to this document.
- 4 general-purpose interrupts (asserted when the GIP flags are set in General-Purpose Interrupt Enable (GIER) and enabled in General-Purpose Interrupt Enable (GIER))
- One non-maskable interrupt (NMI) flag (asserted when the NMI flag is set in Core Control 0 (CCR0))
- MU software and hardware reset (asserted when CR[MUR] and CCR0[HR] are written to and enabled by CR[MURIE] and CIER0[HRIE])

All interrupts except NMI are maskable in the processor control registers: TCR, RCR, GIER, and CR. MU does not assume any internal priority of these interrupts. Multiple interrupts (for example, receive 0 and receive 1, or any transmit or general-purpose interrupts) can be asserted simultaneously. The interrupt controller must resolve the priority of these interrupts at the chip level.

Triggering any enabled interrupt wakes the processor from below mode before servicing the interrupt

The software (as part of the interrupt handler) must clear the general-purpose interrupt pending flags (GSR[GIP*n*]) to deassert the request to the interrupt controller.

The software must also clear the NMI by writing CSSR0[NIMC] (as a part of the interrupt handler) to deassert the NMI request.

When a processor writes to the general-purpose interrupt (GCR[GIR*n*]), MU synchronizes the write event to the other processor to set the general-purpose interrupt request pending flag (GSR[GIP*n*]). When GSR[GIP*n*] is set and the general-purpose interrupt is enabled on the receiving side (GIER[GIE*n*] is 1), the transmitting-side general-purpose interrupt is issued to the receiving processor, which clears this interrupt by writing 1 to GSR[GIP*n*]. The interrupt is deasserted as soon as the write to GSR[GIP*n*] occurs. MU synchronizes the write event of GSR[GIP*n*] to the other processor. The synchronized signal writes 0 to the GIR*n* interrupt.

Before writing 1 to GCR[GIR*n*], verify that the field is 0, which means that a general interrupt is not pending. Generally, MU ignores writing 1 to this field while the field is already 1, but in some cases it may issue a second interrupt.

20.3 External signals

This module has no external signals.

20.4 Initialization

MU does not require initialization.

20.5 Application information

MU facilitates messages between processors. For example, MU passes:

- Short messages. Transmit registers can pass short messages from 1 to 4 words in length for processor A and from 1 to 4 words for processor B. For example, for a four-word message, only one register must have its corresponding interrupt enabled on the receiving side. The first three words of the message are written to the registers with masked interrupts. The fourth word is written to the last register, triggering an interrupt on the receiving side.
- Frame information. Transmit registers can pass frame information for long messages written to the shared system memory. Such frame information normally includes a start address and a number of words. It can include a message type code.
- Event notices and requests. MU can signal events and requests that do not include data words between processors using general-purpose interrupts. For example, one such event is acknowledging that a long message is read from the shared system memory.

- Fixed-length data. Formatted data with a fixed length can be written to predetermined locations in the shared memory. A processor can use general-purpose interrupts to signal to the other processor that the data is ready.
- Announcements. A processor can use the 3 flags to announce its current program state or other billboard messages to the other processor.

20.5.1 Messaging protocols using interrupts

The example below describes a four-word messaging sequence sent between the processor A and processor B.

The transmitting processor writes to the transmit registers sequentially. When *n* = 0, 1, and 2, the interrupts are disabled, so no interrupt goes to the processor (although interrupt conditions occur). For *n* = 3, the interrupt is enabled, and MU generates the last receive interrupt request.

1. Write sequence:
    a. The transmitting processor writes the message information sequentially to its Transmit registers 0, 1, 2.
    b. When the write to Transmit register 3 occurs, RSR[RF3] is set after synchronization. It immediately triggers the receive full 3 interrupt on the receiving processor.
2. Read sequence:
    a. The receiving processor receives the receive full 3 interrupt and starts reading the message transferred from the receive registers.
    b. After the receiving processor reads Receive register 3, MU clears RSR[RF3].

Table 90 and Figure 36 describe the messaging model, using transmit and receive registers and the interrupt messaging protocol.

Table 90. Interrupt messaging protocol (generalized)

Step	Action	Description
1	Processor A writes data.	RR*n* on processor B's side reflects a data write to TR*n* by processor A.
2	Clear the transmitter empty flag and set the receiver full flag.	The data write to TR*n*: • Clears TSR[TE*n*] on the processor A side. • Sets RSR[RF*n*] on the processor B side.
3	Generate the receive interrupt request.	Setting RSR[RF*n*] generates a receive interrupt request to processor B.
4	Processor B reads the data.	After receiving the receive interrupt request, processor B performs a data read of RR*n*.
5	Clear the receiver full flag and set the transmitter empty flag.	Reading the data from the RR*n* register: • Clears RSR[RF*n*] on the processor B side. • Sets TSR[TE*n*] on the processor A side.
6	Generate the transmit interrupt request.	Setting TSR[TE*n*] generates a transmit interrupt request to processor A.

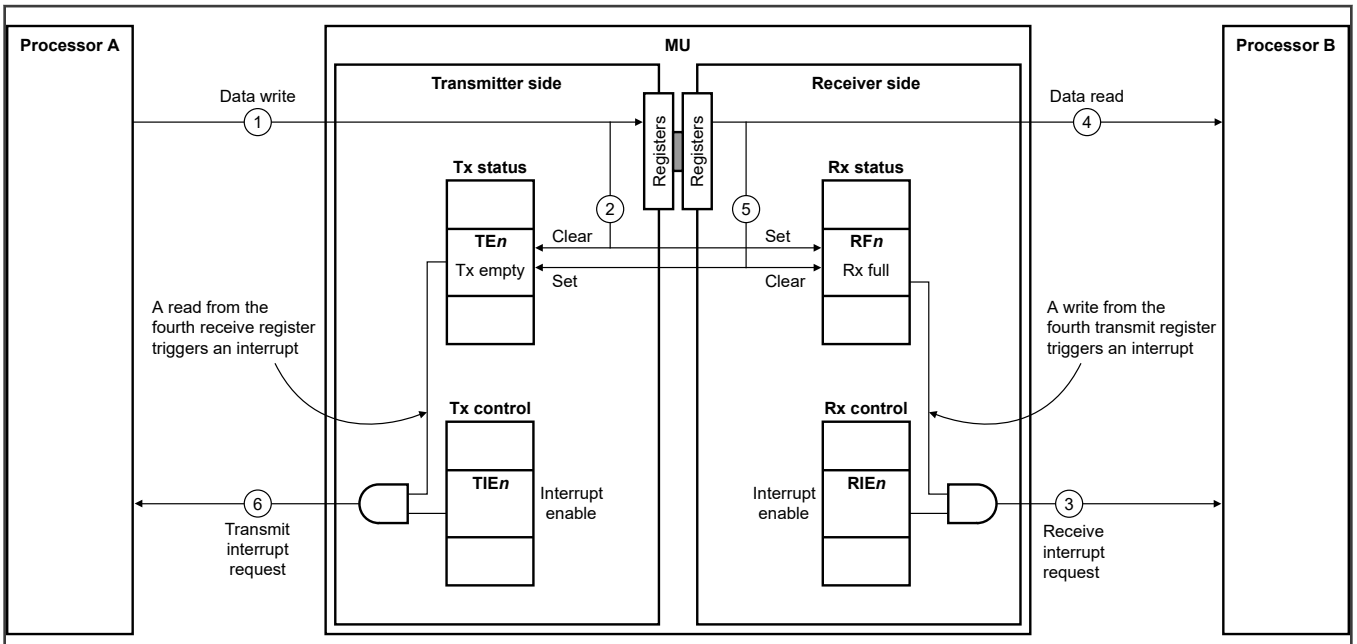


Figure 36. Messaging model using transmit and receive registers

You can use the messaging hardware to implement messaging protocols for an array of message types. MU provides full support for interrupt and polling management schemes.

20.5.2 Messaging protocols using event interrupts

MU can signal events and requests that do not include data words between processors using general-purpose interrupts.

Formatted data with a fixed length can be written to predetermined locations in the shared memory. A processor can use a general-purpose interrupt to signal to the other processor that the data is ready.

A processor can use the 3 flags to announce its current program state (or similar messages) to the other processor.

Table 91 and Figure 37 describe the event steps when the processor triggers a general-purpose interrupt.

Table 91. General-purpose interrupt messaging protocol (generalized)

Step	Action	Description
1	Processor A sets its associated general-purpose interrupt request flag.	Processor A sets GCR[GIR*n*].
2	The general-purpose interrupt request pending status flag is set.	GSR[GIP*n*] is set.
3	MU generates the general-purpose interrupt request to processor B.	Setting GSR[GIP*n*] generates the general-purpose interrupt request to processor B. GIER[GIE*n*] must be set for processor B.
4	Processor B reads the status register.	Processor B reads GSR[GIP*n*].
5	Processor B services the interrupt.	—

Table continues on the next page...

Table 91. General-purpose interrupt messaging protocol (generalized) (continued)

Step	Action	Description
6	Processor B sets GSR[GIP*n*] to clear the interrupt.	Processor B writes 1 to the corresponding GSR[GIP*n*] flag to clear the interrupt.
7	GCR[GIR*n*] is cleared.	Setting GSR[GIP*n*] clears GCR[GIR*n*] on the processor A side.

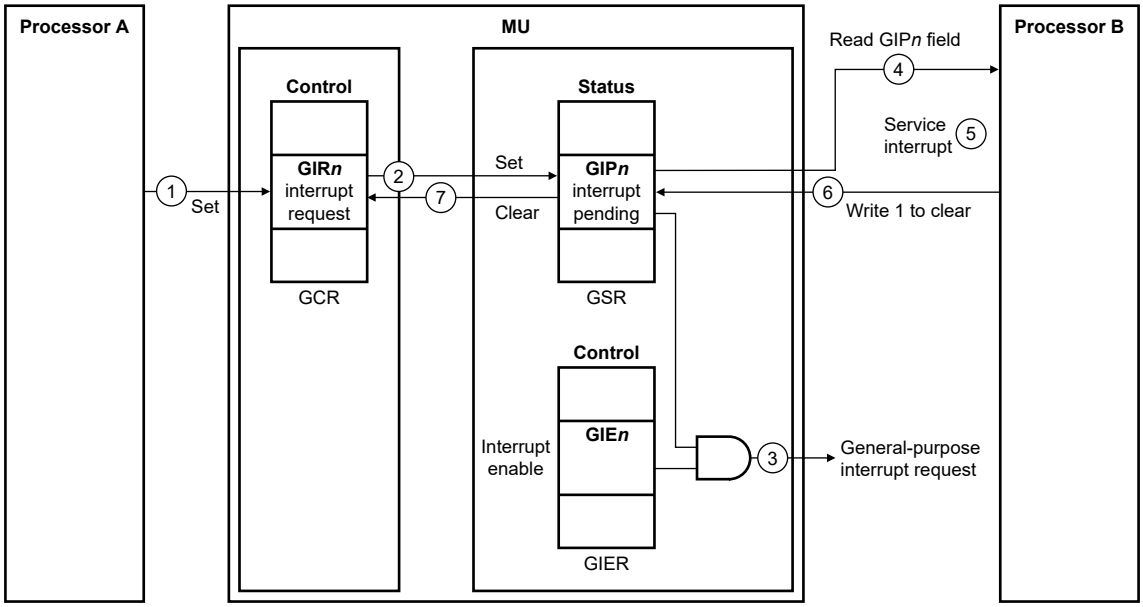


Figure 37. Messaging model using a general-purpose interrupt

20.5.3 Exclusive access to shared memory

MU can signal one processor about its current access to shared memory. This signaling prevents the other processor from overwriting data during the exclusive memory access period.

The following tables describe the signaling protocol that processor A uses to inform processor B about its current access (write) to shared memory:

- Table 92 shows processor A performing an exclusive access to shared memory.
- Table 93 shows processor B scanning for transaction information.
- Table 94 shows processor B accepting exclusive access from processor A.
- Table 95 shows processor B rejecting exclusive access from processor A.

According to the examples shown in the following tables, GCR[GIR*n*], RR*n*, and TR*n* are reserved to support exclusive access to the shared memory protocol.

Table 92. Processor A performs an exclusive access to shared memory

Step	Action	Description
1	Processor A sends the GIR*n* request to processor B using the processor A control register.	Before processor A performs an exclusive access to the shared memory, it sends a GIR*n* request to processor B.
2	Processor A sends an exclusive-access request using a transmit data register (TR*n*).	Processor A sends an exclusive-access request (command, location, and length of target access) to processor B using a selected transmit data register (TR0).
3	Processor A waits for a dedicated interrupt from processor B.	Processor A waits for a dedicated interrupt (as an acknowledgment) triggered by processor B before proceeding.
4	Processor A accesses the shared memory.	After receiving a dedicated interrupt from processor B, processor A proceeds.

Table 93. Processor B scans for transaction information

Step	Action	Description
1	Processor B receives an interrupt from a receive data register (RR*n*).	—
2	Processor B reads the receive data register (RR*n*).	—
3	Processor B scans the receive data register contents.	Processor B scans for transaction information (whether processor A has requested exclusive access).

Table 94. Processor B accepts exclusive access from processor A

Step	Action	Description
1	Processor B triggers a dedicated interrupt.	Processor B acknowledges the request from processor A by triggering a dedicated interrupt (ack) to processor A.
2	Processor B sends a code message to processor A.	With the acknowledge interrupt, processor B sends a code message to processor A through the selected transmit register (TR*n*). The message informs processor A that it can exclusively access the shared memory.

Table 95. Processor B rejects exclusive access from processor A

Step	Action	Description
1	Processor B ignores the request from processor A for exclusive access.	If processor B does not provide permission to processor A, processor B ignores the exclusive-access request.

20.5.4 Packet data transfers

The following table describes an example packet transfer sequence between processor B and processor A subsystems.

Table 96. Packet data transfer sequence

Step	Action	Description
1	Processor B requests DMA.	Processor B sends a DMA request to initiate the packet data transfer.
2	DMA data transfer	DMA acknowledges.
3		DMA starts transferring data from the specified processor B memory location to the specified shared memory.
4		DMA interrupts processor B to signal that the packet transfer has finished.
5	Processor B informs processor A that data is in shared memory.	Using a B-side transmit register, processor B sends a packet information message to processor A about the arrival of new packet data stored in shared memory. The message contains the command, location, and length of packet data.
6	Processor A receives an interrupt.	Processor A receives an interrupt (assuming its corresponding processor A MU-side receive interrupt is enabled). The pending processing task becomes active and processes packet data from memory.
7	Processor A reads data, then writes data.	Processor A reads or processes packet data from shared memory.
8		Processor A writes the result from packet processing to a separate buffer.
9	Processor A informs processor B that the transfer is finished.	After the processing of the packet data finishes, processor A informs processor B (using MU processor A-side transmit register, MUA_TR*n*).
10	Processor A sends an interrupt to processor B (a request for more data).	Processor B receives the next interrupt from processor A, in which processor A requests more packet data.

20.5.5 Freeing a processor from deadlock

During normal operation, one processor may determine that the other processor is not working or is deadlocked. Using Status (SR), the processor can use the methods in the following table to identify and correct the problem.

Table 97. Freeing a processor from deadlock

Processor mode	Technique	Description
—	Processor issues an interrupt.	The other processor can interrupt the processor by issuing one of the 12 (general purpose, receive, or transmit) interrupts.
Normal	Processor issues a non-maskable interrupt.	If a processor is in Normal mode, the other processor can issue a non-maskable interrupt (NMI). You can configure the NMI routine to execute a diagnostic routine that is predefined for such an event.
—	Processor issues a hardware reset.	If the above methods are not helpful, the processor can issue a hardware reset strobe to the other processor.

20.6 Register definition

MU provides transmit and receive data registers (*x*TR0–*n*, *x*RR0–*n*) for communication between processor A and processor B. It also provides control registers (*x*CR) for operations such as interrupts and resets, and status registers (*x*SR) for checking the status of the other MU-side. Figure 38 shows the schematic for MU registers.

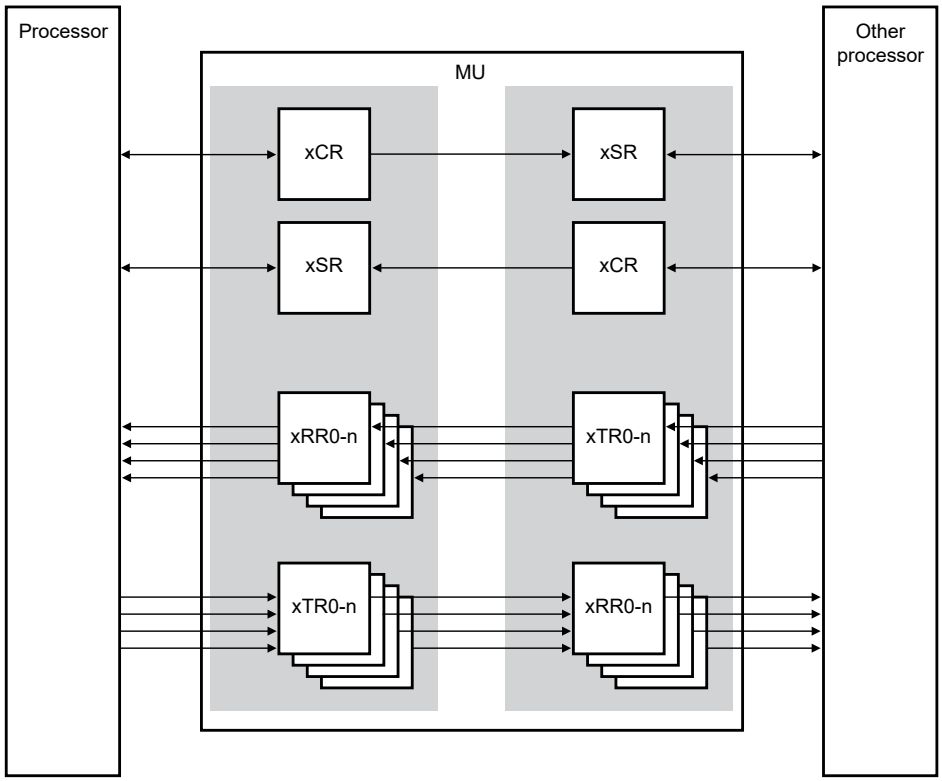


Figure 38. MU registers

20.6.1 MUA register descriptions

This section contains the detailed register descriptions for MUA registers.

NOTE

A module transfer error to processor A or processor B is generated when:

- A read or write access is made to an invalid location.
- A write operation is performed on a read-only register on the processor A side or processor B side of MU.

20.6.1.1 MUA memory map

MU1.MUA base address: 4422_0000h

MU2.MUA base address: 4243_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VER)	32	R	0309_000F
4	Parameter (PAR)	32	R	0304_0404
8	Control (CR)	32	RW	0000_0000
C	Status (SR)	32	RW	See section
10	Core Control 0 (CCR0)	32	RW	0000_0014
14	Core Interrupt Enable 0 (CIER0)	32	RW	0000_0000
18	Core Sticky Status 0 (CSSR0)	32	RW	See section
100	Flag Control (FCR)	32	RW	0000_0000
104	Flag Status (FSR)	32	R	0000_0000
110	General-Purpose Interrupt Enable (GIER)	32	RW	0000_0000
114	General-Purpose Control (GCR)	32	RW	0000_0000
118	General-purpose Status (GSR)	32	RW	0000_0000
120	Transmit Control (TCR)	32	RW	0000_0000
124	Transmit Status (TSR)	32	R	0000_000F
128	Receive Control (RCR)	32	RW	0000_0000
12C	Receive Status (RSR)	32	R	0000_0000
200 - 20C	Transmit (TR0 - TR3)	32	RW	0000_0000
280 - 28C	Receive (RR0 - RR3)	32	R	0000_0000

20.6.1.2 Version ID (VER)

Determines the version ID and feature set number of MUA.

Offset

Register	Offset
VER	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Fields

Field	Description
31-24 MAJOR	Major Version Number
23-16 MINOR	Minor Version Number
15-0 FEATURE	Feature Set Number Indicates the feature set number. MU implements: • Standard features • Expanded number of TR*n*/RR*n* registers

20.6.1.3 Parameter (PAR)

Defines the number of flags, transmit registers, receive registers, and general-purpose interrupt requests available for MU.

Offset

Register	Offset
PAR	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FLAG_WIDTH								GIR_NUM							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RR_NUM								TR_NUM							
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

Fields

Field	Description
31-24 FLAG_WIDTH	Flag Width Specifies the number of flag bits (3) in the Flag Control (FCR) and Flag Status (FSR) registers.
23-16 GIR_NUM	General-Purpose Interrupt Request Number Specifies the number of general-purpose interrupt requests available (4).
15-8 RR_NUM	Receive Register Number Specifies the number of receive registers (4).
7-0 TR_NUM	Transmit Register Number Specifies the number of transmit registers (4).

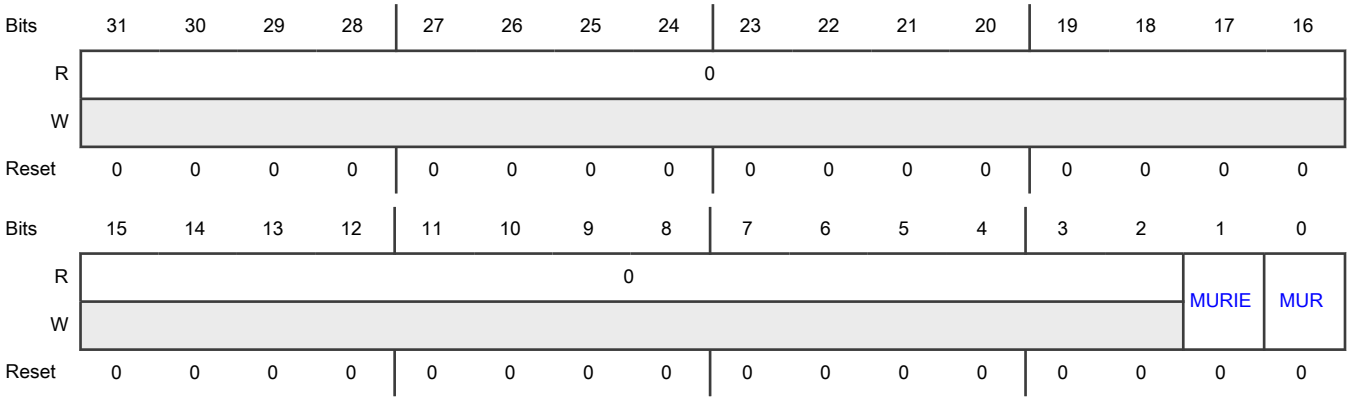
20.6.1.4 Control (CR)

Controls MU reset and reset interrupt enable.

Offset

Register	Offset
CR	8h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 MURIE	MUA Reset Interrupt Enable Enables the processor A-side MU reset interrupt request due to MU reset issued by MUB. If the value of this field is 1, an MU reset interrupt request is issued to processor A when MUA_SR[MURIP] = 1. If the value of this field is 0, MU ignores the value of MURIP and issues no MU reset interrupt request. Only a system reset can reset this field. CR[MUR] cannot reset this field. 0 - Disable 1 - Enable
0 MUR	MU Reset Resets MU. Writing 1 to this field resets the MUA and MUB sides. All internal states are cleared. It forces all control and status registers to return to their default values (except HR, HRM in MUA/B_CCR0 registers; MURIE in MUA/B_CR registers; MURIP and MURS in MUA/B_SR registers). Before writing 1 to this field, interrupt processor B because writing 1 to this field may affect the ongoing processor B program. After writing 1 to this field, monitor the value of MUA_SR[MURS] to know when the reset sequence on the processor B-side has ended. This field always reads 0, and it becomes 0 during the MU reset sequence. 0 - Idle 1 - Reset

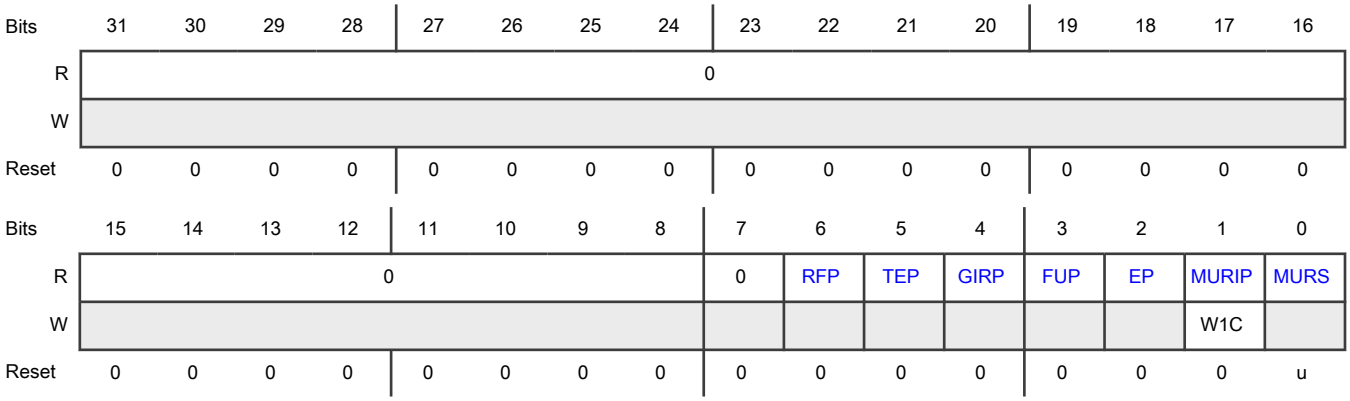
20.6.1.5 Status (SR)

Shows the status of MU resets and the status of pending events and requests.

Offset

Register	Offset
SR	Ch

Diagram



Fields

Field	Description
31-8 —	Reserved
7 —	Reserved
6 RFP	MUA Receive Full Pending Indicates whether a receive full message is pending. This field becomes 1 when MUB writes to a TR*n* register to send data to MUA. After this field becomes 1, MU checks RSR[RF*n*] to determine whether the data in the Receive register is ready for MUA to read it. This field becomes 0 when all MUA RR*n* registers are read, or when MU is reset. 0 - Not pending; MUB is not writing to a Transmit register 1 - Pending; MUB is writing to a Transmit register
5 TEP	MUA Transmit Empty Pending Indicates whether a transmit empty message is pending. This field becomes 1 when any TCR[TIE*n*] field is 1 and TSR[TE*n*] flag is set. After this field becomes 1, MU checks TSR[TE*n*] to determine whether the data in the Transmit (TR*n*) register is ready for MUA to write to it. This field becomes 0 when write operations to all MUA Transmit (TR*n*) registers where TCR[TIE*n*] = 1 (transfer interrupt enabled) are completed, or when MU is reset. 0 - No MUA transmit empty event pending

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Pending; any TCR[TIEn] field is 1 and TSR[TEn] flag is set
4 GIRP	MUA General-Purpose Interrupt Pending Indicates that MUB has sent a general-purpose interrupt request. This field becomes 1 when the MUB side sends a general-purpose interrupt request to the MUA side. GSR[GIP*n*] identifies which general-purpose interrupt request is received. This field becomes 0 when all MUA_GSR[GIP*n*] fields are cleared, or when MU is reset. 0 - No request sent 1 - Request sent
3 FUP	MUA Flag Update Pending Indicates whether a flag update request is pending. MU generates this request when there is a change to the F*n*[2:0] bits of MUA_FCR. This field becomes 1 when the MUA side sends a flag update request to the MUB side. This field becomes 0 when MU acknowledges this flag update request internally (the flag is updated) from the MUB side, or during MU reset. No flag update changes are allowed when this field is 1. When FUP = 1, a write to the F*n*[2:0] bits of MUA_FCR does not generate a flag update event. The F*n*[2:0] bits do not change. If SR[EP] = 1 (event pending), writing to MUA_FCR does not immediately cause this field to become 1. 0 - No pending update flags (initiated by MUA) 1 - Pending update flags (initiated by MUA)
2 EP	MUA Side Event Pending Indicates a pending side event when the MUA side sends an event update request to the MUB side. An event is any hardware message that the Status register on the MUB side reflects. For example, an event occurs when Transmit register 0 is the target of a write operation. During normal operations, the update mechanism for this field operates automatically. MU clears this field automatically when the event update acknowledgment is received, or when MU resets. To ensure that events are posted to MUB, verify that this field is 0. If it is 1, wait and continue to poll this field until it becomes 0. 0 - Not pending 1 - Pending
1 MURIP	MU Reset Interrupt Pending Flag Indicates whether processor B has issued an MU reset. This flag is set after processor B initiates an MU reset by setting MUB_CR[MUR]. If CR[MURIE] = 1, the processor A MU reset interrupt request is issued when processor B writes 1 to MUB_CR[MUR]. Clearing this flag also clears the MU reset interrupt request. Only a system reset can reset this flag. MU reset cannot reset this flag.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - Reset not issued 1 - Reset issued When writing 0 - No effect 1 - Clear the flag
0 MURS	MUA and MUB Reset State Indicates the reset state of MUA and MUB. This field becomes 1 during any system reset or MU reset from the MUA or MUB side. This field becomes 0 when the reset sequence on both MUA and MUB sides ends. After issuing any of the aforementioned reset events, verify that this field is 0 before starting any access. 0 - Out of reset 1 - In reset

20.6.1.6 Core Control 0 (CCR0)

Allows MUA to control the processor on the MUB side.

Also allows control of the processor A hardware reset mask.

Offset

Register	Offset
CCR0	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								BOOT			RSTH	CLKE	HRM	HR	NMI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Fields

Field	Description
31-7 —	Reserved
6-5 BOOT	Target Processor B Boot Configuration Configures the boot source for processor B when it is the target core. See the chip-specific details for more information on boot options. 00 - Processor Boot option 0 01 - Processor Boot option 1 10 - Processor Boot option 2 11 - Processor Boot option 3
4 RSTH	Processor B Reset Hold Holds processor B in reset. **NOTE** Writing 1 to this field does not force processor B into reset. It only holds processor B if the processor resets itself. 0 - Release from reset 1 - Hold in reset
3 CLKE	Processor B Clock Enable Enables platform clock for Processor B to continue running when Processor B enters Stop mode until processor A also enters Stop mode. When this field is 0, the Processor B platform clock is gated when Processor B enters Stop mode. 0 - Disable 1 - Enable
2 HRM	Processor A Hardware Reset Mask Disables hardware reset requests to the other processor side. When this field is 1, MUB_CCR0[HR] is masked, and the hardware reset request to processor A is disabled. When this field is 0, MUB_CCR0[HR] is not masked, and the hardware reset to the processor A is enabled. 0 - Enable 1 - Disable
1 HR	Processor B Hardware Reset Initiates hardware reset. Write 1 to this field to start a hardware reset of processor B. Write 0 to this field to deassert the processor B hardware reset input.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Writing 1 to this field does not cause an internal MU reset. It is routed outside MU to the processor B reset logic. Processor B can mask the HR reset issued by processor A to processor B (according to the value of MUB_CCR0[HRM]). This field does not return to the reset value during MU software reset (MUR). 0 - Deassert hardware reset to processor B. 1 - Assert hardware reset to processor B.
0 NMI	MUB Nonmaskable Interrupt Request Indicates whether Processor A has issued a nonmaskable interrupt to Processor B. When this field becomes 1, it initiates a nonmaskable interrupt to processor B. This field becomes 0 after 1 is written to MUB_CSSR0[NMIC] to clear that field. After this field becomes 0, MUA can initiate another nonmaskable interrupt to MUB. This field is cleared when MU resets. 0 - Nonmaskable interrupt not issued 1 - Nonmaskable interrupt issued

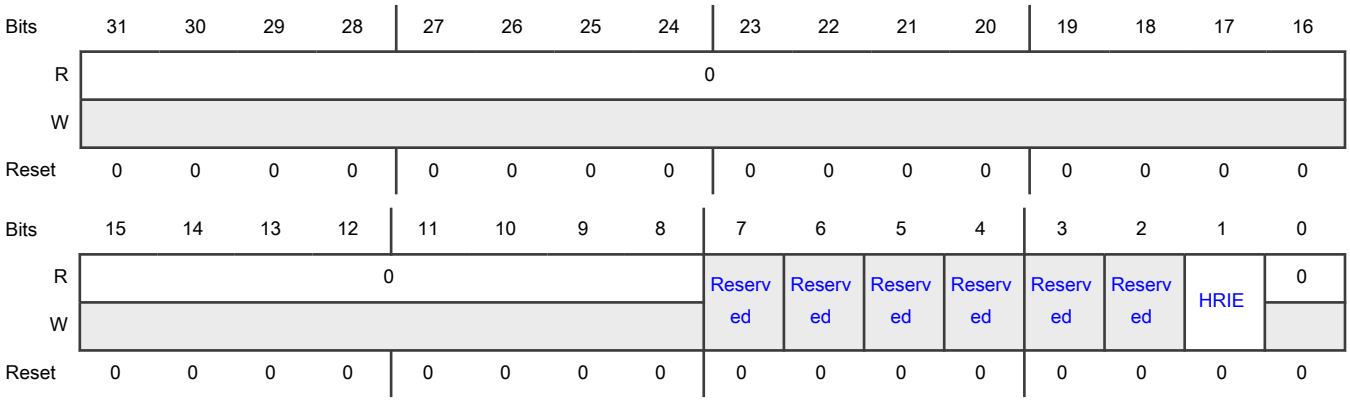
20.6.1.7 Core Interrupt Enable 0 (CIER0)

Controls interrupt enables.

Offset

Register	Offset
CIER0	14h

Diagram



Fields

Field	Description
31-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 HRIE	Processor A Hardware Reset Interrupt Enable Enables Processor A Hardware Reset Interrupt request due to Processor B issuing a hardware reset by writing MUB_CCR0[HR] to Processor A. When the value of this field is 1, MU issues an MU reset interrupt request to Processor A. When the value of this field is 0, MU issues no MU reset interrupt. 0 - Disable 1 - Enable
0 —	Reserved

20.6.1.8 Core Sticky Status 0 (CSSR0)

Shows the status of interrupts pending (W1C).

The reset value is chip-specific.

Offset

Register	Offset
CSSR0	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0	0	0	0	0	HRIP	NMIC
W															W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	u	u

Fields

Field	Description
31-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 HRIP	Processor A Hardware Reset Interrupt Pending Flag Indicates whether Processor A had been reset via MUB_CCR0[HR]. This flag is set when Processor B writes 1 to MUB_CCR0[HR] and MUA_CCR0[HRM] = 0. 0 - No hardware reset initiated 1 - Hardware reset initiated
0	Processor A Nonmaskable Interrupt Clear

Table continues on the next page...

Table continued from the previous page...

Field	Description
NMIC	Clears the nonmaskable interrupt (NMI) request from the MUB side. The MUA-side NMI service routine uses this field. Writing 1 to this field clears MUB_CCR0[NMI], deasserting the NMI request and enabling MUB_CCR0[NMI] to trigger another NMI request. This field always reads as 0, so you cannot poll it. You can only use this field as part of the NMI service routine, in which you must write 1 to this field only once. This field is cleared when MU resets. 0 - Default 1 - Clear MUB_CCR0[NMI]

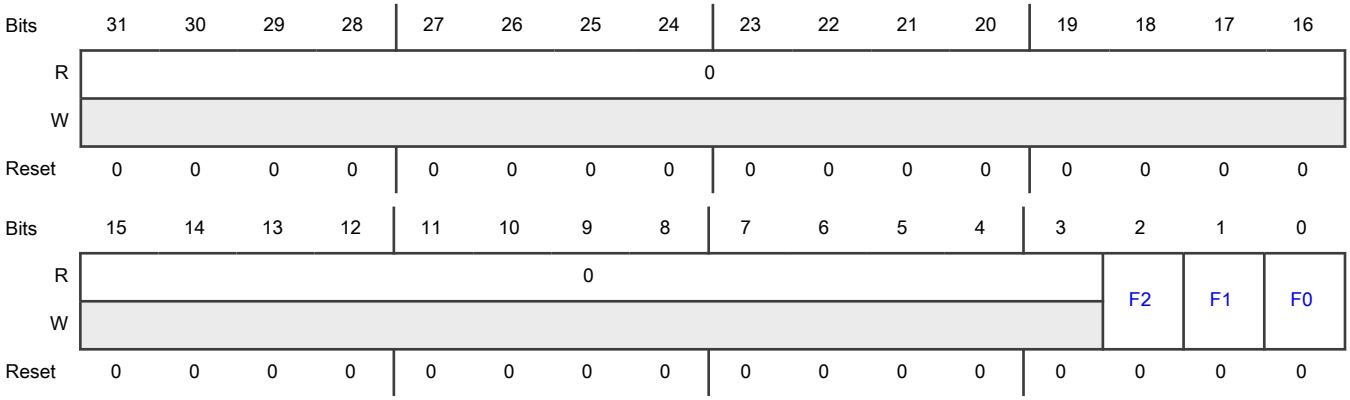
20.6.1.9 Flag Control (FCR)

Configures MUB_FSR[F*n*] flags.

Offset

Register	Offset
FCR	100h

Diagram



Fields

Field	Description
31-3 —	Reserved
2-0	MUA to MUB Flag Configures MUB_FSR[F*n*] flags, where *n* = 0 to 2.

Table continues on the next page...

Table continued from the previous page...

Field	Description
Fn	F*n* configures the corresponding MUB_FSR[F*n*] flag. F*n* becomes 0 when MU resets. 0 - Clear MUB_FSR[Fn] 1 - Set MUB_FSR[Fn]

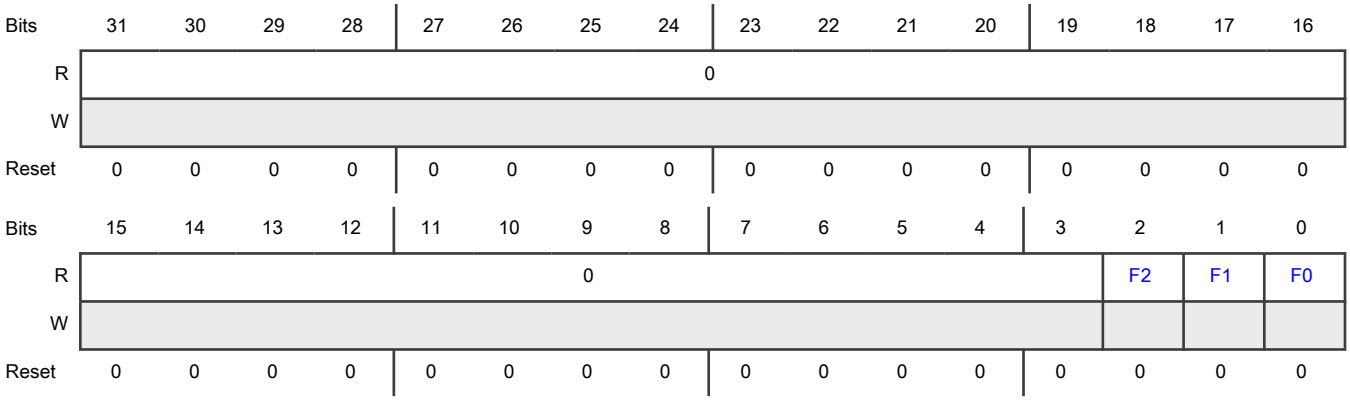
20.6.1.10 Flag Status (FSR)

Contains flags configured by the values written to MUB_FCR[F*n*].

Offset

Register	Offset
FSR	104h

Diagram



Fields

Field	Description
31-3 —	Reserved
2-0 Fn	MUB to MUA-Side Flag Contains flags configured by the values written to MUB_FCR[F*n*], where *n* = 0 to 2. F*n* is the MUA-side flag configured by the values written to MUB_FCR[F*n*]. When MUB_FCR[F*n*] is written to, the write event updates MUA_FSR[F*n*], after the event update latency. 0 - MUB_FCR[Fn] = 0 1 - MUB_FCR[Fn] = 1

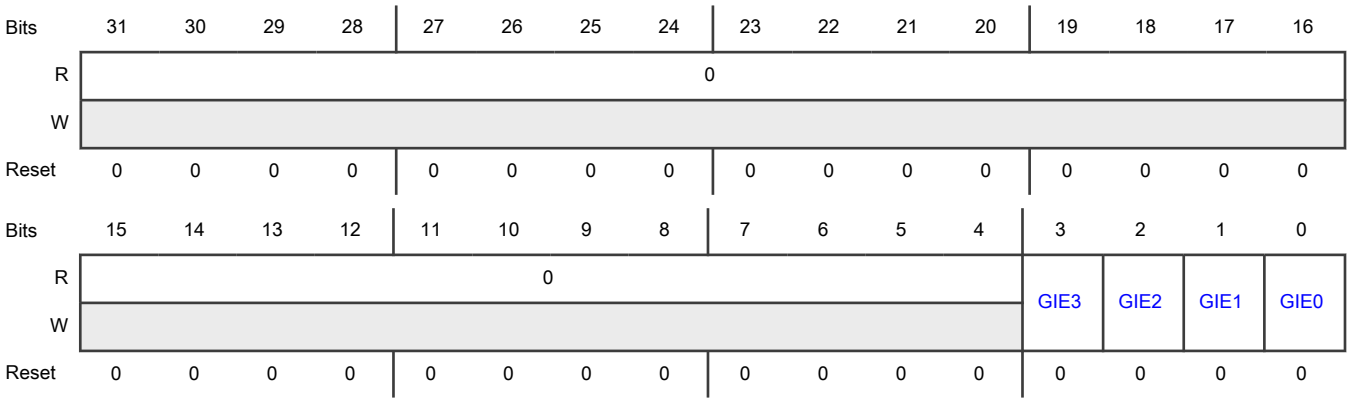
20.6.1.11 General-Purpose Interrupt Enable (GIER)

Contains the MUA general-purpose interrupt enable fields.

Offset

Register	Offset
GIER	110h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 GIEn	MUA General-purpose Interrupt Enable Enables general-purpose interrupt. There are 4 general-purpose interrupts (*n* = 0 to 3). When GIE*n* = 1, a general-purpose interrupt *n* request is issued to processor A when MUA GSR[GIP*n*] = 1. If GIE*n* = 0, the general-purpose interrupt request pending does not trigger the general-purpose interrupt. GIE*n* becomes 0 when MU resets. 0 - Disable 1 - Enable

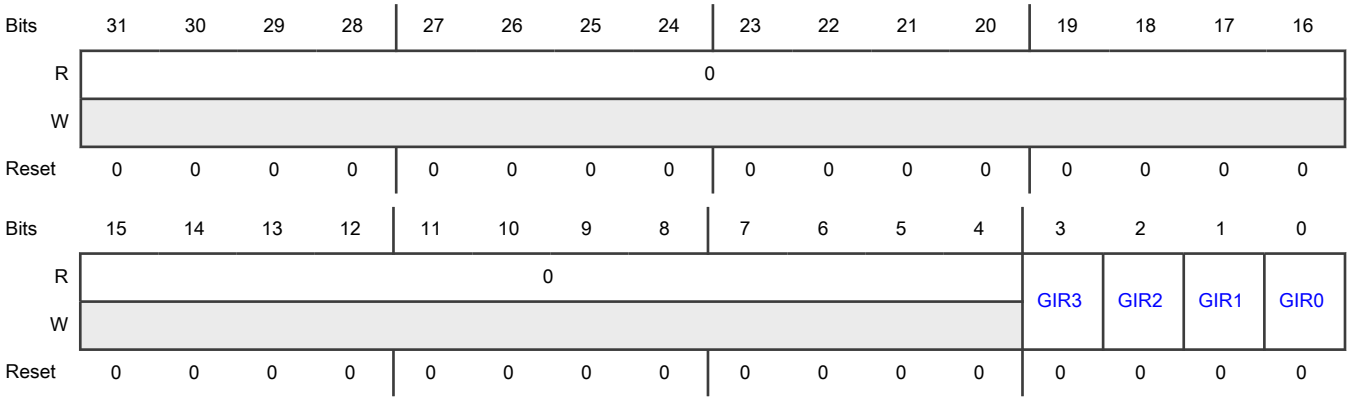
20.6.1.12 General-Purpose Control (GCR)

Contains the MUA general-purpose interrupt request fields.

Offset

Register	Offset
GCR	114h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 GIRn	MUA General-Purpose Interrupt Request Specifies whether general-purpose interrupts are requested to MUB. There are 4 general-purpose interrupts (*n* = 0 to 3). Writing 1 to GIR*n* sets MUB_GSR[GIP*n*]. If MUB_GIER[GIE*n*] = 1, a general-purpose interrupt request is triggered on processor B. This field becomes 0 when MUB_GSR[GIP*n*] is cleared. This clearing informs MUA that the interrupt is accepted (cleared by software). To ensure proper operations, verify that GIR*n* is 0 (no pending interrupt) before writing 1 to it. This field becomes 0 when MU resets. 0 - Not requested 1 - Requested

20.6.1.13 General-purpose Status (GSR)

Contains the status of the MUA general-purpose interrupt pending requests.

Offset

Register	Offset
GSR	118h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												GIP3	GIP2	GIP1	GIP0
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved
3-0 GIPn	MUA General-Purpose Interrupt Request Pending Indicates whether a general-purpose interrupt request is pending. There are 4 general-purpose interrupts (*n* = 0 to 3). GIP*n* informs MUA that MUB_GCR[GIR*n*] changed from 0 to 1. If MUA_GIER[GIE*n*] = 1, a general-purpose interrupt request is issued to processor A. GIP*n* is cleared when MU resets. After GIP*n* is cleared, if MUA_GIER[GIE*n*] = 1, the general-purpose interrupt request is cleared on the MUA side. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not pending 1 - Pending When writing 0 - No effect 1 - Clear the flag

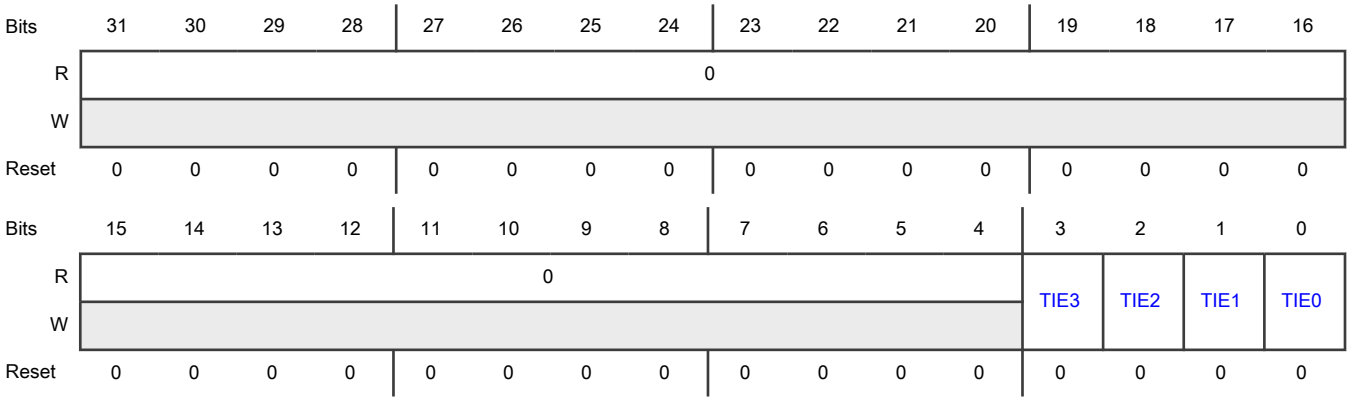
20.6.1.14 Transmit Control (TCR)

Contains the MUA transmit interrupt enable fields.

Offset

Register	Offset
TCR	120h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 TIEn	MUA Transmit Interrupt Enable Enables MUA transmit interrupt *n*, where *n* = 0 to 3. If this field is 1, an MUA transmit interrupt *n* request is issued when MUA_TSR[TE*n*] is set. If this field is 0, MU ignores the value of MUA_TSR[TE*n*], and no MUA transmit interrupt *n* request is issued. 0 - Disable 1 - Enable

20.6.1.15 Transmit Status (TSR)

Indicates whether the MUA transmit registers are empty.

Offset

Register	Offset
TSR	124h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												TE3	TE2	TE1	TE0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Fields

Field	Description
31-4 —	Reserved
3-0 TEn	MUA Transmit Empty Indicates whether MUA Transmit (TR*n*) register is empty, where *n* = 0 to 3. This field becomes 1 after the MUB_RR*n* register is read on the MUB side. When TE*n* = 1, it indicates to the MUA side that the MUA_TR*n* register is ready to be written on the MUA side. If MUA_TCR[TIE*n*] = 1, a transmit *n* interrupt is issued on the MUA side. This field becomes 0 after the MUA_TR*n* register is written to on the MUA side. After this field becomes 0, if MUA_TCR[TIE*n*] = 1, the transmit *n* interrupt request is cleared on the MUA side. This field becomes 1 when MU resets. 0 - Not empty 1 - Empty

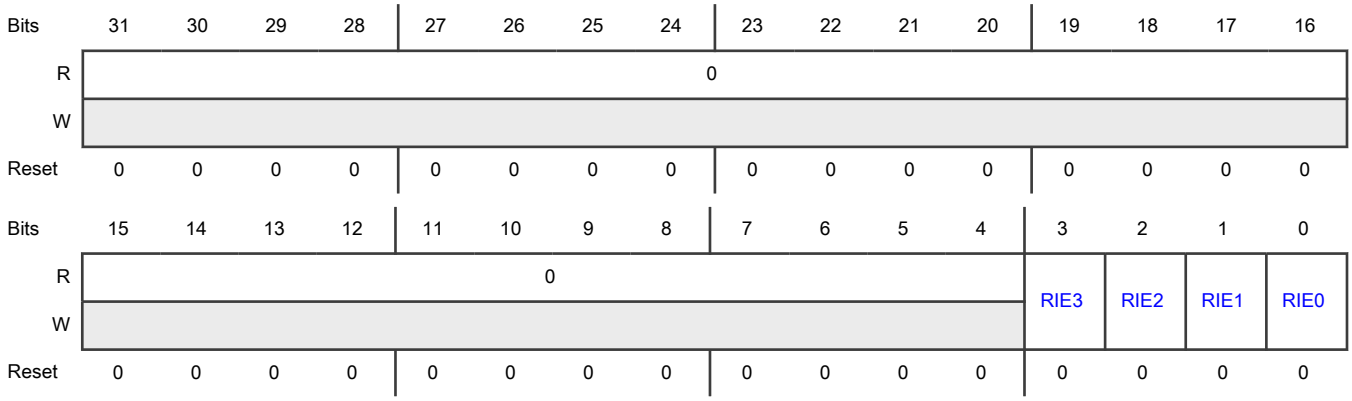
20.6.1.16 Receive Control (RCR)

Contains the MUA receive interrupt enables.

Offset

Register	Offset
RCR	128h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 RIEn	MUA Receive Interrupt Enable Enables MUA receive interrupt *n*, where *n* = 0 to 3. If this field is 1, an MUA receive interrupt *n* request is issued when MUA_RSR[RF*n*] is set. If this field is 0, MU ignores the value of MUA_RSR[RF*n*], and no MUA receive interrupt request is issued. 0 - Disable 1 - Enable

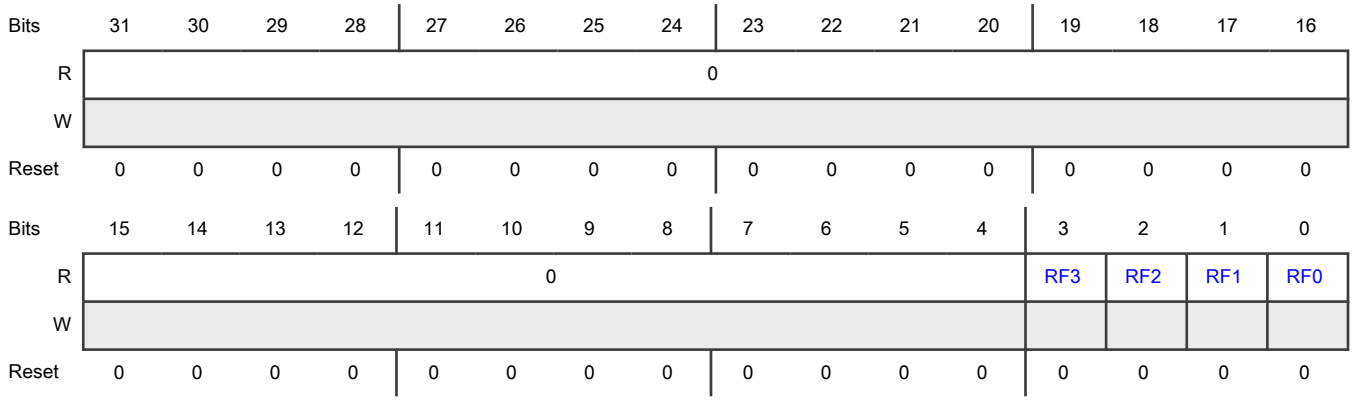
20.6.1.17 Receive Status (RSR)

Indicates whether the MUA receive registers are full.

Offset

Register	Offset
RSR	12Ch

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 RFn	MUA Receive Register Full Indicates whether MUA Receive register (RR*n*) is full, where *n* = 0 to 3. RF*n* becomes 1 when the MUB_TR*n* register is written to on the MUB side. When RF*n* is 1, it indicates to the MUA side that new data in the MUA_RR*n* register is ready for MUA to read it. If MUA_RCR[RIE*n*] = 1, a receive *n* interrupt is issued on the MUA side. RF*n* becomes 0 when the MUA_RR*n* register is read, or when MU is reset. After RF*n* becomes 0, if MUA_RCR[RIE*n*] = 1, the receive *n* interrupt request is cleared on the MUA side. 0 - Not full 1 - Full

20.6.1.18 Transmit (TR0 - TR3)

Contains MUA transmit data.

Offset

Register	Offset
TR0	200h
TR1	204h
TR2	208h
TR3	20Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	TR_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	TR_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TR_DATA	MUA Transmit Data Contains MUA transmit data. MUB_RR*n* reflects the data written to this register. The TR*n* and RR*n* registers are not double-buffered. Writing to MUA_TR*n* overrides the data readable in the MUA_RR*n* register. A write to the Transmit register clears MUA_TSR[TE*n*] on the transmitter side, and sets MUB_RSR[RF*n*] on the receiver side. You can write to this register only when MUA_TSR[TE*n*] = 1. Reading this register returns all zeroes.

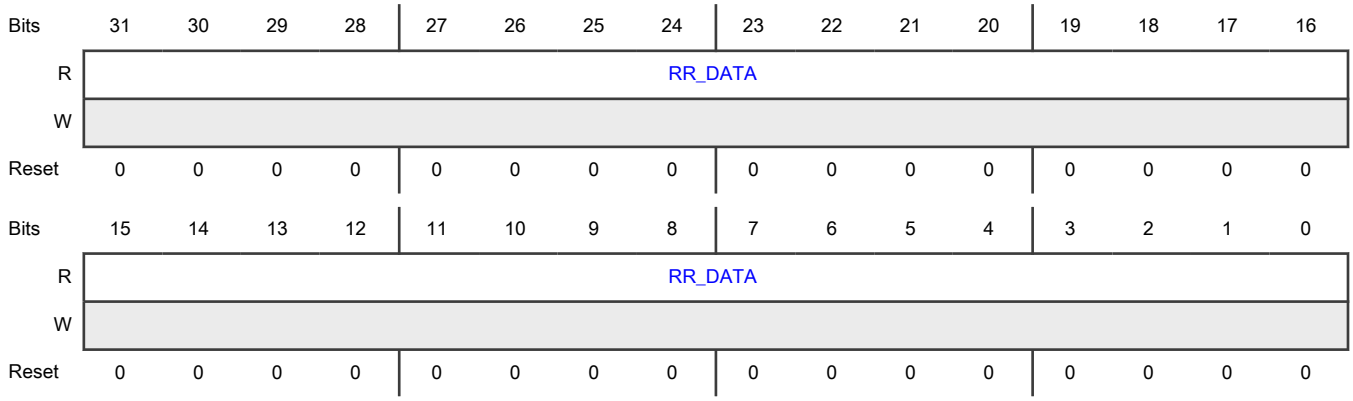
20.6.1.19 Receive (RR0 - RR3)

Contains MUA receive data.

Offset

Register	Offset
RR0	280h
RR1	284h
RR2	288h
RR3	28Ch

Diagram



Fields

Field	Description
31-0 RR_DATA	MUA Receive Data Reflects the data written to MUB TR*n*. Reading this register clears MUA_RSR[RF*n*] on the receiver side, and sets MUB_TSR[TE*n*] on the transmitter side. You can read this register only when MUA_RSR[RF*n*] = 1. Reading it before MUA_RSR[RF*n*] becomes 1 may result in reading incorrect data. Poll MUA_RSR[RF*n*] to confirm it is set before reading RR*n*. Writing to this register generates an error response to MUA.

20.6.2 MUB register descriptions

This section contains the detailed register descriptions for MUB registers.

NOTE

A module transfer error to processor A or processor B is generated when:

- A read or write access is made to an invalid location.
- A write operation is performed on a read-only register on the processor A side or processor B side of MU.

20.6.2.1 MUB memory map

MU1.MUB base address: 4423_0000h

MU2.MUB base address: 4244_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VER)	32	R	0309_000F
4	Parameter (PAR)	32	R	0304_0404
8	Control (CR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
C	Status (SR)	32	RW	See section
10	Core Control 0 (CCR0)	32	RW	0000_0014
14	Core Interrupt Enable 0 (CIER0)	32	RW	0000_0000
18	Core Sticky Status 0 (CSSR0)	32	RW	See section
100	Flag Control (FCR)	32	RW	0000_0000
104	Flag Status (FSR)	32	R	0000_0000
110	General-Purpose Interrupt Enable (GIER)	32	RW	0000_0000
114	General-Purpose Control (GCR)	32	RW	0000_0000
118	General-purpose Status (GSR)	32	RW	0000_0000
120	Transmit Control (TCR)	32	RW	0000_0000
124	Transmit Status (TSR)	32	R	0000_000F
128	Receive Control (RCR)	32	RW	0000_0000
12C	Receive Status (RSR)	32	R	0000_0000
200 - 20C	Transmit (TR0 - TR3)	32	RW	0000_0000
280 - 28C	Receive (RR0 - RR3)	32	R	0000_0000

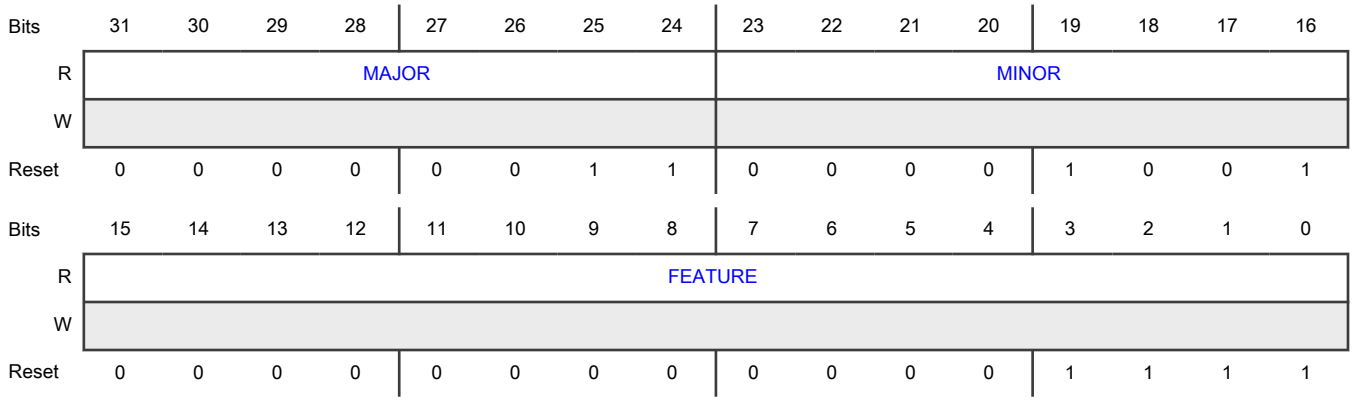
20.6.2.2 Version ID (VER)

Determines the version ID and feature set number of MUB.

Offset

Register	Offset
VER	0h

Diagram



Fields

Field	Description
31-24 MAJOR	Major Version Number
23-16 MINOR	Minor Version Number
15-0 FEATURE	Feature Set Number Indicates the feature set number. MU implements: • Standard features • Expanded number of TR*n*/RR*n* registers

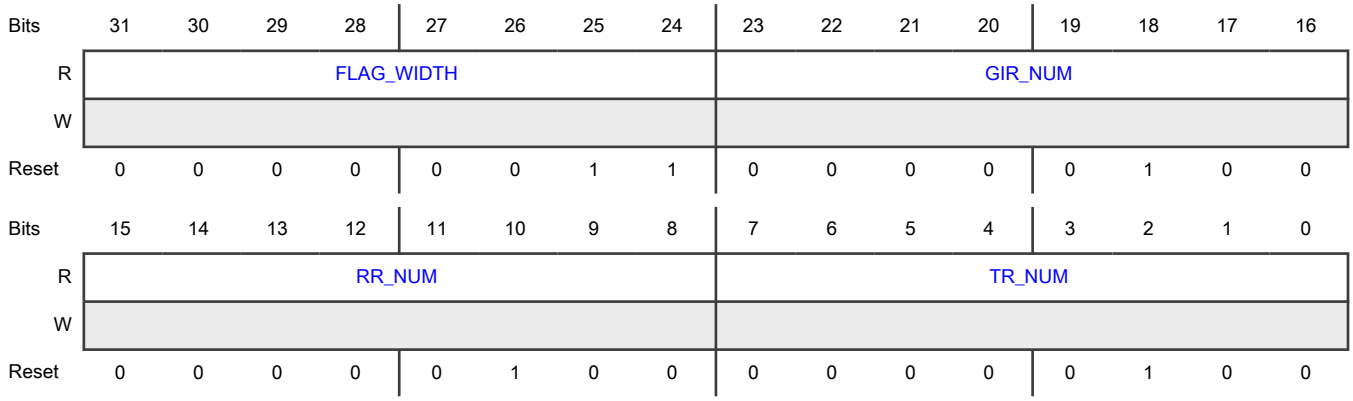
20.6.2.3 Parameter (PAR)

Defines the number of flags, transmit registers, receive registers, and general-purpose interrupt requests available for MU.

Offset

Register	Offset
PAR	4h

Diagram



Fields

Field	Description
31-24 FLAG_WIDTH	Flag Width Specifies the number of flag bits (3) in the Flag Control (FCR) and Flag Status (FSR) registers.

Table continues on the next page...

Table continued from the previous page...

Field	Description
23-16 GIR_NUM	General-Purpose Interrupt Request Number Specifies the number of general-purpose interrupt requests available (4).
15-8 RR_NUM	Receive Register Number Specifies the number of receive registers (4).
7-0 TR_NUM	Transmit Register Number Specifies the number of transmit registers (4).

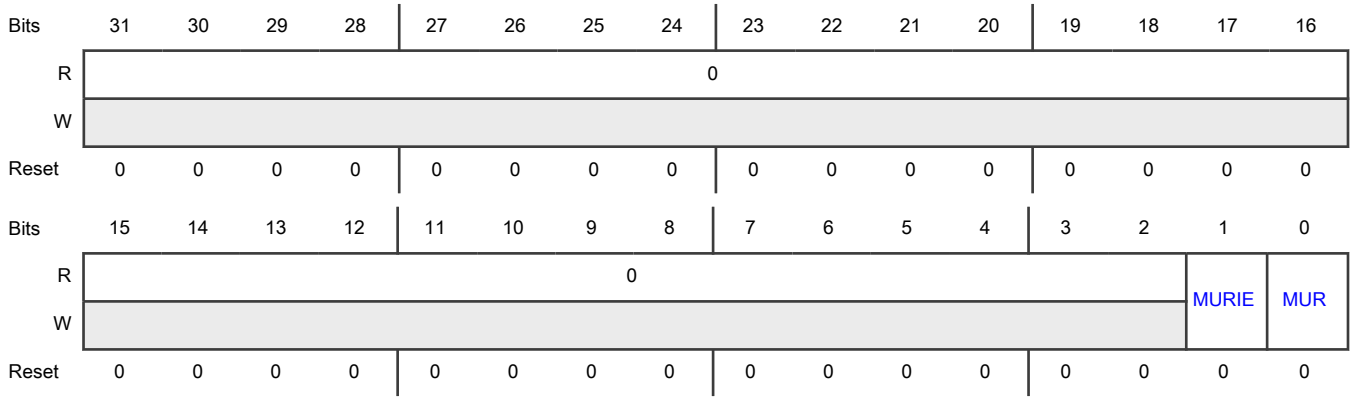
20.6.2.4 Control (CR)

Controls MU reset and reset interrupt enable.

Offset

Register	Offset
CR	8h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 MURIE	MUB Reset Interrupt Enable Enables the processor B-side MU reset interrupt request due to MU reset issued by MUA.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If the value of this field is 1, an MU reset interrupt request is issued to processor B when MUB_SR[MURIP] = 1. If the value of this field is 0, MU ignores the value of MURIP and issues no MU reset interrupt request. Only a system reset can reset this field. CR[MUR] cannot reset this field. 0 - Disable 1 - Enable
0 MUR	MU Reset Resets MU. Writing 1 to this field resets the MUB and MUA sides. All internal states are cleared. It forces all control and status registers to return to their default values (except HR, HRM in MUB/A_CCR0 registers; MURIE in MUB/A_CR registers; MURIP and MURS in MUB/A_SR registers). Before writing 1 to this field, interrupt processor A because writing 1 to this field may affect the ongoing processor A program. After writing 1 to this field, monitor the value of MUB_SR[MURS] to know when the reset sequence on the processor A-side has ended. This field always reads 0, and it becomes 0 during the MU reset sequence. 0 - Idle 1 - Reset

20.6.2.5 Status (SR)

Shows the status of MU resets and the status of pending events and requests.

Offset

Register	Offset
SR	Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	RFP	TEP	GIRP	FUP	EP	MURIP	MURS
W															W1C	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	u

Fields

Field	Description
31-8 —	Reserved
7 —	Reserved
6 RFP	MUB Receive Full Pending Indicates whether a receive full message is pending. This field becomes 1 when MUA writes to a TR*n* register to send data to MUB. After this field becomes 1, MU checks RSR[RF*n*] to determine whether the data in the Receive register is ready for MUB to read it. This field becomes 0 when all MUB RR*n* registers are read, or when MU is reset. 0 - Not pending; MUA is not writing to a Transmit register 1 - Pending; MUA is writing to a Transmit register
5 TEP	MUB Transmit Empty Pending Indicates whether a transmit empty message is pending. This field becomes 1 when any TCR[TIE*n*] field is 1 and TSR[TE*n*] flag is set. After this field becomes 1, MU checks TSR[TE*n*] to determine whether the data in the Transmit (TR*n*) register is ready for MUB to write to it. This field becomes 0 when write operations to all MUB Transmit (TR*n*) registers where TCR[TIE*n*] = 1 (transfer interrupt enabled) are completed, or when MU is reset. 0 - No MUB transmit empty event pending 1 - Pending; any TCR[TIEn] field is 1 and TSR[TEn] flag is set
4 GIRP	MUB General-Purpose Interrupt Pending Indicates that MUA has sent a general-purpose interrupt request. This field becomes 1 when the MUA side sends a general-purpose interrupt request to the MUB side. GSR[GIP*n*] identifies which general-purpose interrupt request is received. This field becomes 0 when all MUB_GSR[GIP*n*] fields are cleared, or when MU is reset. 0 - No request sent 1 - Request sent
3 FUP	MUB Flag Update Pending Indicates whether a flag update request is pending. MU generates this request when there is a change to the F*n*[2:0] bits of MUB_FCR. This field becomes 1 when the MUB side sends a flag update request to the MUA side. This field becomes 0 when MU acknowledges this flag update request internally (the flag is updated) from the MUA side, or during MU reset. No flag update changes are allowed when this field is 1. When FUP = 1, a write to the F*n*[2:0] bits of MUB_FCR does not generate a flag update event. The F*n*[2:0] bits do not change.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If SR[EP] = 1 (event pending), writing to MUB_FCR does not immediately cause this field to become 1. 0 - No pending update flags (initiated by MUB) 1 - Pending update flags (initiated by MUB)
2 EP	MUB Side Event Pending Indicates a pending side event when the MUB side sends an event update request to the MUA side. An event is any hardware message that the Status register on the MUA side reflects. For example, an event occurs when Transmit register 0 is the target of a write operation. During normal operations, the update mechanism for this field operates automatically. MU clears this field automatically when the event update acknowledgment is received, or when MU resets. To ensure that events are posted to MUA, verify that this field is 0. If it is 1, wait and continue to poll this field until it becomes 0. 0 - Not pending 1 - Pending
1 MURIP	MU Reset Interrupt Pending Flag Indicates whether processor A has issued an MU reset. This flag is set after processor A initiates an MU reset by setting MUB_CR[MUR]. If CR[MURIE] = 1, the processor B MU reset interrupt request is issued when processor A writes 1 to MUA_CR[MUR]. Clearing this flag also clears the MU reset interrupt request. Only a system reset can reset this flag. MU reset cannot reset this flag. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Reset not issued 1 - Reset issued When writing 0 - No effect 1 - Clear the flag
0 MURS	MUA and MUB Reset State Indicates the reset state of MUA and MUB. This field becomes 1 during any system reset or MU reset from the MUA or MUB side. This field becomes 0 when the reset sequence on both MUA and MUB sides ends. After issuing any of the aforementioned reset events, verify that this field is 0 before starting any access. 0 - Out of reset 1 - In reset

20.6.2.6 Core Control 0 (CCR0)

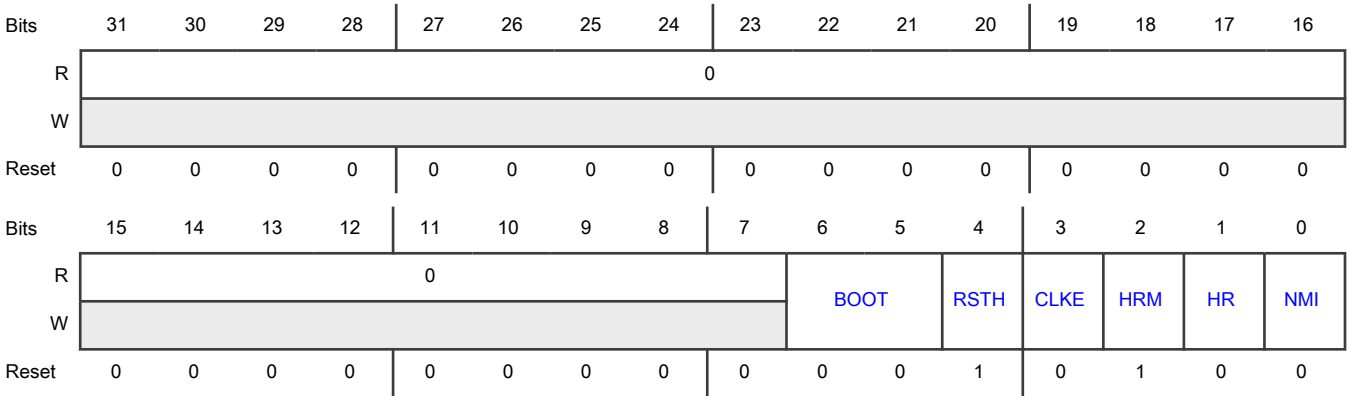
Allows MUB to control the processor on the MUA side.

Also allows control of the processor B hardware reset mask.

Offset

Register	Offset
CCR0	10h

Diagram



Fields

Field	Description
31-7 —	Reserved
6-5 BOOT	Target Processor A Boot Configuration Configures the boot source for processor A when it is the target core. See the chip-specific details for more information on boot options. 00 - Processor Boot option 0 01 - Processor Boot option 1 10 - Processor Boot option 2 11 - Processor Boot option 3
4 RSTH	Processor A Reset Hold Holds processor A in reset. **NOTE** Writing 1 to this field does not force processor A into reset. It only holds processor A if the processor resets itself. 0 - Release from reset

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Hold in reset
3 CLKE	Processor A Clock Enable Enables platform clock for Processor A to continue running when Processor A enters Stop mode until processor B also enters Stop mode. When this field is 0, the Processor A platform clock is gated when Processor A enters Stop mode. 0 - Disable 1 - Enable
2 HRM	Processor B Hardware Reset Mask Disables hardware reset requests to the other processor side. When this field is 1, MUA_CCR0[HR] is masked, and the hardware reset request to processor B is disabled. When this field is 0, MUA_CCR0[HR] is not masked, and the hardware reset to the processor B is enabled. 0 - Enable 1 - Disable
1 HR	Processor A Hardware Reset Initiates hardware reset. Write 1 to this field to start a hardware reset of processor A. Write 0 to this field to deassert the processor A hardware reset input. Writing 1 to this field does not cause an internal MU reset. It is routed outside MU to the processor A reset logic. Processor A can mask the HR reset issued by processor B to processor A (according to the value of MUA_CCR0[HRM]). This field does not return to the reset value during MU software reset (MUR). 0 - Deassert hardware reset to processor A. 1 - Assert hardware reset to processor A.
0 NMI	MUA Nonmaskable Interrupt Request Indicates whether Processor B has issued a nonmaskable interrupt to Processor A. When this field becomes 1, it initiates a nonmaskable interrupt to processor A. This field becomes 0 after 1 is written to MUA_CSSR0[NMIC] to clear that field. After this field becomes 0, MUB can initiate another nonmaskable interrupt to MUA. This field is cleared when MU resets. 0 - Nonmaskable interrupt not issued 1 - Nonmaskable interrupt issued

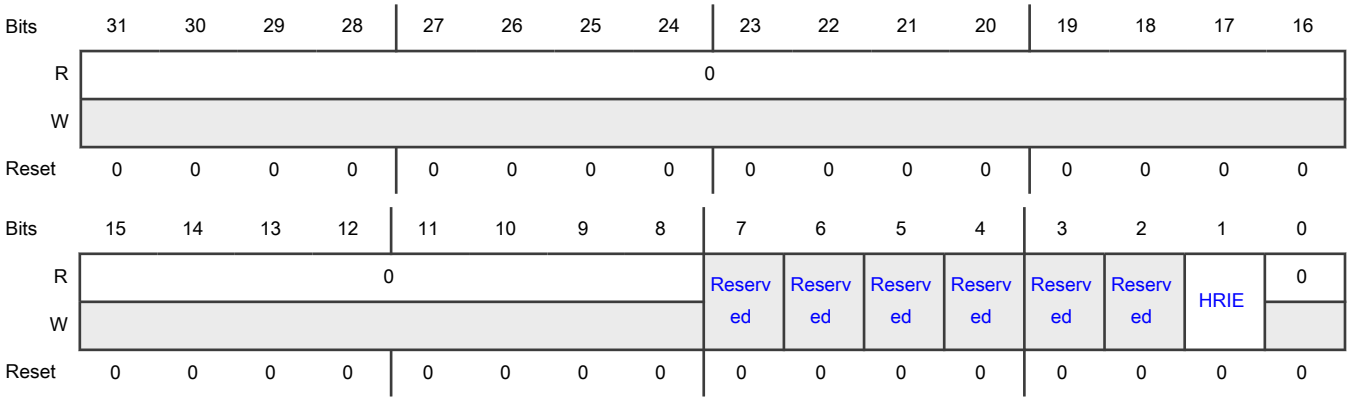
20.6.2.7 Core Interrupt Enable 0 (CIER0)

Controls interrupt enables.

Offset

Register	Offset
CIER0	14h

Diagram



Fields

Field	Description
31-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 HRIE	Processor B Hardware Reset Interrupt Enable Enables Processor B Hardware Reset Interrupt request due to Processor A issuing a hardware reset by writing MUA_CCR0[HR] to Processor B.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When the value of this field is 1, MU issues an MU reset interrupt request to Processor B. When the value of this field is 0, MU issues no MU reset interrupt. 0 - Disable 1 - Enable
0 —	Reserved

20.6.2.8 Core Sticky Status 0 (CSSR0)

Shows the status of interrupts pending (W1C).

The reset value is chip-specific.

Offset

Register	Offset
CSSR0	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0	0	0	0	0	HRIP	NMIC
W															W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	u	u

Fields

Field	Description
31-8 —	Reserved
7 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 HRIP	Processor B Hardware Reset Interrupt Pending Flag Indicates whether Processor B had been reset via MUA_CCR0[HR]. This flag is set when Processor A writes 1 to MUA_CCR0[HR] and MUB_CCR0[HRM] = 0. 0 - No hardware reset initiated 1 - Hardware reset initiated
0 NMIC	Processor B Nonmaskable Interrupt Clear Clears the nonmaskable interrupt (NMI) request from the MUA side. The MUB-side NMI service routine uses this field. Writing 1 to this field clears MUA_CCR0[NMI], deasserting the NMI request and enabling MUA_CCR0[NMI] to trigger another NMI request. This field always reads as 0, so you cannot poll it. You can only use this field as part of the NMI service routine, in which you must write 1 to this field only once. This field is cleared when MU resets. 0 - Default 1 - Clear MUA_CCR0[NMI]

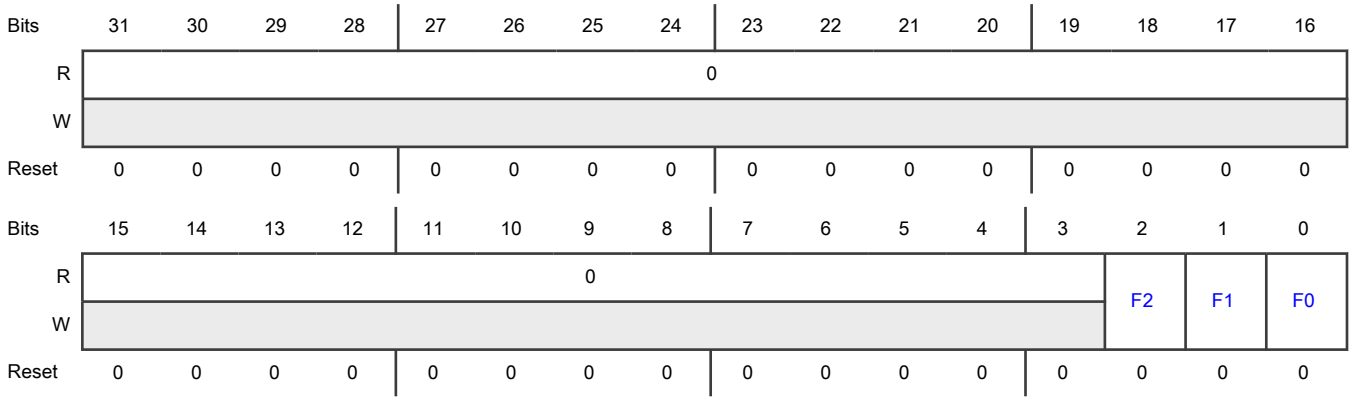
20.6.2.9 Flag Control (FCR)

Configures MUA_FSR[F*n*] flags.

Offset

Register	Offset
FCR	100h

Diagram



Fields

Field	Description
31-3 —	Reserved
2-0 Fn	MUB to MUA Flag Configures MUA_FSR[F*n*] flags, where *n* = 0 to 2. F*n* configures the corresponding MUA_FSR[F*n*] flag. F*n* becomes 0 when MU resets. 0 - Clear MUA_FSR[Fn] 1 - Set MUA_FSR[Fn]

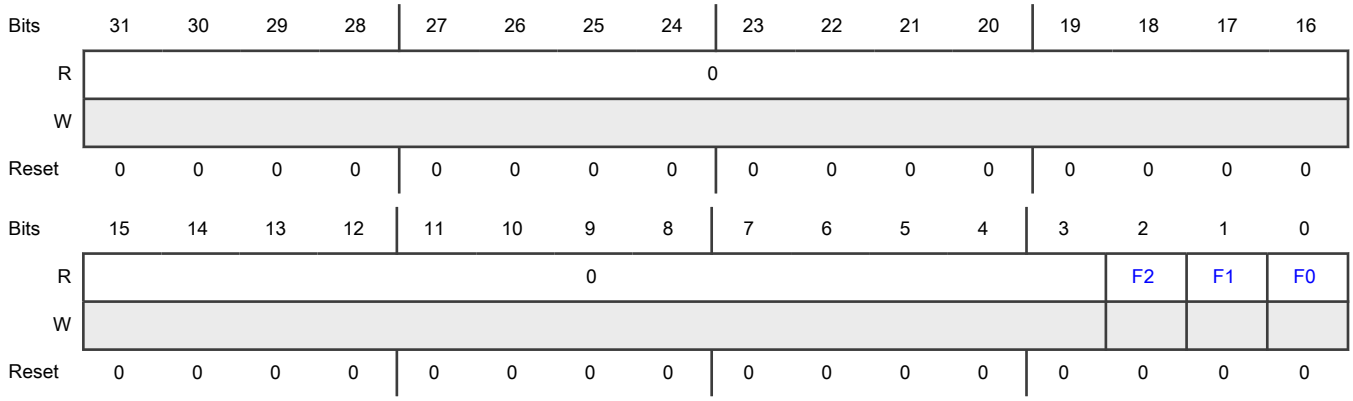
20.6.2.10 Flag Status (FSR)

Contains flags configured by the values written to MUA_FCR[F*n*].

Offset

Register	Offset
FSR	104h

Diagram



Fields

Field	Description
31-3 —	Reserved
2-0 Fn	MUA to MUB-Side Flag Contains flags configured by the values written to MUA_FCR[F*n*], where *n* = 0 to 2. F*n* is the MUB-side flag configured by the values written to MUA_FCR[F*n*]. When MUA_FCR[F*n*] is written to, the write event updates MUB_FSR[F*n*], after the event update latency. 0 - MUA_FCR[Fn] = 0 1 - MUA_FCR[Fn] = 1

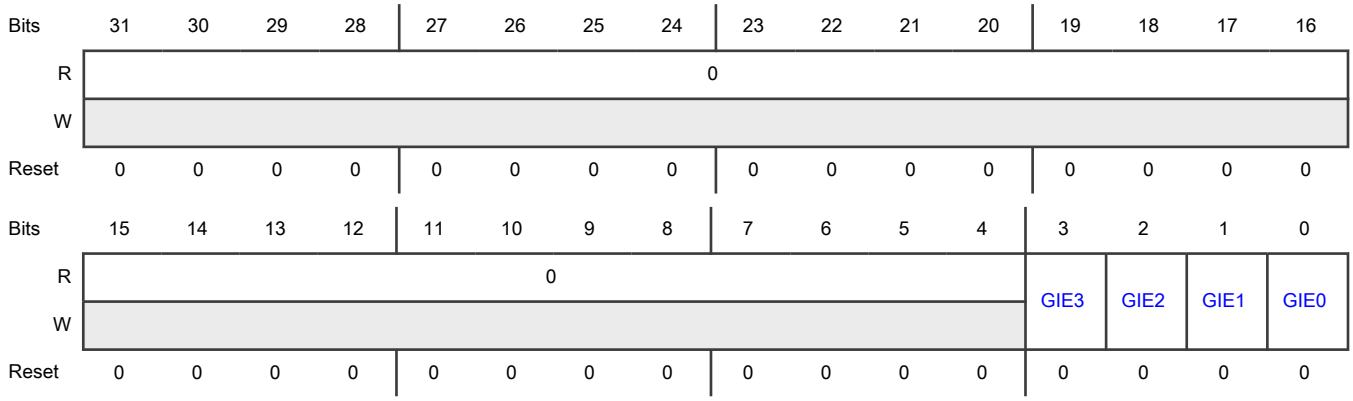
20.6.2.11 General-Purpose Interrupt Enable (GIER)

Contains the MUB general-purpose interrupt enable fields.

Offset

Register	Offset
GIER	110h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 GIEn	MUB General-purpose Interrupt Enable Enables general-purpose interrupt. There are 4 general-purpose interrupts (*n* = 0 to 3). When GIE*n* = 1, a general-purpose interrupt *n* request is issued to processor B when MUB GSR[GIP*n*] = 1. If GIE*n* = 0, the general-purpose interrupt request pending does not trigger the general-purpose interrupt. GIE*n* becomes 0 when MU resets. 0 - Disable 1 - Enable

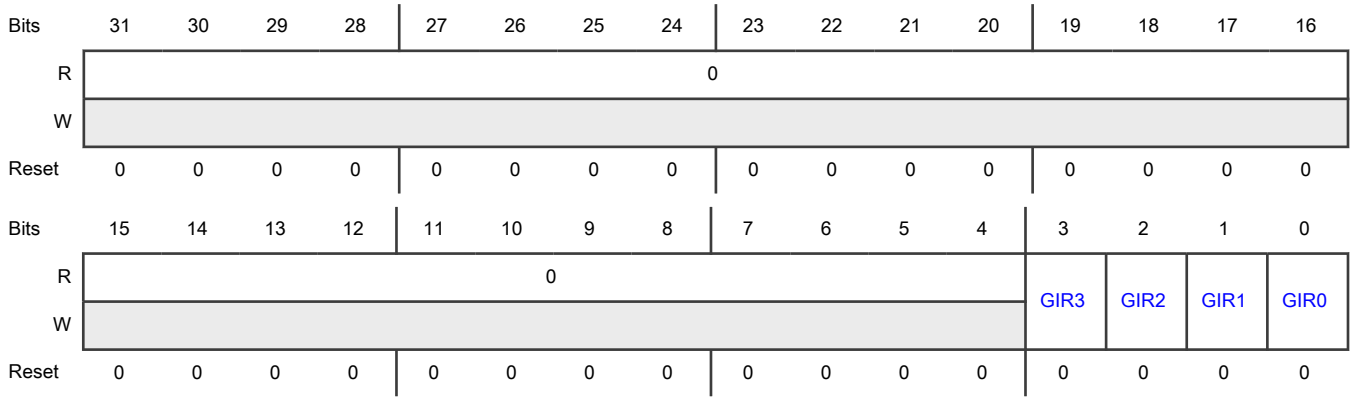
20.6.2.12 General-Purpose Control (GCR)

Contains the MUB general-purpose interrupt request fields.

Offset

Register	Offset
GCR	114h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 GIRn	MUB General-Purpose Interrupt Request Specifies whether general-purpose interrupts are requested to MUA. There are 4 general-purpose interrupts (*n* = 0 to 3). Writing 1 to GIR*n* sets MUA_GSR[GIP*n*]. If MUA_GIER[GIE*n*] = 1, a general-purpose interrupt request is triggered on processor A. This field becomes 0 when MUA_GSR[GIP*n*] is cleared. This clearing informs MUB that the interrupt is accepted (cleared by software). To ensure proper operations, verify that GIR*n* is 0 (no pending interrupt) before writing 1 to it. This field becomes 0 when MU resets. 0 - Not requested 1 - Requested

20.6.2.13 General-purpose Status (GSR)

Contains the status of the MUB general-purpose interrupt pending requests.

Offset

Register	Offset
GSR	118h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												GIP3	GIP2	GIP1	GIP0
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved
3-0 GIPn	MUB General-Purpose Interrupt Request Pending Indicates whether a general-purpose interrupt request is pending. There are 4 general-purpose interrupts (*n* = 0 to 3). GIP*n* informs MUB that MUA_GCR[GIR*n*] changed from 0 to 1. If MUB_GIER[GIE*n*] = 1, a general-purpose interrupt request is issued to processor B. GIP*n* is cleared when MU resets. After GIP*n* is cleared, if MUB_GIER[GIE*n*] = 1, the general-purpose interrupt request is cleared on the MUB side. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not pending 1 - Pending When writing 0 - No effect 1 - Clear the flag

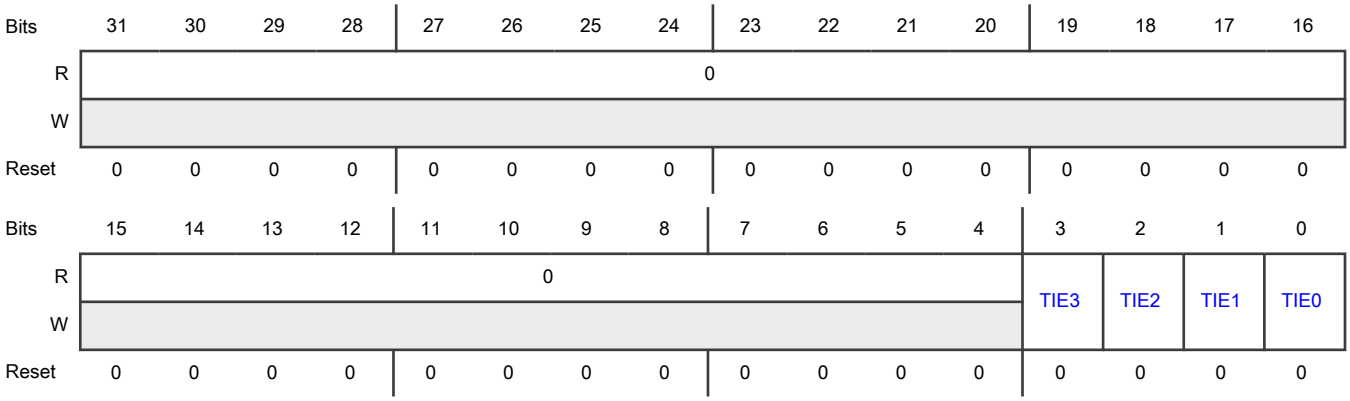
20.6.2.14 Transmit Control (TCR)

Contains the MUB transmit interrupt enable fields.

Offset

Register	Offset
TCR	120h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 TIEn	MUB Transmit Interrupt Enable Enables MUB transmit interrupt *n*, where *n* = 0 to 3. If this field is 1, an MUB transmit interrupt *n* request is issued when MUB_TSR[TE*n*] is set. If this field is 0, MU ignores the value of MUB_TSR[TE*n*], and no MUB transmit interrupt *n* request is issued. 0 - Disable 1 - Enable

20.6.2.15 Transmit Status (TSR)

Indicates whether the MUB transmit registers are empty.

Offset

Register	Offset
TSR	124h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												TE3	TE2	TE1	TE0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Fields

Field	Description
31-4 —	Reserved
3-0 TEn	MUB Transmit Empty Indicates whether MUB Transmit (TR*n*) register is empty, where *n* = 0 to 3. This field becomes 1 after the MUA_RR*n* register is read on the MUA side. When TE*n* = 1, it indicates to the MUB side that the MUB_TR*n* register is ready to be written on the MUB side. If MUB_TCR[TIE*n*] = 1, a transmit *n* interrupt is issued on the MUB side. This field becomes 0 after the MUB_TR*n* register is written to on the MUB side. After this field becomes 0, if MUB_TCR[TIE*n*] = 1, the transmit *n* interrupt request is cleared on the MUB side. This field becomes 1 when MU resets. 0 - Not empty 1 - Empty

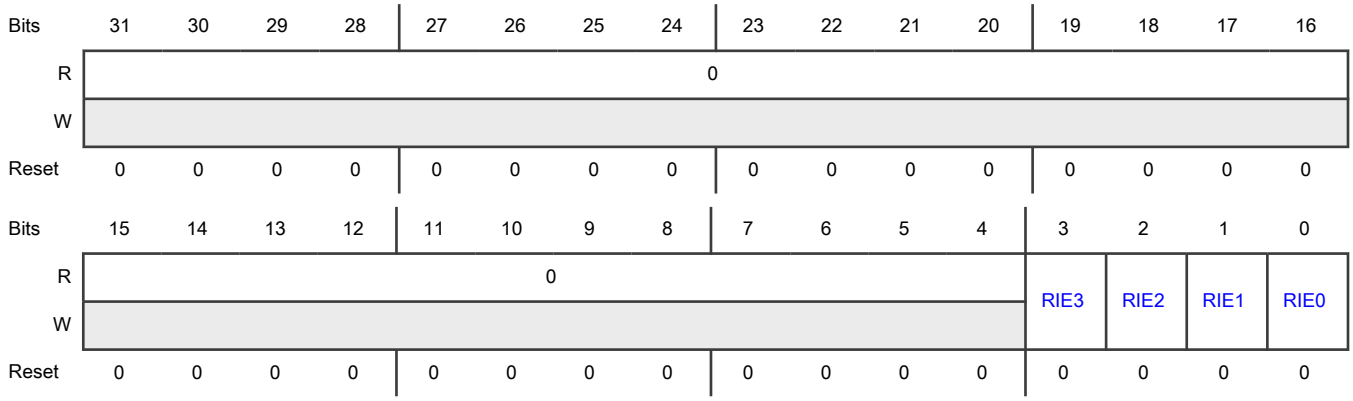
20.6.2.16 Receive Control (RCR)

Contains the MUB receive interrupt enables.

Offset

Register	Offset
RCR	128h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 RIEn	MUB Receive Interrupt Enable Enables MUB receive interrupt *n*, where *n* = 0 to 3. If this field is 1, an MUB receive interrupt *n* request is issued when MUB_RSR[RF*n*] is set. If this field is 0, MU ignores the value of MUB_RSR[RF*n*], and no MUB receive interrupt request is issued. 0 - Disable 1 - Enable

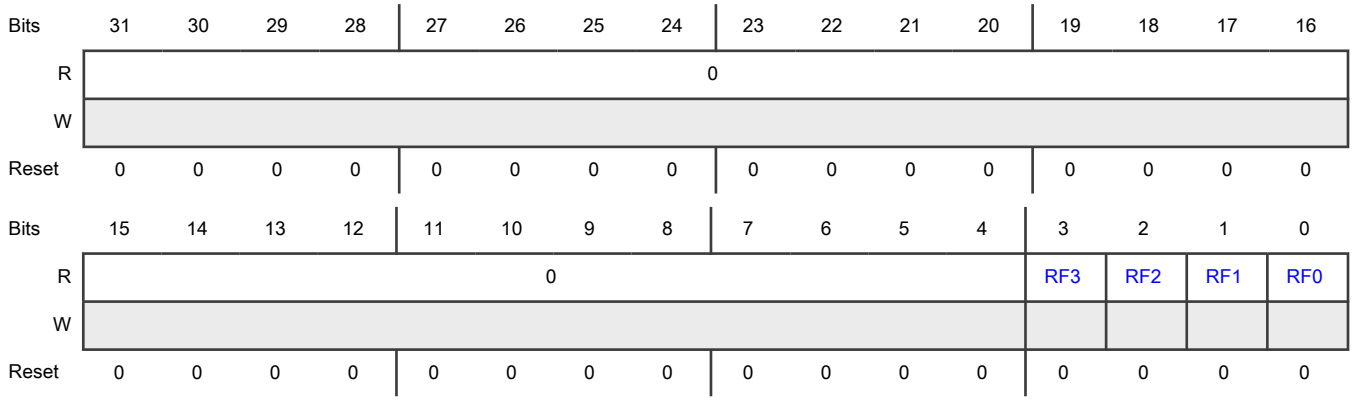
20.6.2.17 Receive Status (RSR)

Indicates whether the MUB receive registers are full.

Offset

Register	Offset
RSR	12Ch

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 RFn	MUB Receive Register Full Indicates whether MUB Receive register (RR*n*) is full, where *n* = 0 to 3. RF*n* becomes 1 when the MUA_TR*n* register is written to on the MUA side. When RF*n* is 1, it indicates to the MUB side that new data in the MUB_RR*n* register is ready for MUB to read it. If MUB_RCR[RIE*n*] = 1, a receive *n* interrupt is issued on the MUB side. RF*n* becomes 0 when the MUB_RR*n* register is read, or when MU is reset. After RF*n* becomes 0, if MUB_RCR[RIE*n*] = 1, the receive *n* interrupt request is cleared on the MUB side. 0 - Not full 1 - Full

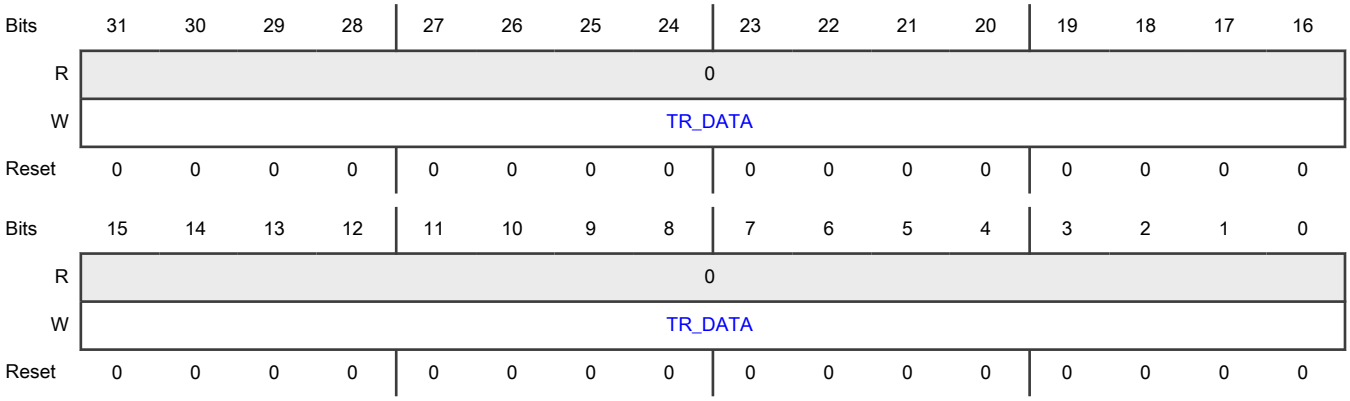
20.6.2.18 Transmit (TR0 - TR3)

Contains MUB transmit data.

Offset

Register	Offset
TR0	200h
TR1	204h
TR2	208h
TR3	20Ch

Diagram



Fields

Field	Description
31-0 TR_DATA	MUB Transmit Data Contains MUB transmit data. MUA_RR*n* reflects the data written to this register. The TR*n* and RR*n* registers are not double-buffered. Writing to MUB_TR*n* overrides the data readable in the MUA_RR*n* register. A write to the Transmit register clears MUB_TSR[TE*n*] on the transmitter side, and sets MUA_RSR[RF*n*] on the receiver side. You can write to this register only when MUB_TSR[TE*n*] = 1. Reading this register returns all zeroes.

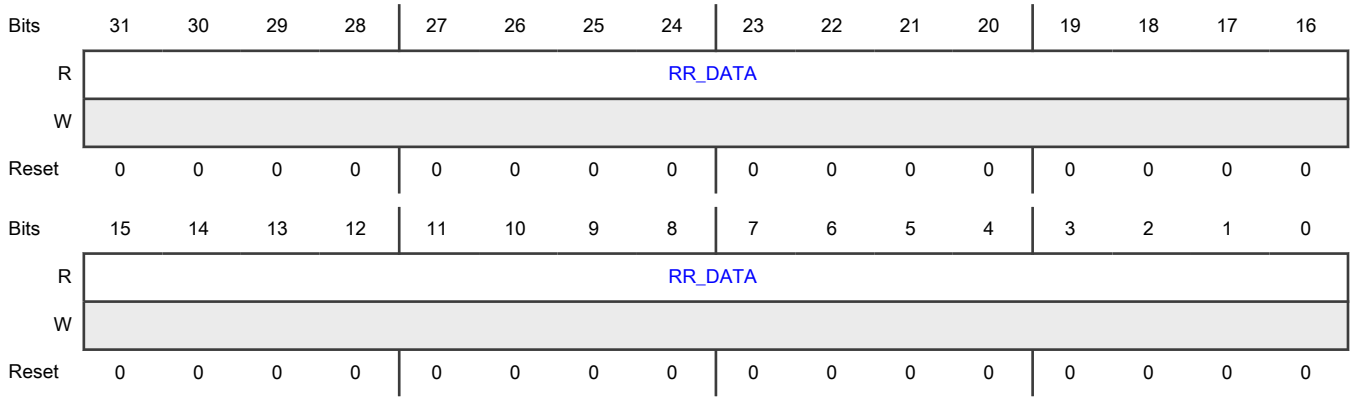
20.6.2.19 Receive (RR0 - RR3)

Contains MUB receive data.

Offset

Register	Offset
RR0	280h
RR1	284h
RR2	288h
RR3	28Ch

Diagram



Fields

Field	Description
31-0 RR_DATA	MUB Receive Data Reflects the data written to MUA TR*n*. Reading this register clears MUB_RSR[RF*n*] on the receiver side, and sets MUA_TSR[TE*n*] on the transmitter side. You can read this register only when MUB_RSR[RF*n*] = 1. Reading it before MUB_RSR[RF*n*] becomes 1 may result in reading incorrect data. Poll MUB_RSR[RF*n*] to confirm it is set before reading RR*n*. Writing to this register generates an error response to MUB.

20.7 Glossary

EP	Event Pending
GIR	General-purpose Interrupt Request
GIP	General-purpose Interrupt Pending
MUR	Messaging Unit Reset
NMI	Non-Maskable Interrupt
RF	Receiver Full
RFP	Receive Full Pending
TE	Transmitter Empty
TEP	Transmit Empty Pending
MUA	Messaging Unit A
MUB	Messaging Unit B

Chapter 21
Semaphores2 (SEMA42)

21.1 Overview

SEMA42 is a memory-mapped module that provides robust hardware support needed in multi-core systems for implementing semaphores and provides a simple mechanism to achieve "lock and unlock" operations via a single - write access. The hardware semaphore module provides hardware-enforced gates as well as other useful system functions related to the gating mechanisms.

NOTE

Terminology in this chapter has been updated as follows:

Table 98. Updated terms

Updated term	Deprecated term
Initiator	Master
Target	Slave
Stop	Abort

21.1.1 Block diagram

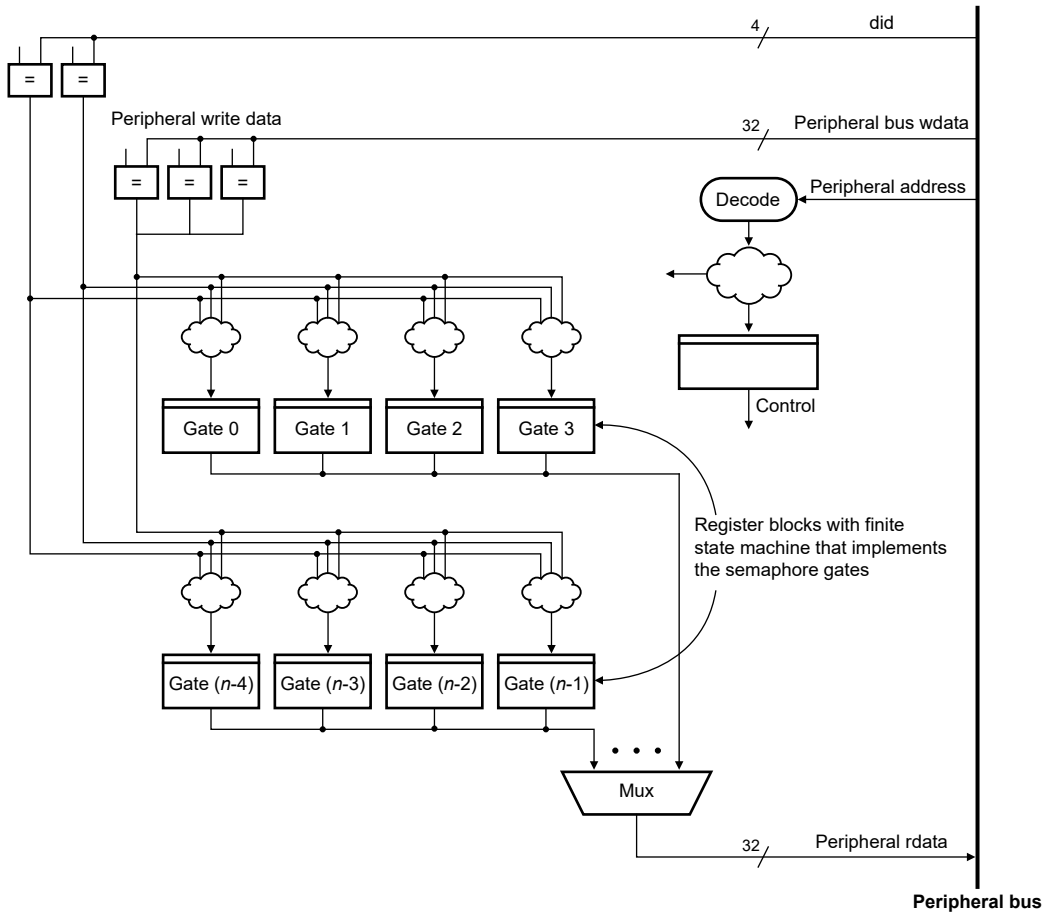


Figure 39. SEMA42 block diagram

21.1.2 Features

SEMA42 implements hardware-enforced semaphores as an IPS-mapped target peripheral device. The feature set includes:

- Support for 64 hardware-enforced gates in a multi-domain configuration that supports up to 15 domains.
- Each hardware gate appears as a 16-state, 4-bit state machine.
- Support for secure reset mechanisms to clear the contents of individual gates, as well as a clear-all capability.
- Memory-mapped target peripheral that offers programming-model accesses.

21.2 Functional description

The intent of the hardware gate implementation is to specify a protocol where the locking domain must unlock the gate. However, some systems may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset. To support this special gate reset requirement, SEMA42 implements a secure reset mechanism that allows you to initialize a hardware gate (or all the gates) by following a specific dual-write access pattern. The secure-gate reset:

- Uses a technique similar to that required for the servicing of a software watchdog timer
- Requires two consecutive writes with pre-defined data patterns from the same domain

You must do this to force the clearing of the specified gate(s). The required access pattern as follows:

1. A domain performs a 16-bit write to the RSTGT memory location. The most significant byte (RSTGT_W[RSTGDP]) must be E2h. The value of least significant byte is irrelevant for this reference and can be anything.
2. The same domain then performs a second 16-bit write to the RSTGT location. For this write, the upper byte (RSTGT_W[RSTGDP]) is the logical complement of the first data pattern (1Dh) and the lower byte (RSTGT_W[RSTGTN]) specifies the gate(s) to be reset. This gate field can specify a single gate or all gates to be cleared. If the same domain writes incorrect data on the second access or another domain performs the second write access, SEMA42 stops the special gate reset sequence and does not assert an error signal.
3. Reads of the RSTGT location return information on the 2-bit reset state machine (RSTGT_R[RSTGSM]) that implements these functions:
   - The domain performing the reset (RSTGT_R[RSTGMS])
   - The last-cleared gate number(s) (RSTGT_R[RSTGTN])

   Reads of the RSTGT register do not affect the secure-reset finite state machine in any manner.

21.2.1 Multi-core programming: software gates

Multi-processor systems require a function that can be used to safely and easily provide a locking mechanism for system software to control access to shared data structures, shared hardware resources, and so on. The software uses the gating mechanisms to serialize (and synchronize) accesses to shared data and/or resources to prevent race conditions and preserve memory coherency between different processes and domains.

Consider the following description of a typical use-case: dm*X* enters a section of code, where shared data values are to be updated. The domain must first acquire a semaphore. Think of this as the locking (or closing) of a software gate. After the gate locks, a properly-architected software system does not allow other processes (or domains) to execute the same code segment or modify the shared data structure protected by the gate. In other words, the system locks out other processes/domains. Many software implementations include a spin-wait loop within the lock function until the gate locks. After domain X obtains the lock, domain X continues execution and updates the data values protected by the particular lock. After domain X completes the updates, it unlocks (or opens) the software gate, allowing other processes/domains access to the updated data values.

A correctly-implemented system solution must follow these important rules:

- A gate variable must protect all writes to shared data values or shared hardware resources.
- After a domain locks a gate, the system must block other processes/domains from accessing the shared data or resources. Software conventions enforce this.
- The domain that locks a particular gate is the only domain that can open (unlock) that gate.

Information in the hardware gate identifying the locking domain can be extremely useful for system-level debugging.

21.2.2 64 Hardware-enforced gates

Gates appear as a 64-entry byte-size array with read and write accesses.

Domains lock gates by writing "domainID_number+1" to the appropriate gate and must read back the gate value to verify that the lock operation succeeded.

After the gate locks, the locking domain unlocks the gate by writing zeroes.

- 16-state implementation

  If gate = 0h, then state = unlocked

  if gate = 1h, then state = locked by domain (initiator) 0

  if gate = 2h, then state = locked by domain (initiator) 1

  …

  if gate = Fh, then state = locked by domain (initiator) 14

SEMA42 uses the logical domain number and the specified data patterns as reference attributes to validate all write operation.

After a gate locks, the locking domain must unlock the gate by writing zeroes.

21.2.3 State machine of the GATE*n* registers

This section describes more about the SEMA42 functional operation and specific details of the state machines of the GATE*n* registers.

As described previously, each of the GATE*n* registers implements a 4-bit, 16-state machine. The following figure shows a simplified diagram of the state transitions for each gate.

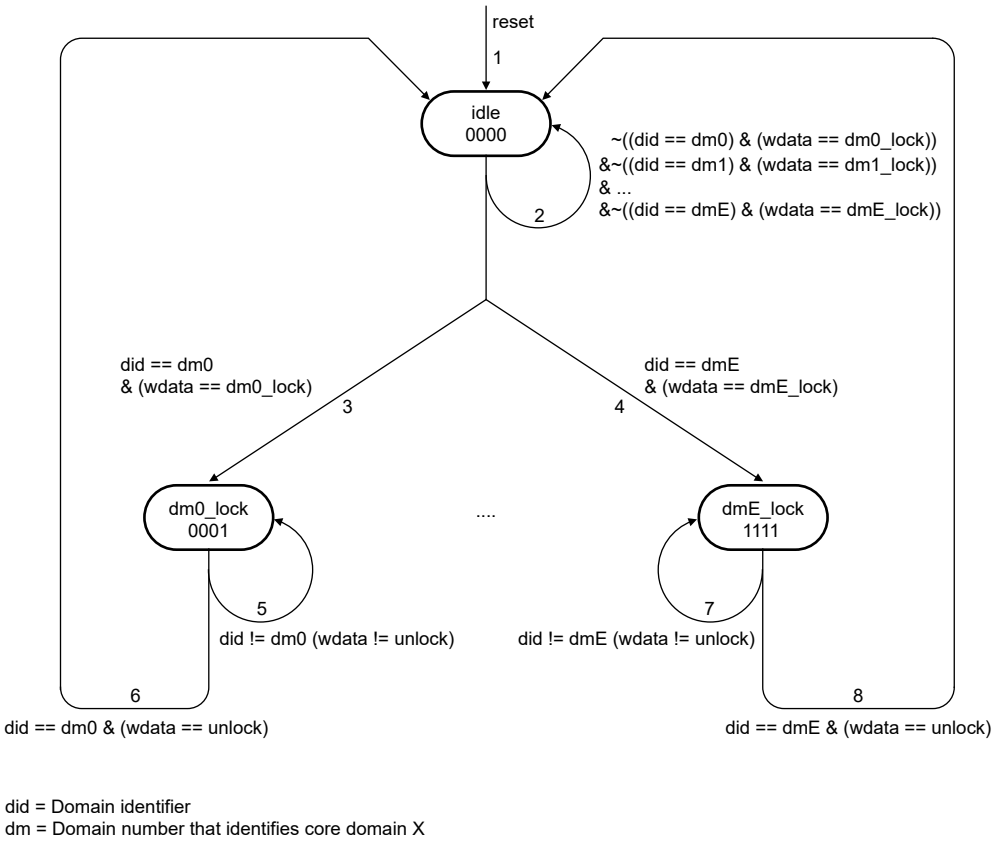


Figure 40. GATE*n* state machine

In the figure above, "dmE" represents domain 14 (E in hexadecimal). The platform passes the domain number to SEMA42.

The following table defines the GATE*n* state transitions.

Table 99. GATE*n* state transitions

Current state	Next state	Transition	Description
–	idle	1	Any reset, whether a system reset or a software-initiated gate reset, unconditionally forces the gate into the idle state.
idle	idle	2	Unless a write of the appropriate lock value from the corresponding domain occurs, the gate remains in the idle state.
idle	dm0_lock	3	When domain 0h initiates a write of the dm0_lock data value, the gate transitions into the dm0_lock state.

Table continues on the next page...

Table 99. GATE*n* state transitions (continued)

Current state	Next state	Transition	Description
idle	dmE_lock	4	When domain Eh initiates a write of the dmE_lock value, the gate transitions into the dmE_lock state.
dm0_lock	dm0_lock	5	When in this state, the gate remains here if any attempted write is not from domain 0h with the unlock data value.
dm0_lock	idle	6	The gate returns to the idle (unlocked) state after a write from domain 0h with the unlock data value occurs.
dmE_lock	dmE_lock	7	When in this state, the gate remains here if any attempted write is not from domain Eh with the unlock data value.
dmE_lock	idle	8	The gate returns to the idle (unlocked) state after a write from domain Eh with the unlock data value occurs.

SEMA42 uses these gate data values:

- The lock data value is (domain number) + 1.
- The unlock data value is 00h.

21.2.4 Clocking

Type of clock	Description
Module clock (clk)	Functions the module and controls the access to the registers.

21.2.5 Interrupts

This module has no interrupts.

21.3 External signals

This module has no external signals.

21.4 Initialization

This module does not require initialization.

21.5 Memory map/register definition

You can access these registers only in Supervisor mode. User accesses terminate with an error.

21.5.1 SEMA42 register descriptions

21.5.1.1 SEMA42 memory map

SEMA42_1 base address: 4426_0000h

SEMA42_2 base address: 4245_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0 - 3F	Gate (GATE0 - GATE63) [1]	8	RW	00
42	Reset Gate Read (RSTGT_R)	16	R	0000
42	Reset Gate Write (RSTGT_W)	16	W	See section

1. In this array, the index and offset values of the registers do not increment in direct alignment. For details, see the register description.

21.5.1.2 Gate (GATE0 - GATE63)

Implements each semaphore gate in a 4-bit finite state machine, right-justified in a byte data structure. The hardware uses the logical domain-identifier number in conjunction with the data patterns to validate all attempted write operations. Only domain initiators can modify the gate registers. After a gate locks, only the locking domain must open (unlock) the gate.

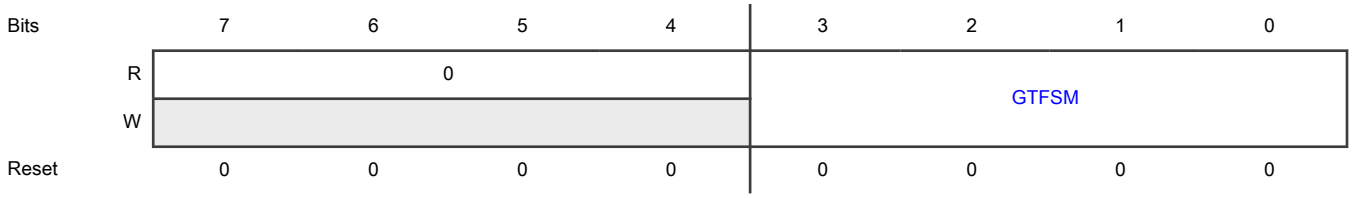
You can read multiple gate values in a single access. However, you can update only a single gate at a time, via a write operation. If you attempt to write a byte-wide value that is neither the unlock value (00h) nor the appropriate lock value (domainID_number + 1), SEMA42 considers this as "no operation" and does not change any gate state. Attempts to write multiple gates in a single-aligned access with a size larger than 8 bits (byte) generate an error termination and do not allow any gate state changes.

Offset

For n = 0 to 63:

Register	Offset
GATEn	0h + (n + 3 - 2 × (n mod 4))

Diagram



Fields

Field	Description
7-4 —	Reserved
3-0 GTFSM	Gate Finite State Machine Indicates the state of the gate for the last domain that locked the gate. This can be useful during system debug. The hardware gate has a 16-state implementation, defined as:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0000 - The gate is unlocked (free). 0001 - Domain 0 locked the gate. 0010 - Domain 1 locked the gate. 0011 - Domain 2 locked the gate. 0100 - Domain 3 locked the gate. 0101 - Domain 4 locked the gate. 0110 - Domain 5 locked the gate. 0111 - Domain 6 locked the gate. 1000 - Domain 7 locked the gate. 1001 - Domain 8 locked the gate. 1010 - Domain 9 locked the gate. 1011 - Domain 10 locked the gate. 1100 - Domain 11 locked the gate. 1101 - Domain 12 locked the gate. 1110 - Domain 13 locked the gate. 1111 - Domain 14 locked the gate.

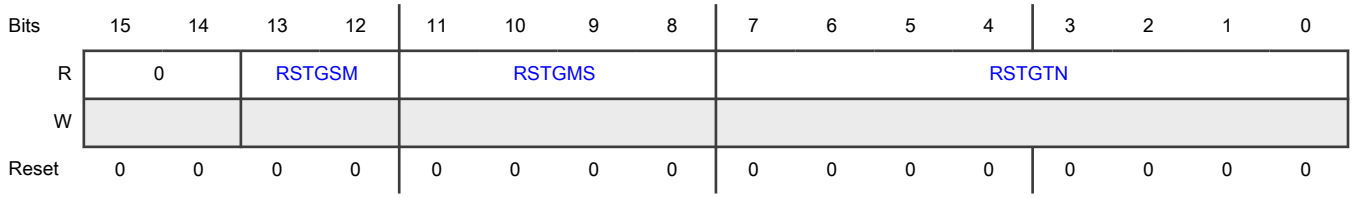
21.5.1.3 Reset Gate Read (RSTGT_R)

Describes the specific hardware gate to be reset and records the logic number of domain. Reset Gate Write (RSTGT_W) also describe the same register showing the fields when you write it.

Offset

Register	Offset
RSTGT_R	42h

Diagram



Fields

Field	Description
15-14 —	Reserved
13-12 RSTGSM	Reset Gate Finite State Machine Indicates the encoded state machine value when you read the register. RSTGSM = 10b is valid for only a single machine cycle, so a read can never return this value. SEMA42 maintains the reset state machine in a 2-bit, 3-state implementation, defined as follows: 00 - Idle, waiting for the first data pattern write. 01 - Waiting for the second data pattern write 10 - The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. 11 - This state encoding is never used and therefore reserved.
11-8 RSTGMS	Reset Gate Domain Records the logical number of the domain performing the gate reset function. To succeed, this function requires that the same domain initiate the two consecutive writes to this register. SEMA42 updates the field each time a write to this register occurs.
7-0 RSTGTN	Reset Gate Number Specifies the specific hardware gate to be reset. The second write updates this field. If RSTGTN < 64, SEMA42 resets the single gate defined by RSTGTN. Otherwise, SEMA42 resets all the gates.

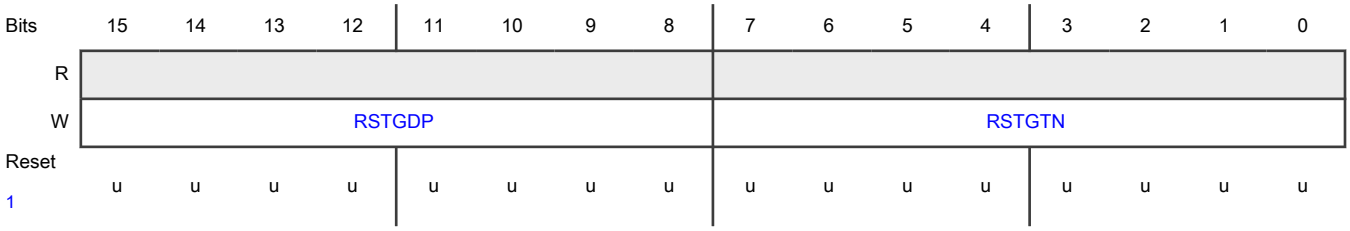
21.5.1.4 Reset Gate Write (RSTGT_W)

Specifies the hardware gate to reset when data patterns are specified.

Offset

Register	Offset
RSTGT_W	42h

Diagram



1. Reset value is not applicable for writes.

Fields

Field	Description
15-8 RSTGDP	Reset Gate Data Pattern You access this field with the specified data patterns on the two consecutive writes to enable the gate-reset mechanism. For the first write, RSTGDP must be E2h. For the second write, RSTGDP must be 1Dh.
7-0 RSTGTN	Reset Gate Number Specifies the specific hardware gate to be reset. The second write updates this field. If RSTGTN < 64, SEMA42 resets the single gate defined by RSTGTN. Otherwise, SEMA42 resets all the gates.

21.6 Glossary

IPS Protocol used for peripheral accesses to the programming model

dm*X* Domain X

Chapter 22
Trusted Resource Domain Controller (TRDC)

22.1 Chip-specific TRDC information

Table 100. Related information in i.MX 93 Applications Processor Reference Manual

Topic	Related module
CM33	Arm Cortex CM33 Platform (CM33)
A55	Arm Cortex CA55 Platform (CA55)
Full description	Trusted Resource Domain Controller (TRDC)
System memory map	Memory Maps
Clocking	Clock Controller Module (CCM)
Signal multiplexing	Port control

NOTE

A core complex should not rely on TRDC and should use the internal MMU for memory management. In addition, initiators like Cortex-A55, that implement the DSU architecture, do support AxPROT internally, but do not drive AxPROT[2] and AxPROT[0] on the system bus. For Cortex-A55, after boot, it is recommended to use the internal MMU for access control.

22.1.1 Module instances

This device has six instances of the TRDC module:

- TRDC AONMIX
- TRDC WAKEUPMIX
- TRDC MEGAMIX
- TRDC NICMIX
- TRDC MEDIAMIX
- TRDC HSIOMIX

NOTE

TRDC settings will be lost if the mix power is off.

NOTE

This chip does not support Flash Logical Window (FLW). The registers for FLW exists but have no effect on device operation.

NOTE

TZ-M is enabled on this chip. TRDC components - MGR, MDAC, MRC and MBC cannot be accessed in a non-secure mode.

22.1.2 Chip-specific TRDC information

Table 101. TRDC MGR Configuration

Location	Parameter Name	Configured Value	AIPS Memory Allocation (KB)	Comments
AONMIX	Number of DIDs	16		Support 16 domains
	Number of MDACs	5		Support 5 MDACs
	Number of MRCs	1	4	Support 1 Region Checkers
	Number of MBCs	2	16	Support 2 Block Checkers
	Total MRC, MBC AIPS memory (KB)		20	
	Number 64K AIPS slots for MRC,MBC		1	
WAKEUPMIX	Number of DIDs	16		Support 16 domains
	Number of MDACs	5		Support 5 MDACs
	Number of MRCs	2	8	Support 2 Region Checkers
	Number of MBCs	2	16	Support 2 Block Checkers
	Total MRC, MBC AIPS memory (KB)		24	
	Number 64K AIPS slots for MRC,MBC		1	
MEGAMIX	Number of DIDs	16		Support 16 domains
	Number of MDACs	7		Support 7 MDACs
	Number of MRCs	0	0	Support 0 Region Checkers
	Number of MBCs	0	0	Support 0 Block Checkers
	Total MRC, MBC AIPS memory (KB)		0	
	Number 64K AIPS slots for MRC,MBC		0	
NICMIX	Number of DIDs	16		Support 16 domains
	Number of MDACs	4		Support 4 MDACs, remove the mdac for gic600 axi master
	Number of MRCs	1	4	Support 1 Region Checker

Table continues on the next page...

Table 101. TRDC MGR Configuration (continued)

	Number of MBCs	4	32	Support 4 Block Checkers
	Total MRC, MBC AIPS memory (KB)		36	
	Number 64K AIPS slots for MRC,MBC		1	
HSIOMIX	Number of DIDs	16		Support 16 domains
	Number of MDACs	2		Support 2 MDACs
	Number of MRCs	0	0	Support 0 Region Checkers
	Number of MBCs	0	0	Support 0 Block Checkers
	Total MRC, MBC AIPS memory (KB)		0	
	Number 64K AIPS slots for MRC,MBC		0	
MEDIAMIX	Number of DIDs	16		Support 16 domains
	Number of MDACs	4		Support 4 MDACs
	Number of MRCs	0	0	Support 0 Region Checkers
	Number of MBCs	0	0	Support 0 Block Checkers
	Total MRC, MBC AIPS memory (KB)		0	
	Number 64K AIPS slots for MRC,MBC		0	

Table 102. Default DID Assignments for TRDC components

MDAC DEFAULT_DID	TRDC_BDB_DID_BIT[x] 1=Deny, 0=Allow, N/A=Not Used	
0	bit[0] = N/A	EdgeLock Enclave
1	bit[1] = N/A	MTR_MSTR
2	bit[2] = N/A	CM33_I, CM33_S
3	bit[3] = N/A	CA55, GIC600
4	bit[4] = N/A	Reserved
5	bit[5] = N/A	uSDHC1
6	bit[6] = N/A	uSDHC2
7	bit[7] = N/A	eDMA3_1, eDMA4_2

Table continues on the next page...

Table 102. Default DID Assignments for TRDC components (continued)

		NOTE Security Requirement: Only assign eDMA instances assigned DEFAULT_DID=7 so we have a single master DID number only for DMA in non-DMA ID replication mode of operation, this will prevent DMA DID from spoofing another master.
8	bit[8] = N/A	CoreSight ETR, TESTPORT
9	bit[9] = N/A	DAP AHB_AP_SYS (JTAG access to System Bus)
10	bit[10] = N/A	For improved security glitch protection assign all other devices - including but not limited to: ENET1, ENET2, uSDHC3, NPU, USB2, LCDIFs, ISI
11	bit[11] = N/A	USB1
12	bit[12] = N/A	Not Assigned
13	bit[13] = N/A	Not Assigned
14	bit[14] = N/A	Not Assigned
15	bit[15] = N/A	Not Assigned

22.1.3 TRDC MDAC Configuration

Table 103. TRDC Configuration

DAC_INST_NUM	This value is the instance number of the MDAC for a give TRDC master.
MASTER	Specifies the master to perform region/domain checking.
Switch/Port	Connect to NIC, NoC system fabric port.
Location	Specifies which subsystem the TRDC master and MRC component is located in.
BUS Type	Specifies the Slave Memories Bus Type to connect to the MRC.
DAC_REG_NUM	For CPU DACs, this typically depends on the number of processing IDs that the connected CPU will generate, or

Table continues on the next page...

Table 103. TRDC Configuration (continued)

	the number of PID values that software will use for context switches. For non-CPU DACs, this is set to 1. Only one register is needed for non-CPU DAC to source the DID/Secure/Priv attributes for the master.
DAC_CPU	Set to 1 for any "CPU" master that could do context switching.
DEFAULT_DID	This is the DID value that is driven out by the MDAC when MDAC_EN_DDID is set and the MDAC is not yet valid/enabled.
DAC_INST_PID_IN	For DACs connected to CPUs, there's an optional PID input and PID registers. If the processor/CPU connected to DAC sources a process-ID, the integrator would connect that bus to the MDAC and enable this parameter. If the CPU does not source a PID, the user can program the PID registers in the MDAC. Software saves and restores this PID with any context switch.
MAX_DOMAIN_MAPPING	Specifies the number of domains.

Table 104. TRDC MDAC Configuration

TRDC location	MDAC	DAC_INST_NUM	MASTER	DAC_REG_NUM	DAC_CPU	DEFAULT_DID	DAC_INST_PID_IN
AONMIX	MDAC_A0	0	Reserved	1	1	0	0
	MDAC_A1	1	CM33_I	3	1	2	0
			CM33_S	3	1	2	0
	MDAC_A2	2	eDMA3_1	1	0	7	0
	MDAC_A3	3	MTR_FBX_MSTR	1	0	1	0
	MDAC_A4	4	MTR_MSTR	1	0	1	0
WAKEUPMIX	MDAC_W0	0	Reserved	1	0	4	0
	MDAC_W1	1	Reserved	1	0	4	0
	MDAC_W2	2	DAP AHB_AP_SYS	1	0	9	0
	MDAC_W3	3	CoreSight ETR	1	0	8	0
	MDAC_W4	4	eDMA4_2	1	0	7	0
MEGAMIX	MDAC_M0	0	uSDHC1	1	0	5	0
	MDAC_M1	1	uSDHC2	1	0	6	0
	MDAC_M2	2	TESTPORT	1	0	8	0
	MDAC_M3	3	uSDHC3	1	0	10	0

Table continues on the next page...

Table 104. TRDC MDAC Configuration (continued)

	MDAC_M4	4	ENET (Ethernet MAC)	1	0	10	0
	MDAC_M5	5	ENET (Ethernet MAC)	1	0	10	0
	MDAC_M6	6	ENET_QOS (Ethernet Controller with TSN)	1	0	10	0
NICMIX	MDAC_N0	0	CA55 Read Channel	4	1	3	1
	MDAC_N0	1	CA55 Write Channel	4	1	3	1
	MDAC_N1	2	NPU m0	1	0	10	0
	MDAC_N2	3	NPU m1	1	0	10	0
HSIOMIX	MDAC_H0	0	USB 1	1	0	11	0
	MDAC_H1	1	USB 2	1	0	10	0
MEDIAMIX	MDAC_C0	0	ISI M	1	0	10	0
	MDAC_C1	1	ISI U	1	0	10	0
			ISI V	1	0	10	0
	MDAC_C2	2	PXP	1	0	10	0
	MDAC_C3	3	LCDIF	1	0	10	0

NOTE

Access on WAKEUPMIX MDAC_C5 will not return transfer error.

22.1.4 TRDC MBC Configuration

Location	MBC	MBC Instance Number	Number of Slaves	Slave Port	Peripherals (Slave Memory)	Number of Blocks	Block Size (KB)
AONMIX	MBC_A0	0	3	SLV0	AIPS1	128	64
				SLV1	Edgelock Secure Enclave	8	64
				SLV2	GPIO1	1	64
				SLV3	Reserved	0	0
	MBC_A1	1	2	SLV0	CM33 Code-TCM	32	4

Table continues on the next page...

Table continued from the previous page...

				SLV1	CM33 System-TCM	32	4
				SLV2	Reserved	0	0
				SLV3	Reserved	0	0
WAKEUPMIX	MBC_W0	0	4	SLV0	AIPS2	128	64
				SLV1	GPIO2	1	64
				SLV2	GPIO3	1	64
				SLV3	DAP (Debug)	17	1024
	MBC_W1	1	4	SLV0	AIPS3	128	64
				SLV1	AHB_ISPAP	1	64
				SLV2	NIC_MAIN GPV	1	1024
				SLV3	GPIO4	1	64
NICMIX	MBC_N0	0	4	SLV0	DDR CFG	49	64
				SLV1	AIPS4	11	64
				SLV2	MEDIAMIX	37	64
				SLV3	HSIOMIX	48	64
	MBC_N1	1	4	SLV0	CA55 MTR_DCA, TCU, TROUT Read Channel	5	64
				SLV1	CA55 MTR_DCA, TCU, TROUT Write Channel	5	64
				SLV2	MLMIX	17	64
				SLV3	MLMIX	17	64
	MBC_N2	2	2	SLV0	GIC	9	64
				SLV1	GIC	9	64
				SLV2	Reserved	0	0
				SLV3	Reserved	0	0
	MBC_N3[1]	3	2	SLV0	OCRAM	40	16
				SLV1	OCRAM	40	16
				SLV2	Reserved	0	0

Table continues on the next page...

Table continued from the previous page...

				SLV3	Reserved	0	0

1. For this instance, OCRAM address is remapped from 0x2048_0000 to 0x8000_0000. When the MBC Domain Error Word0 register (MBC-N3_DERR_W0) is read, the actual address is reg value + 0x2048_0000 - 0x8000_0000.

22.1.4.1 TRDC_WAKEUPMIX MBC Global Configuration Register

The reset values of MBC Global Configuration Register (MBC0_MEM0_GLBCFG - MBC1_MEM3_GLBCFG) for TRDC_WAKEUPMIX are as below -

Table 105. Reset Values of MBC0_MEM0_GLBCFG - MBC1_MEM3_GLBCFG

Register	Reset Value
MBC0_MEM0_GLBCFG	0010_0080h
MBC0_MEM1_GLBCFG	0010_0001h
MBC0_MEM2_GLBCFG	0010_0001h
MBC0_MEM3_GLBCFG	0014_0011h
MBC1_MEM0_GLBCFG	0010_0080h
MBC1_MEM1_GLBCFG	0010_0001h
MBC1_MEM2_GLBCFG	0014_0001h
MBC1_MEM3_GLBCFG	0010_0001h

22.1.5 TRDC MRC Configuration

Table 106. TRDC MRC Configuration

Location	MRC	MRC Instance Number	Number of Slaves	Slave Memory	Number of MRC descriptors
AONMIX	MRC_A0	0	1	CM33 ROM	8
WAKEUPMIX	MRC_W0	0	1	CA55 ROM	8
	MRC_W1	1	1	FlexSPI1	8
NICMIX	MRC_N0	0	2	DRAM	16

NOTE

Access on WAKEUPMIX MRC_W2 will not return transfer error.

NOTE

MRCs should be used to check non-contiguous memory space.

22.2 Overview

TRDC provides an integrated and scalable architectural framework for access control, system memory protection, and peripheral isolation. It allows software to assign chip resources including processor cores, non-core bus initiators, memory regions, and peripherals to processing domains to support enforcement of robust operational environments.

- First, each bus initiating resource is assigned to a domain identifier (domainID, DID). Typically, each processor is assigned to a unique domainID and all remaining non-processor bus initiators assigned to a different domainID.

- Next, the access control policies for the individual domains are programmed into any number of registers implemented in the target memory block and region checkers.
- Finally, all accesses throughout the device are monitored concurrently to determine the validity of each access. If a reference from a given domain has sufficient access rights, it is allowed to continue, else the access is aborted and error information captured.

The access control scheme that TRDC defines supports a 4-level model, combining the traditional privileged (also known as supervisor) and user modes with an additional signal defining the secure, nonsecure attributes of each memory reference. The result is a 4-level hierarchical access control mechanism with different access control policies based on read, write, and execute references, where:

SecurePriv > SecureUser > NonsecurePriv > NonsecureUser

Combined with the secure/nonsecure and privileged/user attributes, a domainID is associated with every system bus transaction and forms the hardware basis for the implementation of TRDC's access control mechanisms.

NOTE

The terminology in this chapter has been updated to align with NXP's inclusive language standards, as shown in the table below.

Table 107. Updated terms

Updated term	Deprecated term
Initiator	Master
Target	Slave

22.2.1 Block diagram

As previously noted, the TRDC implementation is distributed across multiple submodules instantiated throughout the device. The TRDC submodules include:

- TRDC_MGR
  - The Manager (MGR) submodule coordinates all programming model reads and writes.
- TRDC_DAC
  - The Domain Assignment Controller (DAC) submodule handles resource assignments and generation of the domain identifiers.
- TRDC_MBC
  - The Memory Block Checker (MBC) submodule implements access controls for on-chip internal memories and peripherals based on a fixed-sized block format.
- TRDC_MRC
  - The Memory Region Checker (MRC) submodule implements the access controls for external off-chip memories and peripherals based on the pre-programmed region descriptor registers.

See Figure 41 for a simplified TRDC block diagram focusing on topology and connections.

NOTE

TRDC typically consists of various submodules like DAC, MGR, MBC, and MRC. However, all the submodules as shown in the block diagram below may not be present for this configuration. For chip-specific implementation details of this module's instances, see the chip-specific information.

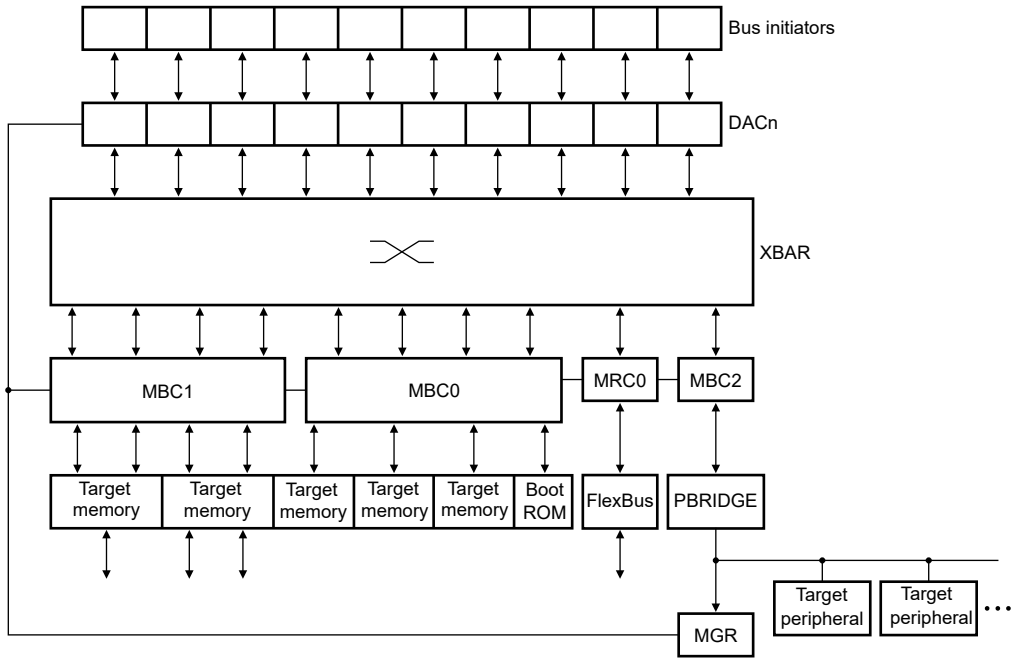


Figure 41. Block diagram

22.2.2 Features

- Ability to assign chip resources to processing "domains":
  - Processor cores, non-core bus initiators, target memories, and target peripherals
  - Each processing domain is assigned a unique domain identifier (domainID, DID)
  - DomainID is an attribute associated with every system bus transaction
  - Used in conjunction with secure/nonsecure, privileged/user attributes
- Defines access rights to targets in MBCs for on-chip memories and peripherals and in MRCs for external memories and peripherals
- Built upon a 4-level hierarchical access control model:
  - SecurePriv > SecureUser > NonsecurePriv > NonsecureUser
  - Defined by multiple sets of user-programmable R/W/X (Read, Write, Execute) flags per access state
- Distributes programming model and hardware implementation across multiple submodules:
  - Supports a broad, highly-configurable architecture definition
  - Supports efficient mechanisms for memory space context switch state changes using nonsecure enables (NSE) for memory blocks and regions

22.3 Functional description

This section provides more details on the operation and implementation of the TRDC submodules.

22.3.1 Modes of operation

TRDC does not support any special modes of operation. As a memory-mapped device located in the core platform's clock domain, it responds strictly on the basis of memory addresses of the connected system buses. TRDC evaluates domain assignment

and access control functions on a reference-by-reference basis using the addresses and attributes connected to the system bus port(s).

22.3.2 Manager (TRDC_MGR)

The MGR submodule is responsible for coordinating TRDC's programming model reads and writes.

The complete programming model is large. Additionally, the programming model is distributed across the various submodule instances (DAC, MBC, MRC).

MGR provides:

- The required hardware management for all programming model accesses.
- Generating and distributing the appropriate address decodes and write data to the DAC, MBC, and MRC submodule instances.
- Collecting and combining all the register read data buses and responses.

MGR implements:

- The Control Register
- All the read-only hardware configuration registers (HWCFG[0-1], DACFGn, MBCm_MEMs_GLBCFG, MRC_GLBCFG)
- The program-visible versions of the domain error capture registers.

To support the domain error reporting functionality, MGR collects the individual captured error indicator output signals from all the submodule instances and remaps them so that they are organized into an instance bitmap by domain that the DERRLOCn register array specifies.

22.3.3 Domain Assignment Controller (TRDC_DAC)

The DAC submodule is responsible for the generation of the domainID on every memory transaction for every bus initiator in the device. The resulting domainID, and {nonsecure, privileged} signals are generated and then treated as address attributes and associated with each transaction as it moves through the system.

DAC modules that drive the domainIDs (DIDs) for processor cores include an option of processor-identifier (PID) matching to select between DIDs. You can accomplish this by first enabling PID checking by writing TRDC_MDA_Wm_n[PE] = [2,3]. With PID matching enabled, a domain hit is determined by matching TRDC_MDA_Wm_n[PID] with the PID input or PID in the TRDC_PID register. The PIDM field can be used to match specific groups of PIDs.

As a simple example, assume a system with two domainIDs.

- domainID 0 is for critical tasks.
- domainID 1 is for non-critical tasks.
- Two domainIDs require two MDA_Wm_n registers, MDA_W0 and MDA_W1.

Further assume the processor’s task identifier defines whether a task is critical or non-critical as defined by system software.

- PIDs [0-15] are assigned to critical tasks.
- PIDs != [0-15] are assigned to noncritical tasks.

During startup, software initializes the MDA_W[0-1] registers. MDA_W0 defines DID = 1 for PIDs [0-15] and MDA_W1 defines DID = 2 for PIDs != [0-15].

1. MDA_W0 = 0x8000_0F81
    - VLD = 1
    - PID = 0x0
    - PIDM = 0xF
    - PE = 2

- DIDS = 0
- DID = 1

2. MDA_W1 = 0x8000_0FC2
   - VLD = 1
   - PID = 0x0
   - PIDM = 0xF
   - PE = 3
   - DIDS = 0
   - DID = 2

As the processor executes, it loads the appropriate task ID into its PID register as the task is started. The processor's PID register value is input to the corresponding DAC module and used by the MDA_Wr_m comparison logic. The system then dynamically generates two domainIDs and the downstream TRDC access check logic can distinguish and enforce different access control rights based on the different domainIDs.

A special mode is provided for driving the DID of non-processor initiators. This is enabled with the DID bypass (DIDB) bit. The DAC has a DID input that can be driven by a initiator and used directly as the DID output. This allows variability in the DID as dictated by the initiator and is intended to be used by DMA initiators that masquerade as, (that is, drive the DID of) the initiator that programmed it.

22.3.4 Access evaluation

Fundamental to the TRDC's operation is the actual access violation check performed by the memory block checker (MBC) and memory region (MRC).

Previous descriptions have detailed the domain assignment mechanisms and the tracking of the domain identifier as an address attribute through the system bus switching fabric(s). As transactions reach the target memory and peripheral controllers, the address is used to select the appropriate memory region descriptor or block configuration register. Once the appropriate register has been selected, the next function is the actual access evaluation.

For this step, the 3-bit M{B,R}ACSEL field of the selected RGD or BLK_CFG register selects the appropriate GLBAC (global access control policy). The GLBAC register defines the access rights for the domain based on the 4-level model combined with the transaction's access attributes (read = 0/write = 1, secure = 0/nonsecure = 1, execute =1/non-execute=0, user = 0/privileged = 1).

The 4-level model is defined by the concatenation of two address attributes: secure/nonsecure and user/privileged attributes. These two attribute signals (nonsecure, privileged) and the resulting access levels are defined as:

```
if {nonsecure, privileged} == 0b00, then level = SecureUser
if {nonsecure, privileged} == 0b01, then level = SecurePriv(ileged)
if {nonsecure, privileged} == 0b10, then level = NonsecureUser
if {nonsecure, privileged} == 0b11, then level = NonsecurePriv(ileged)
```

The resulting hierarchical access control model specifies:

```
SecurePriv , SecureUser , NonsecurePriv , NonsecureUser
```

After the access evaluation is complete and the access error signal generated, the transaction is allowed to continue if it has sufficient access rights, else the access is aborted with the address and attribute information captured in the appropriate error registers.

22.3.5 Memory Block Checker (TRDC_MBC)

The TRDC's Memory Block Checker provides domain-based access control for all system bus references targeted to on-chip internal memories and peripherals. Using programmed block configuration registers which define the access rights per domain and block, the MBC concurrently monitors system bus transactions and evaluates the appropriateness of each transfer. Memory

references that have sufficient access rights are allowed to complete, while references that have insufficient rights are terminated with an access error response.

The TRDC architectural framework supports up to 8 MBC instances where each MBC can support up to 4 sub-regions (MEM[s]). Note monitoring of an AXI memory counts as 2 sub-regions since there are independent read and write channels per AXI memory. Conversely, an AHB connection counts as a single subregion.

Each sub-memory is divided up in equal sized consecutive blocks. Sub-memory 0 (MEM0) can support up to 512 blocks while sub-memory 1-3 (MEM1-MEM3) can each support up to 64 blocks. The number of blocks and size of the block is defined in MBCm_MEMs_GLBCFG[NBLKS,SIZE_LOG2] register fields, where m is the MBC instance number, and s is the sub-memory number. For example, Sub-memory 0 (MEM0) may have 16 consecutive blocks, each of size 64 KB, and corresponding MBCm_DOMd_BLK_CFG_W register to a block will control access to that block.

Each MBCm_DOMd_BLK_CFG_W contains eight access control structures. Each structure is comprised of a 1-bit NSE (NonSecure Enable) plus a 3-bit MBACSEL (Memory Block Access Control Select). The NSE controls secure/nonsecure access to the corresponding block, while the MBACSEL field selects a MBCm_MEMN_GLBACr register which controls the read/write/execute access to the corresponding block. There are 8 programmable MBCm_MEMn_GLBACr registers per block checker, each containing contain read-write-execute control flags for each of the four operating modes:

```
SPR, SPW, SPX - Secure    Privileged {Read, Write, Execute}
SUR, SUW, SUX - Secure    User       {Read, Write, Execute}
NPR, NPW, NPX - Nonsecure Privileged {Read, Write, Execute}
NUR, NUW, NUX - Nonsecure User       {Read, Write, Execute}
```

GLBAC1-7 also have a lock bit MBCm_MEMn_GLBACr[LK]. When LK=1, the GLBACr register is read-only until the next reset. Furthermore, if MBCm_DOMd_BLK_CFG_W[MBACSEL] selects a GLBAC that is locked, the 3-bit MBCACSEL field is also locked until the next reset. GLBAC0 cannot be locked.

The MBC also has MBCm_DOMd_MEMs_BLK_NSE_W registers. These registers are a bit map of the block NSE bits. Each word supports up to 32 blocks.

```
MBCm_DOMd_MEM0_BLK_NSE_W[0-15] has a maximum of 16 words to support 512 blocks
MBCm_DOMd_MEM[1-3]_BLK_NSE_W[0-1] has a maximum of 2 words to support 64 blocks.
```

The block NSE bits can be programmed by writing these registers or by writing the MBCm_MEMN_BLK_{INDEX, SET, CLR, CLR_ALL} registers.

The MBCm_MEMn_NSE_BLK_{INDEX, SET, CLR, CLR_ALL} registers provide a mechanism to efficiently and quickly manipulate the NSE bitmap with bitmasks to set or clear individual blocks within the bitmap and a single operation to perform a clear all across multiple DIDs. The BLK_INDEX register provides the control definition for the BLK_SET and BLK_CLR operations.

22.3.5.1 Memory block hit determination

MBCm_MEMs_GLBCFG[NBLK, SIZE_LOG2] is used to determine which bits of the address correspond to block number.

```
block_n_hit = (addr[MSB:LSB] == n)
where LSB=SIZE_LOG2
MSB=f{NBLKS,LSB}
```

Note that the block hit determination is based only on the address comparison of the first byte being accessed; that is, the MBC does not check the size of the access to make sure it entirely fits within the region.

22.3.5.2 Memory block access evaluation

For a block n hit, the MBC logic evaluates the access rights defined by the MBCm_DOMd_BLK_CFG_Ww registers. The domainID attribute and block number hit selects the appropriate MBCm_DOMd_BLK_CFG_Ww register to use in the access evaluation. The {R,W,X}, nonsecure and privilege attributes of the access are compared with the MBC global access control policy defined in MBCm_DOMd_BLK_CFG_W[MBACSEL, NSE] fields.

While TRDC_CR[GVLDB] = 1, and for each sub-memory being monitored, the MBC performs a reduction-AND of all the individual error terms from each access evaluation.

Unlike the MRC access evaluation and termination, the MBC access evaluation terminates the access with a bus error and reports an access error for only one condition - when the access doesn't have sufficient access rights. By nature, the block checker can only hit in ONE block, and by implementation, there are no "miss" accesses. Accesses outside of the range covered by the MBC are not sent to the MBC. There also aren't any individual block valid bits. When TRDC_CR[GVLDB] = 1, checking on all blocks is always enabled.

22.3.6 Memory Region Checker (TRDC_MRC)

The TRDC's Memory Region Controller provides domain-based, hardware access control for all system bus references targeted at non-peripheral memory spaces. Using pre-programmed *region descriptors* which define memory spaces and their associated access rights per domain identifier, the MRC concurrently monitors system bus transactions and evaluates the appropriateness of each transfer. Memory references that have sufficient access control rights are allowed to complete, while references that are not mapped to any region descriptor or have insufficient rights are terminated with an access error response.

The TRDC architectural framework supports up to 16 MRC instances, where each MRC instance can support [1-16] region descriptors, concurrently monitoring up to 8 target memory ports. Note, monitoring an AXI protocol port counts as two ports since there are independent read address and write address channels. Conversely, an AHB connection counts as a single port.

A partial, simplified one-dimensional block diagram of an instance of the MRC module is shown in Figure 42. In this figure, a single target bus port is shown. When the remaining target ports are considered, the MRC hardware's two-dimensional connection matrix, broadcasting each system bus target port access across the implemented memory region descriptors, is the resulting structure with the basic access evaluation macro shown as the replicated submodule block. In Figure 42, the system bus fabric target port is shown on the left, the memory region descriptor registers are in the middle, and the peripheral bus interface on the right side. The evaluation macro contains two magnitude comparators connected to the start and end address registers from each region descriptor as well as the combinational logic blocks to determine the region hit and, based on the domainID associated with the reference, access violations.

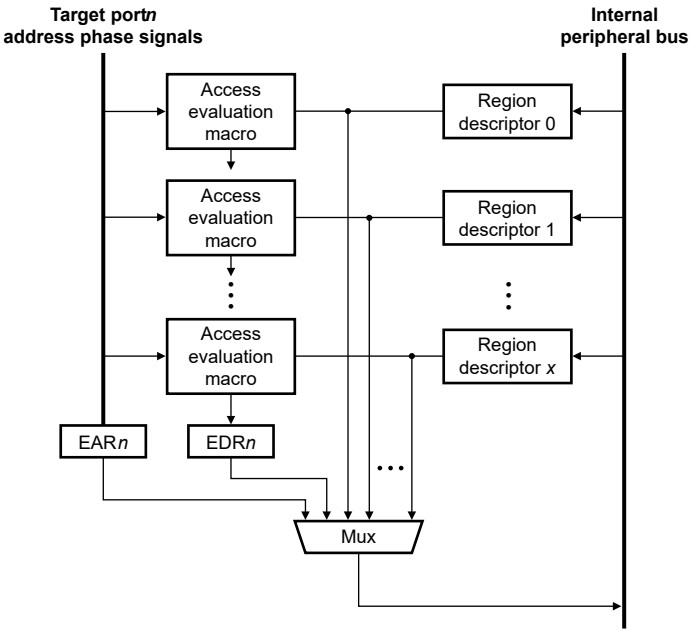


Figure 42. MRC access evaluation macro hardware structure

For each access, the MRC hardware performs a two-step evaluation process. First, the access is compared to all memory region descriptors to determine hit/miss status, and then, using the domainID associated with the reference plus the address attributes {write, nonsecure, privileged}, the access rights are examined using the method detailed in Access evaluation.

22.3.6.1 Memory region descriptor hit determination

To determine if the current reference hits in a given region, two magnitude comparators are used in conjunction with the region's start and end addresses. The boolean equation for this portion of the hit determination is defined as:

```
region_hit_n =
   ((addr >= start_addr) & (addr <= end_addr)) & valid
```

where `addr` is the current system reference address, `start_addr` and `end_addr` are the start and end addresses, and `valid` is the valid bit, all from memory region descriptor n.

Note there are *no hardware checks* to verify that `end_addr ≥ start_addr`, and it is software's responsibility to properly load appropriate values into these fields of the region descriptor.

Also, the region hit determination is based only on an address comparison of the first byte being accessed; that is, the MRC does *not* check the size of the access to make sure it entirely fits within the region.

22.3.6.2 Memory region access evaluation

For each region descriptor hit, the MRC logic evaluates the access rights defined by the MRCm_DOMd_RGDr_Ww registers. Specifically, the domainID attribute selects the appropriate MRCm_DOMd_RGDr_Ww register to use in the access evaluation. The {R,W,X}, nonsecure and privilege attributes of the access are compared with the MRC global access control policy defined in MRCm_DOMd_RGDr_W0[MRACSEL] and MRCm_DOMd_RGDr_W1[NSE] fields. See Access evaluation for additional details on this function.

While TRDC_CR[GVLDR] = 1, and for each bus target port being monitored, the MRC performs a **reduction-AND** of all the individual *(no_hit | error)* terms from each access evaluation macro. Recall as specified in Memory region descriptor hit determination, an *invalid* MRC_DOMd_RGDr_Ww (MRCm_DOMd_RGDr_W1[VLD] = 0) forces the region_hit_n evaluation to be negated.

This expression then terminates the bus cycle with an error and reports an access error for three conditions:

1. If the access does not hit in *any* region descriptor, an access error is reported.
2. If the access hits in a single region descriptor and that region signals a domain violation, then an access error is reported.
3. If the access hits in multiple (overlapping) regions and one region signals a violation, then an access error is reported.

The third condition reflects that priority is given to access denying over access allowing for overlapping regions.

Unimplemented domain IDs (DIDs) do not have any associated region descriptors and therefore have no access rights.

22.3.7 Interrupts

This module outputs an interrupt signal which can be connected to the system's interrupt controller. Please check chip-specific interrupt assignment for details. Interrupt is asserted on detection of access violation by any checker, and it remains asserted until DERRLOCn registers are cleared. This interrupt is in addition to interrupt generated by initiator after receiving bus error due to access violation by memory block checker or memory region checker. So, system may receive interrupt through different channels but from the same access violation at checker.

22.4 External signals

The TRDC module does not directly support any external interfaces.

The *internal* interfaces include a standard 32-bit target bus for all programming model accesses, connections to the address phase signals associated with AHB and/or AXI system buses and connections to the target peripheral buses as shown in TRDC block diagram.

22.5 Initialization

Out of reset, TRDC_CR[GVLDR, GVLDB, GVLDM] bits are cleared, TRDC is disabled and in deny by default mode allowing secure privileged startup code to configure the entire programming model. In deny by default mode, only the DIDs that aren't denied by default are allowed to access memory and peripherals. The allowed DIDs are implementation defined. The initialization process typically is performed in 3 steps:

1. Read the various hardware configuration registers (HWCFG{0,1}, DACFGn, MBCm_MEMs_GLBCFG, MRC_GLBCFG) to obtain the implemented hardware capabilities for the device.
2. Using the information retrieved in Step 1 coupled with the desired domain architecture, program all the register data structures for domain assignment (MDA_Wm_n), memory block configuration (MBCm_DOMd_MEMs_BLK_CFG_Ww) and memory region descriptors (MRCm_DOMd_RGDr_Ww). There are individual valid bits included in these registers which typically would be asserted. Additionally, there are also lock mechanisms available if register settings need to be configured and marked as read-only.
3. Once the TRDC programming model is loaded, the TRDC_CR is written with GVLD {R, B, M} asserted. Upon the assertion of the 3 GVLD bits, deny by default mode is exited and at that point, the TRDC is fully operational. The TRDC remains enabled until the next reset.

22.6 Application information

As described in Overview, the TRDC architecture is intended to provide a broad, highly-capable framework for access control, system memory protection and peripheral isolation. The resulting microarchitecture implementation is distributed throughout the core platform, highly configurable via hardware design parameters and software programmability.

22.6.1 Initiator domain assignments

The typical use case related to initiator domain assignments is to include one (or more) processor core(s) in a single domain, possibly coupled with other bus initiator devices like DMA, etc. The definition of a domain may be static, based simply on the combination of a processor and other optional bus initiators. Alternatively, the processor's operation may be configured to dynamically select between a very small number of domains. In particular, the optional inclusion of PID value(s) to be used to create multiple classes of CPU with different domain values.

For example, "critical" tasks, whether safety-critical, performance-critical, etc. can be grouped together in one domain and all other tasks into a second domain. Non-processor bus initiators typically have a single MDA_Wm_n configuration register associated with them. The domainID and {nonsecure, privileged} attributes are usually statically assigned, unless the module can "inherit" attributes from a processor programming it (certain modules like DMAs).

The MDA_Wm_n registers, along with the memory region descriptors and the peripheral domain access control registers all have lock features that allow the register resources to be converted into read-only resources to protect the written configuration.

22.6.2 Memory region descriptor management

There are important concepts to consider in managing the memory region descriptors.

Recall that the association between each MRCn instance and the target memory regions it monitors is SoC-dependent. The device-specific configuration specifies the number of implemented memory region descriptors in a given instance and the specific port numbers associated with the target memories being monitored.

22.6.3 Domain error capture management

Recall when a domain access violation is detected by either a memory region checker or a memory block checker, address and attribute information of the offending access is captured. The DERRLOCn read-only registers provide additional information signaling the instance number of the submodule where the access violation(s) occurred. Since the resulting exception handler needs the submodule instance to retrieve the captured address and attribute information from DERR_W0_n and DERR_W1_n, the DERRLOCn registers provide the instance number details.

These registers are organized as a word array, *indexed by the faulting domain number*, with the contents of each register providing a bitmap signaling the instance number(s) associated with all submodules containing captured error information for that domain.

It is recommended that the exception handler begin by reading the HWCFG1 register to determine its domainID. Next, it uses the just-retrieved domainID to index into the DERRLOCn array. The resulting DERRLOCn value is then examined to determine the instance number of the reporting MBC and/or MRC submodule.

There may be multiple access violations, across multiple instances, pending for a given domain. It is suggested that exception handler use a "find first one" instruction (alternatively known as "count leading zeroes") to quickly and efficiently find the instance number containing access violation details. Once the instance number has been determined, the captured error address and attribute information can easily be retrieved from the domain error registers. For 32 bits address, the captured error address can be retrieved from DERR_W0_n and the attribute information can be retrieved from DERR_W1_n. For 36 bits address, the captured error address can be retrieved from DERR_W0_n and DERR_W1_n and the attribute information can be retrieved from DERR_W2_n. The 2-bit DERR_W1_n[EST] or DERR_W2_n[EST] field provides information signaling whether one or more errors have been detected by the submodule instance. A special write to DERR_W3_n is required to reset (and rearm) the EST error capture state machine.

This process is repeated until all errors associated with a given domainID have been processed.

A graphical representation of this 2-step retrieval of the domain error address and attributes is shown in Figure 43.

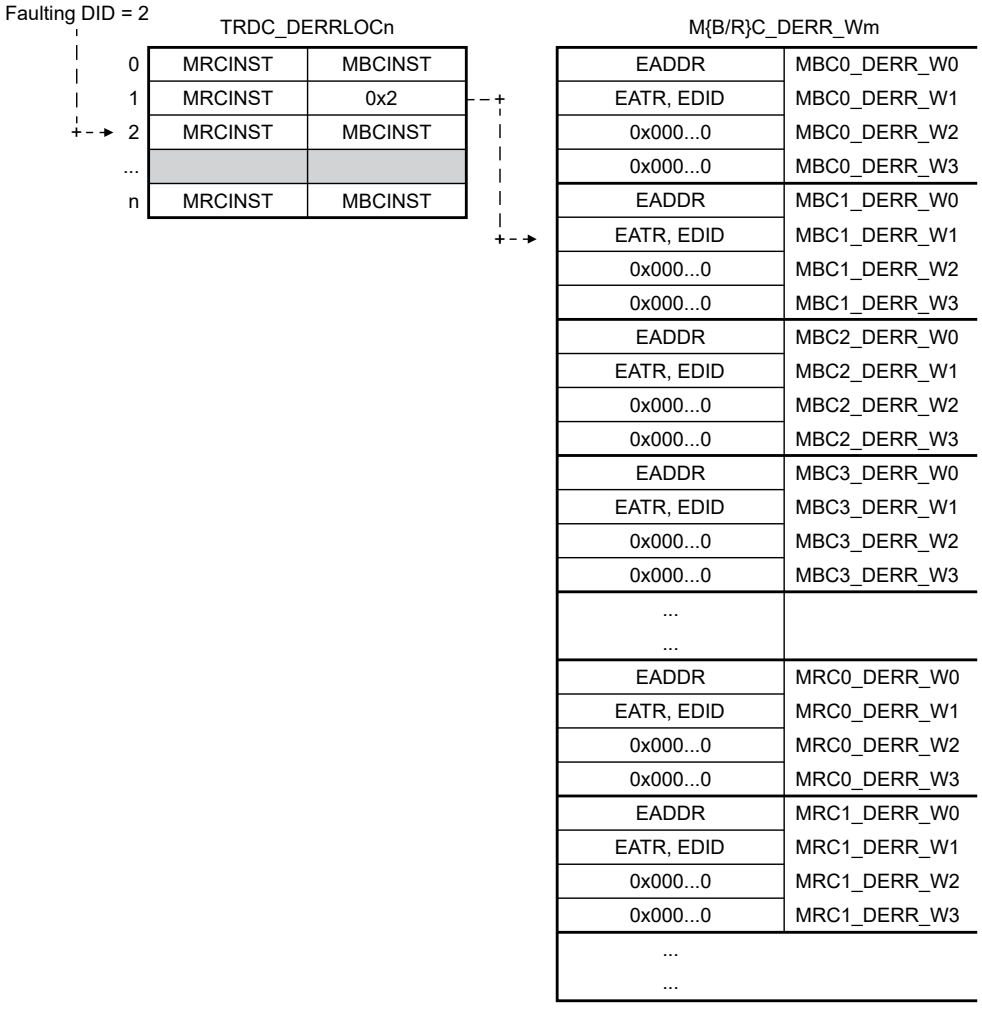


Figure 43. Two-step retrieval of domain error address and attributes

22.7 Register descriptions

The MBCm_NSE_BLK_SET, MBCm_NSE_BLK_CLR and MBCm_NSE_BLK_CLR_ALL registers exist at offset for only domain ID value 0 even when more than 1 domains exist, and address offsets for non-zero domain ID value should be considered reserved space. So, this reserved address space in MBC space does not return transfer error: 0xn14 to 0xn1C where n = (1 to NUM_DID-1) * 2 for each MBC.

If TZ-M is enabled, the TRDC programming model can only be accessed in SecurePrivileged mode. If TZ-M is disabled, it can only be accessed in Privileged mode regardless of Secure or NonSecure; that is it can't be accessed in User mode. Unless noted otherwise, the programming model registers can be accessed via 8-, 16- or 32-bit reads and 32-bit write references. Attempted accesses in a different operating mode, using unsupported write data sizes, writes to read-only resources, or access to reserved spaces are terminated with an error unless noted otherwise. The TRDC programming model is partitioned into these groups of registers:

- Basic hardware control and configuration
- Domain errors: location and details
- Initiator domain assignments
- Memory block and region checkers

It should be noted that many of the programming model registers in TRDC are organized as 2-, 3- or 4- dimensional data structures. For the 2-dimensional structures, the generic arrays contain "m" words representing the "columns" and "n" instances of the structure representing the "rows". These may be described as `structure[n][m]`. They appear in the address space of the programming model memory map in the standard C language row-major layout, that is, the "m" words representing the "column" appear sequentially with the entire row replicated "n" times. The TRDC register structure uses the naming convention of *TRDC_<regname>_Wm_n*, where the column number "m" appears as a numeric suffix on the W (32-bit word) identifier, and the row identifier "n" appears as the final numerical suffix.

For the 4-dimensional structures, the generic arrays contain "m", "d", "s" or "r", and "w" indices corresponding to MBC/MRC instance, domain index, sub-memory or region index and word index. The naming convention for the MBC block configuration and MRC region descriptor word registers are: *MBCm_DOMd_MEMs_BLK_CFG_Ww* and *MRCm_DOMd_RGDr_Ww*

For the reset values in this section, a "*" indicates an initial value determined by the hardware configuration.

22.7.1 TRDC register descriptions

22.7.1.1 TRDC memory map

TRDC_AONMIX base address: 4427_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	TRDC Register (TRDC_CR)	32	RW	0000_0010
F0	TRDC Hardware Configuration Register 0 (TRDC_HWCFG0)	32	R	2102_0510
F4	TRDC Hardware Configuration Register 1 (TRDC_HWCFG1)	32	R	See section
F8	TRDC Hardware Configuration Register 2 (TRDC_HWCFG2)	32	R	0000_0000
FC	TRDC Hardware Configuration Register 3 (TRDC_HWCFG3)	32	R	0000_0000
101 - 104	Domain Assignment Configuration Register (DACFG1 - DACFG4)	8	R	See section
1C0	TRDC IDAU Control Register (TRDC_IDAU_CR)	32	RW	0000_0008
1E0	TRDC FLW Control (TRDC_FLW_CTL)	32	RW	0000_0000
1E4	TRDC FLW Physical Base (TRDC_FLW_PBASE)	32	R	0100_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1E8	TRDC FLW Array Base (TRDC_FLW_ABASE)	32	RW	0000_0000
1EC	TRDC FLW Block Count (TRDC_FLW_BCNT)	32	RW	0000_0000
1FC	TRDC Fault Domain ID (TRDC_FDID)	32	RW	0000_0000
200 - 23C	TRDC Domain Error Location Register (TRDC_DERRLOC0 - TRDC_DERRLOC15)	32	R	0000_0000
400	MBC Domain Error Word0 Register (MBC0_DERR_W0)	32	R	0000_0000
404	MBC Domain Error Word1 Register (MBC0_DERR_W1)	32	R	0000_0000
40C	MBC Domain Error Word3 Register (MBC0_DERR_W3)	32	RW	0000_0000
410	MBC Domain Error Word0 Register (MBC1_DERR_W0)	32	R	0000_0000
414	MBC Domain Error Word1 Register (MBC1_DERR_W1)	32	R	0000_0000
41C	MBC Domain Error Word3 Register (MBC1_DERR_W3)	32	RW	0000_0000
480	MRC Domain Error Word0 Register (MRC0_DERR_W0)	32	R	0000_0000
484	MRC Domain Error Word1 Register (MRC0_DERR_W1)	32	R	0000_0000
48C	MRC Domain Error Word3 Register (MRC0_DERR_W3)	32	RW	0000_0000
704	Process Identifier (PID1)	32	RW	0000_0000
820	DAC Initiator Domain Assignment Register (MDA_W0_1_DFMT0)	32	RW	0000_0000
824	DAC Initiator Domain Assignment Register (MDA_W1_1_DFMT0)	32	RW	0000_0000
828	DAC Initiator Domain Assignment Register (MDA_W2_1_DFMT0)	32	RW	0000_0000
840	DAC Initiator Domain Assignment Register (MDA_W0_2_DFMT1)	32	RW	2000_0000
860	DAC Initiator Domain Assignment Register (MDA_W0_3_DFMT1)	32	RW	2000_0000
880	DAC Initiator Domain Assignment Register (MDA_W0_4_DFMT1)	32	RW	2000_0000
1_0000	MBC Global Configuration Register (MBC0_MEM0_GLBCFG)	32	R	0010_0080
1_0004	MBC Global Configuration Register (MBC0_MEM1_GLBCFG)	32	R	0010_0008
1_0008	MBC Global Configuration Register (MBC0_MEM2_GLBCFG)	32	R	0010_0001
1_000C	MBC Global Configuration Register (MBC0_MEM3_GLBCFG)	32	R	000C_0000
1_0010	MBC NonSecure Enable Block Index (MBC0_NSE_BLK_INDEX)	32	RW	0000_0000
1_0014	MBC NonSecure Enable Block Set (MBC0_NSE_BLK_SET)	32	RW	0000_0000
1_0018	MBC NonSecure Enable Block Clear (MBC0_NSE_BLK_CLR)	32	RW	0000_0000
1_001C	MBC NonSecure Enable Block Clear All (MBC0_NSE_BLK_CLR_ALL)	32	RW	0000_0000
1_0020	MBC Global Access Control (MBC0_MEMN_GLBAC0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0024	MBC Global Access Control (MBC0_MEMN_GLBAC1)	32	RW	0000_0000
1_0028	MBC Global Access Control (MBC0_MEMN_GLBAC2)	32	RW	0000_0000
1_002C	MBC Global Access Control (MBC0_MEMN_GLBAC3)	32	RW	0000_0000
1_0030	MBC Global Access Control (MBC0_MEMN_GLBAC4)	32	RW	0000_0000
1_0034	MBC Global Access Control (MBC0_MEMN_GLBAC5)	32	RW	0000_0000
1_0038	MBC Global Access Control (MBC0_MEMN_GLBAC6)	32	RW	0000_0000
1_003C	MBC Global Access Control (MBC0_MEMN_GLBAC7)	32	RW	0000_0000
1_0040	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0044	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0048	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_004C	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0050	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0054	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0058	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_005C	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0060	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0064	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0068	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_006C	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0070	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0074	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W13)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0078	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_007C	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0140	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0144	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0148	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_014C	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0180	MBC Memory Block Configuration Word (MBC0_DOM0_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_01A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_01A8	MBC Memory Block Configuration Word (MBC0_DOM0_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_01C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0240	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0244	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0248	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_024C	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0250	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0254	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0258	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_025C	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W7)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0260	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0264	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0268	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_026C	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0270	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0274	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0278	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_027C	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0340	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0344	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0348	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_034C	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0380	MBC Memory Block Configuration Word (MBC0_DOM1_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_03A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_03A8	MBC Memory Block Configuration Word (MBC0_DOM1_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_03C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0440	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0444	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0448	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_044C	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0450	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0454	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0458	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_045C	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0460	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0464	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0468	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_046C	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0470	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0474	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0478	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_047C	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0540	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0544	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0548	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_054C	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM0_BLK_NSE_W3)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0580	MBC Memory Block Configuration Word (MBC0_DOM2_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_05A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_05A8	MBC Memory Block Configuration Word (MBC0_DOM2_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_05C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0640	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0644	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0648	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_064C	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0650	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0654	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0658	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_065C	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0660	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0664	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0668	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_066C	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0670	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0674	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W13)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0678	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_067C	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0740	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0744	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0748	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_074C	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0780	MBC Memory Block Configuration Word (MBC0_DOM3_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_07A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_07A8	MBC Memory Block Configuration Word (MBC0_DOM3_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_07C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0840	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0844	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0848	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_084C	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0850	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0854	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0858	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_085C	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W7)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0860	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0864	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0868	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_086C	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0870	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0874	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0878	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_087C	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0940	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0944	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0948	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_094C	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0980	MBC Memory Block Configuration Word (MBC0_DOM4_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_09A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_09A8	MBC Memory Block Configuration Word (MBC0_DOM4_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_09C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0A40	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0A44	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0A48	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_0A4C	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0A50	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0A54	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0A58	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0A5C	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0A60	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0A64	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0A68	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_0A6C	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0A70	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0A74	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0A78	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_0A7C	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0B40	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0B44	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0B48	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_0B4C	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM0_BLK_NSE_W3)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0B80	MBC Memory Block Configuration Word (MBC0_DOM5_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0BA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_0BA8	MBC Memory Block Configuration Word (MBC0_DOM5_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_0BC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0C40	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0C44	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0C48	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_0C4C	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0C50	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0C54	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0C58	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0C5C	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0C60	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0C64	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0C68	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_0C6C	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0C70	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0C74	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W13)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0C78	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_0C7C	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0D40	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0D44	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0D48	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_0D4C	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0D80	MBC Memory Block Configuration Word (MBC0_DOM6_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0DA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_0DA8	MBC Memory Block Configuration Word (MBC0_DOM6_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_0DC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0E40	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0E44	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0E48	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_0E4C	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0E50	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0E54	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0E58	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0E5C	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W7)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0E60	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0E64	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0E68	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_0E6C	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0E70	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0E74	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0E78	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_0E7C	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0F40	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0F44	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0F48	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_0F4C	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0F80	MBC Memory Block Configuration Word (MBC0_DOM7_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0FA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_0FA8	MBC Memory Block Configuration Word (MBC0_DOM7_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_0FC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1040	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1044	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1048	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_104C	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1050	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1054	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1058	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_105C	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1060	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1064	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1068	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_106C	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1070	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1074	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1078	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_107C	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1140	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1144	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1148	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_114C	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM0_BLK_NSE_W3)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1180	MBC Memory Block Configuration Word (MBC0_DOM8_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_11A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_11A8	MBC Memory Block Configuration Word (MBC0_DOM8_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_11C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1240	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1244	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1248	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_124C	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1250	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1254	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1258	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_125C	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1260	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1264	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1268	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_126C	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1270	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1274	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W13)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1278	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_127C	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1340	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1344	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1348	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_134C	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1380	MBC Memory Block Configuration Word (MBC0_DOM9_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_13A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_13A8	MBC Memory Block Configuration Word (MBC0_DOM9_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_13C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1440	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1444	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1448	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_144C	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1450	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1454	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1458	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_145C	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W7)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1460	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1464	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1468	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_146C	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1470	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1474	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1478	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_147C	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1540	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1544	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1548	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_154C	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1580	MBC Memory Block Configuration Word (MBC0_DOM10_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_15A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_15A8	MBC Memory Block Configuration Word (MBC0_DOM10_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_15C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1640	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1644	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1648	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_164C	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1650	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1654	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1658	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_165C	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1660	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1664	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1668	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_166C	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1670	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1674	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1678	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_167C	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1740	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1744	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1748	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_174C	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM0_BLK_NSE_W3)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1780	MBC Memory Block Configuration Word (MBC0_DOM11_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_17A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_17A8	MBC Memory Block Configuration Word (MBC0_DOM11_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_17C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1840	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1844	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1848	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_184C	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1850	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1854	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1858	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_185C	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1860	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1864	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1868	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_186C	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1870	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1874	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W13)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1878	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_187C	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1940	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1944	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1948	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_194C	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1980	MBC Memory Block Configuration Word (MBC0_DOM12_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_19A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_19A8	MBC Memory Block Configuration Word (MBC0_DOM12_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_19C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1A40	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1A44	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1A48	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_1A4C	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1A50	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1A54	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1A58	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1A5C	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W7)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1A60	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1A64	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1A68	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_1A6C	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1A70	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1A74	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1A78	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_1A7C	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1B40	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1B44	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1B48	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_1B4C	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1B80	MBC Memory Block Configuration Word (MBC0_DOM13_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1BA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_1BA8	MBC Memory Block Configuration Word (MBC0_DOM13_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_1BC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1C40	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1C44	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1C48	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_1C4C	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1C50	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1C54	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1C58	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1C5C	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1C60	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1C64	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1C68	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_1C6C	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1C70	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1C74	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1C78	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_1C7C	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1D40	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1D44	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1D48	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_1D4C	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM0_BLK_NSE_W3)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1D80	MBC Memory Block Configuration Word (MBC0_DOM14_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1DA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_1DA8	MBC Memory Block Configuration Word (MBC0_DOM14_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_1DC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1E40	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1E44	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1E48	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_1E4C	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1E50	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1E54	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1E58	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1E5C	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1E60	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1E64	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1E68	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_1E6C	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1E70	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1E74	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W13)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1E78	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_1E7C	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1F40	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1F44	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1F48	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_1F4C	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1F80	MBC Memory Block Configuration Word (MBC0_DOM15_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1FA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_1FA8	MBC Memory Block Configuration Word (MBC0_DOM15_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_1FC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_2000	MBC Global Configuration Register (MBC1_MEM0_GLBCFG)	32	R	000C_0020
1_2004	MBC Global Configuration Register (MBC1_MEM1_GLBCFG)	32	R	000C_0020
1_2008	MBC Global Configuration Register (MBC1_MEM2_GLBCFG)	32	R	000C_0000
1_200C	MBC Global Configuration Register (MBC1_MEM3_GLBCFG)	32	R	000C_0000
1_2010	MBC NonSecure Enable Block Index (MBC1_NSE_BLK_INDEX)	32	RW	0000_0000
1_2014	MBC NonSecure Enable Block Set (MBC1_NSE_BLK_SET)	32	RW	0000_0000
1_2018	MBC NonSecure Enable Block Clear (MBC1_NSE_BLK_CLR)	32	RW	0000_0000
1_201C	MBC NonSecure Enable Block Clear All (MBC1_NSE_BLK_CLR_ALL)	32	RW	0000_0000
1_2020	MBC Global Access Control (MBC1_MEMN_GLBAC0)	32	RW	0000_0000
1_2024	MBC Global Access Control (MBC1_MEMN_GLBAC1)	32	RW	0000_0000
1_2028	MBC Global Access Control (MBC1_MEMN_GLBAC2)	32	RW	0000_0000
1_202C	MBC Global Access Control (MBC1_MEMN_GLBAC3)	32	RW	0000_0000
1_2030	MBC Global Access Control (MBC1_MEMN_GLBAC4)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2034	MBC Global Access Control (MBC1_MEMN_GLBAC5)	32	RW	0000_0000
1_2038	MBC Global Access Control (MBC1_MEMN_GLBAC6)	32	RW	0000_0000
1_203C	MBC Global Access Control (MBC1_MEMN_GLBAC7)	32	RW	0000_0000
1_2040	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2044	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2048	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_204C	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2140	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2180	MBC Memory Block Configuration Word (MBC1_DOM0_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2184	MBC Memory Block Configuration Word (MBC1_DOM0_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_2188	MBC Memory Block Configuration Word (MBC1_DOM0_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_218C	MBC Memory Block Configuration Word (MBC1_DOM0_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_21A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2240	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2244	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2248	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_224C	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2340	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2380	MBC Memory Block Configuration Word (MBC1_DOM1_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2384	MBC Memory Block Configuration Word (MBC1_DOM1_MEM1_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2388	MBC Memory Block Configuration Word (MBC1_DOM1_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_238C	MBC Memory Block Configuration Word (MBC1_DOM1_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_23A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2440	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2444	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2448	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_244C	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2540	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2580	MBC Memory Block Configuration Word (MBC1_DOM2_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2584	MBC Memory Block Configuration Word (MBC1_DOM2_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_2588	MBC Memory Block Configuration Word (MBC1_DOM2_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_258C	MBC Memory Block Configuration Word (MBC1_DOM2_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_25A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2640	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2644	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2648	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_264C	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2740	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM0_BLK_NSE_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2780	MBC Memory Block Configuration Word (MBC1_DOM3_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2784	MBC Memory Block Configuration Word (MBC1_DOM3_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_2788	MBC Memory Block Configuration Word (MBC1_DOM3_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_278C	MBC Memory Block Configuration Word (MBC1_DOM3_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_27A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2840	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2844	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2848	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_284C	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2940	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2980	MBC Memory Block Configuration Word (MBC1_DOM4_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2984	MBC Memory Block Configuration Word (MBC1_DOM4_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_2988	MBC Memory Block Configuration Word (MBC1_DOM4_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_298C	MBC Memory Block Configuration Word (MBC1_DOM4_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_29A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2A40	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2A44	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2A48	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2A4C	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2B40	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2B80	MBC Memory Block Configuration Word (MBC1_DOM5_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2B84	MBC Memory Block Configuration Word (MBC1_DOM5_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_2B88	MBC Memory Block Configuration Word (MBC1_DOM5_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_2B8C	MBC Memory Block Configuration Word (MBC1_DOM5_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_2BA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2C40	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2C44	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2C48	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_2C4C	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2D40	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2D80	MBC Memory Block Configuration Word (MBC1_DOM6_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2D84	MBC Memory Block Configuration Word (MBC1_DOM6_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_2D88	MBC Memory Block Configuration Word (MBC1_DOM6_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_2D8C	MBC Memory Block Configuration Word (MBC1_DOM6_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_2DA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2E40	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2E44	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2E48	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_2E4C	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2F40	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2F80	MBC Memory Block Configuration Word (MBC1_DOM7_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2F84	MBC Memory Block Configuration Word (MBC1_DOM7_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_2F88	MBC Memory Block Configuration Word (MBC1_DOM7_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_2F8C	MBC Memory Block Configuration Word (MBC1_DOM7_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_2FA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3040	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3044	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3048	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_304C	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3140	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3180	MBC Memory Block Configuration Word (MBC1_DOM8_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3184	MBC Memory Block Configuration Word (MBC1_DOM8_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_3188	MBC Memory Block Configuration Word (MBC1_DOM8_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_318C	MBC Memory Block Configuration Word (MBC1_DOM8_MEM1_BLK_CFG_W3)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_31A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3240	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3244	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3248	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_324C	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3340	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3380	MBC Memory Block Configuration Word (MBC1_DOM9_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3384	MBC Memory Block Configuration Word (MBC1_DOM9_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_3388	MBC Memory Block Configuration Word (MBC1_DOM9_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_338C	MBC Memory Block Configuration Word (MBC1_DOM9_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_33A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3440	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3444	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3448	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_344C	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3540	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3580	MBC Memory Block Configuration Word (MBC1_DOM10_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3584	MBC Memory Block Configuration Word (MBC1_DOM10_MEM1_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_3588	MBC Memory Block Configuration Word (MBC1_DOM10_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_358C	MBC Memory Block Configuration Word (MBC1_DOM10_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_35A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3640	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3644	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3648	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_364C	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3740	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3780	MBC Memory Block Configuration Word (MBC1_DOM11_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3784	MBC Memory Block Configuration Word (MBC1_DOM11_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_3788	MBC Memory Block Configuration Word (MBC1_DOM11_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_378C	MBC Memory Block Configuration Word (MBC1_DOM11_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_37A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3840	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3844	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3848	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_384C	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3940	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM0_BLK_NSE_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_3980	MBC Memory Block Configuration Word (MBC1_DOM12_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3984	MBC Memory Block Configuration Word (MBC1_DOM12_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_3988	MBC Memory Block Configuration Word (MBC1_DOM12_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_398C	MBC Memory Block Configuration Word (MBC1_DOM12_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_39A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3A40	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3A44	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3A48	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_3A4C	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3B40	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3B80	MBC Memory Block Configuration Word (MBC1_DOM13_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3B84	MBC Memory Block Configuration Word (MBC1_DOM13_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_3B88	MBC Memory Block Configuration Word (MBC1_DOM13_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_3B8C	MBC Memory Block Configuration Word (MBC1_DOM13_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_3BA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3C40	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3C44	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3C48	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_3C4C	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3D40	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3D80	MBC Memory Block Configuration Word (MBC1_DOM14_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3D84	MBC Memory Block Configuration Word (MBC1_DOM14_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_3D88	MBC Memory Block Configuration Word (MBC1_DOM14_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_3D8C	MBC Memory Block Configuration Word (MBC1_DOM14_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_3DA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3E40	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3E44	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3E48	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_3E4C	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3F40	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3F80	MBC Memory Block Configuration Word (MBC1_DOM15_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3F84	MBC Memory Block Configuration Word (MBC1_DOM15_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_3F88	MBC Memory Block Configuration Word (MBC1_DOM15_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_3F8C	MBC Memory Block Configuration Word (MBC1_DOM15_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_3FA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_4000	MRC Global Configuration Register (MRC0_GLBCFG)	32	R	0000_0008
1_4010	MRC NonSecure Enable Region Indirect (MRC0_NSE_RGN_INDIRECT)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4014	MRC NonSecure Enable Region Set (MRC0_NSE_RGN_SET)	32	RW	0000_0000
1_4018	MRC NonSecure Enable Region Clear (MRC0_NSE_RGN_CLR)	32	RW	0000_0000
1_401C	MRC NonSecure Enable Region Clear All (MRC0_NSE_RGN_CLR_ALL)	32	RW	0000_0000
1_4020	MRC Global Access Control (MRC0_GLBAC0)	32	RW	0000_0000
1_4024	MRC Global Access Control (MRC0_GLBAC1)	32	RW	0000_0000
1_4028	MRC Global Access Control (MRC0_GLBAC2)	32	RW	0000_0000
1_402C	MRC Global Access Control (MRC0_GLBAC3)	32	RW	0000_0000
1_4030	MRC Global Access Control (MRC0_GLBAC4)	32	RW	0000_0000
1_4034	MRC Global Access Control (MRC0_GLBAC5)	32	RW	0000_0000
1_4038	MRC Global Access Control (MRC0_GLBAC6)	32	RW	0000_0000
1_403C	MRC Global Access Control (MRC0_GLBAC7)	32	RW	0000_0000
1_4040	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD0_W0)	32	RW	0000_0000
1_4044	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD0_W1)	32	RW	0000_0000
1_4048	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD1_W0)	32	RW	0000_0000
1_404C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD1_W1)	32	RW	0000_0000
1_4050	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD2_W0)	32	RW	0000_0000
1_4054	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD2_W1)	32	RW	0000_0000
1_4058	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD3_W0)	32	RW	0000_0000
1_405C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD3_W1)	32	RW	0000_0000
1_4060	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD4_W0)	32	RW	0000_0000
1_4064	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD4_W1)	32	RW	0000_0000
1_4068	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD5_W0)	32	RW	0000_0000
1_406C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD5_W1)	32	RW	0000_0000
1_4070	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD6_W0)	32	RW	0000_0000
1_4074	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD6_W1)	32	RW	0000_0000
1_4078	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD7_W0)	32	RW	0000_0000
1_407C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD7_W1)	32	RW	0000_0000
1_40C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM0_RGD_NSE)	32	RW	0000_0000
1_4140	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD0_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4144	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD0_W1)	32	RW	0000_0000
1_4148	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD1_W0)	32	RW	0000_0000
1_414C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD1_W1)	32	RW	0000_0000
1_4150	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD2_W0)	32	RW	0000_0000
1_4154	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD2_W1)	32	RW	0000_0000
1_4158	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD3_W0)	32	RW	0000_0000
1_415C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD3_W1)	32	RW	0000_0000
1_4160	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD4_W0)	32	RW	0000_0000
1_4164	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD4_W1)	32	RW	0000_0000
1_4168	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD5_W0)	32	RW	0000_0000
1_416C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD5_W1)	32	RW	0000_0000
1_4170	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD6_W0)	32	RW	0000_0000
1_4174	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD6_W1)	32	RW	0000_0000
1_4178	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD7_W0)	32	RW	0000_0000
1_417C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD7_W1)	32	RW	0000_0000
1_41C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM1_RGD_NSE)	32	RW	0000_0000
1_4240	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD0_W0)	32	RW	0000_0000
1_4244	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD0_W1)	32	RW	0000_0000
1_4248	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD1_W0)	32	RW	0000_0000
1_424C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD1_W1)	32	RW	0000_0000
1_4250	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD2_W0)	32	RW	0000_0000
1_4254	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD2_W1)	32	RW	0000_0000
1_4258	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD3_W0)	32	RW	0000_0000
1_425C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD3_W1)	32	RW	0000_0000
1_4260	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD4_W0)	32	RW	0000_0000
1_4264	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD4_W1)	32	RW	0000_0000
1_4268	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD5_W0)	32	RW	0000_0000
1_426C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD5_W1)	32	RW	0000_0000
1_4270	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD6_W0)	32	RW	0000_0000
1_4274	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD6_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4278	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD7_W0)	32	RW	0000_0000
1_427C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD7_W1)	32	RW	0000_0000
1_42C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM2_RGD_NSE)	32	RW	0000_0000
1_4340	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD0_W0)	32	RW	0000_0000
1_4344	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD0_W1)	32	RW	0000_0000
1_4348	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD1_W0)	32	RW	0000_0000
1_434C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD1_W1)	32	RW	0000_0000
1_4350	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD2_W0)	32	RW	0000_0000
1_4354	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD2_W1)	32	RW	0000_0000
1_4358	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD3_W0)	32	RW	0000_0000
1_435C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD3_W1)	32	RW	0000_0000
1_4360	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD4_W0)	32	RW	0000_0000
1_4364	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD4_W1)	32	RW	0000_0000
1_4368	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD5_W0)	32	RW	0000_0000
1_436C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD5_W1)	32	RW	0000_0000
1_4370	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD6_W0)	32	RW	0000_0000
1_4374	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD6_W1)	32	RW	0000_0000
1_4378	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD7_W0)	32	RW	0000_0000
1_437C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD7_W1)	32	RW	0000_0000
1_43C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM3_RGD_NSE)	32	RW	0000_0000
1_4440	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD0_W0)	32	RW	0000_0000
1_4444	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD0_W1)	32	RW	0000_0000
1_4448	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD1_W0)	32	RW	0000_0000
1_444C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD1_W1)	32	RW	0000_0000
1_4450	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD2_W0)	32	RW	0000_0000
1_4454	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD2_W1)	32	RW	0000_0000
1_4458	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD3_W0)	32	RW	0000_0000
1_445C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD3_W1)	32	RW	0000_0000
1_4460	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD4_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4464	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD4_W1)	32	RW	0000_0000
1_4468	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD5_W0)	32	RW	0000_0000
1_446C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD5_W1)	32	RW	0000_0000
1_4470	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD6_W0)	32	RW	0000_0000
1_4474	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD6_W1)	32	RW	0000_0000
1_4478	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD7_W0)	32	RW	0000_0000
1_447C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD7_W1)	32	RW	0000_0000
1_44C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM4_RGD_NSE)	32	RW	0000_0000
1_4540	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD0_W0)	32	RW	0000_0000
1_4544	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD0_W1)	32	RW	0000_0000
1_4548	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD1_W0)	32	RW	0000_0000
1_454C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD1_W1)	32	RW	0000_0000
1_4550	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD2_W0)	32	RW	0000_0000
1_4554	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD2_W1)	32	RW	0000_0000
1_4558	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD3_W0)	32	RW	0000_0000
1_455C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD3_W1)	32	RW	0000_0000
1_4560	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD4_W0)	32	RW	0000_0000
1_4564	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD4_W1)	32	RW	0000_0000
1_4568	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD5_W0)	32	RW	0000_0000
1_456C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD5_W1)	32	RW	0000_0000
1_4570	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD6_W0)	32	RW	0000_0000
1_4574	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD6_W1)	32	RW	0000_0000
1_4578	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD7_W0)	32	RW	0000_0000
1_457C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD7_W1)	32	RW	0000_0000
1_45C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM5_RGD_NSE)	32	RW	0000_0000
1_4640	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD0_W0)	32	RW	0000_0000
1_4644	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD0_W1)	32	RW	0000_0000
1_4648	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD1_W0)	32	RW	0000_0000
1_464C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD1_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4650	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD2_W0)	32	RW	0000_0000
1_4654	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD2_W1)	32	RW	0000_0000
1_4658	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD3_W0)	32	RW	0000_0000
1_465C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD3_W1)	32	RW	0000_0000
1_4660	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD4_W0)	32	RW	0000_0000
1_4664	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD4_W1)	32	RW	0000_0000
1_4668	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD5_W0)	32	RW	0000_0000
1_466C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD5_W1)	32	RW	0000_0000
1_4670	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD6_W0)	32	RW	0000_0000
1_4674	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD6_W1)	32	RW	0000_0000
1_4678	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD7_W0)	32	RW	0000_0000
1_467C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD7_W1)	32	RW	0000_0000
1_46C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM6_RGD_NSE)	32	RW	0000_0000
1_4740	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD0_W0)	32	RW	0000_0000
1_4744	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD0_W1)	32	RW	0000_0000
1_4748	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD1_W0)	32	RW	0000_0000
1_474C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD1_W1)	32	RW	0000_0000
1_4750	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD2_W0)	32	RW	0000_0000
1_4754	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD2_W1)	32	RW	0000_0000
1_4758	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD3_W0)	32	RW	0000_0000
1_475C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD3_W1)	32	RW	0000_0000
1_4760	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD4_W0)	32	RW	0000_0000
1_4764	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD4_W1)	32	RW	0000_0000
1_4768	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD5_W0)	32	RW	0000_0000
1_476C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD5_W1)	32	RW	0000_0000
1_4770	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD6_W0)	32	RW	0000_0000
1_4774	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD6_W1)	32	RW	0000_0000
1_4778	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD7_W0)	32	RW	0000_0000
1_477C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD7_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_47C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM7_RGD_NSE)	32	RW	0000_0000
1_4840	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD0_W0)	32	RW	0000_0000
1_4844	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD0_W1)	32	RW	0000_0000
1_4848	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD1_W0)	32	RW	0000_0000
1_484C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD1_W1)	32	RW	0000_0000
1_4850	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD2_W0)	32	RW	0000_0000
1_4854	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD2_W1)	32	RW	0000_0000
1_4858	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD3_W0)	32	RW	0000_0000
1_485C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD3_W1)	32	RW	0000_0000
1_4860	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD4_W0)	32	RW	0000_0000
1_4864	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD4_W1)	32	RW	0000_0000
1_4868	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD5_W0)	32	RW	0000_0000
1_486C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD5_W1)	32	RW	0000_0000
1_4870	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD6_W0)	32	RW	0000_0000
1_4874	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD6_W1)	32	RW	0000_0000
1_4878	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD7_W0)	32	RW	0000_0000
1_487C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD7_W1)	32	RW	0000_0000
1_48C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM8_RGD_NSE)	32	RW	0000_0000
1_4940	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD0_W0)	32	RW	0000_0000
1_4944	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD0_W1)	32	RW	0000_0000
1_4948	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD1_W0)	32	RW	0000_0000
1_494C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD1_W1)	32	RW	0000_0000
1_4950	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD2_W0)	32	RW	0000_0000
1_4954	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD2_W1)	32	RW	0000_0000
1_4958	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD3_W0)	32	RW	0000_0000
1_495C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD3_W1)	32	RW	0000_0000
1_4960	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD4_W0)	32	RW	0000_0000
1_4964	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD4_W1)	32	RW	0000_0000
1_4968	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD5_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_496C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD5_W1)	32	RW	0000_0000
1_4970	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD6_W0)	32	RW	0000_0000
1_4974	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD6_W1)	32	RW	0000_0000
1_4978	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD7_W0)	32	RW	0000_0000
1_497C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD7_W1)	32	RW	0000_0000
1_49C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM9_RGD_NSE)	32	RW	0000_0000
1_4A40	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD0_W0)	32	RW	0000_0000
1_4A44	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD0_W1)	32	RW	0000_0000
1_4A48	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD1_W0)	32	RW	0000_0000
1_4A4C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD1_W1)	32	RW	0000_0000
1_4A50	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD2_W0)	32	RW	0000_0000
1_4A54	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD2_W1)	32	RW	0000_0000
1_4A58	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD3_W0)	32	RW	0000_0000
1_4A5C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD3_W1)	32	RW	0000_0000
1_4A60	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD4_W0)	32	RW	0000_0000
1_4A64	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD4_W1)	32	RW	0000_0000
1_4A68	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD5_W0)	32	RW	0000_0000
1_4A6C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD5_W1)	32	RW	0000_0000
1_4A70	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD6_W0)	32	RW	0000_0000
1_4A74	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD6_W1)	32	RW	0000_0000
1_4A78	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD7_W0)	32	RW	0000_0000
1_4A7C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD7_W1)	32	RW	0000_0000
1_4AC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM10_RGD_NSE)	32	RW	0000_0000
1_4B40	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD0_W0)	32	RW	0000_0000
1_4B44	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD0_W1)	32	RW	0000_0000
1_4B48	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD1_W0)	32	RW	0000_0000
1_4B4C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD1_W1)	32	RW	0000_0000
1_4B50	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD2_W0)	32	RW	0000_0000
1_4B54	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD2_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4B58	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD3_W0)	32	RW	0000_0000
1_4B5C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD3_W1)	32	RW	0000_0000
1_4B60	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD4_W0)	32	RW	0000_0000
1_4B64	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD4_W1)	32	RW	0000_0000
1_4B68	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD5_W0)	32	RW	0000_0000
1_4B6C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD5_W1)	32	RW	0000_0000
1_4B70	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD6_W0)	32	RW	0000_0000
1_4B74	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD6_W1)	32	RW	0000_0000
1_4B78	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD7_W0)	32	RW	0000_0000
1_4B7C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD7_W1)	32	RW	0000_0000
1_4BC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM11_RGD_NSE)	32	RW	0000_0000
1_4C40	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD0_W0)	32	RW	0000_0000
1_4C44	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD0_W1)	32	RW	0000_0000
1_4C48	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD1_W0)	32	RW	0000_0000
1_4C4C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD1_W1)	32	RW	0000_0000
1_4C50	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD2_W0)	32	RW	0000_0000
1_4C54	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD2_W1)	32	RW	0000_0000
1_4C58	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD3_W0)	32	RW	0000_0000
1_4C5C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD3_W1)	32	RW	0000_0000
1_4C60	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD4_W0)	32	RW	0000_0000
1_4C64	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD4_W1)	32	RW	0000_0000
1_4C68	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD5_W0)	32	RW	0000_0000
1_4C6C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD5_W1)	32	RW	0000_0000
1_4C70	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD6_W0)	32	RW	0000_0000
1_4C74	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD6_W1)	32	RW	0000_0000
1_4C78	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD7_W0)	32	RW	0000_0000
1_4C7C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD7_W1)	32	RW	0000_0000
1_4CC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM12_RGD_NSE)	32	RW	0000_0000
1_4D40	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD0_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4D44	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD0_W1)	32	RW	0000_0000
1_4D48	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD1_W0)	32	RW	0000_0000
1_4D4C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD1_W1)	32	RW	0000_0000
1_4D50	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD2_W0)	32	RW	0000_0000
1_4D54	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD2_W1)	32	RW	0000_0000
1_4D58	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD3_W0)	32	RW	0000_0000
1_4D5C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD3_W1)	32	RW	0000_0000
1_4D60	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD4_W0)	32	RW	0000_0000
1_4D64	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD4_W1)	32	RW	0000_0000
1_4D68	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD5_W0)	32	RW	0000_0000
1_4D6C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD5_W1)	32	RW	0000_0000
1_4D70	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD6_W0)	32	RW	0000_0000
1_4D74	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD6_W1)	32	RW	0000_0000
1_4D78	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD7_W0)	32	RW	0000_0000
1_4D7C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD7_W1)	32	RW	0000_0000
1_4DC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM13_RGD_NSE)	32	RW	0000_0000
1_4E40	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD0_W0)	32	RW	0000_0000
1_4E44	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD0_W1)	32	RW	0000_0000
1_4E48	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD1_W0)	32	RW	0000_0000
1_4E4C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD1_W1)	32	RW	0000_0000
1_4E50	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD2_W0)	32	RW	0000_0000
1_4E54	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD2_W1)	32	RW	0000_0000
1_4E58	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD3_W0)	32	RW	0000_0000
1_4E5C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD3_W1)	32	RW	0000_0000
1_4E60	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD4_W0)	32	RW	0000_0000
1_4E64	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD4_W1)	32	RW	0000_0000
1_4E68	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD5_W0)	32	RW	0000_0000
1_4E6C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD5_W1)	32	RW	0000_0000
1_4E70	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD6_W0)	32	RW	0000_0000
1_4E74	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD6_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4E78	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD7_W0)	32	RW	0000_0000
1_4E7C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD7_W1)	32	RW	0000_0000
1_4EC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM14_RGD_NSE)	32	RW	0000_0000
1_4F40	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD0_W0)	32	RW	0000_0000
1_4F44	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD0_W1)	32	RW	0000_0000
1_4F48	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD1_W0)	32	RW	0000_0000
1_4F4C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD1_W1)	32	RW	0000_0000
1_4F50	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD2_W0)	32	RW	0000_0000
1_4F54	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD2_W1)	32	RW	0000_0000
1_4F58	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD3_W0)	32	RW	0000_0000
1_4F5C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD3_W1)	32	RW	0000_0000
1_4F60	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD4_W0)	32	RW	0000_0000
1_4F64	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD4_W1)	32	RW	0000_0000
1_4F68	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD5_W0)	32	RW	0000_0000
1_4F6C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD5_W1)	32	RW	0000_0000
1_4F70	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD6_W0)	32	RW	0000_0000
1_4F74	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD6_W1)	32	RW	0000_0000
1_4F78	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD7_W0)	32	RW	0000_0000
1_4F7C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD7_W1)	32	RW	0000_0000
1_4FC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM15_RGD_NSE)	32	RW	0000_0000

22.7.1.2 TRDC Register (TRDC_CR)

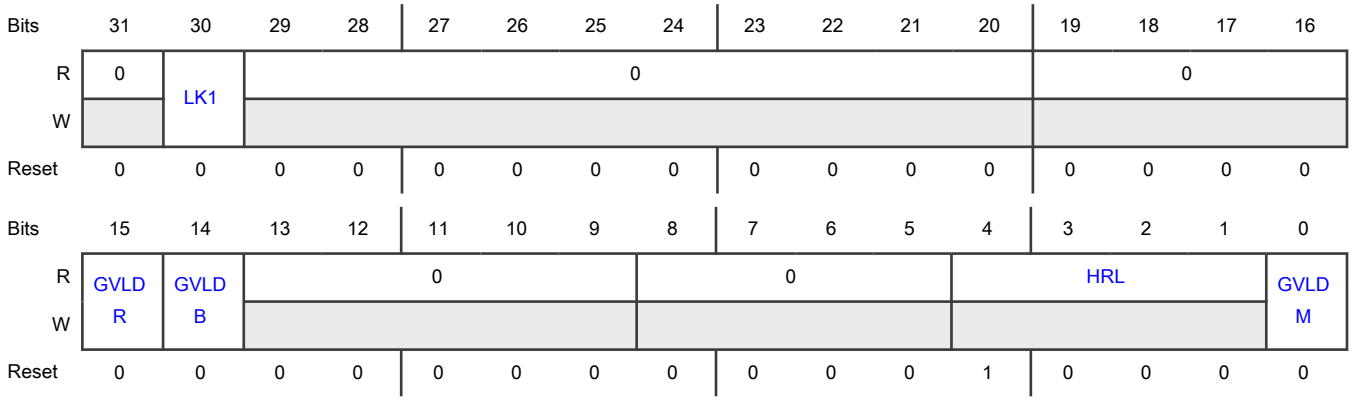
This register provides status about the TRDC and global enable bits for the entire module's operation. A fully operational TRDC requires all global bits GVLD {R,B,M} to be asserted. Undefined behavior results if they are not all asserted. If there are no region checkers (MRCs) present in the configuration, GVLDR need not be asserted for a fully operational TRDC.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_CR	0h

Diagram



Fields

Field	Description
31 —	Reserved
30 LK1	Lock Status This field is the lock status of the TRDC_CR. This field is set when all GVLD bits are set; GVLDR=GVLDB=GVLDM=1b1. Once set, this bit remains asserted until the next reset. 0 - The CR can be written by any secure privileged write. 1 - The CR is locked (read-only) until the next reset.
29-20 —	Reserved
19-16 —	Reserved
15 GVLDR	Global Valid for Memory Region Checkers TRDC global MRC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC MRCs are disabled. 1 - TRDC MRCs are enabled.
14 GVLDB	Global Valid for Memory Block Checkers TRDC global MBC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC MBCs are disabled. 1 - TRDC MBCs are enabled.
13-9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
8-5 —	Reserved
4-1 HRL	Hardware Revision Level This read-only field specifies the TRDC's hardware and definition revision level. It can be read by software to determine the functional definition of the module.
0 GVLDM	Global Valid for Domain Assignment Controllers TRDC global DAC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC DACs are disabled. 1 - TRDC DACs are enabled.

22.7.1.3 TRDC Hardware Configuration Register 0 (TRDC_HWCFG0)

This read-only register contains information on the TRDC's hardware configuration. Specifically, it defines the number of implemented domains and bus initiators along with the number of instances of memory block checkers (MBCs) and memory region checkers (MRCs). The register value at reset is device-specific. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG0	F0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MID				NMRC				0				NMBC			
W																
Reset	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NMSTR								0				NDID			
W																
Reset	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0

Fields

Field	Description
31-29 MID	Module ID This field defines major version ID of this module.
28-24 NMRC	Number of MRCs This field defines the number of MRCs on the device [1-16].
23-20 —	Reserved
19-16 NMBC	Number of MBCs This field defines the number of MBCs on the device [1-8].
15-8 NMSTR	Number of bus initiators This read-only field defines the number of bus initiators.
7-5 —	Reserved
4-0 NDID	Number of domains This read-only field defines the number of domains on the device [1-16].

22.7.1.4 TRDC Hardware Configuration Register 1 (TRDC_HWCFG1)

This register contains information on the TRDC's hardware configuration. It provides a mechanism for software to determine its domain number by simply reading the register. See Domain error capture management for more details on typical usage. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG1	F4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												DID			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	u	u	u	u

Fields

Field	Description
31-4 —	Reserved
3-0 DID	Domain identifier number This field provides the domain number [0-15] of the requesting bus initiator.

22.7.1.5 TRDC Hardware Configuration Register 2 (TRDC_HWCFG2)

This register contains information on the TRDC's hardware configuration. It provides a bitmap signaling the presence of a process identifier (PID) register sourced from the given bus initiator. The HWCFG2 register is associated with bus initiators [31-0]. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG2	F8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PIDPn															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIDPn															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PIDPn	Process identifier present Process identifier present from bus initiator n, where n = [31-0]. This field provides a bitmap to signal that bus initiator (where n is defined as bit number) sources a process identifier register to the TRDC_DAC logic. For each bit in this field: • 0 - Bus initiator n does not source a process identifier register. The TRDC_DAC logic provides the needed PID for processor cores. • 1 - Bus initiator n sources a process identifier register.

22.7.1.6 TRDC Hardware Configuration Register 3 (TRDC_HWCFG3)

This register contains information on the TRDC's hardware configuration. Specifically, it provides a bitmap signaling the presence of a process identifier (PID) register sourced from the given bus initiator. The HWCFG3 register is associated with bus initiators [63-32], while the HWCFG2 register covers bus initiators [31-0]. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG3	FCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PIDPn															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIDPn															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PIDPn	Process identifier present Process identifier present from bus initiator n, where n = [63-32]. This field provides a bitmap to signal that bus initiator n (where n is defined as 32 + bit number) sources a process identifier register to the TRDC_DAC logic.

Table continues on the next page...

Field	Description
	Process identifier present from bus initiator n, where n = [31-0]. This field provides a bitmap to signal that bus initiator (where n is defined as bit number) sources a process identifier register to the TRDC_DAC logic. For each bit in this field: • 0 - Bus initiator n does not source a process identifier register. The TRDC_DAC logic provides the needed PID for processor cores. • 1 - Bus initiator n sources a process identifier register.

22.7.1.7 Domain Assignment Configuration Register (DACFG1 - DACFG4)

This register defines the number of implemented domain assignment registers for bus initiator m, where m+1 can specify from 1 to 64 bus initiators. These registers are organized as a byte-sized data array and can be read using 8-, 16- or 32-bit accesses. An all-zero value (NCM = 0, NMDAR = 0) indicates a non-existent bus initiator. Attempted writes are error terminated.

Register read will not return transfer error when DACFG for a initiator doesn't exist but it is in the same 32-bit group as DACFG for an existing initiator. For example, when there are only 2 DACFG0-1 registers for 2 initiator, and initiators for DACFG2-3 don't exist, then access to DACFG2-3 won't return transfer error.

Typically, processor bus initiators have one or more domain assignment registers, while non-processor initiators have a single domain assignment register.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
DACFG1	101h
DACFG2	102h
DACFG3	103h
DACFG4	104h

Diagram

Bits	7	6	5	4	3	2	1	0
R	NCM	0			NMDAR			
W								
Reset	See Register reset values.							

Register reset values

Register	Reset value
DACFG1	03h
DACFG2–DACFG4	81h

Fields

Field	Description
7 NCM	Non-CPU Initiator This read-only field signals that bus initiator m is a non-CPU initiator. It specifies that the format of the associated MDA_Wr_m register defines a non-processor domain assignment. This field is zero for a non-existent bus initiator. 0 - Bus initiator is a processor. 1 - Bus initiator is a non-processor.
6-4 —	Reserved
3-0 NMDAR	Number of initiator domain assignment registers for bus initiator m This read-only field specifies the number of registers associated with the initiator domain assignment register for a given bus initiator. The value is limited to the range [0-8], where zero indicates a non-existent bus initiator and non-zero values indicate the number of implemented registers associated with this MDAm.

22.7.1.8 TRDC IDAU Control Register (TRDC_IDAU_CR)

This register defines the configuration for the Implementation-Defined Attribution Unit which is tightly-coupled to the TZ-M extensions in the processor core. Many of the fields in this register explicitly control processor TZ-M functions.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_IDAU_CR	1C0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PCURRNS	0	0	0	0	0	0						0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			LKSAU	LKNSMPU	LKSMPU	LKNSVTOR	LKSVTAI...	0	SAUDIS	MPUNSDIS	MPUSDIS	CFGSECE...	0		VLD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Fields

Field	Description
31 PCURRNS	Processor current security Current security state of the processor. 0 - Processor is in Secure state 1 - Processor is in Nonsecure state
30 —	Reserved
29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25-19 —	Reserved
18-16 —	Reserved
15-13 —	Reserved
12 LKSAU	Lock SAU Asserting this bit prevents changes to the processor's Secure SAU memory regions already programmed. All processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the SAU_CTRL, SAU_RNR, SAU_RBAR and SAU_RLAR registers from software or from a debug agent connected to the processor
11 LKNSMPU	Lock Nonsecure MPU Asserting this bit prevents changes to the processor's Nonsecure MPU memory regions already programmed. All processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Disable writes to the MPU_CTRL_NS, MPU_RNR_NS, MPU_RBAR_NS, MPU_RLAR_NS, MPU_RBAR_A_NSn and MPU_RLAR_A_NSn from software or from a debug agent connected to the processor
10 LKSMPU	Lock Secure MPU Asserting this signal prevents changes to the processor's programmed Secure MPU memory regions and all processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the MPU_CTRL, MPU_RNR, MPU_RBAR, MPU_RLAR, MPU_RBAR_An and MPU_RLAR_An from software or from a debug agent connected to the processor in Secure state
9 LKNSVTOR	Lock Nonsecure Vector Table Offset Register Asserting this signal prevents changes to the processor's Nonsecure vector table base address. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock this register 1 - Disable writes to the VTOR_NS register
8 LKSVTAIRCR	Lock Secure VTOR, Application interrupt and Reset Control Registers This bit is sticky, once set, it can only be cleared with a reset. Asserting this signal prevents processor changes to: • The Secure vector table base address. • Handling of Secure interrupt priority. • BusFault, HardFault, and NMI security target settings in the processor. 0 - Unlock these registers 1 - Disable writes to the VTOR_S, AIRCR[PRIS], and AIRCR[BFHFNMINS] registers
7 —	Reserved
6 SAUDIS	Security Attribution Unit Disable 0 - SAU is enabled 1 - SAU is disabled
5 MPUNSDIS	NonSecure Memory Protection Unit Disabled 0 - Nonsecure MPU is enabled 1 - Nonsecure MPU is disabled
4 MPUSDIS	Secure Memory Protection Unit Disabled 0 - Secure MPU is enabled 1 - Secure MPU is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
3 CFGSECEXT	Configure Security Extension 0 - Armv8M Security Extension is disabled 1 - Armv8-M Security Extension is enabled
2-1 —	Reserved
0 VLD	Valid When VLD =0, all address attribute from IDAU is nonSecure. When VLD=1, values in other fields of this register are valid.

22.7.1.9 TRDC FLW Control (TRDC_FLW_CTL)

This register provides control of the FLW = Flash Logical Window operation.

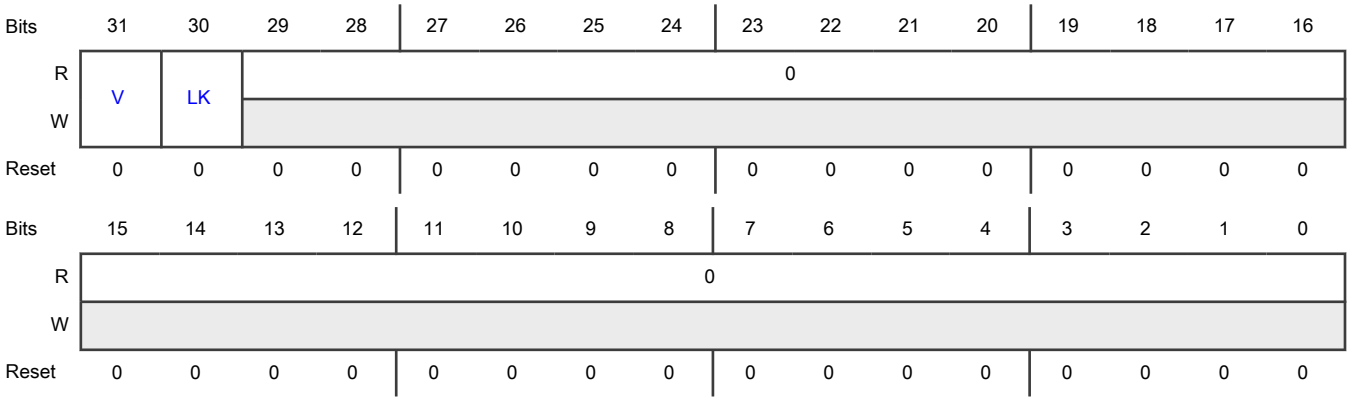
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_FLW_CTL	1E0h

Diagram



Fields

Field	Description
31 V	Valid bit 0 - FLW function is disabled.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - FLW function is enabled.
30 LK	Lock bit 0 - FLW registers may be modified. 1 - FLW registers are locked until the next reset.
29-0 —	Reserved

22.7.1.10 TRDC FLW Physical Base (TRDC_FLW_PBASE)

This read-only register gives the physical base address of the Flash Logical Window.

This register exists for this device but has not effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_FLW_PBASE	1E4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PBASE															
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PBASE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PBASE	Physical base address Physical address of the base of the FLW (mod 32KBytes).

22.7.1.11 TRDC FLW Array Base (TRDC_FLW_ABASE)

This register gives the flash array base address of the Flash Logical Window.

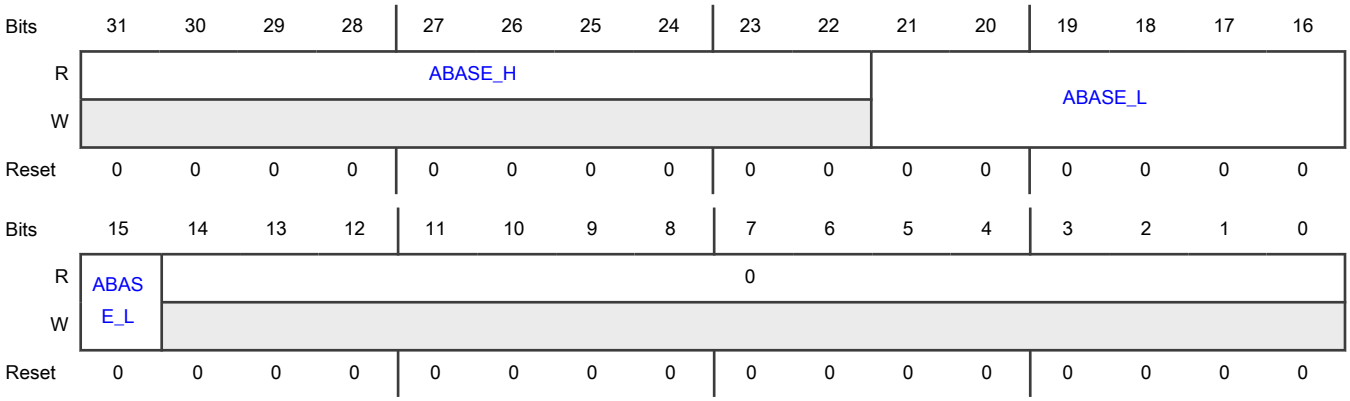
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write. This register is not writable if TRDC_FLW_CTL[LK]=1.

Offset

Register	Offset
TRDC_FLW_ABASE	1E8h

Diagram



Fields

Field	Description
31-22 ABASE_H	Array base address high Flash array address upper bits of the base of the FLW (mod 32KBytes).
21-15 ABASE_L	Array base address low Flash array address lower bits of the base of the FLW (mod 32KBytes).
14-0 —	Reserved

22.7.1.12 TRDC FLW Block Count (TRDC_FLW_BCNT)

This register gives the size of the Flash Logic Window in 32KByte blocks.

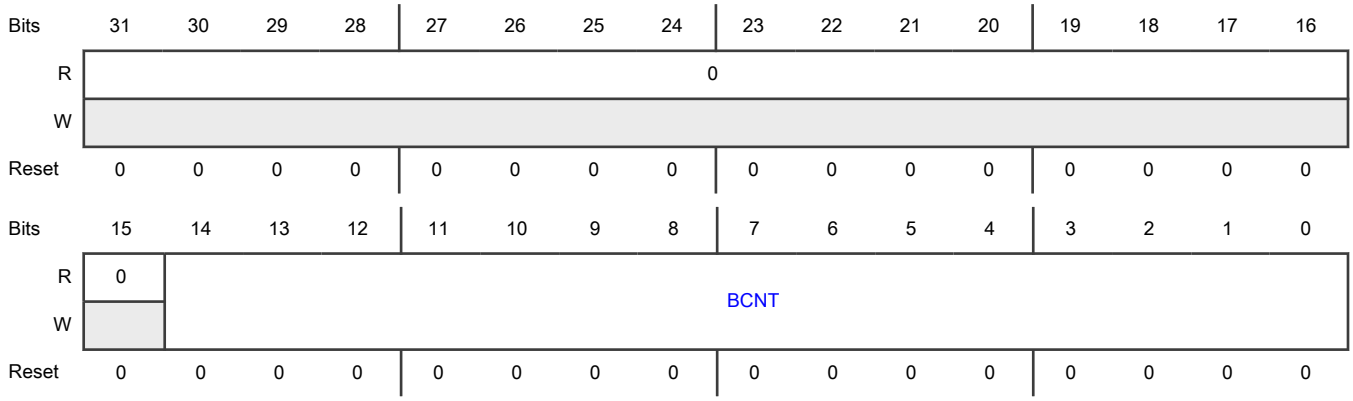
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write. This register is not writable if TRDC_FLW_CTL[LK]=1.

Offset

Register	Offset
TRDC_FLW_BCNT	1ECh

Diagram



Fields

Field	Description
31-15 —	Reserved
14-0 BCNT	Block Count Size of FLW in number of 32KByte blocks.

22.7.1.13 TRDC Fault Domain ID (TRDC_FDID)

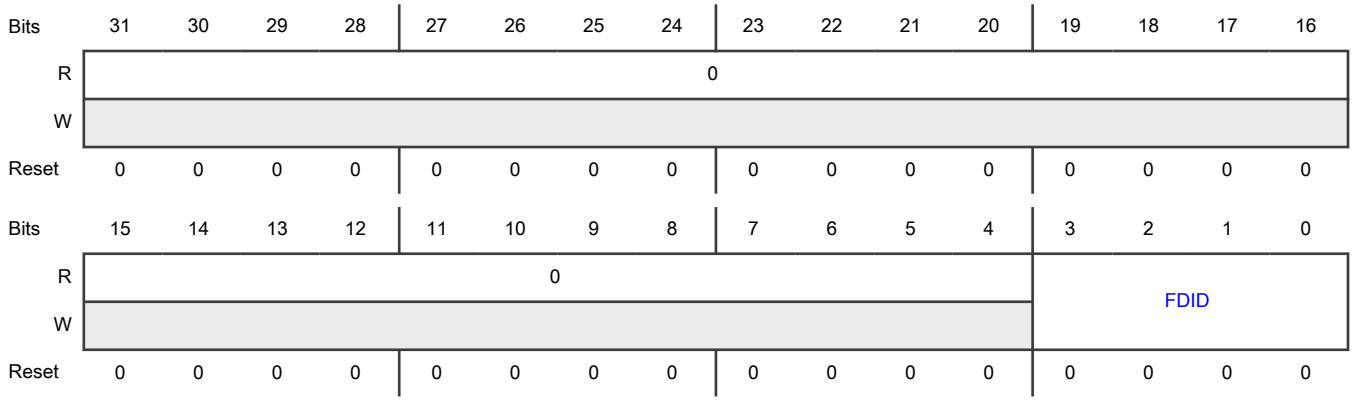
In the event of an access error, this register is used to specify the domainID of the faulting reference before indexing into the Domain Error registers.

If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_FDID	1FCh

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 FDID	Domain ID of Faulted Access This field indicates the domainID of the fault used to index the array of domain error information. The user queries the DERRLOCx registers to find the DomainID of the faulting access and must write this register with the domain ID before reading the Domain Error registers.

22.7.1.14 TRDC Domain Error Location Register (TRDC_DERRLOC0 - TRDC_DERRLOC15)

This array of read-only registers provide the instance number of the submodule where (an) access violation(s) occurred. These registers are organized as a word array, indexed by the faulting domain number, d. The two fields of this register provide a bitmap of instances associated with all submodules containing captured error information for that domain. These instance numbers are then used as indices into the DERR_W0_*i*, DERR_W1_*i*, and DERR_W3_*i*register arrays. See Domain error capture management for more details.

When an access violation is detected by either a Memory Region Checker (MRC) or a Memory Block Checker (MBC), address and attribute information of the offending access is captured. Using the faulting domainID number as the index, d, this array of read-only registers provide additional information signaling the instance number of the submodule where the access violation(s) occurred. Since the resulting exception handler needs the submodule instance to retrieve the captured address and attribute information from DERR_W0_*i* and DERR_W1_*i*, these registers provide the instance number details.

Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

For d = 0 to 15:

Register	Offset
TRDC_DERRLOCd	200h + (d × 4h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MRCINST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								MBCINST							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 MRCINST	MRC instance This field is a bitmap indicating the presence of a detected access violation for domain d in the implemented instances of the MRC. The least-significant bit of this field (bit 16) corresponds to MRC instance 0. The most-significant bit of this field (bit 31) corresponds to MRC instance 15 (MRC instance = i-16 where i is the register bit number), and so on. Multiple bits can be set at any time indicating access violations for this domain have been detected across multiple instances of the MRCs. For each bit in this field: • 0 - The memory region checker has not detected an access violation or is not physically present. • 1 - The memory region checker has detected one or more access violations for this domain.
15-8 —	Reserved
7-0 MBCINST	MBC instance This field is a bitmap indicating the presence of a detected access violation for domain d in the implemented instances of the MBC. The least-significant bit of this field (bit 0) corresponds to MBC instance 0. The most-significant bit of this field (bit 7) corresponds to MBC instance 7 Multiple bits can be set at any time indicating access violations for this domain have been detected across multiple instances of the MBCs. For each bit in this field: • 0 - The memory block checker has not detected an access violation or is not physically present. • 1 - The memory block checker has detected one or more access violations for this domain.

22.7.1.15 MBC Domain Error Word0 Register (MBC0_DERR_W0 - MBC1_DERR_W0)

This read-only register array provides the address of an access violation detected by memory block checker (MBC). These registers are organized as a word array, which is indexed by the MBC instance number. That is, the index, i, of this array is the instance number of MBC with the access violation. The submodule instance numbers are provided by the DERRLOC registers. The memory mapped error capture detail registers are organized as 24 sequential 16 byte entries.

When an access violation is detected and the offending information captured, subsequent updates to this register are disabled until the required data pattern is written to the DERR_W3_i register. At that time, this register is cleared and re-enabled to capture the next access violation.

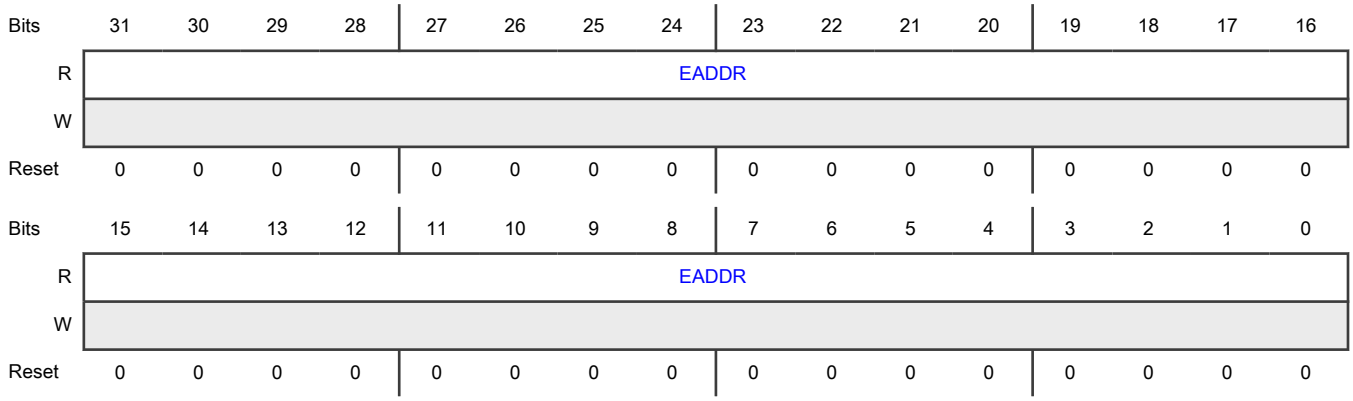
Attempted writes are error terminated as are attempted reads of an MBC instance that is not physically present.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
MBC0_DERR_W0	400h
MBC1_DERR_W0	410h

Diagram



Fields

Field	Description
31-0 EADDR	Error address This is the unaliased virtual address of the access that generated the access violation.

22.7.1.16 MBC Domain Error Word1 Register (MBC0_DERR_W1 - MBC1_DERR_W1)

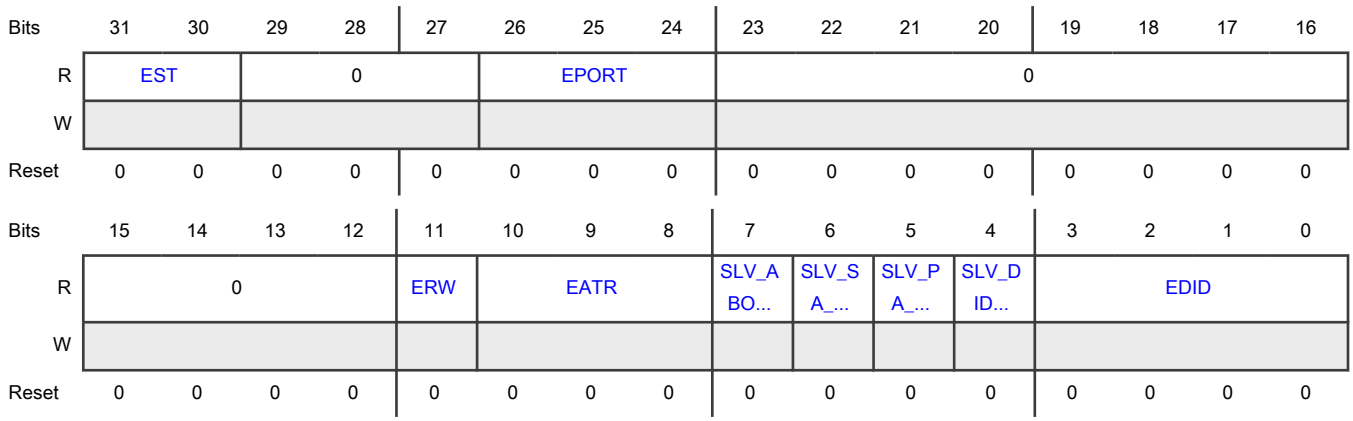
This read-only register array provides the attributes of an access violation detected by Memory Block Checker (MBC). These registers are organized as a word array, which is indexed by the violating submodule instance number. Refer to register Domain Error Location(DERRLOCd), Domain Error Word0 Register(DERR_W0_i), and Domain error capture management for more information.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
MBC0_DERR_W1	404h
MBC1_DERR_W1	414h

Diagram



Fields

Field	Description
31-30 EST	Error state This field signals the state of access violations for this domain in this instance of the block checker or region checker. Once an access violation has been detected and the faulting address and attribute information stored, subsequent errors are simply recorded as an overrun condition without any data captured. After retrieving the faulting address and attributes, the error capture mechanism must be rearmed by performing a write to DERR_W3_i. 00 - No access violation has been detected. 01 - No access violation has been detected. 10 - A single access violation has been detected. 11 - Multiple access violations for this domain have been detected by this submodule instance. Only the address and attribute information for the first error have been captured in DERR_W0_i and DERR_W1_i.
29-27 —	Reserved
26-24 EPORT	Error port This field identifies the encoded port number of the MBC that detected the access violation. The MBC port number connection is device-specific. See the chip configuration details for more information. This is the MBC target that has the violation. 000 - mbcxslv0 001 - mbcxslv1 010 - mbcxslv2 011 - mbcxslv3
23-12 —	Reserved
11 ERW	Error read/write This field signals whether the captured access violation occurred on a read or write reference. 0 - Read access 1 - Write access
10-8 EATR	Error attributes This field captures attributes of the access violation. 000 - Secure user mode, instruction fetch access. 001 - Secure user mode, data access. 010 - Secure privileged mode, instruction fetch access. 011 - Secure privileged mode, data access. 100 - Nonsecure user mode, instruction fetch access.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - Nonsecure user mode, data access. 110 - Nonsecure privileged mode, instruction fetch access. 111 - Nonsecure privileged mode, data access.
7 SLV_ABORT	Bus protect error This field captures the bus protect error from target which cause access abort.
6 SLV_SA_ERR	Secure attribute check error This field captures the secure attribute check error from target.
5 SLV_PA_ERR	Privilege attribute check error This field captures the privilege attribute check error from target.
4 SLV_DID_ERR	DID check error This field captures the DID check error from target.
3-0 EDID	Error domain identifier This field captures the domain identifier of the access violation.

22.7.1.17 MBC Domain Error Word3 Register (MBC0_DERR_W3 - MBC1_DERR_W3)

This register is used to rearm the error capture logic and clear the DERR_W0_i and DERR_W1_i registers. After the domain access violation error details have been read, typically in an exception service routine, a 32-bit word write to this register is required to rearm the error capture logic.

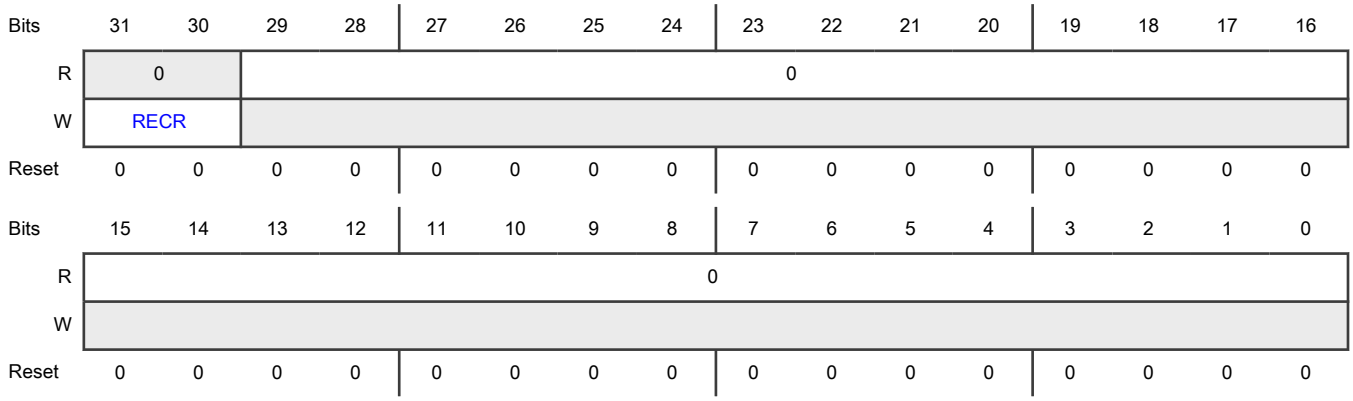
A read of this location returns zeroes. Attempted reads of a Memory Region Checker (MRC) or Memory Block Checker (MBC) instance that is not physically present are error terminated.

See Domain error capture management for more details.

Offset

Register	Offset
MBC0_DERR_W3	40Ch
MBC1_DERR_W3	41Ch

Diagram



Fields

Field	Description
31-30 RECR	Rearm Error Capture Registers This 2-bit, write-only field controls the rearming of the domain error capture registers. Once an access violation has been detected with the faulting address and attribute information stored, subsequent errors are simply recorded as an overrun condition without any data captured. Writing 01b to this field rearms the error capture mechanism and clears the DERR_W0_i and DERR_W1_i registers. A write of any value other than 01b has no effect.
29-0 —	Reserved

22.7.1.18 MRC Domain Error Word0 Register (MRC0_DERR_W0)

This read-only register array provides the address of an access violation detected by memory region checker (MRC). These registers are organized as a word array, which is indexed by the MRC instance number. That is, the index, i, of this array is the instance number of MRC with the access violation. The submodule instance numbers are provided by the DERRLOC registers. The memory mapped error capture detail registers are organized as 24 sequential 16 byte entries.

When an access violation is detected and the offending information captured, subsequent updates to this register are disabled until the required data pattern is written to the DERR_W3_i register. At that time, this register is cleared and re-enabled to capture the next access violation.

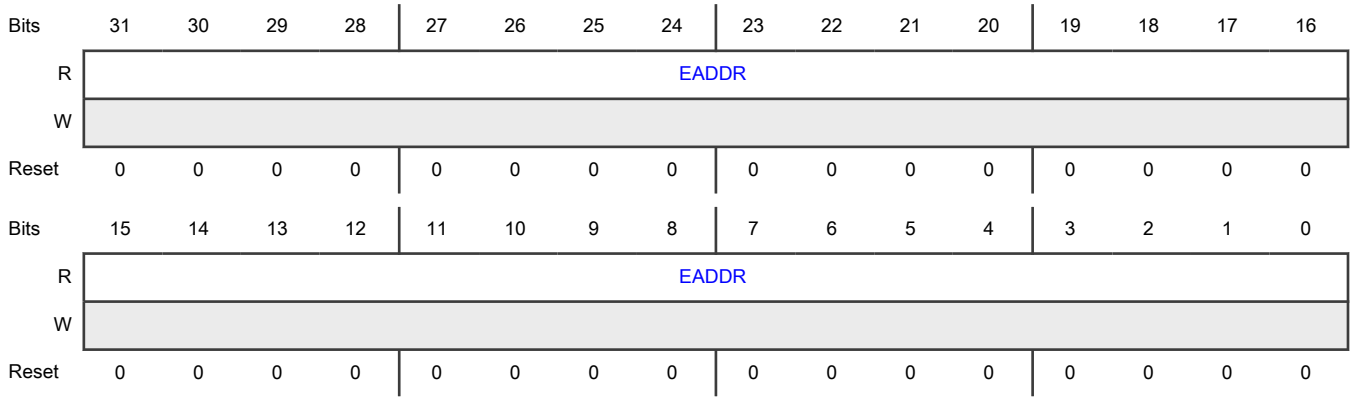
Attempted writes are error terminated as are attempted reads of an MRC instance that is not physically present.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
MRC0_DERR_W0	480h

Diagram



Fields

Field	Description
31-0 EADDR	Error address This is the unaliased virtual address of the access that generated the access violation.

22.7.1.19 MRC Domain Error Word1 Register (MRC0_DERR_W1)

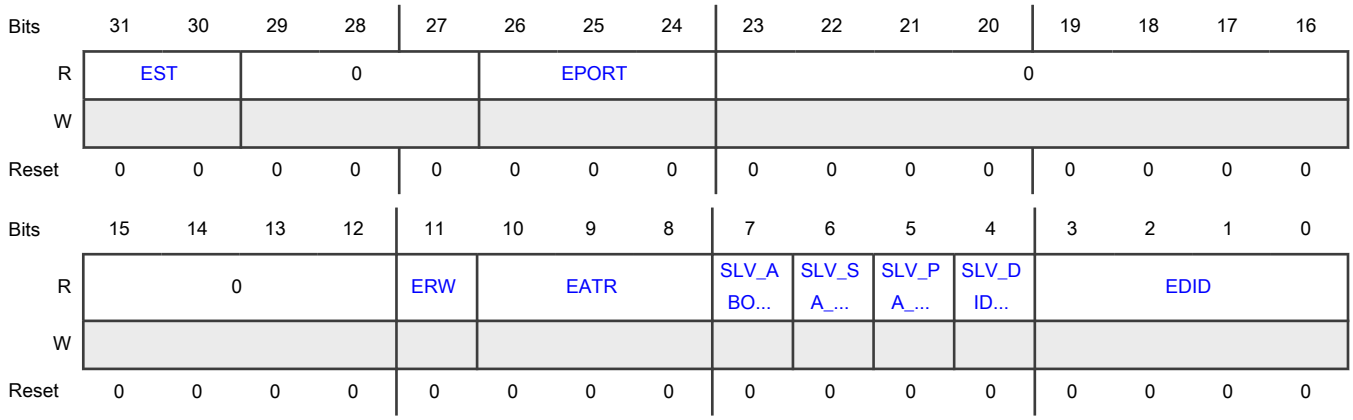
This read-only register array provides the attributes of an access violation detected by memory region checker (MRC). These registers are organized as a word array, which is indexed by the violating submodule instance number. Refer to register Domain Error Location(DERRLOCd), Domain Error Word0 Register(DERR_W0_i), and Domain error capture management for more information.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
MRC0_DERR_W1	484h

Diagram



Fields

Field	Description
31-30 EST	Error state This field signals the state of access violations for this domain in this instance of the block checker or region checker. Once an access violation has been detected and the faulting address and attribute information stored, subsequent errors are simply recorded as an overrun condition without any data captured. After retrieving the faulting address and attributes, the error capture mechanism must be rearmed by performing a write to DERR_W3_i. 00 - No access violation has been detected. 01 - No access violation has been detected. 10 - A single access violation has been detected. 11 - Multiple access violations for this domain have been detected by this submodule instance. Only the address and attribute information for the first error have been captured in DERR_W0_i and DERR_W1_i.
29-27 —	Reserved
26-24 EPORT	Error port This field identifies the encoded port number of the MRC that detected the access violation. The MRC port number connection is device-specific. See the chip configuration details for more information.
23-12 —	Reserved
11 ERW	Error read/write This field signals whether the captured access violation occurred on a read or write reference. 0 - Read access 1 - Write access
10-8 EATR	Error attributes This field captures certain attributes of the access violation. 000 - Secure user mode, instruction fetch access. 001 - Secure user mode, data access. 010 - Secure privileged mode, instruction fetch access. 011 - Secure privileged mode, data access. 100 - Nonsecure user mode, instruction fetch access. 101 - Nonsecure user mode, data access. 110 - Nonsecure privileged mode, instruction fetch access. 111 - Nonsecure privileged mode, data access.
7	Bus protect error

Table continues on the next page...

Table continued from the previous page...

Field	Description
SLV_ABORT	This field captures the bus protect error from target which cause access abort.
6 SLV_SA_ERR	Secure attribute check error This field captures the secure attribute check error from target.
5 SLV_PA_ERR	Privilege attribute check error This field captures the privilege attribute check error from target.
4 SLV_DID_ERR	DID check error This field captures the DID check error from target.
3-0 EDID	Error domain identifier This field captures the domain identifier of the access violation.

22.7.1.20 MRC Domain Error Word3 Register (MRC0_DERR_W3)

This register is used to rearm the error capture logic and clear the DERR_W0_i and DERR_W1_i registers. After the domain access violation error details have been read, typically in an exception service routine, a 32-bit word write to this register is required to rearm the error capture logic.

A read of this location returns zeroes. Attempted reads of a memory region checker (MRC) or Memory Block Checker (MBC) instance that is not physically present are error terminated.

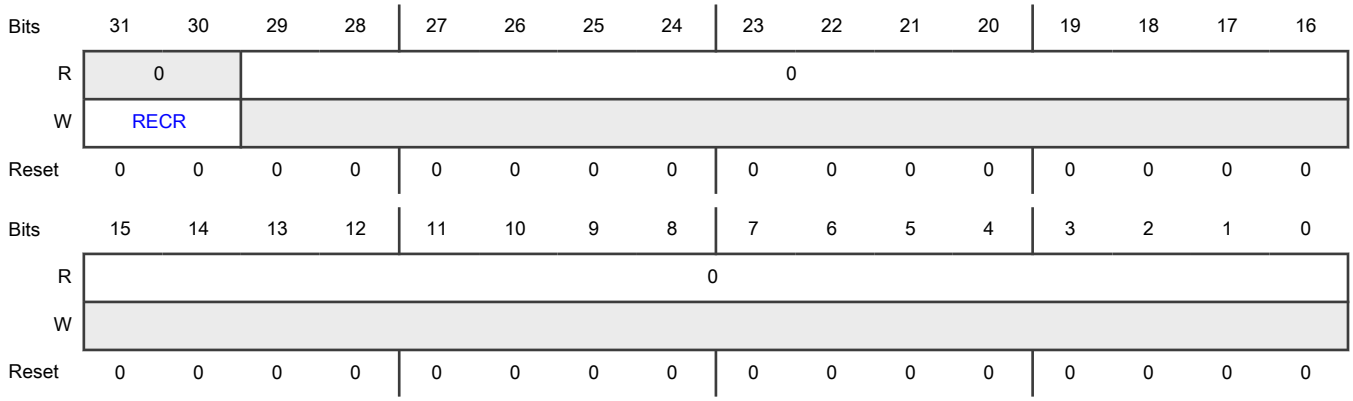
See Domain error capture management for more details.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
MRC0_DERR_W3	48Ch

Diagram



Fields

Field	Description
31-30 RECR	Rearm Error Capture Registers This 2-bit, write-only field controls the rearming of the domain error capture registers. Once an access violation has been detected with the faulting address and attribute information stored, subsequent errors are simply recorded as an overrun condition without any data captured. Writing 01b to this field rearms the error capture mechanism and clears the DERR_W0_i and DERR_W1_i registers. A write of any value other than 01b has no effect.
29-0 —	Reserved

22.7.1.21 Process Identifier (PID1)

In the TRDC's access control definition, each processor has a corresponding process identifier (PID) which performs the important functions: the entire value can be used to group tasks into different domains.

While certain processors include this register in their programming model definitions, others do not. This register is provided for those processors that do *not* include a PID register in their programming models. Secure privileged software saves and restores the PID as part of any context switch.

This data structure defines an array of 32-bit values, one per MDA module, that define the PID. Since this register resource is only applicable to processor cores, the data structure is typically sparsely populated. The HWCFG[2-3] registers provide a bitmap of the implemented PID*n* registers. This data structure is indexed using the corresponding MDA instance number.

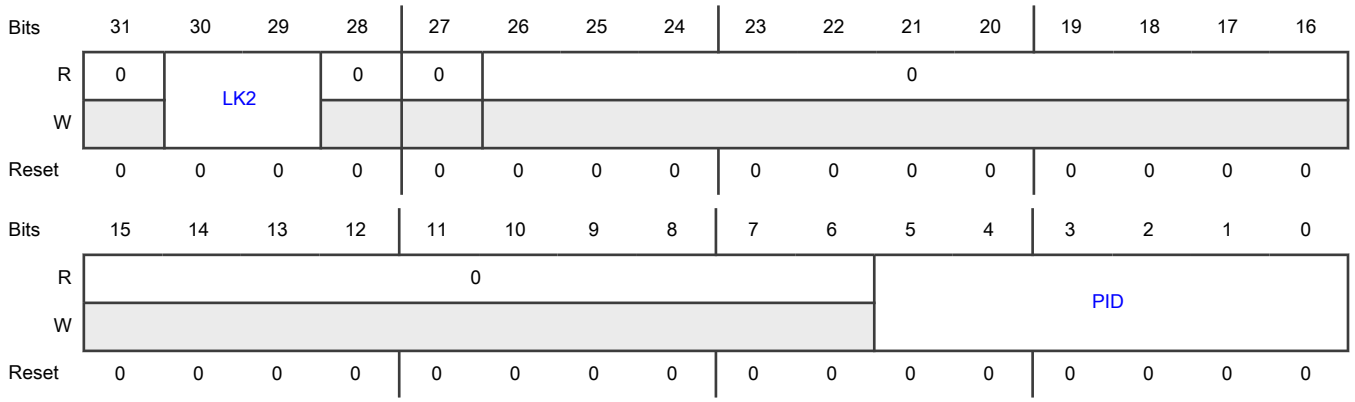
Depending on the operating clock domain of each DAC instance, there may be optional information stored in the corresponding PIDm register to properly implement the LK2 = 2 functionality.

Reads of the PID*n* register return the contents of this register, or the PID*n* value directly sourced from a processor. For non-processor bus initiators, this register does not exist and any attempted read is error terminated.

Offset

Register	Offset
PID1	704h

Diagram



Fields

Field	Description
31 —	Reserved
30-29 LK2	Lock This field provides a locking mechanism that can be used to limit the ability to write the register. Once set, these bits individually remain asserted until the next reset. If the processor sources its PIDm directly, that is, HWCFG{2-3}[PIDPm = 1], then privileged reads of this memory location return zero for this field. 00 - Register can be written by any secure privileged write. 01 - Register can be written by any secure privileged write. 10 - Register can only be written by a secure privileged write from the bus initiator that locked the register. 11 - Register is locked (read-only) until the next reset.
28 —	Reserved
27 —	Reserved
26-6 —	Reserved
5-0 PID	Process identifier This field determines the secure (0) or nonsecure (1) attribute for transactions associated with the corresponding processor. If the processor sources its PID*n* directly, then secure privileged reads of this memory location return the processor register for this field.

22.7.1.22 DAC Initiator Domain Assignment Register (MDA_W0_1_DFMT0 - MDA_W2_1_DFMT0)

The MDA_Wr_m registers provide a 2-dimensional data structure for assigning bus initiators to domains. The number of implemented registers is defined by DACFGm[NMDAR]. This per-initiator domain assignment is then repeated for each bus initiator (MDAm). Thus, m specifies the initiator number and r refers to the specific MDA register for a given bus initiator.

Each Wr within the MDAm structure is a word-sized definition; there are two formats supported, one for processor cores and another for non-processors. Processor initiators typically support one or more Wr domain definitions, while non-processor initiators support a single Wr. Processor initiators Domain Assignment register names use a _DFMT0 suffix. Non-processor initiators Domain Assignment register names use a _DFMT1 suffix.

The DAC submodule is responsible for the generation of domain identifiers for every transaction from every bus initiator. If there is a single Wr for a given initiator, then the specified domain identifier is used directly. If there are multiple Wr values for a given initiator, then the DAC evaluates the conditional terms to determine a “hit”. For all Wr hits, their corresponding domain identifiers are simply logically summed together (boolean OR). Use cases are typically expected to *hit in a single Wr* for a processor initiator. Special care is needed if *none* of the conditional terms hit in any Wr evaluation; for this case, the generated DID = 0 and software needs to be aware of any potential access rights granted for this DID.

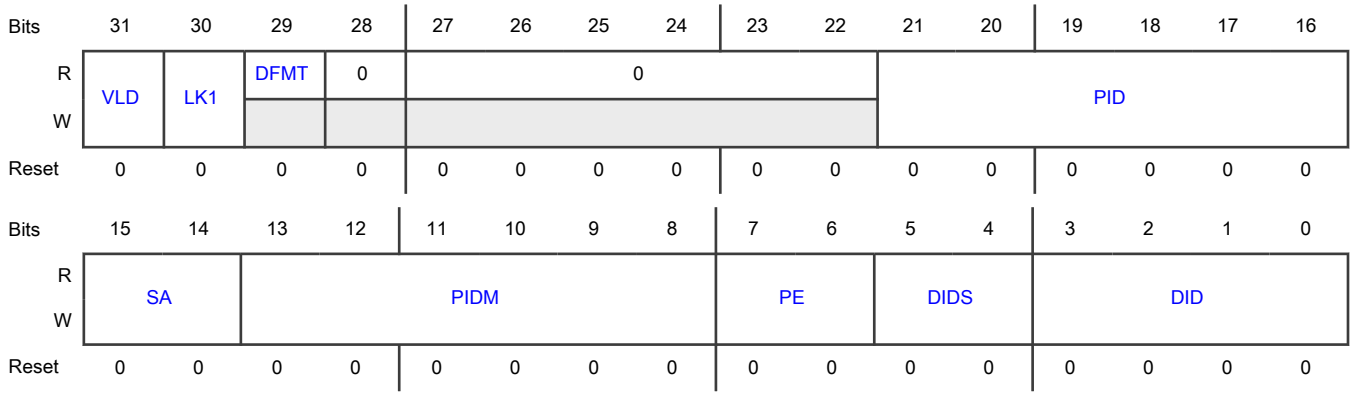
Each MDA_Wr_m register has one of two programming models depending on the state of the domain format field, DFMT. The model described in this section is for DFMT = 0. This definition allows three different specifications of the DID for processors. The DFMT = 1 model is described in Initiator Domain Assignment (MDA_Wr_m_DFMT1) register.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/write.

Offset

Register	Offset
MDA_W0_1_DFMT0	820h
MDA_W1_1_DFMT0	824h
MDA_W2_1_DFMT0	828h

Diagram



Fields

Field	Description
31 VLD	Valid This field indicates the domain assignment is valid. It is further qualified by TRDC_CR[GVLDM] = 1. If TRDC_CR[GVLDM] is cleared, the DID output is defined by the SoC-specific *default DID* value. See the TRDC chip configuration section for more details. If both MDA_Wr_m[VLD] and TRDC_CR[GVLDM] are asserted, the DID output is defined by the remaining contents of this register. 0 - The Wr domain assignment is invalid. 1 - The Wr domain assignment is valid.
30 LK1	1-bit Lock This field provides a locking mechanism that can be used to limit the ability to write the register. Once set, this bit remains asserted until the next reset. 0 - Register can be written by any secure privileged write. 1 - Register is locked (read-only) until the next reset.
29	Domain format

Table continues on the next page...

Table continued from the previous page...

Field	Description
DFMT	Identifies this register's domain assignment format. **NOTE** This bitfield access is ROZ 0 - Processor-core domain assignment 1 - Non-processor domain assignment
28 —	Reserved
27-22 —	Reserved
21-16 PID	Process Identifier This field specifies that the process identifier is to be combined with the PIDM field and included in the domain hit determination. The optional inclusion of the PID and PIDM is controlled by the MDA_Wr_m[PE] field.
15-14 SA	Secure attribute This field defines the secure/nonsecure attribute for processor cores. **NOTE** The bus initiator’s input secure/nonsecure attribute is used if SA = 1X, or this VLD = 0. If TZM_ENB = 0, the initiator attribute is forced to nonsecure. A nonsecure write cannot program this field to 2'b00, which is a security level higher than the mode of the process that is writing it. Reset value of SA: • if TZM_ENB=0, SA reset value = 2'b01 • if TZM_ENB=1, SA reset value = 2'b00 00 - Force the bus attribute for this initiator to secure. 01 - Force the bus attribute for this initiator to nonsecure. 10 - Use the bus initiator’s secure/nonsecure attribute directly. 11 - Use the bus initiator’s secure/nonsecure attribute directly.
13-8 PIDM	Process Identifier Mask This field provides a masking capability so that multiple process identifiers can be included as part of the domain hit determination. If a bit in the PIDM is set, then the corresponding bit of the PID is ignored in the comparison. The optional inclusion of the PID and PIDM is controlled by the MDA_Wr_m[PE] field.
7-6 PE	Process identifier enable Controls the optional inclusion of the PID, qualified by PIDM, into the domain hit evaluation. It provides the ability to include inclusive or exclusive sets of masked PID values.

Table continues on the next page...

Table continued from the previous page...

Field	Description

	00 - No process identifier is included in the domain hit evaluation.
	01 - No process identifier is included in the domain hit evaluation.
	10 - PE = 2. The process identifier is included in the domain hit evaluation as defined by the expression:
	`partial_domain_hit = (PE == 2) && ((PID & ~PIDM) == (TRDC_PIDn[PID] & ~PIDM))`
	11 - PE = 3. The process identifier is included in the domain hit evaluation as defined by the expression:
	`partial_domain_hit = (PE == 3) && ~((PID & ~PIDM) == (TRDC_PIDn[PID] & ~PIDM))`

5-4 DIDS	DID Select This field selects the source of the domain identifier. 00 - Use MDAm[3:0] as the domain identifier. 01 - Use the input DID as the domain identifier. 10 - Use MDAm[3:2] concatenated with the low-order 2 bits of the input DID (DID_in[1:0]) as the domain identifier. 11 - Reserved for future use.
3-0 DID	Domain identifier This 4-bit field is the domain ID attribute that is sent on the accesses from the bus initiator connected to the DAC when DIDS=00b. DID[3:2] is used when DIDS=10b. Write a value greater than the number of DIDs is illegal.

22.7.1.23 DAC Initiator Domain Assignment Register (MDA_W0_2_DFMT1 - MDA_W0_4_DFMT1)

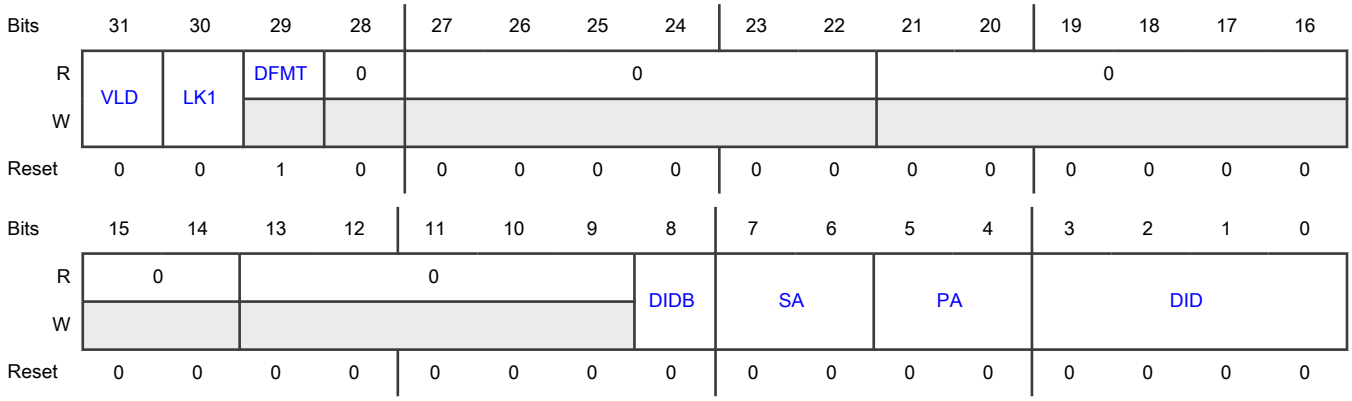
This register is identical to Initiator Domain Assignment (MDA_Wr_m_DFMT0) register except that the domain format field, DFMT, is 1 and the PID field is not used. This format supports two different specifications of the DID for non-core bus initiators.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/write.

Offset

Register	Offset
MDA_W0_2_DFMT1	840h
MDA_W0_3_DFMT1	860h
MDA_W0_4_DFMT1	880h

Diagram



Fields

Field	Description
31 VLD	Valid This field indicates the domain assignment is valid. It is further qualified by TRDC_CR[GVLDM] = 1. If TRDC_CR[GVLDM] is cleared, the DID output is defined by the SoC-specific *default DID* value. See the TRDC chip configuration section for more details. If both MDA_Wr_m[VLD] and TRDC_CR[GVLDM] are asserted, the DID output is defined by the remaining contents of this register. 0 - The Wr domain assignment is invalid. 1 - The Wr domain assignment is valid.
30 LK1	1-bit Lock This field provides a locking mechanism that can be used to limit the ability to write the register. Once set, this bit remains asserted until the next reset. 0 - Register can be written by any secure privileged write. 1 - Register is locked (read-only) until the next reset.
29 DFMT	Domain format Identifies this register's domain assignment format. **NOTE** This bitfield access is ROO 0 - Processor-core domain assignment 1 - Non-processor domain assignment
28 —	Reserved
27-22 —	Reserved
21-16	Reserved

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>—</td><td></td></tr>
<tr><td>15-14
—</td><td>Reserved</td></tr>
<tr><td>13-9
—</td><td>Reserved</td></tr>
<tr><td>8
DIDB</td><td>DID Bypass
If asserted, this bit enables the bypassing of an input DID value as the domain identifier for this non-processor bus initiator. This capability allows non-processor bus initiators, for example, a DMA to masquerade as a processor.
Once set, this field is “sticky” and remains set until the next reset.
<table><tr><th>DAC</th><th>DID Input</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>DMA Domain ID</td></tr><tr><td>2</td><td>USB Domain ID</td></tr></table>
0 - Use MDAn[3:0] as the domain identifier.
1 - Use the DID input as the domain identifier.</td></tr>
<tr><td>7-6
SA</td><td>Secure attribute
This field defines the secure/nonsecure attribute for non-processor cores.
NOTE
The bus initiator’s input secure/nonsecure attribute is used if SA = 1X, or this VLD = 0. If TZM_ENB = 0, the initiator attribute is forced to nonsecure. A nonsecure write cannot program this field to 2'b00, which is a security level higher than the mode of the process that is writing it.
Reset value of SA:
• if TZM_ENB=0, SA reset value = 2'b01
• if TZM_ENB=1, SA reset value = 2'b00
00 - Force the bus attribute for this initiator to secure.
01 - Force the bus attribute for this initiator to nonsecure.
10 - Use the bus initiator’s secure/nonsecure attribute directly.
11 - Use the bus initiator’s secure/nonsecure attribute directly.</td></tr>
<tr><td>5-4
PA</td><td>Privileged attribute
This field defines the privileged/user attribute for non-processor cores.</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE The bus initiator's input privileged/user attribute is used if PA = 1X, or this VLD = 0. 00 - Force the bus attribute for this initiator to user. 01 - Force the bus attribute for this initiator to privileged. 10 - Use the bus initiator's privileged/user attribute directly. 11 - Use the bus initiator's privileged/user attribute directly.
3-0 DID	Domain identifier This 4-bit field is the domain ID attribute that is sent on the accesses from the bus initiator connected to the DAC when DIDB=0. Write a value greater than the number of DIDs is illegal.

22.7.1.24 MBC Global Configuration Register (MBC0_MEM0_GLBCFG - MBC1_MEM3_GLBCFG)

These MBC global configuration read-only registers contain information on the MBC's hardware configuration. Specifically, it defines the number of memory blocks and the size of each block in each MBC mem (r).

NOTE

An all zero value (SIZE_LOG2 = 0, NBLKS = 0) indicates a non-existent memory. For the number of memories implemented in the MBCs, refer to chip-specific information

Offset

For m = 0 to 1; r = 0 to 3:

Register	Offset
MBCm_MEMr_GLBCFG	1_0000h + (m × 2000h) + (r × 4h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				0								SIZE_LOG2			
W																
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								NBLKS							
W																
Reset	See Register reset values.															

Register reset values

Register	Reset value
MBC0_MEM0_GLBCFG	0010_0080h
MBC0_MEM1_GLBCFG	0010_0008h
MBC0_MEM2_GLBCFG	0010_0001h
MBC0_MEM3_GLBCFG	000C_0000h
MBC1_MEM0_GLBCFG–MBC1_MEM1_GLBCFG	000C_0020h
MBC1_MEM2_GLBCFG–MBC1_MEM3_GLBCFG	000C_0000h

Fields

Field	Description
31-30 —	Reserved
29-21 —	Reserved
20-16 SIZE_LOG2	Log2 size per block For example SIZE_LOG2=0x0C is 2^12=4 KB blocks.
15-10 —	Reserved
9-0 NBLKS	Number of blocks in this memory

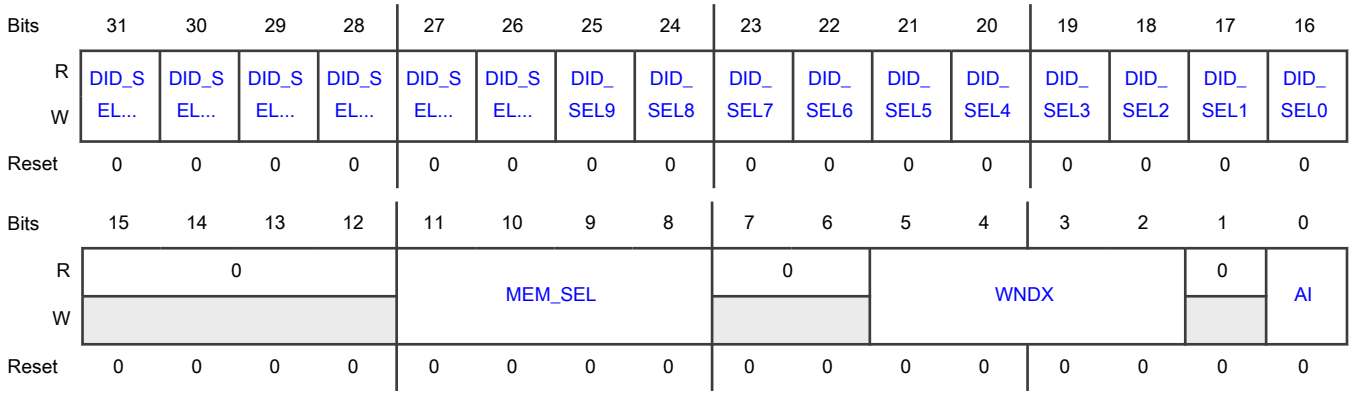
22.7.1.25 MBC NonSecure Enable Block Index (MBC0_NSE_BLK_INDEX - MBC1_NSE_BLK_INDEX)

This R/W register defines the selected memories that are affected by the writes to the NSE_BLK_SET and NSE_BLK_CLR registers.

Offset

Register	Offset
MBC0_NSE_BLK_INDEX	1_0010h
MBC1_NSE_BLK_INDEX	1_2010h

Diagram



Fields

Field	Description
31-16 DID_SELn	DID Select Destination domain bitmap select. 0 - No effect. 1 - Selects NSE bits for this domain.
15-12 —	Reserved
11-8 MEM_SEL	Memory Select Destination memory bitmap select. • Bit [11] - MBC MEM 3 • Bit [10] - MBC MEM 2 • Bit [9] - MBC MEM 1 • Bit [8] - MBC MEM 0
7-6 —	Reserved
5-2 WNDX	Word index into the block NSE bitmap. It selects the BLK_NSE_Wn register, where WNDX determines the value of n.
1 —	Reserved
0 AI	Auto Increment 0 - No effect. 1 - Add 1 to the WNDX field after the register write.

22.7.1.26 MBC NonSecure Enable Block Set (MBC0_NSE_BLK_SET - MBC1_NSE_BLK_SET)

A write to this register sets the appropriate NSE Bits for the selected domains and memories defined at the word location NSE_BLK_INDEX[WDNX]. If NSE_BLK_INDEX[AI] = 1, then the NSE_BLK_INDEX[WDNX] field is incremented by 1 (modulo-16) after the write is completed. This register reads as zero.

Offset

Register	Offset
MBC0_NSE_BLK_SET	1_0014h
MBC1_NSE_BLK_SET	1_2014h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	W1SET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	W1SET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 W1SET	Write-1 Set If set to 1, sets appropriate NSE bit for the selected domains and memories defined at the word location NSE_BLK_INDEX[WDNX]. Write with value 0 has no effect.

22.7.1.27 MBC NonSecure Enable Block Clear (MBC0_NSE_BLK_CLR - MBC1_NSE_BLK_CLR)

A write to this location clears the appropriate NSE bits as defined by the W1CLR[n]=1 for the selected domains and memories Defined at the word location NSE_BLK_INDEX[WDNX]. If NSE_BLK_INDEX[AI] = 1, then the NSE_BLK_INDEX[WDNX] field is incremented by 1 (modulo-16) after the write is completed. This register reads as zero.

Offset

Register	Offset
MBC0_NSE_BLK_CLR	1_0018h
MBC1_NSE_BLK_CLR	1_2018h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	W1CLR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	W1CLR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 W1CLR	Write-1 Clear If set to 1, Clear the appropriate NSE bit for the selected domains and memories defined at the word location NSE_BLK_INDEX[WDNX]. Write with value 0 has no effect.

22.7.1.28 MBC NonSecure Enable Block Clear All (MBC0_NSE_BLK_CLR_ALL - MBC1_NSE_BLK_CLR_ALL)

A write to this location clears all the block NSE bits for the selected domains and memories defined in the write data. This register reads as zero.

Offset

Register	Offset
MBC0_NSE_BLK_CLR_ALL	1_001Ch
MBC1_NSE_BLK_CLR_ALL	1_201Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL9	DID_SEL8	DID_SEL7	DID_SEL6	DID_SEL5	DID_SEL4	DID_SEL3	DID_SEL2	DID_SEL1	DID_SEL0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0				0							
W					MEMSEL											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 DID_SELn	DID Select Destination domain bitmap select. 0 - No effect. 1 - Clear all NSE bits for this domain.
15-12 —	Reserved
11-8 MEMSEL	Memory Select Destination memory bitmap select. • Bit [11] - clear all MEM 3 NSE bits • Bit [10] - clear all MEM 2 NSE bits • Bit [9] - clear all MEM 1 NSE bits • Bit [8] - clear all MEM 0 NSE bits
7-0 —	Reserved

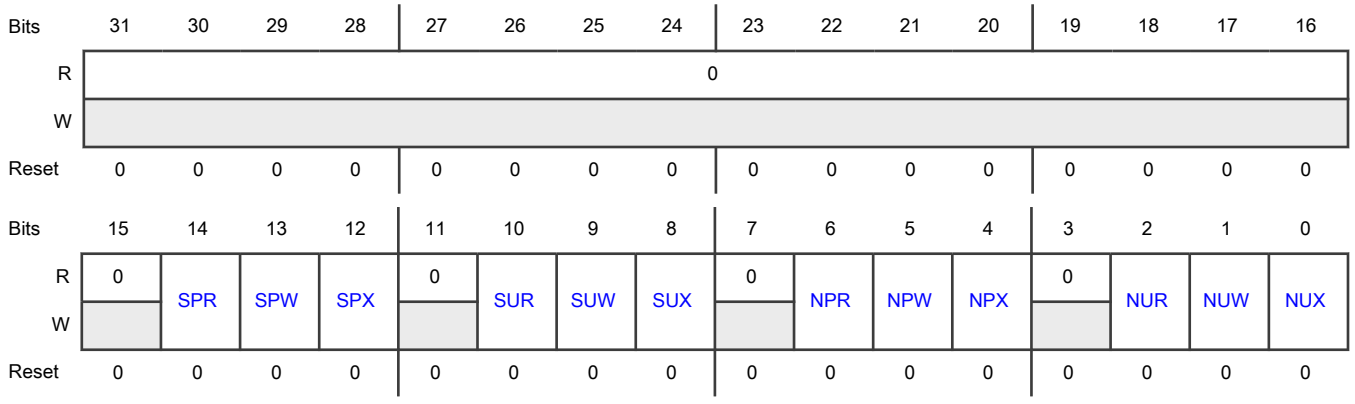
22.7.1.29 MBC Global Access Control (MBC0_MEMN_GLBAC0)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User.

Offset

Register	Offset
MBC0_MEMN_GLBAC0	1_0020h

Diagram



Fields

Field	Description
31-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6	NonsecurePriv Read

Table continues on the next page...

Table continued from the previous page...

Field	Description
NPR	NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

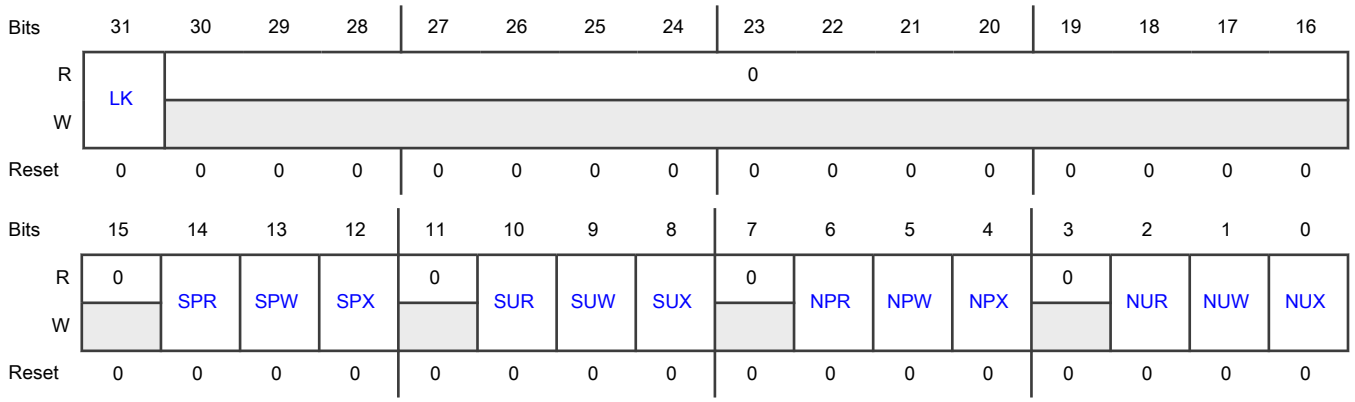
22.7.1.30 MBC Global Access Control (MBC0_MEMN_GLBAC1 - MBC0_MEMN_GLBAC7)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

Offset

Register	Offset
MBC0_MEMN_GLBAC1	1_0024h
MBC0_MEMN_GLBAC2	1_0028h
MBC0_MEMN_GLBAC3	1_002Ch
MBC0_MEMN_GLBAC4	1_0030h
MBC0_MEMN_GLBAC5	1_0034h
MBC0_MEMN_GLBAC6	1_0038h
MBC0_MEMN_GLBAC7	1_003Ch

Diagram



Fields

Field	Description
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0 - This register is not locked and can be altered. 1 - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13	SecurePriv Write

Table continues on the next page...

Table continued from the previous page...

Field	Description
SPW	SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

22.7.1.31 MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W0 - MBC0_DOM15_MEM2_BLK_CFG_W0)

MBC[m]_DOM[d]_MEM[s]_BLK_CFG_W[w], where,

- m - mbc index
- d - domain index
- s - memory target index
- w - word index

These registers are read/write for both the alternate view of the NSE bit and the 3-bit MBACSEL field. This is an array of 4 bit fields defining the block configuration for the given submemory. Each 4-bit field includes an alternative view of the associated NSE bit plus a 3-bit access control select that selects the global access control value that applies to the referenced memory block.

For a given memory block, B, if the value programmed in the MBACSELn field selects a locked MBC_MEMN_GLBACr register, the MBACSEL field is locked until the next reset. Depending on BLK_CFG_W, the NSEn/MBASELn fields correspond to different blocks. The following table describes the relationship between W (word index) and B (block number).

MEM0 supports up to 512 blocks ; W0 - W63

MEM1-3 supports up to 64 block ; W0 - W7

NOTE

This register is shown with fields such that it covers entire 32 bit register. However, actual fields may be less if number of blocks are such that fields don't align to 32 bits. These absent fields may be shown but user can refer to table below to deduce actual fields and treat absent fields as reserved.

Table 108. Block Config Word to Block Number Relationship

Block Config Word	NSE7/ MBACSEL7	NSE6/ MBACSEL6	NSE5/ MBACSEL5	NSE4/ MBACSEL4	NSE3/ MBACSEL3	NSE2/ MBACSEL2	NSE1/ MBACSEL1	NSE0/ MBACSEL0
W0	block 7	block 6	block 5	block 4	block 3	block 2	block 1	block 0
W1	block 15	block 14	block 13	block 12	block 11	block 10	block 9	block 8
W2	block 23	block 22	block 21	block 20	block 19	block 18	block 17	block 16
W3	block 31	block 30	block 29	block 28	block 27	block 26	block 25	block 24
W4	block 39	block 38	block 37	block 36	block 35	block 34	block 33	block 32
W5	block 47	block 46	block 45	block 44	block 43	block 42	block 41	block 40
W6	block 55	block 54	block 53	block 52	block 51	block 50	block 49	block 48
W7	block 63	block 62	block 61	block 60	block 59	block 58	block 57	block 56
For MEM0, block 64 - block 511								
W8	block 71	block 70	block 69	block 68	block 67	block 66	block 65	block 64
. . .								
W63	block 511	block 510	block 509	block 508	block 507	block 506	block 505	block 504

Offset

Register	Offset
MBC0_DOM0_MEM0_BLK_CFG_W0	1_0040h
MBC0_DOM0_MEM0_BLK_CFG_W1	1_0044h
MBC0_DOM0_MEM0_BLK_CFG_W2	1_0048h
MBC0_DOM0_MEM0_BLK_CFG_W3	1_004Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM0_MEM0_BLK_CFG_W4	1_0050h
MBC0_DOM0_MEM0_BLK_CFG_W5	1_0054h
MBC0_DOM0_MEM0_BLK_CFG_W6	1_0058h
MBC0_DOM0_MEM0_BLK_CFG_W7	1_005Ch
MBC0_DOM0_MEM0_BLK_CFG_W8	1_0060h
MBC0_DOM0_MEM0_BLK_CFG_W9	1_0064h
MBC0_DOM0_MEM0_BLK_CFG_W10	1_0068h
MBC0_DOM0_MEM0_BLK_CFG_W11	1_006Ch
MBC0_DOM0_MEM0_BLK_CFG_W12	1_0070h
MBC0_DOM0_MEM0_BLK_CFG_W13	1_0074h
MBC0_DOM0_MEM0_BLK_CFG_W14	1_0078h
MBC0_DOM0_MEM0_BLK_CFG_W15	1_007Ch
MBC0_DOM0_MEM1_BLK_CFG_W0	1_0180h
MBC0_DOM0_MEM2_BLK_CFG_W0	1_01A8h
MBC0_DOM1_MEM0_BLK_CFG_W0	1_0240h
MBC0_DOM1_MEM0_BLK_CFG_W1	1_0244h
MBC0_DOM1_MEM0_BLK_CFG_W2	1_0248h
MBC0_DOM1_MEM0_BLK_CFG_W3	1_024Ch
MBC0_DOM1_MEM0_BLK_CFG_W4	1_0250h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM1_MEM0_BLK_CFG_W5	1_0254h
MBC0_DOM1_MEM0_BLK_CFG_W6	1_0258h
MBC0_DOM1_MEM0_BLK_CFG_W7	1_025Ch
MBC0_DOM1_MEM0_BLK_CFG_W8	1_0260h
MBC0_DOM1_MEM0_BLK_CFG_W9	1_0264h
MBC0_DOM1_MEM0_BLK_CFG_W10	1_0268h
MBC0_DOM1_MEM0_BLK_CFG_W11	1_026Ch
MBC0_DOM1_MEM0_BLK_CFG_W12	1_0270h
MBC0_DOM1_MEM0_BLK_CFG_W13	1_0274h
MBC0_DOM1_MEM0_BLK_CFG_W14	1_0278h
MBC0_DOM1_MEM0_BLK_CFG_W15	1_027Ch
MBC0_DOM1_MEM1_BLK_CFG_W0	1_0380h
MBC0_DOM1_MEM2_BLK_CFG_W0	1_03A8h
MBC0_DOM2_MEM0_BLK_CFG_W0	1_0440h
MBC0_DOM2_MEM0_BLK_CFG_W1	1_0444h
MBC0_DOM2_MEM0_BLK_CFG_W2	1_0448h
MBC0_DOM2_MEM0_BLK_CFG_W3	1_044Ch
MBC0_DOM2_MEM0_BLK_CFG_W4	1_0450h
MBC0_DOM2_MEM0_BLK_CFG_W5	1_0454h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM2_MEM0_BLK_CFG_W6	1_0458h
MBC0_DOM2_MEM0_BLK_CFG_W7	1_045Ch
MBC0_DOM2_MEM0_BLK_CFG_W8	1_0460h
MBC0_DOM2_MEM0_BLK_CFG_W9	1_0464h
MBC0_DOM2_MEM0_BLK_CFG_W10	1_0468h
MBC0_DOM2_MEM0_BLK_CFG_W11	1_046Ch
MBC0_DOM2_MEM0_BLK_CFG_W12	1_0470h
MBC0_DOM2_MEM0_BLK_CFG_W13	1_0474h
MBC0_DOM2_MEM0_BLK_CFG_W14	1_0478h
MBC0_DOM2_MEM0_BLK_CFG_W15	1_047Ch
MBC0_DOM2_MEM1_BLK_CFG_W0	1_0580h
MBC0_DOM2_MEM2_BLK_CFG_W0	1_05A8h
MBC0_DOM3_MEM0_BLK_CFG_W0	1_0640h
MBC0_DOM3_MEM0_BLK_CFG_W1	1_0644h
MBC0_DOM3_MEM0_BLK_CFG_W2	1_0648h
MBC0_DOM3_MEM0_BLK_CFG_W3	1_064Ch
MBC0_DOM3_MEM0_BLK_CFG_W4	1_0650h
MBC0_DOM3_MEM0_BLK_CFG_W5	1_0654h
MBC0_DOM3_MEM0_BLK_CFG_W6	1_0658h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM3_MEM0_BLK_CFG_W7	1_065Ch
MBC0_DOM3_MEM0_BLK_CFG_W8	1_0660h
MBC0_DOM3_MEM0_BLK_CFG_W9	1_0664h
MBC0_DOM3_MEM0_BLK_CFG_W10	1_0668h
MBC0_DOM3_MEM0_BLK_CFG_W11	1_066Ch
MBC0_DOM3_MEM0_BLK_CFG_W12	1_0670h
MBC0_DOM3_MEM0_BLK_CFG_W13	1_0674h
MBC0_DOM3_MEM0_BLK_CFG_W14	1_0678h
MBC0_DOM3_MEM0_BLK_CFG_W15	1_067Ch
MBC0_DOM3_MEM1_BLK_CFG_W0	1_0780h
MBC0_DOM3_MEM2_BLK_CFG_W0	1_07A8h
MBC0_DOM4_MEM0_BLK_CFG_W0	1_0840h
MBC0_DOM4_MEM0_BLK_CFG_W1	1_0844h
MBC0_DOM4_MEM0_BLK_CFG_W2	1_0848h
MBC0_DOM4_MEM0_BLK_CFG_W3	1_084Ch
MBC0_DOM4_MEM0_BLK_CFG_W4	1_0850h
MBC0_DOM4_MEM0_BLK_CFG_W5	1_0854h
MBC0_DOM4_MEM0_BLK_CFG_W6	1_0858h
MBC0_DOM4_MEM0_BLK_CFG_W7	1_085Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM4_MEM0_BLK_CFG_W8	1_0860h
MBC0_DOM4_MEM0_BLK_CFG_W9	1_0864h
MBC0_DOM4_MEM0_BLK_CFG_W10	1_0868h
MBC0_DOM4_MEM0_BLK_CFG_W11	1_086Ch
MBC0_DOM4_MEM0_BLK_CFG_W12	1_0870h
MBC0_DOM4_MEM0_BLK_CFG_W13	1_0874h
MBC0_DOM4_MEM0_BLK_CFG_W14	1_0878h
MBC0_DOM4_MEM0_BLK_CFG_W15	1_087Ch
MBC0_DOM4_MEM1_BLK_CFG_W0	1_0980h
MBC0_DOM4_MEM2_BLK_CFG_W0	1_09A8h
MBC0_DOM5_MEM0_BLK_CFG_W0	1_0A40h
MBC0_DOM5_MEM0_BLK_CFG_W1	1_0A44h
MBC0_DOM5_MEM0_BLK_CFG_W2	1_0A48h
MBC0_DOM5_MEM0_BLK_CFG_W3	1_0A4Ch
MBC0_DOM5_MEM0_BLK_CFG_W4	1_0A50h
MBC0_DOM5_MEM0_BLK_CFG_W5	1_0A54h
MBC0_DOM5_MEM0_BLK_CFG_W6	1_0A58h
MBC0_DOM5_MEM0_BLK_CFG_W7	1_0A5Ch
MBC0_DOM5_MEM0_BLK_CFG_W8	1_0A60h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM5_MEM0_BLK_CFG_W9	1_0A64h
MBC0_DOM5_MEM0_BLK_CFG_W10	1_0A68h
MBC0_DOM5_MEM0_BLK_CFG_W11	1_0A6Ch
MBC0_DOM5_MEM0_BLK_CFG_W12	1_0A70h
MBC0_DOM5_MEM0_BLK_CFG_W13	1_0A74h
MBC0_DOM5_MEM0_BLK_CFG_W14	1_0A78h
MBC0_DOM5_MEM0_BLK_CFG_W15	1_0A7Ch
MBC0_DOM5_MEM1_BLK_CFG_W0	1_0B80h
MBC0_DOM5_MEM2_BLK_CFG_W0	1_0BA8h
MBC0_DOM6_MEM0_BLK_CFG_W0	1_0C40h
MBC0_DOM6_MEM0_BLK_CFG_W1	1_0C44h
MBC0_DOM6_MEM0_BLK_CFG_W2	1_0C48h
MBC0_DOM6_MEM0_BLK_CFG_W3	1_0C4Ch
MBC0_DOM6_MEM0_BLK_CFG_W4	1_0C50h
MBC0_DOM6_MEM0_BLK_CFG_W5	1_0C54h
MBC0_DOM6_MEM0_BLK_CFG_W6	1_0C58h
MBC0_DOM6_MEM0_BLK_CFG_W7	1_0C5Ch
MBC0_DOM6_MEM0_BLK_CFG_W8	1_0C60h
MBC0_DOM6_MEM0_BLK_CFG_W9	1_0C64h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM6_MEM0_BLK_CFG_W10	1_0C68h
MBC0_DOM6_MEM0_BLK_CFG_W11	1_0C6Ch
MBC0_DOM6_MEM0_BLK_CFG_W12	1_0C70h
MBC0_DOM6_MEM0_BLK_CFG_W13	1_0C74h
MBC0_DOM6_MEM0_BLK_CFG_W14	1_0C78h
MBC0_DOM6_MEM0_BLK_CFG_W15	1_0C7Ch
MBC0_DOM6_MEM1_BLK_CFG_W0	1_0D80h
MBC0_DOM6_MEM2_BLK_CFG_W0	1_0DA8h
MBC0_DOM7_MEM0_BLK_CFG_W0	1_0E40h
MBC0_DOM7_MEM0_BLK_CFG_W1	1_0E44h
MBC0_DOM7_MEM0_BLK_CFG_W2	1_0E48h
MBC0_DOM7_MEM0_BLK_CFG_W3	1_0E4Ch
MBC0_DOM7_MEM0_BLK_CFG_W4	1_0E50h
MBC0_DOM7_MEM0_BLK_CFG_W5	1_0E54h
MBC0_DOM7_MEM0_BLK_CFG_W6	1_0E58h
MBC0_DOM7_MEM0_BLK_CFG_W7	1_0E5Ch
MBC0_DOM7_MEM0_BLK_CFG_W8	1_0E60h
MBC0_DOM7_MEM0_BLK_CFG_W9	1_0E64h
MBC0_DOM7_MEM0_BLK_CFG_W10	1_0E68h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM7_MEM0_BLK_CFG_W11	1_0E6Ch
MBC0_DOM7_MEM0_BLK_CFG_W12	1_0E70h
MBC0_DOM7_MEM0_BLK_CFG_W13	1_0E74h
MBC0_DOM7_MEM0_BLK_CFG_W14	1_0E78h
MBC0_DOM7_MEM0_BLK_CFG_W15	1_0E7Ch
MBC0_DOM7_MEM1_BLK_CFG_W0	1_0F80h
MBC0_DOM7_MEM2_BLK_CFG_W0	1_0FA8h
MBC0_DOM8_MEM0_BLK_CFG_W0	1_1040h
MBC0_DOM8_MEM0_BLK_CFG_W1	1_1044h
MBC0_DOM8_MEM0_BLK_CFG_W2	1_1048h
MBC0_DOM8_MEM0_BLK_CFG_W3	1_104Ch
MBC0_DOM8_MEM0_BLK_CFG_W4	1_1050h
MBC0_DOM8_MEM0_BLK_CFG_W5	1_1054h
MBC0_DOM8_MEM0_BLK_CFG_W6	1_1058h
MBC0_DOM8_MEM0_BLK_CFG_W7	1_105Ch
MBC0_DOM8_MEM0_BLK_CFG_W8	1_1060h
MBC0_DOM8_MEM0_BLK_CFG_W9	1_1064h
MBC0_DOM8_MEM0_BLK_CFG_W10	1_1068h
MBC0_DOM8_MEM0_BLK_CFG_W11	1_106Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM8_MEM0_BLK_CFG_W12	1_1070h
MBC0_DOM8_MEM0_BLK_CFG_W13	1_1074h
MBC0_DOM8_MEM0_BLK_CFG_W14	1_1078h
MBC0_DOM8_MEM0_BLK_CFG_W15	1_107Ch
MBC0_DOM8_MEM1_BLK_CFG_W0	1_1180h
MBC0_DOM8_MEM2_BLK_CFG_W0	1_11A8h
MBC0_DOM9_MEM0_BLK_CFG_W0	1_1240h
MBC0_DOM9_MEM0_BLK_CFG_W1	1_1244h
MBC0_DOM9_MEM0_BLK_CFG_W2	1_1248h
MBC0_DOM9_MEM0_BLK_CFG_W3	1_124Ch
MBC0_DOM9_MEM0_BLK_CFG_W4	1_1250h
MBC0_DOM9_MEM0_BLK_CFG_W5	1_1254h
MBC0_DOM9_MEM0_BLK_CFG_W6	1_1258h
MBC0_DOM9_MEM0_BLK_CFG_W7	1_125Ch
MBC0_DOM9_MEM0_BLK_CFG_W8	1_1260h
MBC0_DOM9_MEM0_BLK_CFG_W9	1_1264h
MBC0_DOM9_MEM0_BLK_CFG_W10	1_1268h
MBC0_DOM9_MEM0_BLK_CFG_W11	1_126Ch
MBC0_DOM9_MEM0_BLK_CFG_W12	1_1270h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM9_MEM0_BLK_CFG_W13	1_1274h
MBC0_DOM9_MEM0_BLK_CFG_W14	1_1278h
MBC0_DOM9_MEM0_BLK_CFG_W15	1_127Ch
MBC0_DOM9_MEM1_BLK_CFG_W0	1_1380h
MBC0_DOM9_MEM2_BLK_CFG_W0	1_13A8h
MBC0_DOM10_MEM0_BLK_CFG_W0	1_1440h
MBC0_DOM10_MEM0_BLK_CFG_W1	1_1444h
MBC0_DOM10_MEM0_BLK_CFG_W2	1_1448h
MBC0_DOM10_MEM0_BLK_CFG_W3	1_144Ch
MBC0_DOM10_MEM0_BLK_CFG_W4	1_1450h
MBC0_DOM10_MEM0_BLK_CFG_W5	1_1454h
MBC0_DOM10_MEM0_BLK_CFG_W6	1_1458h
MBC0_DOM10_MEM0_BLK_CFG_W7	1_145Ch
MBC0_DOM10_MEM0_BLK_CFG_W8	1_1460h
MBC0_DOM10_MEM0_BLK_CFG_W9	1_1464h
MBC0_DOM10_MEM0_BLK_CFG_W10	1_1468h
MBC0_DOM10_MEM0_BLK_CFG_W11	1_146Ch
MBC0_DOM10_MEM0_BLK_CFG_W12	1_1470h
MBC0_DOM10_MEM0_BLK_CFG_W13	1_1474h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM10_MEM0_BLK_CFG_W14	1_1478h
MBC0_DOM10_MEM0_BLK_CFG_W15	1_147Ch
MBC0_DOM10_MEM1_BLK_CFG_W0	1_1580h
MBC0_DOM10_MEM2_BLK_CFG_W0	1_15A8h
MBC0_DOM11_MEM0_BLK_CFG_W0	1_1640h
MBC0_DOM11_MEM0_BLK_CFG_W1	1_1644h
MBC0_DOM11_MEM0_BLK_CFG_W2	1_1648h
MBC0_DOM11_MEM0_BLK_CFG_W3	1_164Ch
MBC0_DOM11_MEM0_BLK_CFG_W4	1_1650h
MBC0_DOM11_MEM0_BLK_CFG_W5	1_1654h
MBC0_DOM11_MEM0_BLK_CFG_W6	1_1658h
MBC0_DOM11_MEM0_BLK_CFG_W7	1_165Ch
MBC0_DOM11_MEM0_BLK_CFG_W8	1_1660h
MBC0_DOM11_MEM0_BLK_CFG_W9	1_1664h
MBC0_DOM11_MEM0_BLK_CFG_W10	1_1668h
MBC0_DOM11_MEM0_BLK_CFG_W11	1_166Ch
MBC0_DOM11_MEM0_BLK_CFG_W12	1_1670h
MBC0_DOM11_MEM0_BLK_CFG_W13	1_1674h
MBC0_DOM11_MEM0_BLK_CFG_W14	1_1678h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM11_MEM0_BLK_CFG_W15	1_167Ch
MBC0_DOM11_MEM1_BLK_CFG_W0	1_1780h
MBC0_DOM11_MEM2_BLK_CFG_W0	1_17A8h
MBC0_DOM12_MEM0_BLK_CFG_W0	1_1840h
MBC0_DOM12_MEM0_BLK_CFG_W1	1_1844h
MBC0_DOM12_MEM0_BLK_CFG_W2	1_1848h
MBC0_DOM12_MEM0_BLK_CFG_W3	1_184Ch
MBC0_DOM12_MEM0_BLK_CFG_W4	1_1850h
MBC0_DOM12_MEM0_BLK_CFG_W5	1_1854h
MBC0_DOM12_MEM0_BLK_CFG_W6	1_1858h
MBC0_DOM12_MEM0_BLK_CFG_W7	1_185Ch
MBC0_DOM12_MEM0_BLK_CFG_W8	1_1860h
MBC0_DOM12_MEM0_BLK_CFG_W9	1_1864h
MBC0_DOM12_MEM0_BLK_CFG_W10	1_1868h
MBC0_DOM12_MEM0_BLK_CFG_W11	1_186Ch
MBC0_DOM12_MEM0_BLK_CFG_W12	1_1870h
MBC0_DOM12_MEM0_BLK_CFG_W13	1_1874h
MBC0_DOM12_MEM0_BLK_CFG_W14	1_1878h
MBC0_DOM12_MEM0_BLK_CFG_W15	1_187Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM12_MEM1_BLK_CFG_W0	1_1980h
MBC0_DOM12_MEM2_BLK_CFG_W0	1_19A8h
MBC0_DOM13_MEM0_BLK_CFG_W0	1_1A40h
MBC0_DOM13_MEM0_BLK_CFG_W1	1_1A44h
MBC0_DOM13_MEM0_BLK_CFG_W2	1_1A48h
MBC0_DOM13_MEM0_BLK_CFG_W3	1_1A4Ch
MBC0_DOM13_MEM0_BLK_CFG_W4	1_1A50h
MBC0_DOM13_MEM0_BLK_CFG_W5	1_1A54h
MBC0_DOM13_MEM0_BLK_CFG_W6	1_1A58h
MBC0_DOM13_MEM0_BLK_CFG_W7	1_1A5Ch
MBC0_DOM13_MEM0_BLK_CFG_W8	1_1A60h
MBC0_DOM13_MEM0_BLK_CFG_W9	1_1A64h
MBC0_DOM13_MEM0_BLK_CFG_W10	1_1A68h
MBC0_DOM13_MEM0_BLK_CFG_W11	1_1A6Ch
MBC0_DOM13_MEM0_BLK_CFG_W12	1_1A70h
MBC0_DOM13_MEM0_BLK_CFG_W13	1_1A74h
MBC0_DOM13_MEM0_BLK_CFG_W14	1_1A78h
MBC0_DOM13_MEM0_BLK_CFG_W15	1_1A7Ch
MBC0_DOM13_MEM1_BLK_CFG_W0	1_1B80h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM13_MEM2_BLK_CFG_W0	1_1BA8h
MBC0_DOM14_MEM0_BLK_CFG_W0	1_1C40h
MBC0_DOM14_MEM0_BLK_CFG_W1	1_1C44h
MBC0_DOM14_MEM0_BLK_CFG_W2	1_1C48h
MBC0_DOM14_MEM0_BLK_CFG_W3	1_1C4Ch
MBC0_DOM14_MEM0_BLK_CFG_W4	1_1C50h
MBC0_DOM14_MEM0_BLK_CFG_W5	1_1C54h
MBC0_DOM14_MEM0_BLK_CFG_W6	1_1C58h
MBC0_DOM14_MEM0_BLK_CFG_W7	1_1C5Ch
MBC0_DOM14_MEM0_BLK_CFG_W8	1_1C60h
MBC0_DOM14_MEM0_BLK_CFG_W9	1_1C64h
MBC0_DOM14_MEM0_BLK_CFG_W10	1_1C68h
MBC0_DOM14_MEM0_BLK_CFG_W11	1_1C6Ch
MBC0_DOM14_MEM0_BLK_CFG_W12	1_1C70h
MBC0_DOM14_MEM0_BLK_CFG_W13	1_1C74h
MBC0_DOM14_MEM0_BLK_CFG_W14	1_1C78h
MBC0_DOM14_MEM0_BLK_CFG_W15	1_1C7Ch
MBC0_DOM14_MEM1_BLK_CFG_W0	1_1D80h
MBC0_DOM14_MEM2_BLK_CFG_W0	1_1DA8h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM15_MEM0_BLK_CFG_W0	1_1E40h
MBC0_DOM15_MEM0_BLK_CFG_W1	1_1E44h
MBC0_DOM15_MEM0_BLK_CFG_W2	1_1E48h
MBC0_DOM15_MEM0_BLK_CFG_W3	1_1E4Ch
MBC0_DOM15_MEM0_BLK_CFG_W4	1_1E50h
MBC0_DOM15_MEM0_BLK_CFG_W5	1_1E54h
MBC0_DOM15_MEM0_BLK_CFG_W6	1_1E58h
MBC0_DOM15_MEM0_BLK_CFG_W7	1_1E5Ch
MBC0_DOM15_MEM0_BLK_CFG_W8	1_1E60h
MBC0_DOM15_MEM0_BLK_CFG_W9	1_1E64h
MBC0_DOM15_MEM0_BLK_CFG_W10	1_1E68h
MBC0_DOM15_MEM0_BLK_CFG_W11	1_1E6Ch
MBC0_DOM15_MEM0_BLK_CFG_W12	1_1E70h
MBC0_DOM15_MEM0_BLK_CFG_W13	1_1E74h
MBC0_DOM15_MEM0_BLK_CFG_W14	1_1E78h
MBC0_DOM15_MEM0_BLK_CFG_W15	1_1E7Ch
MBC0_DOM15_MEM1_BLK_CFG_W0	1_1F80h
MBC0_DOM15_MEM2_BLK_CFG_W0	1_1FA8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NSE7	MBACSEL7			NSE6	MBACSEL6			NSE5	MBACSEL5			NSE4	MBACSEL4		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NSE3	MBACSEL3			NSE2	MBACSEL2			NSE1	MBACSEL1			NSE0	MBACSEL0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 NSE7	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
30-28 MBACSEL7	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL7 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
27 NSE6	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
26-24	Memory Block Access Control Select for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
MBACSEL6	This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL6 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
23 NSE5	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
22-20 MBACSEL5	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL5 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
19 NSE4	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
18-16 MBACSEL4	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL4 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
15 NSE3	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
14-12 MBACSEL3	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL3 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
11	NonSecure Enable for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
NSE2	0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
10-8 MBACSEL2	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL2 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
7 NSE1	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
6-4 MBACSEL1	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL1 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
	111 - select MBC_MEMN_GLBAC7 access control policy for block B
3 NSE0	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
2-0 MBACSEL0	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL0 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B

22.7.1.32 MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W0 - MBC0_DOM15_MEM2_BLK_NSE_W0)

MBC[m]_DOM[d]_MEM[s]_BLK_NSE_W[w], where,

- m - mbc index
- d - domain index
- s - memory target index
- w - word index

This is the bitmap of the individual NSE bits for the memory block.

NOTE

This register is shown with fields such that it covers entire 32 bit register. However, actual fields may be less if number of blocks are such that fields don't align to 32 bits. These absent fields may be shown but user can refer to table below to deduce actual fields and treat absent fields as reserved.

The following table describes the relationship between W (word index) and B (block number).

Table 109. Block Config Word to Block Number Relationship

Block NSE Word	BIT31	BIT30-BIT1	BIT0
W0	block 31	block 30 - block 1	block 0
W1	block 63	block 62 - block 33	block 32
For MEM0, block 64 - block 511			
W2	block 95	block 94 - block 65	block 64
. . .			
W15	block 511	block 510 - block 481	block 480

The NSE bitmap can be accessed directly by register reads and writes to these locations, or "indirectly" through writes to NSE_SET, NSE_CLR, NSE_CLR_ALL, and BLK_CFG[NSEn] registers.

Offset

Register	Offset
MBC0_DOM0_MEM0_BLK_NSE_W0	1_0140h
MBC0_DOM0_MEM0_BLK_NSE_W1	1_0144h
MBC0_DOM0_MEM0_BLK_NSE_W2	1_0148h
MBC0_DOM0_MEM0_BLK_NSE_W3	1_014Ch
MBC0_DOM0_MEM1_BLK_NSE_W0	1_01A0h
MBC0_DOM0_MEM2_BLK_NSE_W0	1_01C8h
MBC0_DOM1_MEM0_BLK_NSE_W0	1_0340h
MBC0_DOM1_MEM0_BLK_NSE_W1	1_0344h
MBC0_DOM1_MEM0_BLK_NSE_W2	1_0348h
MBC0_DOM1_MEM0_BLK_NSE_W3	1_034Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM1_MEM1_BLK_NSE_W0	1_03A0h
MBC0_DOM1_MEM2_BLK_NSE_W0	1_03C8h
MBC0_DOM2_MEM0_BLK_NSE_W0	1_0540h
MBC0_DOM2_MEM0_BLK_NSE_W1	1_0544h
MBC0_DOM2_MEM0_BLK_NSE_W2	1_0548h
MBC0_DOM2_MEM0_BLK_NSE_W3	1_054Ch
MBC0_DOM2_MEM1_BLK_NSE_W0	1_05A0h
MBC0_DOM2_MEM2_BLK_NSE_W0	1_05C8h
MBC0_DOM3_MEM0_BLK_NSE_W0	1_0740h
MBC0_DOM3_MEM0_BLK_NSE_W1	1_0744h
MBC0_DOM3_MEM0_BLK_NSE_W2	1_0748h
MBC0_DOM3_MEM0_BLK_NSE_W3	1_074Ch
MBC0_DOM3_MEM1_BLK_NSE_W0	1_07A0h
MBC0_DOM3_MEM2_BLK_NSE_W0	1_07C8h
MBC0_DOM4_MEM0_BLK_NSE_W0	1_0940h
MBC0_DOM4_MEM0_BLK_NSE_W1	1_0944h
MBC0_DOM4_MEM0_BLK_NSE_W2	1_0948h
MBC0_DOM4_MEM0_BLK_NSE_W3	1_094Ch
MBC0_DOM4_MEM1_BLK_NSE_W0	1_09A0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM4_MEM2_BLK_NSE_W0	1_09C8h
MBC0_DOM5_MEM0_BLK_NSE_W0	1_0B40h
MBC0_DOM5_MEM0_BLK_NSE_W1	1_0B44h
MBC0_DOM5_MEM0_BLK_NSE_W2	1_0B48h
MBC0_DOM5_MEM0_BLK_NSE_W3	1_0B4Ch
MBC0_DOM5_MEM1_BLK_NSE_W0	1_0BA0h
MBC0_DOM5_MEM2_BLK_NSE_W0	1_0BC8h
MBC0_DOM6_MEM0_BLK_NSE_W0	1_0D40h
MBC0_DOM6_MEM0_BLK_NSE_W1	1_0D44h
MBC0_DOM6_MEM0_BLK_NSE_W2	1_0D48h
MBC0_DOM6_MEM0_BLK_NSE_W3	1_0D4Ch
MBC0_DOM6_MEM1_BLK_NSE_W0	1_0DA0h
MBC0_DOM6_MEM2_BLK_NSE_W0	1_0DC8h
MBC0_DOM7_MEM0_BLK_NSE_W0	1_0F40h
MBC0_DOM7_MEM0_BLK_NSE_W1	1_0F44h
MBC0_DOM7_MEM0_BLK_NSE_W2	1_0F48h
MBC0_DOM7_MEM0_BLK_NSE_W3	1_0F4Ch
MBC0_DOM7_MEM1_BLK_NSE_W0	1_0FA0h
MBC0_DOM7_MEM2_BLK_NSE_W0	1_0FC8h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM8_MEM0_BLK_NSE_W0	1_1140h
MBC0_DOM8_MEM0_BLK_NSE_W1	1_1144h
MBC0_DOM8_MEM0_BLK_NSE_W2	1_1148h
MBC0_DOM8_MEM0_BLK_NSE_W3	1_114Ch
MBC0_DOM8_MEM1_BLK_NSE_W0	1_11A0h
MBC0_DOM8_MEM2_BLK_NSE_W0	1_11C8h
MBC0_DOM9_MEM0_BLK_NSE_W0	1_1340h
MBC0_DOM9_MEM0_BLK_NSE_W1	1_1344h
MBC0_DOM9_MEM0_BLK_NSE_W2	1_1348h
MBC0_DOM9_MEM0_BLK_NSE_W3	1_134Ch
MBC0_DOM9_MEM1_BLK_NSE_W0	1_13A0h
MBC0_DOM9_MEM2_BLK_NSE_W0	1_13C8h
MBC0_DOM10_MEM0_BLK_NSE_W0	1_1540h
MBC0_DOM10_MEM0_BLK_NSE_W1	1_1544h
MBC0_DOM10_MEM0_BLK_NSE_W2	1_1548h
MBC0_DOM10_MEM0_BLK_NSE_W3	1_154Ch
MBC0_DOM10_MEM1_BLK_NSE_W0	1_15A0h
MBC0_DOM10_MEM2_BLK_NSE_W0	1_15C8h
MBC0_DOM11_MEM0_BLK_NSE_W0	1_1740h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM11_MEM0_BLK_NSE_W1	1_1744h
MBC0_DOM11_MEM0_BLK_NSE_W2	1_1748h
MBC0_DOM11_MEM0_BLK_NSE_W3	1_174Ch
MBC0_DOM11_MEM1_BLK_NSE_W0	1_17A0h
MBC0_DOM11_MEM2_BLK_NSE_W0	1_17C8h
MBC0_DOM12_MEM0_BLK_NSE_W0	1_1940h
MBC0_DOM12_MEM0_BLK_NSE_W1	1_1944h
MBC0_DOM12_MEM0_BLK_NSE_W2	1_1948h
MBC0_DOM12_MEM0_BLK_NSE_W3	1_194Ch
MBC0_DOM12_MEM1_BLK_NSE_W0	1_19A0h
MBC0_DOM12_MEM2_BLK_NSE_W0	1_19C8h
MBC0_DOM13_MEM0_BLK_NSE_W0	1_1B40h
MBC0_DOM13_MEM0_BLK_NSE_W1	1_1B44h
MBC0_DOM13_MEM0_BLK_NSE_W2	1_1B48h
MBC0_DOM13_MEM0_BLK_NSE_W3	1_1B4Ch
MBC0_DOM13_MEM1_BLK_NSE_W0	1_1BA0h
MBC0_DOM13_MEM2_BLK_NSE_W0	1_1BC8h
MBC0_DOM14_MEM0_BLK_NSE_W0	1_1D40h
MBC0_DOM14_MEM0_BLK_NSE_W1	1_1D44h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM14_MEM0_BLK_NSE_W2	1_1D48h
MBC0_DOM14_MEM0_BLK_NSE_W3	1_1D4Ch
MBC0_DOM14_MEM1_BLK_NSE_W0	1_1DA0h
MBC0_DOM14_MEM2_BLK_NSE_W0	1_1DC8h
MBC0_DOM15_MEM0_BLK_NSE_W0	1_1F40h
MBC0_DOM15_MEM0_BLK_NSE_W1	1_1F44h
MBC0_DOM15_MEM0_BLK_NSE_W2	1_1F48h
MBC0_DOM15_MEM0_BLK_NSE_W3	1_1F4Ch
MBC0_DOM15_MEM1_BLK_NSE_W0	1_1FA0h
MBC0_DOM15_MEM2_BLK_NSE_W0	1_1FC8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BIT31	BIT30	BIT29	BIT28	BIT27	BIT26	BIT25	BIT24	BIT23	BIT22	BIT21	BIT20	BIT19	BIT18	BIT17	BIT16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BITb	Bit b NonSecure Enable [b = 0 - 31]

Table continues on the next page...

Field	Description
	0 - Secure read and write to block B are based on corresponding MBACSEL field in register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).

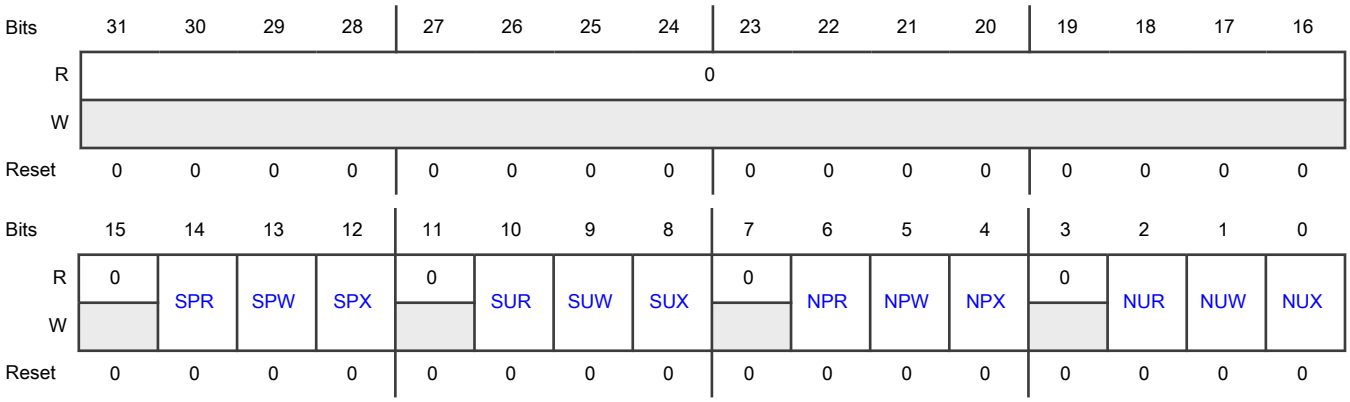
22.7.1.33 MBC Global Access Control (MBC1_MEMN_GLBAC0)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User.

Offset

Register	Offset
MBC1_MEMN_GLBAC0	1_2020h

Diagram



Fields

Field	Description
31-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13	SecurePriv Write

Table continues on the next page...

Table continued from the previous page...

Field	Description
SPW	SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

22.7.1.34 MBC Global Access Control (MBC1_MEMN_GLBAC1 - MBC1_MEMN_GLBAC7)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

Offset

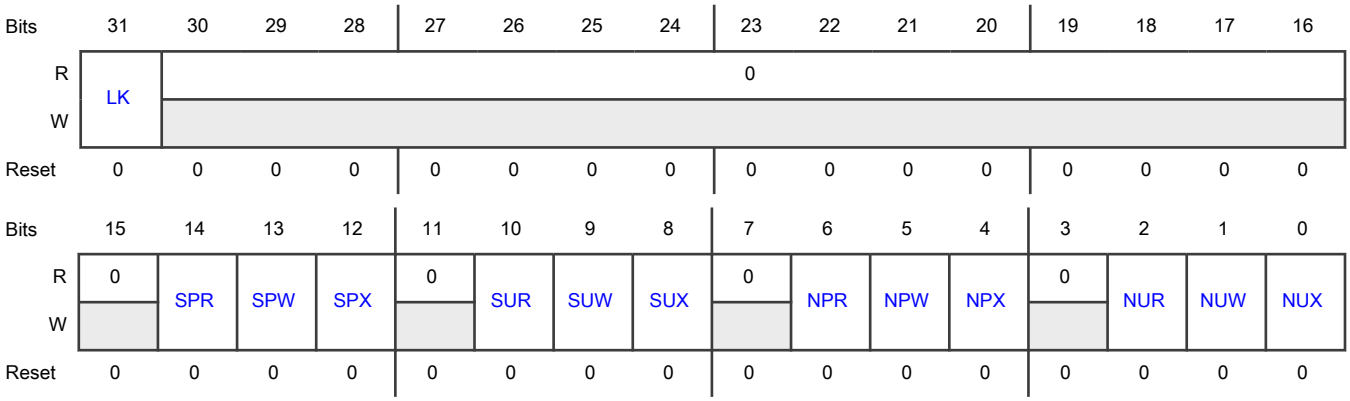
Register	Offset
MBC1_MEMN_GLBAC1	1_2024h
MBC1_MEMN_GLBAC2	1_2028h
MBC1_MEMN_GLBAC3	1_202Ch
MBC1_MEMN_GLBAC4	1_2030h
MBC1_MEMN_GLBAC5	1_2034h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_MEMN_GLBAC6	1_2038h
MBC1_MEMN_GLBAC7	1_203Ch

Diagram



Fields

Field	Description
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0 - This register is not locked and can be altered. 1 - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12	SecurePriv Execute SecurePriv execute access control flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
SPX	0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

22.7.1.35 MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W0 - MBC1_DOM15_MEM1_BLK_CFG_W3)

MBC[m]_DOM[d]_MEM[s]_BLK_CFG_W[w], where,

- m - mbc index
- d - domain index
- s - memory target index
- w - word index

These registers are read/write for both the alternate view of the NSE bit and the 3-bit MBACSEL field. This is an array of 4 bit fields defining the block configuration for the given submemory. Each 4-bit field includes an alternative view of the associated NSE bit plus a 3-bit access control select that selects the global access control value that applies to the referenced memory block.

For a given memory block, B, if the value programmed in the MBACSELn field selects a locked MBC_MEMN_GLBACr register, the MBACSEL field is locked until the next reset. Depending on BLK_CFG_W, the NSEn/MBASELn fields correspond to different blocks. The following table describes the relationship between W (word index) and B (block number).

MEM0 supports up to 512 blocks ; W0 - W63

MEM1-3 supports up to 64 block ; W0 - W7

NOTE

This register is shown with fields such that it covers entire 32 bit register. However, actual fields may be less if number of blocks are such that fields don't align to 32 bits. These absent fields may be shown but user can refer to table below to deduce actual fields and treat absent fields as reserved.

Table 110. Block Config Word to Block Number Relationship

Block Config Word	NSE7/ MBACSEL7	NSE6/ MBACSEL6	NSE5/ MBACSEL5	NSE4/ MBACSEL4	NSE3/ MBACSEL3	NSE2/ MBACSEL2	NSE1/ MBACSEL1	NSE0/ MBACSEL0
W0	block 7	block 6	block 5	block 4	block 3	block 2	block 1	block 0
W1	block 15	block 14	block 13	block 12	block 11	block 10	block 9	block 8
W2	block 23	block 22	block 21	block 20	block 19	block 18	block 17	block 16
W3	block 31	block 30	block 29	block 28	block 27	block 26	block 25	block 24
W4	block 39	block 38	block 37	block 36	block 35	block 34	block 33	block 32
W5	block 47	block 46	block 45	block 44	block 43	block 42	block 41	block 40
W6	block 55	block 54	block 53	block 52	block 51	block 50	block 49	block 48
W7	block 63	block 62	block 61	block 60	block 59	block 58	block 57	block 56
For MEM0, block 64 - block 511								
W8	block 71	block 70	block 69	block 68	block 67	block 66	block 65	block 64
. . .								
W63	block 511	block 510	block 509	block 508	block 507	block 506	block 505	block 504

Offset

Register	Offset
MBC1_DOM0_MEM0_BLK_CFG_W0	1_2040h
MBC1_DOM0_MEM0_BLK_CFG_W1	1_2044h
MBC1_DOM0_MEM0_BLK_CFG_W2	1_2048h
MBC1_DOM0_MEM0_BLK_CFG_W3	1_204Ch
MBC1_DOM0_MEM1_BLK_CFG_W0	1_2180h
MBC1_DOM0_MEM1_BLK_CFG_W1	1_2184h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM0_MEM1_BLK_CFG_W2	1_2188h
MBC1_DOM0_MEM1_BLK_CFG_W3	1_218Ch
MBC1_DOM1_MEM0_BLK_CFG_W0	1_2240h
MBC1_DOM1_MEM0_BLK_CFG_W1	1_2244h
MBC1_DOM1_MEM0_BLK_CFG_W2	1_2248h
MBC1_DOM1_MEM0_BLK_CFG_W3	1_224Ch
MBC1_DOM1_MEM1_BLK_CFG_W0	1_2380h
MBC1_DOM1_MEM1_BLK_CFG_W1	1_2384h
MBC1_DOM1_MEM1_BLK_CFG_W2	1_2388h
MBC1_DOM1_MEM1_BLK_CFG_W3	1_238Ch
MBC1_DOM2_MEM0_BLK_CFG_W0	1_2440h
MBC1_DOM2_MEM0_BLK_CFG_W1	1_2444h
MBC1_DOM2_MEM0_BLK_CFG_W2	1_2448h
MBC1_DOM2_MEM0_BLK_CFG_W3	1_244Ch
MBC1_DOM2_MEM1_BLK_CFG_W0	1_2580h
MBC1_DOM2_MEM1_BLK_CFG_W1	1_2584h
MBC1_DOM2_MEM1_BLK_CFG_W2	1_2588h
MBC1_DOM2_MEM1_BLK_CFG_W3	1_258Ch
MBC1_DOM3_MEM0_BLK_CFG_W0	1_2640h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM3_MEM0_BLK_CFG_W1	1_2644h
MBC1_DOM3_MEM0_BLK_CFG_W2	1_2648h
MBC1_DOM3_MEM0_BLK_CFG_W3	1_264Ch
MBC1_DOM3_MEM1_BLK_CFG_W0	1_2780h
MBC1_DOM3_MEM1_BLK_CFG_W1	1_2784h
MBC1_DOM3_MEM1_BLK_CFG_W2	1_2788h
MBC1_DOM3_MEM1_BLK_CFG_W3	1_278Ch
MBC1_DOM4_MEM0_BLK_CFG_W0	1_2840h
MBC1_DOM4_MEM0_BLK_CFG_W1	1_2844h
MBC1_DOM4_MEM0_BLK_CFG_W2	1_2848h
MBC1_DOM4_MEM0_BLK_CFG_W3	1_284Ch
MBC1_DOM4_MEM1_BLK_CFG_W0	1_2980h
MBC1_DOM4_MEM1_BLK_CFG_W1	1_2984h
MBC1_DOM4_MEM1_BLK_CFG_W2	1_2988h
MBC1_DOM4_MEM1_BLK_CFG_W3	1_298Ch
MBC1_DOM5_MEM0_BLK_CFG_W0	1_2A40h
MBC1_DOM5_MEM0_BLK_CFG_W1	1_2A44h
MBC1_DOM5_MEM0_BLK_CFG_W2	1_2A48h
MBC1_DOM5_MEM0_BLK_CFG_W3	1_2A4Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM5_MEM1_BLK_CFG_W0	1_2B80h
MBC1_DOM5_MEM1_BLK_CFG_W1	1_2B84h
MBC1_DOM5_MEM1_BLK_CFG_W2	1_2B88h
MBC1_DOM5_MEM1_BLK_CFG_W3	1_2B8Ch
MBC1_DOM6_MEM0_BLK_CFG_W0	1_2C40h
MBC1_DOM6_MEM0_BLK_CFG_W1	1_2C44h
MBC1_DOM6_MEM0_BLK_CFG_W2	1_2C48h
MBC1_DOM6_MEM0_BLK_CFG_W3	1_2C4Ch
MBC1_DOM6_MEM1_BLK_CFG_W0	1_2D80h
MBC1_DOM6_MEM1_BLK_CFG_W1	1_2D84h
MBC1_DOM6_MEM1_BLK_CFG_W2	1_2D88h
MBC1_DOM6_MEM1_BLK_CFG_W3	1_2D8Ch
MBC1_DOM7_MEM0_BLK_CFG_W0	1_2E40h
MBC1_DOM7_MEM0_BLK_CFG_W1	1_2E44h
MBC1_DOM7_MEM0_BLK_CFG_W2	1_2E48h
MBC1_DOM7_MEM0_BLK_CFG_W3	1_2E4Ch
MBC1_DOM7_MEM1_BLK_CFG_W0	1_2F80h
MBC1_DOM7_MEM1_BLK_CFG_W1	1_2F84h
MBC1_DOM7_MEM1_BLK_CFG_W2	1_2F88h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM7_MEM1_BLK_CFG_W3	1_2F8Ch
MBC1_DOM8_MEM0_BLK_CFG_W0	1_3040h
MBC1_DOM8_MEM0_BLK_CFG_W1	1_3044h
MBC1_DOM8_MEM0_BLK_CFG_W2	1_3048h
MBC1_DOM8_MEM0_BLK_CFG_W3	1_304Ch
MBC1_DOM8_MEM1_BLK_CFG_W0	1_3180h
MBC1_DOM8_MEM1_BLK_CFG_W1	1_3184h
MBC1_DOM8_MEM1_BLK_CFG_W2	1_3188h
MBC1_DOM8_MEM1_BLK_CFG_W3	1_318Ch
MBC1_DOM9_MEM0_BLK_CFG_W0	1_3240h
MBC1_DOM9_MEM0_BLK_CFG_W1	1_3244h
MBC1_DOM9_MEM0_BLK_CFG_W2	1_3248h
MBC1_DOM9_MEM0_BLK_CFG_W3	1_324Ch
MBC1_DOM9_MEM1_BLK_CFG_W0	1_3380h
MBC1_DOM9_MEM1_BLK_CFG_W1	1_3384h
MBC1_DOM9_MEM1_BLK_CFG_W2	1_3388h
MBC1_DOM9_MEM1_BLK_CFG_W3	1_338Ch
MBC1_DOM10_MEM0_BLK_CFG_W0	1_3440h
MBC1_DOM10_MEM0_BLK_CFG_W1	1_3444h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM10_MEM0_BLK_CFG_W2	1_3448h
MBC1_DOM10_MEM0_BLK_CFG_W3	1_344Ch
MBC1_DOM10_MEM1_BLK_CFG_W0	1_3580h
MBC1_DOM10_MEM1_BLK_CFG_W1	1_3584h
MBC1_DOM10_MEM1_BLK_CFG_W2	1_3588h
MBC1_DOM10_MEM1_BLK_CFG_W3	1_358Ch
MBC1_DOM11_MEM0_BLK_CFG_W0	1_3640h
MBC1_DOM11_MEM0_BLK_CFG_W1	1_3644h
MBC1_DOM11_MEM0_BLK_CFG_W2	1_3648h
MBC1_DOM11_MEM0_BLK_CFG_W3	1_364Ch
MBC1_DOM11_MEM1_BLK_CFG_W0	1_3780h
MBC1_DOM11_MEM1_BLK_CFG_W1	1_3784h
MBC1_DOM11_MEM1_BLK_CFG_W2	1_3788h
MBC1_DOM11_MEM1_BLK_CFG_W3	1_378Ch
MBC1_DOM12_MEM0_BLK_CFG_W0	1_3840h
MBC1_DOM12_MEM0_BLK_CFG_W1	1_3844h
MBC1_DOM12_MEM0_BLK_CFG_W2	1_3848h
MBC1_DOM12_MEM0_BLK_CFG_W3	1_384Ch
MBC1_DOM12_MEM1_BLK_CFG_W0	1_3980h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM12_MEM1_BLK_CFG_W1	1_3984h
MBC1_DOM12_MEM1_BLK_CFG_W2	1_3988h
MBC1_DOM12_MEM1_BLK_CFG_W3	1_398Ch
MBC1_DOM13_MEM0_BLK_CFG_W0	1_3A40h
MBC1_DOM13_MEM0_BLK_CFG_W1	1_3A44h
MBC1_DOM13_MEM0_BLK_CFG_W2	1_3A48h
MBC1_DOM13_MEM0_BLK_CFG_W3	1_3A4Ch
MBC1_DOM13_MEM1_BLK_CFG_W0	1_3B80h
MBC1_DOM13_MEM1_BLK_CFG_W1	1_3B84h
MBC1_DOM13_MEM1_BLK_CFG_W2	1_3B88h
MBC1_DOM13_MEM1_BLK_CFG_W3	1_3B8Ch
MBC1_DOM14_MEM0_BLK_CFG_W0	1_3C40h
MBC1_DOM14_MEM0_BLK_CFG_W1	1_3C44h
MBC1_DOM14_MEM0_BLK_CFG_W2	1_3C48h
MBC1_DOM14_MEM0_BLK_CFG_W3	1_3C4Ch
MBC1_DOM14_MEM1_BLK_CFG_W0	1_3D80h
MBC1_DOM14_MEM1_BLK_CFG_W1	1_3D84h
MBC1_DOM14_MEM1_BLK_CFG_W2	1_3D88h
MBC1_DOM14_MEM1_BLK_CFG_W3	1_3D8Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM15_MEM0_BLK_CFG_W0	1_3E40h
MBC1_DOM15_MEM0_BLK_CFG_W1	1_3E44h
MBC1_DOM15_MEM0_BLK_CFG_W2	1_3E48h
MBC1_DOM15_MEM0_BLK_CFG_W3	1_3E4Ch
MBC1_DOM15_MEM1_BLK_CFG_W0	1_3F80h
MBC1_DOM15_MEM1_BLK_CFG_W1	1_3F84h
MBC1_DOM15_MEM1_BLK_CFG_W2	1_3F88h
MBC1_DOM15_MEM1_BLK_CFG_W3	1_3F8Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NSE7	MBACSEL7			NSE6	MBACSEL6			NSE5	MBACSEL5			NSE4	MBACSEL4		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NSE3	MBACSEL3			NSE2	MBACSEL2			NSE1	MBACSEL1			NSE0	MBACSEL0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 NSE7	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
30-28	Memory Block Access Control Select for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
MBACSEL7	This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL7 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
27 NSE6	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
26-24 MBACSEL6	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL6 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
23 NSE5	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
22-20 MBACSEL5	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL5 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
19 NSE4	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
18-16 MBACSEL4	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL4 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
15	NonSecure Enable for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
NSE3	0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
14-12 MBACSEL3	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL3 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
11 NSE2	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
10-8 MBACSEL2	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL2 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
	111 - select MBC_MEMN_GLBAC7 access control policy for block B
7 NSE1	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
6-4 MBACSEL1	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL1 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
3 NSE0	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
2-0 MBACSEL0	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL0 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B

22.7.1.36 MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM0_BLK_NSE_W0 - MBC1_DOM15_MEM1_BLK_NSE_W0)

MBC[m]_DOM[d]_MEM[s]_BLK_NSE_W[w], where,

- m - mbc index
- d - domain index
- s - memory target index
- w - word index

This is the bitmap of the individual NSE bits for the memory block.

NOTE

This register is shown with fields such that it covers entire 32 bit register. However, actual fields may be less if number of blocks are such that fields don't align to 32 bits. These absent fields may be shown but user can refer to table below to deduce actual fields and treat absent fields as reserved.

The following table describes the relationship between W (word index) and B (block number).

Table 111. Block Config Word to Block Number Relationship

Block NSE Word	BIT31	BIT30-BIT1	BIT0
W0	block 31	block 30 - block 1	block 0
W1	block 63	block 62 - block 33	block 32
For MEM0, block 64 - block 511			
W2	block 95	block 94 - block 65	block 64
.			
.			
.			
W15	block 511	block 510 - block 481	block 480

The NSE bitmap can be accessed directly by register reads and writes to these locations, or "indirectly" through writes to NSE_SET, NSE_CLR, NSE_CLR_ALL, and BLK_CFG[NSEn] registers.

Offset

Register	Offset
MBC1_DOM0_MEM0_BLK_NSE_W0	1_2140h
MBC1_DOM0_MEM1_BLK_NSE_W0	1_21A0h
MBC1_DOM1_MEM0_BLK_NSE_W0	1_2340h
MBC1_DOM1_MEM1_BLK_NSE_W0	1_23A0h
MBC1_DOM2_MEM0_BLK_NSE_W0	1_2540h
MBC1_DOM2_MEM1_BLK_NSE_W0	1_25A0h
MBC1_DOM3_MEM0_BLK_NSE_W0	1_2740h
MBC1_DOM3_MEM1_BLK_NSE_W0	1_27A0h
MBC1_DOM4_MEM0_BLK_NSE_W0	1_2940h
MBC1_DOM4_MEM1_BLK_NSE_W0	1_29A0h
MBC1_DOM5_MEM0_BLK_NSE_W0	1_2B40h
MBC1_DOM5_MEM1_BLK_NSE_W0	1_2BA0h
MBC1_DOM6_MEM0_BLK_NSE_W0	1_2D40h
MBC1_DOM6_MEM1_BLK_NSE_W0	1_2DA0h
MBC1_DOM7_MEM0_BLK_NSE_W0	1_2F40h
MBC1_DOM7_MEM1_BLK_NSE_W0	1_2FA0h
MBC1_DOM8_MEM0_BLK_NSE_W0	1_3140h
MBC1_DOM8_MEM1_BLK_NSE_W0	1_31A0h
MBC1_DOM9_MEM0_BLK_NSE_W0	1_3340h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM9_MEM1_BLK_NSE_W0	1_33A0h
MBC1_DOM10_MEM0_BLK_NSE_W0	1_3540h
MBC1_DOM10_MEM1_BLK_NSE_W0	1_35A0h
MBC1_DOM11_MEM0_BLK_NSE_W0	1_3740h
MBC1_DOM11_MEM1_BLK_NSE_W0	1_37A0h
MBC1_DOM12_MEM0_BLK_NSE_W0	1_3940h
MBC1_DOM12_MEM1_BLK_NSE_W0	1_39A0h
MBC1_DOM13_MEM0_BLK_NSE_W0	1_3B40h
MBC1_DOM13_MEM1_BLK_NSE_W0	1_3BA0h
MBC1_DOM14_MEM0_BLK_NSE_W0	1_3D40h
MBC1_DOM14_MEM1_BLK_NSE_W0	1_3DA0h
MBC1_DOM15_MEM0_BLK_NSE_W0	1_3F40h
MBC1_DOM15_MEM1_BLK_NSE_W0	1_3FA0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BIT31	BIT30	BIT29	BIT28	BIT27	BIT26	BIT25	BIT24	BIT23	BIT22	BIT21	BIT20	BIT19	BIT18	BIT17	BIT16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BITb	Bit b NonSecure Enable [b = 0 - 31] 0 - Secure read and write to block B are based on corresponding MBACSEL field in register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).

22.7.1.37 MRC Global Configuration Register (MRC0_GLBCFG)

This read-only register describes the number of region descriptors for this memory.

Offset

Register	Offset
MRC0_GLBCFG	1_4000h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												NRGNS			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Fields

Field	Description
31-5 —	Reserved
4-0 NRGNS	Number of regions [1-16]

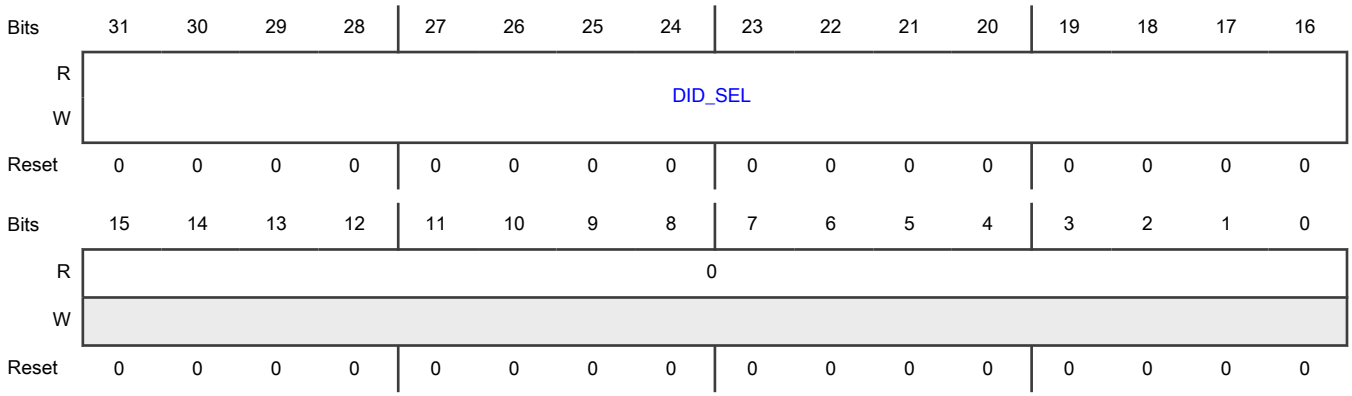
22.7.1.38 MRC NonSecure Enable Region Indirect (MRC0_NSE_RGN_INDIRECT)

This R/W register defines the selected domains that are affected by writes to the NSE_RGN_SET and NSE_RGN_CLR registers.

Offset

Register	Offset
MRC0_NSE_RGN_INDIRECT	1_4010h

Diagram



Fields

Field	Description
31-16 DID_SEL	DID Select Destination domain bitmap select. • Bit [31] = 1, update domain 15 • Bit [30] = 1, update domain 14 • Bit [29] = 1, update domain 13 • Bit [28] = 1, update domain 12 • Bit [27] = 1, update domain 11 • Bit [26] = 1, update domain 10 • Bit [25] = 1, update domain 9 • Bit [24] = 1, update domain 8 • Bit [23] = 1, update domain 7 • Bit [22] = 1, update domain 6 • Bit [21] = 1, update domain 5 • Bit [20] = 1, update domain 4 • Bit [19] = 1, update domain 3 • Bit [18] = 1, update domain 2 • Bit [17] = 1, update domain 1 • Bit [16] = 1, update domain 0

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-0 —	Reserved

22.7.1.39 MRC NonSecure Enable Region Set (MRC0_NSE_RGN_SET)

A write to this location sets the appropriate NSE bits in region descriptors for the selected domain in MRC NonSecure Enable Region Indirect register. This register reads as 0.

Offset

Register	Offset
MRC0_NSE_RGN_SET	1_4014h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	W1SET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 W1SET	Write-1 Set • Bit [15] = Set the NSE bits for RGD15 • Bit [14] = Set the NSE bits for RGD14 • Bit [13] = Set the NSE bits for RGD13 • Bit [12] = Set the NSE bits for RGD12 • Bit [11] = Set the NSE bits for RGD11

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Bit [10] = Set the NSE bits for RGD10 • Bit [9] = Set the NSE bits for RGD9 • Bit [8] = Set the NSE bits for RGD8 • Bit [7] = Set the NSE bits for RGD7 • Bit [6] = Set the NSE bits for RGD6 • Bit [5] = Set the NSE bits for RGD5 • Bit [4] = Set the NSE bits for RGD4 • Bit [3] = Set the NSE bits for RGD3 • Bit [2] = Set the NSE bits for RGD2 • Bit [1] = Set the NSE bits for RGD1 • Bit [0] = Set the NSE bits for RGD0

22.7.1.40 MRC NonSecure Enable Region Clear (MRC0_NSE_RGN_CLR)

A write to this location clears the appropriate NSE bits in region descriptors for the selected domain in MRC NonSecure Enable Region Indirect register. This register reads as 0.

Offset

Register	Offset
MRC0_NSE_RGN_CLR	1_4018h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	W1CLR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 W1CLR	Write-1 Clear • Bit [15] = Clear the NSE bits for RGD15 • Bit [14] = Clear the NSE bits for RGD14 • Bit [13] = Clear the NSE bits for RGD13 • Bit [12] = Clear the NSE bits for RGD12 • Bit [11] = Clear the NSE bits for RGD11 • Bit [10] = Clear the NSE bits for RGD10 • Bit [9] = Clear the NSE bits for RGD9 • Bit [8] = Clear the NSE bits for RGD8 • Bit [7] = Clear the NSE bits for RGD7 • Bit [6] = Clear the NSE bits for RGD6 • Bit [5] = Clear the NSE bits for RGD5 • Bit [4] = Clear the NSE bits for RGD4 • Bit [3] = Clear the NSE bits for RGD3 • Bit [2] = Clear the NSE bits for RGD2 • Bit [1] = Clear the NSE bits for RGD1 • Bit [0] = Clear the NSE bits for RGD0

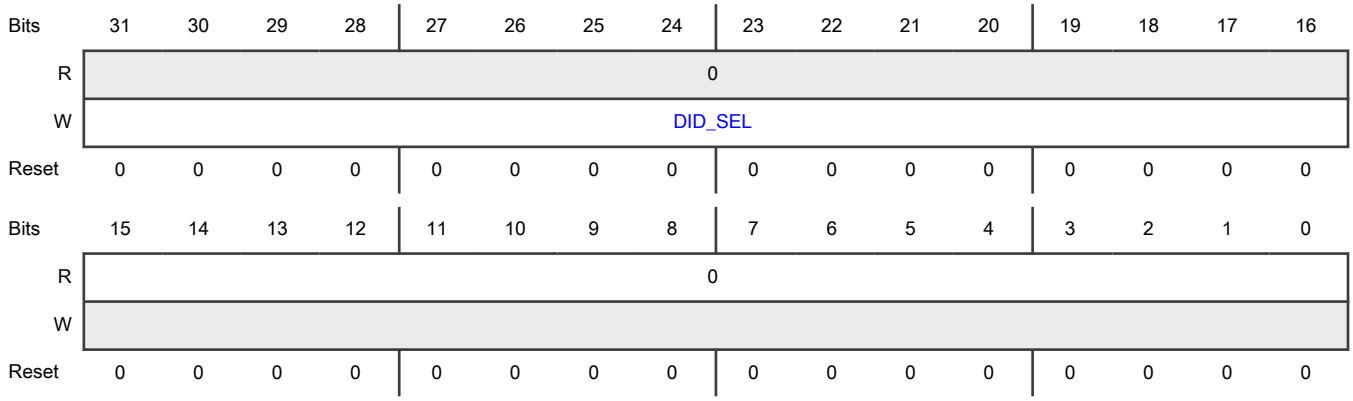
22.7.1.41 MRC NonSecure Enable Region Clear All (MRC0_NSE_RGN_CLR_ALL)

A write to this register clears all the region NSE bits for the selected domains defined in the write data.

Offset

Register	Offset
MRC0_NSE_RGN_CLR_ALL	1_401Ch

Diagram



Fields

Field	Description
31-16 DID_SEL	DID Select Destination domain bitmap select. • Bit [31] = 1 clear NSE bits for domain 15 • Bit [30] = 1 clear NSE bits for domain 14 • Bit [29] = 1 clear NSE bits for domain 13 • Bit [28] = 1 clear NSE bits for domain 12 • Bit [27] = 1 clear NSE bits for domain 11 • Bit [26] = 1 clear NSE bits for domain 10 • Bit [25] = 1 clear NSE bits for domain 9 • Bit [24] = 1 clear NSE bits for domain 8 • Bit [23] = 1 clear NSE bits for domain 7 • Bit [22] = 1 clear NSE bits for domain 6 • Bit [21] = 1 clear NSE bits for domain 5 • Bit [20] = 1 clear NSE bits for domain 4 • Bit [19] = 1 clear NSE bits for domain 3 • Bit [18] = 1 clear NSE bits for domain 2 • Bit [17] = 1 clear NSE bits for domain 1 • Bit [16] = 1 clear NSE bits for domain 0
15-0 —	Reserved

22.7.1.42 MRC Global Access Control (MRC0_GLBAC0)

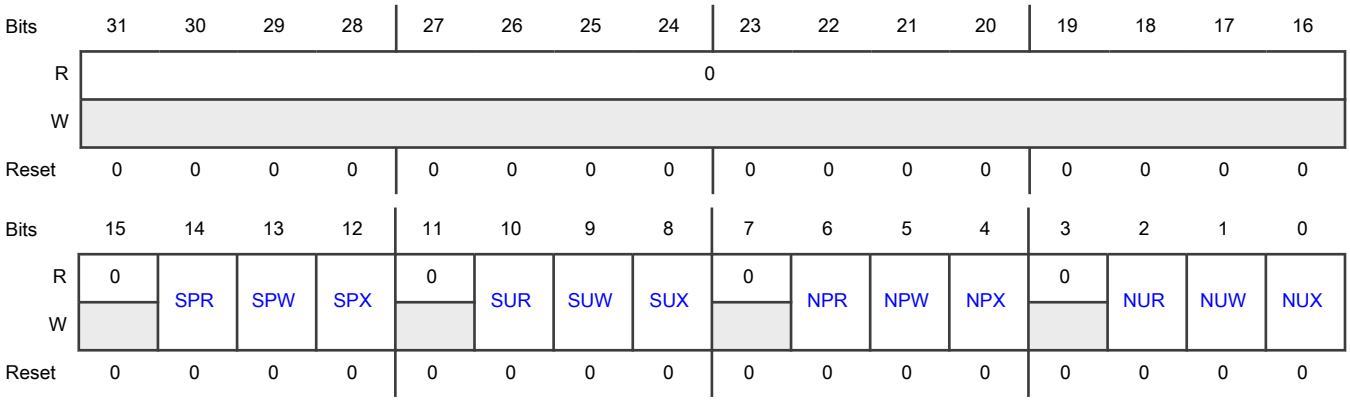
These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, NonsecureUser. The lock bit provides the ability to lock these

control access flags, to lock the MRC_BLK_CFG[MRACSEL] field, and to lock the MRC_RGD[STRT_ADDR, END_ADDR, VLD] fields when a locked MRC_GLBAC register is selected.

Offset

Register	Offset
MRC0_GLBAC0	1_4020h

Diagram



Fields

Field	Description
31-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2	NonsecureUser Read NonsecureUser read access control flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
NUR	0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

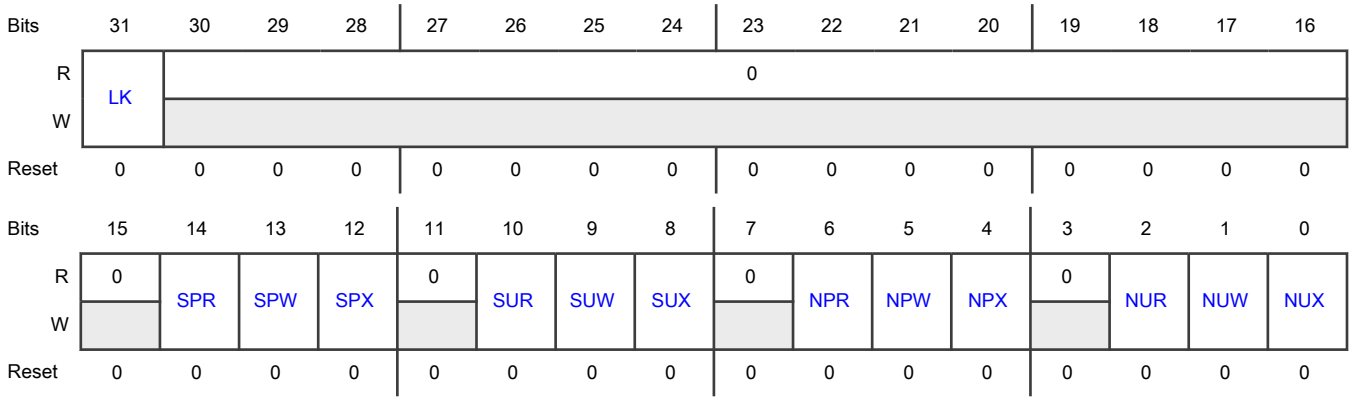
22.7.1.43 MRC Global Access Control (MRC0_GLBAC1 - MRC0_GLBAC7)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, NonsecureUser. The lock bit provides the ability to lock these control access flags, to lock the MRC_BLK_CFG[MRACSEL] field, and to lock the MRC_RGD[STRT_ADDR, END_ADDR, VLD] fields when a locked MRC_GLBAC register is selected.

Offset

Register	Offset
MRC0_GLBAC1	1_4024h
MRC0_GLBAC2	1_4028h
MRC0_GLBAC3	1_402Ch
MRC0_GLBAC4	1_4030h
MRC0_GLBAC5	1_4034h
MRC0_GLBAC6	1_4038h
MRC0_GLBAC7	1_403Ch

Diagram



Fields

Field	Description
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0 - This register is not locked and can be altered. 1 - This register is locked (read-only) and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10	SecureUser Read

Table continues on the next page...

Table continued from the previous page...

Field	Description
SUR	SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

22.7.1.44 MRC Region Descriptor Word 0 (MRC0_DOM0_RGD0_W0 - MRC0_DOM15_RGD7_W0)

MRC[m]_DOM[d]_RGD[r]_W[w], where,

- m - mrc index
- d - domain index
- r - region descriptor index
- w - word index

The MRC[m]_DOM[d]_RGD[r]_W[w] registers provide a 3-dimensional data structure for defining access control policies per domain for each supported memory region.

Memory map/register definition:

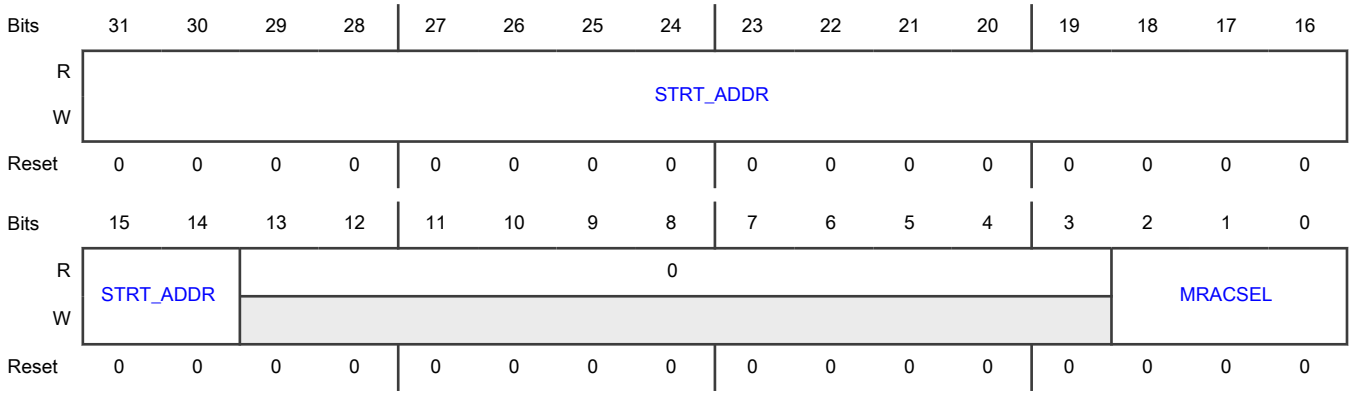
There two word-size registers (W[w]) defined in each memory region. There are up to 16 memory region checkers (MRCs), m, each supporting up to 16 memory region descriptors (RGD), r. The number of MRCs is defined by HWCFG0[NMRC].

Offset

For d = 0 to 15; r = 0 to 7:

Register	Offset
MRC0_DOMd_RGDr_W0	1_4040h + (d × 100h) + (r × 8h)

Diagram



Fields

Field	Description
31-14 STRT_ADDR	Start Address 0-mod-16K physical start address [31:14] of region r. **NOTE** This field is locked (read-only) if MRACSEL = r and MRC_GLBACr[LK] = 1.
13-3 —	Reserved
2-0 MRACSEL	Memory Region Access Control Select This field selects the global access control associated with region r. **NOTE** This field is locked (read-only) if MRACSEL = r and MRC_GLBACr[LK] = 1. 000 - Select MRC_GLBAC0 access control policy 001 - Select MRC_GLBAC1 access control policy 010 - Select MRC_GLBAC2 access control policy 011 - Select MRC_GLBAC3 access control policy 100 - Select MRC_GLBAC4 access control policy 101 - Select MRC_GLBAC5 access control policy 110 - Select MRC_GLBAC6 access control policy 111 - Select MRC_GLBAC7 access control policy

22.7.1.45 MRC Region Descriptor Word 1 (MRC0_DOM0_RGD0_W1 - MRC0_DOM15_RGD7_W1)

MRC[m]_DOM[d]_RGD[r]_W[w], where,

- m - mrc index
- d - domain index
- r - region descriptor index

- w - word index

The MRC[m]_[d]_rgd[r]_w[w] registers provide a 3-dimensional data structure for defining access control policies per domain for each supported memory region.

Memory map/register definition:

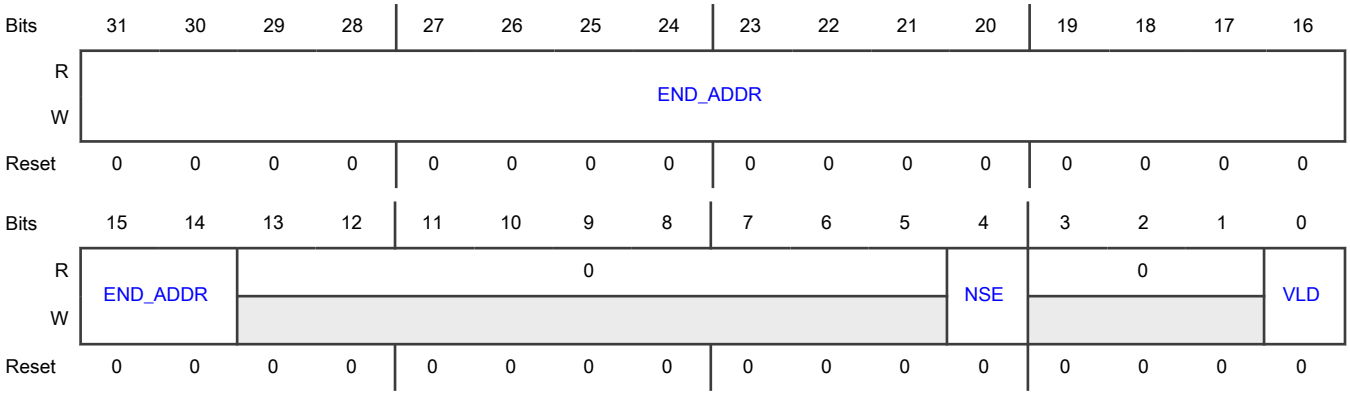
There two word-size registers (W[w]) defined in each memory region. There are up to 16 memory region checkers (MRCs), m, each supporting up to 16 region descriptors (RGD), r. The number of MRCs is defined by HWCFG0[NMRC].

Offset

For d = 0 to 15; r = 0 to 7:

Register	Offset
MRC0_DOMd_RGDr_W1	1_4044h + (d × 100h) + (r × 8h)

Diagram



Fields

Field	Description
31-14 END_ADDR	End Address (16K-1)-mod-16K physical end address [31:14] of region r. **NOTE** This field is locked (read-only) if MRACSEL = r and MRC_GLBACr[LK] = 1.
13-5 —	Reserved
4 NSE	NonSecure Enable Alternate view of NSE flag for Region r. 0 - Secure accesses to region r are based on corresponding MRACSEL field in this register (MRCm_DOMd_RGDr_Ww[MRACSEL]), nonsecure accesses to region r are not allowed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Secure accesses to region r are not allowed, nonsecure accesses to region r are based on corresponding MRACSEL field in this register (MRCm_DOMd_RGDr_Ww[MRACSEL]).
3-1 —	Reserved
0 VLD	Valid If set, this region is valid and accesses are checked for this region. **NOTE** This field is locked (read-only) if MRACSEL = r and MRC_GLBACr[LK] = 1.

22.7.1.46 MRC Region Descriptor NonSecure Enable (MRC0_DOM0_RGD_NSE - MRC0_DOM15_RGD_NSE)

This register is the bitmap of the individual NSE bits for the region descriptors. Bit n corresponds to region n.

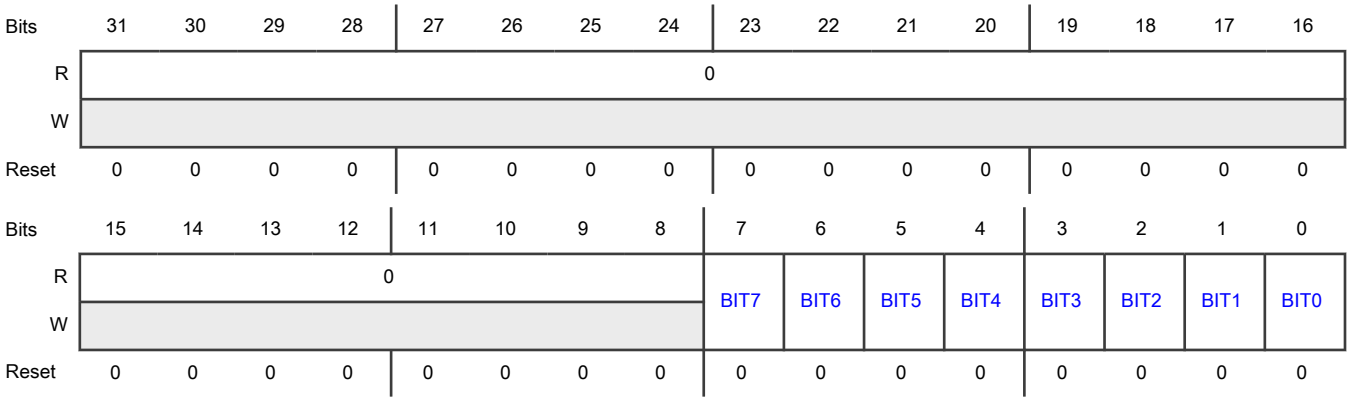
These bits can also be written or "indirectly" through writes to NSE_SET, NSE_CLR, NSE_CLR_ALL, and RGD_W0[NSE] registers.

Offset

For d = 0 to 15:

Register	Offset
MRC0_DOMd_RGD_NSE	1_40C0h + (d × 100h)

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 BITn	Bit n NonSecure Enable [n = 0 - 15] 0 - Secure accesses to region r are based on corresponding MRACSEL field in this register (MRCm_DOMd_RGDr_Ww[MRACSEL]), nonsecure accesses to region r are not allowed. 1 - Secure accesses to region r are not allowed, nonsecure accesses to region r are based on corresponding MRACSEL field in register (MRCm_DOMd_RGDr_Ww[MRACSEL]).

22.7.2 TRDC register descriptions

22.7.2.1 TRDC memory map

TRDC_HSIOMIX base address: 4C03_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	TRDC Register (TRDC_CR)	32	RW	0000_0010
F0	TRDC Hardware Configuration Register 0 (TRDC_HWCFG0)	32	R	2000_0210
F4	TRDC Hardware Configuration Register 1 (TRDC_HWCFG1)	32	R	See section
100 - 101	Domain Assignment Configuration Register (DACFG0 - DACFG1)	8	R	81
1C0	TRDC IDAU Control Register (TRDC_IDAU_CR)	32	RW	0000_0008
1E0	TRDC FLW Control (TRDC_FLW_CTL)	32	RW	0000_0000
1E4	TRDC FLW Physical Base (TRDC_FLW_PBASE)	32	R	0100_0000
1E8	TRDC FLW Array Base (TRDC_FLW_ABASE)	32	RW	0000_0000
1EC	TRDC FLW Block Count (TRDC_FLW_BCNT)	32	RW	0000_0000
1FC	TRDC Fault Domain ID (TRDC_FDID)	32	RW	0000_0000
200 - 23C	TRDC Domain Error Location Register (TRDC_DERRLOC0 - TRDC_DERRLOC15)	32	R	0000_0000
800	DAC Initiator Domain Assignment Register (MDA_W0_0_DFMT1)	32	RW	2000_0000
820	DAC Initiator Domain Assignment Register (MDA_W0_1_DFMT1)	32	RW	2000_0000

22.7.2.2 TRDC Register (TRDC_CR)

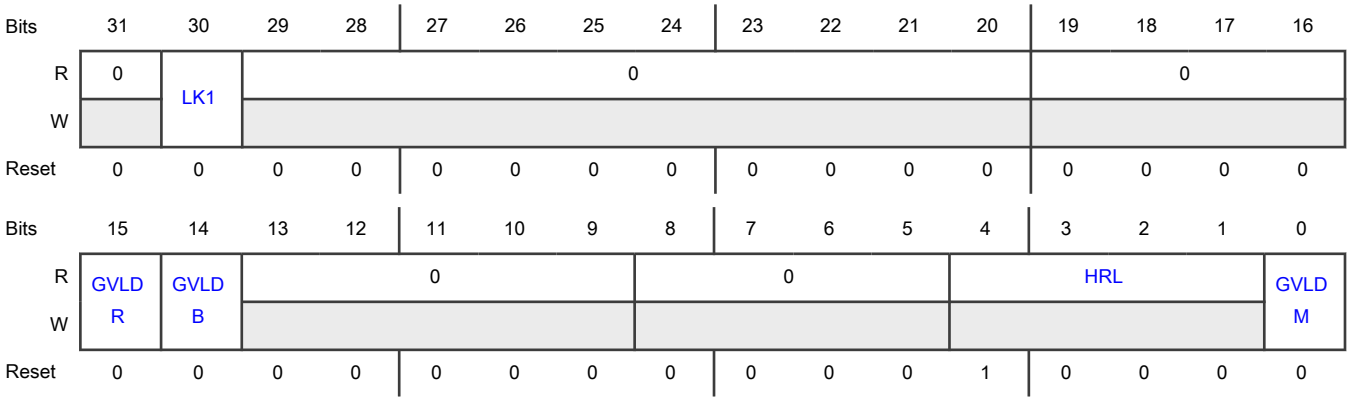
This register provides status about the TRDC and global enable bits for the entire module's operation. A fully operational TRDC requires all global bits GVLD {R,B,M} to be asserted. Undefined behavior results if they are not all asserted. If there are no region checkers (MRCs) present in the configuration, GVLDR need not be asserted for a fully operational TRDC.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_CR	0h

Diagram



Fields

Field	Description
31 —	Reserved
30 LK1	Lock Status This field is the lock status of the TRDC_CR. This field is set when all GVLD bits are set; GVLDR=GVLDB=GVLDM=1b1. Once set, this bit remains asserted until the next reset. 0 - The CR can be written by any secure privileged write. 1 - The CR is locked (read-only) until the next reset.
29-20 —	Reserved
19-16 —	Reserved
15 GVLDR	Global Valid for Memory Region Checkers TRDC global MRC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC MRCs are disabled. 1 - TRDC MRCs are enabled.
14 GVLDB	Global Valid for Memory Block Checkers TRDC global MBC enable/disable. Once set, this bit remains set until the next reset.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - TRDC MBCs are disabled. 1 - TRDC MBCs are enabled.
13-9 —	Reserved
8-5 —	Reserved
4-1 HRL	Hardware Revision Level This read-only field specifies the TRDC's hardware and definition revision level. It can be read by software to determine the functional definition of the module.
0 GVLDM	Global Valid for Domain Assignment Controllers TRDC global DAC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC DACs are disabled. 1 - TRDC DACs are enabled.

22.7.2.3 TRDC Hardware Configuration Register 0 (TRDC_HWCFG0)

This read-only register contains information on the TRDC's hardware configuration. Specifically, it defines the number of implemented domains and bus initiators along with the number of instances of memory block checkers (MBCs) and memory region checkers (MRCs). The register value at reset is device-specific. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG0	F0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MID				NMRC				0				NMBC			
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NMSTR								0				NDID			
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

Fields

Field	Description
31-29 MID	Module ID This field defines major version ID of this module.
28-24 NMRC	Number of MRCs This field defines the number of MRCs on the device [1-16].
23-20 —	Reserved
19-16 NMBC	Number of MBCs This field defines the number of MBCs on the device [1-8].
15-8 NMSTR	Number of bus initiators This read-only field defines the number of bus initiators.
7-5 —	Reserved
4-0 NDID	Number of domains This read-only field defines the number of domains on the device [1-16].

22.7.2.4 TRDC Hardware Configuration Register 1 (TRDC_HWCFG1)

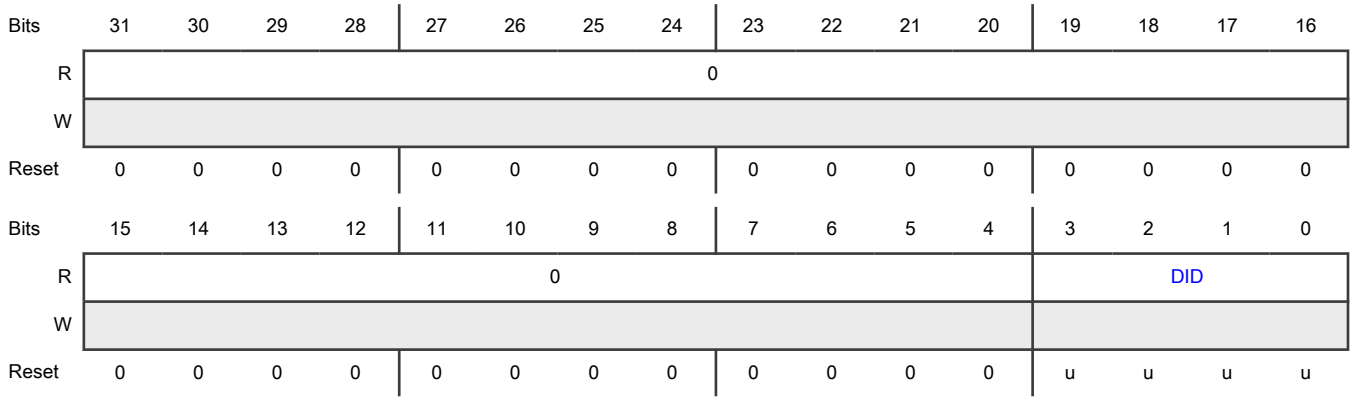
This register contains information on the TRDC's hardware configuration. It provides a mechanism for software to determine its domain number by simply reading the register. See Domain error capture management for more details on typical usage. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG1	F4h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 DID	Domain identifier number This field provides the domain number [0-15] of the requesting bus initiator.

22.7.2.5 Domain Assignment Configuration Register (DACFG0 - DACFG1)

This register defines the number of implemented domain assignment registers for bus initiator m, where m+1 can specify from 1 to 64 bus initiators. These registers are organized as a byte-sized data array and can be read using 8-, 16- or 32-bit accesses. An all-zero value (NCM = 0, NMDAR = 0) indicates a non-existent bus initiator. Attempted writes are error terminated.

Register read will not return transfer error when DACFG for a initiator doesn't exist but it is in the same 32-bit group as DACFG for an existing initiator. For example, when there are only 2 DACFG0-1 registers for 2 initiator, and initiators for DACFG2-3 don't exist, then access to DACFG2-3 won't return transfer error.

Typically, processor bus initiators have one or more domain assignment registers, while non-processor initiators have a single domain assignment register.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
DACFG0	100h
DACFG1	101h

Diagram

Bits	7	6	5	4	3	2	1	0
R	NCM	0			NMDAR			
W								
Reset	1	0	0	0	0	0	0	1

Fields

Field	Description
7 NCM	Non-CPU Initiator This read-only field signals that bus initiator m is a non-CPU initiator. It specifies that the format of the associated MDA_Wr_m register defines a non-processor domain assignment. This field is zero for a non-existent bus initiator. 0 - Bus initiator is a processor. 1 - Bus initiator is a non-processor.
6-4 —	Reserved
3-0 NMDAR	Number of initiator domain assignment registers for bus initiator m This read-only field specifies the number of registers associated with the initiator domain assignment register for a given bus initiator. The value is limited to the range [0-8], where zero indicates a non-existent bus initiator and non-zero values indicate the number of implemented registers associated with this MDAm.

22.7.2.6 TRDC IDAU Control Register (TRDC_IDAU_CR)

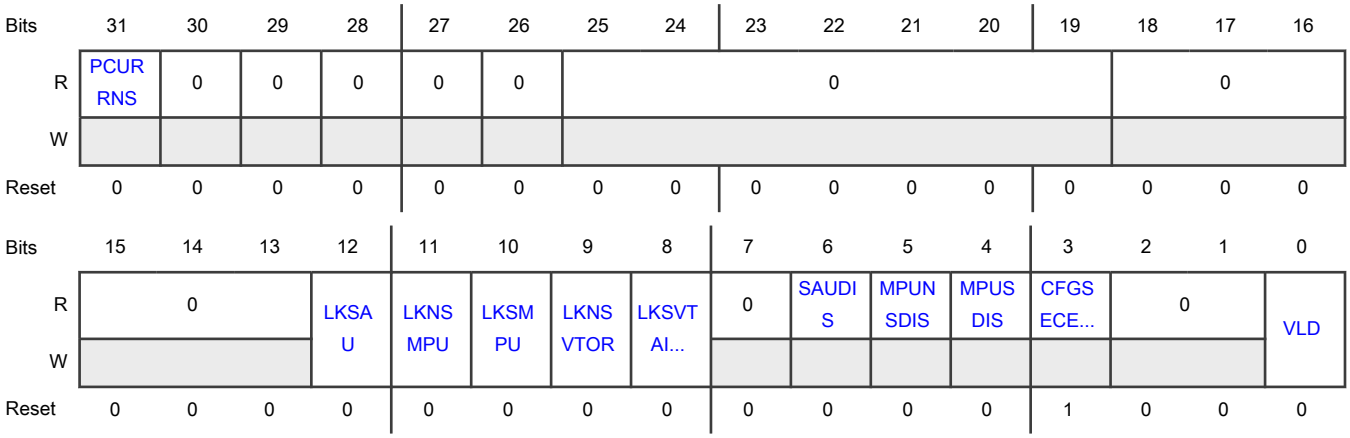
This register defines the configuration for the Implementation-Defined Attribution Unit which is tightly-coupled to the TZ-M extensions in the processor core. Many of the fields in this register explicitly control processor TZ-M functions.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_IDAU_CR	1C0h

Diagram



Fields

Field	Description
31 PCURRNS	Processor current security Current security state of the processor. 0 - Processor is in Secure state 1 - Processor is in Nonsecure state
30 —	Reserved
29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25-19 —	Reserved
18-16 —	Reserved
15-13 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 LKSAU	Lock SAU Asserting this bit prevents changes to the processor's Secure SAU memory regions already programmed. All processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the SAU_CTRL, SAU_RNR, SAU_RBAR and SAU_RLAR registers from software or from a debug agent connected to the processor
11 LKNSMPU	Lock Nonsecure MPU Asserting this bit prevents changes to the processor's Nonsecure MPU memory regions already programmed. All processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the MPU_CTRL_NS, MPU_RNR_NS, MPU_RBAR_NS, MPU_RLAR_NS, MPU_RBAR_A_NSn and MPU_RLAR_A_NSn from software or from a debug agent connected to the processor
10 LKSMPU	Lock Secure MPU Asserting this signal prevents changes to the processor's programmed Secure MPU memory regions and all processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the MPU_CTRL, MPU_RNR, MPU_RBAR, MPU_RLAR, MPU_RBAR_An and MPU_RLAR_An from software or from a debug agent connected to the processor in Secure state
9 LKNSVTOR	Lock Nonsecure Vector Table Offset Register Asserting this signal prevents changes to the processor's Nonsecure vector table base address. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock this register 1 - Disable writes to the VTOR_NS register
8 LKSVTAIRCR	Lock Secure VTOR, Application interrupt and Reset Control Registers This bit is sticky, once set, it can only be cleared with a reset. Asserting this signal prevents processor changes to: • The Secure vector table base address. • Handling of Secure interrupt priority. • BusFault, HardFault, and NMI security target settings in the processor. 0 - Unlock these registers 1 - Disable writes to the VTOR_S, AIRCR[PRIS], and AIRCR[BFHFNMINS] registers
7	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
6 SAUDIS	Security Attribution Unit Disable 0 - SAU is enabled 1 - SAU is disabled
5 MPUNSDIS	NonSecure Memory Protection Unit Disabled 0 - Nonsecure MPU is enabled 1 - Nonsecure MPU is disabled
4 MPUSDIS	Secure Memory Protection Unit Disabled 0 - Secure MPU is enabled 1 - Secure MPU is disabled
3 CFGSECEXT	Configure Security Extension 0 - Armv8M Security Extension is disabled 1 - Armv8-M Security Extension is enabled
2-1 —	Reserved
0 VLD	Valid When VLD =0, all address attribute from IDAU is nonSecure. When VLD=1, values in other fields of this register are valid.

22.7.2.7 TRDC FLW Control (TRDC_FLW_CTL)

This register provides control of the FLW = Flash Logical Window operation.

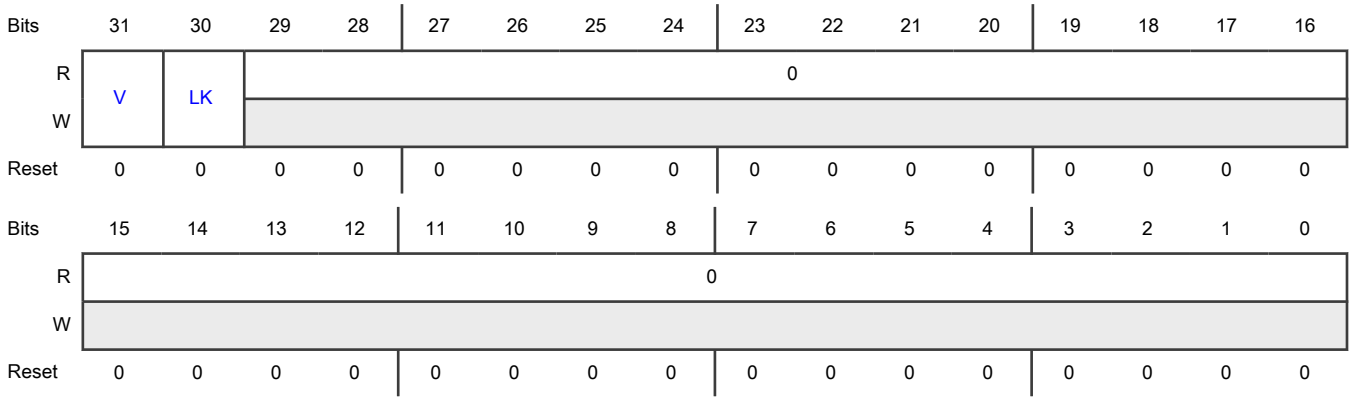
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_FLW_CTL	1E0h

Diagram



Fields

Field	Description
31 V	Valid bit 0 - FLW function is disabled. 1 - FLW function is enabled.
30 LK	Lock bit 0 - FLW registers may be modified. 1 - FLW registers are locked until the next reset.
29-0 —	Reserved

22.7.2.8 TRDC FLW Physical Base (TRDC_FLW_PBASE)

This read-only register gives the physical base address of the Flash Logical Window.

This register exists for this device but has not effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_FLW_PBASE	1E4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PBASE															
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PBASE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PBASE	Physical base address Physical address of the base of the FLW (mod 32KBytes).

22.7.2.9 TRDC FLW Array Base (TRDC_FLW_ABASE)

This register gives the flash array base address of the Flash Logical Window.

This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write. This register is not writable if TRDC_FLW_CTL[LK]=1.

Offset

Register	Offset
TRDC_FLW_ABASE	1E8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ABASE_H										ABASE_L					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ABASE_L	0														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-22 ABASE_H	Array base address high Flash array address upper bits of the base of the FLW (mod 32KBytes).
21-15 ABASE_L	Array base address low Flash array address lower bits of the base of the FLW (mod 32KBytes).
14-0 —	Reserved

22.7.2.10 TRDC FLW Block Count (TRDC_FLW_BCNT)

This register gives the size of the Flash Logic Window in 32KByte blocks.

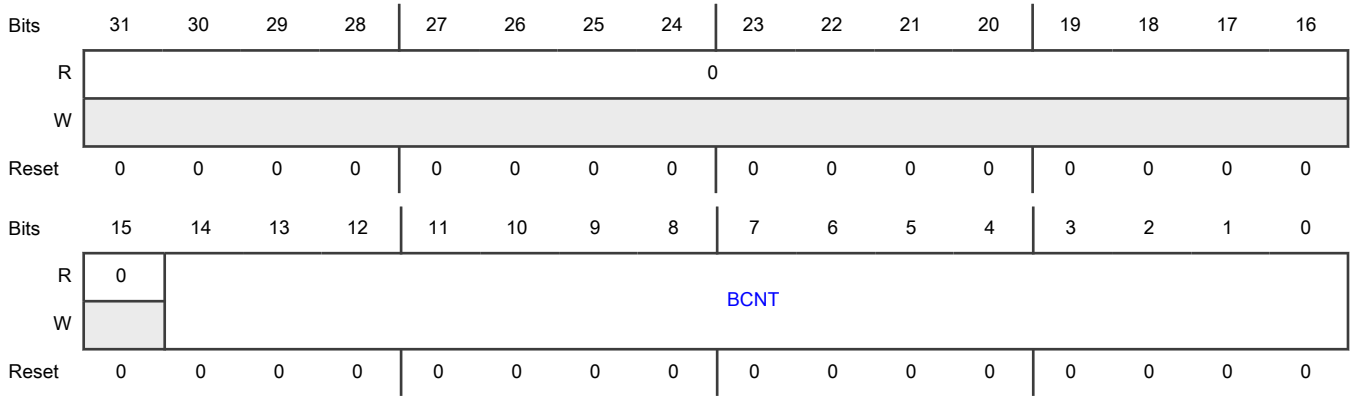
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write. This register is not writable if TRDC_FLW_CTL[LK]=1.

Offset

Register	Offset
TRDC_FLW_BCNT	1ECh

Diagram



Fields

Field	Description
31-15 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
14-0 BCNT	Block Count Size of FLW in number of 32KByte blocks.

22.7.2.11 TRDC Fault Domain ID (TRDC_FDID)

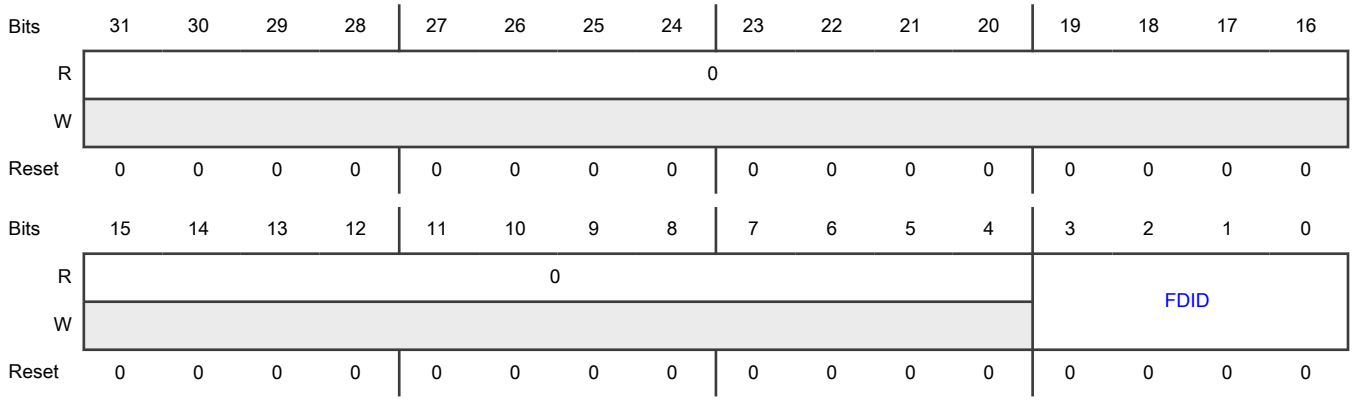
In the event of an access error, this register is used to specify the domainID of the faulting reference before indexing into the Domain Error registers.

If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_FDID	1FCh

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 FDID	Domain ID of Faulted Access This field indicates the domainID of the fault used to index the array of domain error information. The user queries the DERRLOCx registers to find the DomainID of the faulting access and must write this register with the domain ID before reading the Domain Error registers.

22.7.2.12 TRDC Domain Error Location Register (TRDC_DERRLOC0 - TRDC_DERRLOC15)

This array of read-only registers provide the instance number of the submodule where (an) access violation(s) occurred. These registers are organized as a word array, indexed by the faulting domain number, d. The two fields of this register provide a bitmap

of instances associated with all submodules containing captured error information for that domain. These instance numbers are then used as indices into the DERR_W0_*i*, DERR_W1_*i*, and DERR_W3_*i*register arrays. See Domain error capture management for more details.

When an access violation is detected by either a Memory Region Checker (MRC) or a Memory Block Checker (MBC), address and attribute information of the offending access is captured. Using the faulting domainID number as the index, d, this array of read-only registers provide additional information signaling the instance number of the submodule where the access violation(s) occurred. Since the resulting exception handler needs the submodule instance to retrieve the captured address and attribute information from DERR_W0_*i* and DERR_W1_*i*, these registers provide the instance number details.

Attempted writes are error terminated.

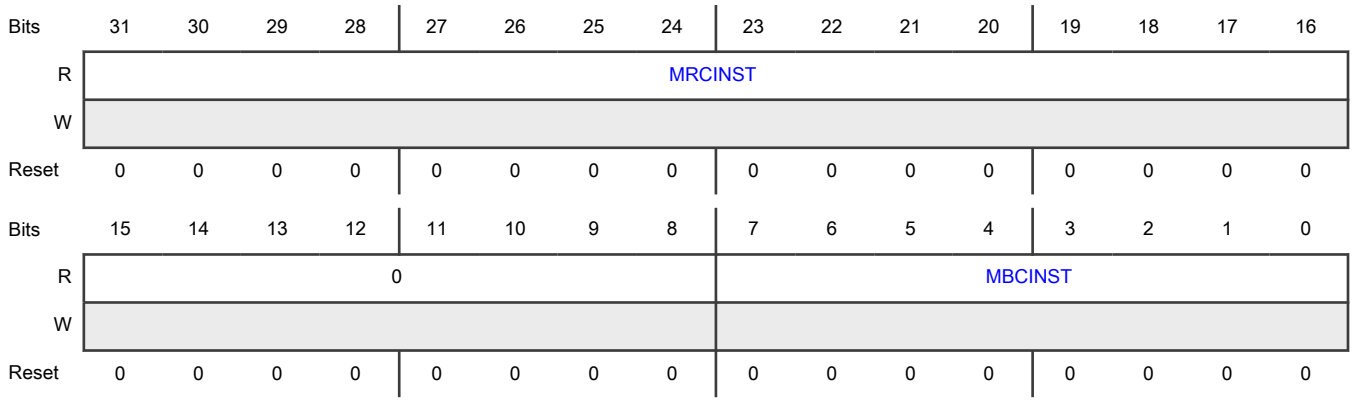
Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

For d = 0 to 15:

Register	Offset
TRDC_DERRLOCd	200h + (d × 4h)

Diagram



Fields

Field	Description
31-16 MRCINST	MRC instance This field is a bitmap indicating the presence of a detected access violation for domain d in the implemented instances of the MRC. The least-significant bit of this field (bit 16) corresponds to MRC instance 0. The most-significant bit of this field (bit 31) corresponds to MRC instance 15 (MRC instance = i-16 where i is the register bit number), and so on. Multiple bits can be set at any time indicating access violations for this domain have been detected across multiple instances of the MRCs. For each bit in this field: • 0 - The memory region checker has not detected an access violation or is not physically present. • 1 - The memory region checker has detected one or more access violations for this domain.
15-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7-0 MBCINST	MBC instance This field is a bitmap indicating the presence of a detected access violation for domain d in the implemented instances of the MBC. The least-significant bit of this field (bit 0) corresponds to MBC instance 0. The most-significant bit of this field (bit 7) corresponds to MBC instance 7 Multiple bits can be set at any time indicating access violations for this domain have been detected across multiple instances of the MBCs. For each bit in this field: • 0 - The memory block checker has not detected an access violation or is not physically present. • 1 - The memory block checker has detected one or more access violations for this domain.

22.7.2.13 DAC Initiator Domain Assignment Register (MDA_W0_0_DFMT1 - MDA_W0_1_DFMT1)

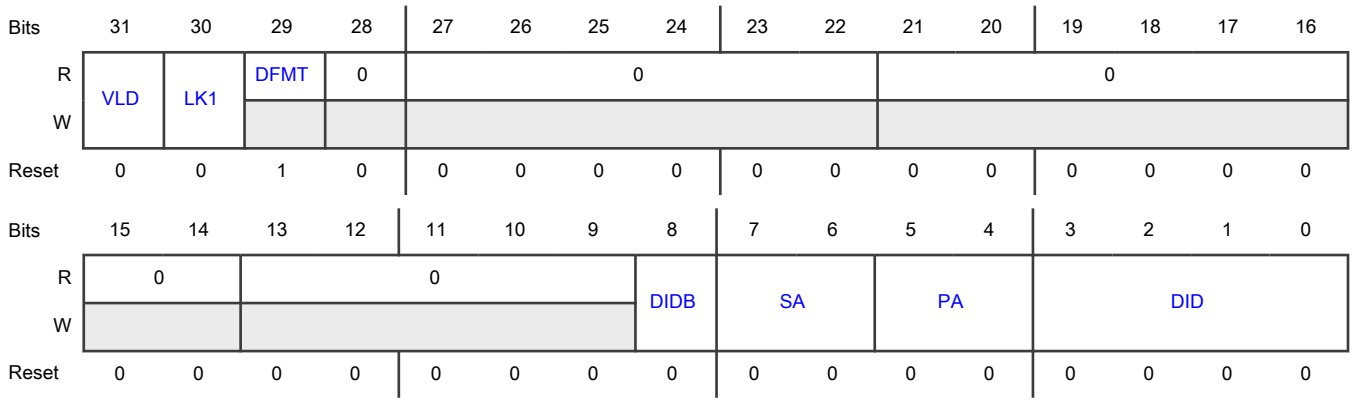
This register is identical to Initiator Domain Assignment (MDA_Wr_m_DFMT0) register except that the domain format field, DFMT, is 1 and the PID field is not used. This format supports two different specifications of the DID for non-core bus initiators.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/write.

Offset

Register	Offset
MDA_W0_0_DFMT1	800h
MDA_W0_1_DFMT1	820h

Diagram



Fields

Field	Description
31	Valid

Table continues on the next page...

Table continued from the previous page...

Field	Description
VLD	This field indicates the domain assignment is valid. It is further qualified by TRDC_CR[GVLDM] = 1. If TRDC_CR[GVLDM] is cleared, the DID output is defined by the SoC-specific *default DID* value. See the TRDC chip configuration section for more details. If both MDA_Wr_m[VLD] and TRDC_CR[GVLDM] are asserted, the DID output is defined by the remaining contents of this register. 0 - The Wr domain assignment is invalid. 1 - The Wr domain assignment is valid.
30 LK1	1-bit Lock This field provides a locking mechanism that can be used to limit the ability to write the register. Once set, this bit remains asserted until the next reset. 0 - Register can be written by any secure privileged write. 1 - Register is locked (read-only) until the next reset.
29 DFMT	Domain format Identifies this register's domain assignment format. **NOTE** This bitfield access is ROO 0 - Processor-core domain assignment 1 - Non-processor domain assignment
28 —	Reserved
27-22 —	Reserved
21-16 —	Reserved
15-14 —	Reserved
13-9 —	Reserved
8 DIDB	DID Bypass If asserted, this bit enables the bypassing of an input DID value as the domain identifier for this non-processor bus initiator. This capability allows non-processor bus initiators, for example, a DMA to masquerade as a processor. Once set, this field is “sticky” and remains set until the next reset.

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>DAC</th><th>DID Input</th></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>DMA Domain ID</td></tr>
<tr><td>2</td><td>USB Domain ID</td></tr>
</table>
0 - Use MDAn[3:0] as the domain identifier.
1 - Use the DID input as the domain identifier.
</td></tr>
<tr><td>7-6
SA</td><td>Secure attribute
This field defines the secure/nonsecure attribute for non-processor cores.
NOTE
The bus initiator’s input secure/nonsecure attribute is used if SA = 1X, or this VLD = 0. If TZM_ENB = 0, the initiator attribute is forced to nonsecure. A nonsecure write cannot program this field to 2'b00, which is a security level higher than the mode of the process that is writing it.
Reset value of SA:
• if TZM_ENB=0, SA reset value = 2'b01
• if TZM_ENB=1, SA reset value = 2'b00
00 - Force the bus attribute for this initiator to secure.
01 - Force the bus attribute for this initiator to nonsecure.
10 - Use the bus initiator’s secure/nonsecure attribute directly.
11 - Use the bus initiator’s secure/nonsecure attribute directly.</td></tr>
<tr><td>5-4
PA</td><td>Privileged attribute
This field defines the privileged/user attribute for non-processor cores.
NOTE
The bus initiator’s input privileged/user attribute is used if PA = 1X, or this VLD = 0.
00 - Force the bus attribute for this initiator to user.
01 - Force the bus attribute for this initiator to privileged.
10 - Use the bus initiator’s privileged/user attribute directly.
11 - Use the bus initiator’s privileged/user attribute directly.</td></tr>
<tr><td>3-0
DID</td><td>Domain identifier
This 4-bit field is the domain ID attribute that is sent on the accesses from the bus initiator connected to the DAC when DIDB=0.
Write a value greater than the number of DIDs is illegal.</td></tr>
</table>

22.7.3 TRDC register descriptions

22.7.3.1 TRDC memory map

TRDC_MEDIAMIX base address: 4AC3_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	TRDC Register (TRDC_CR)	32	RW	0000_0010
F0	TRDC Hardware Configuration Register 0 (TRDC_HWCFG0)	32	R	2000_0410
F4	TRDC Hardware Configuration Register 1 (TRDC_HWCFG1)	32	R	See section
100 - 103	Domain Assignment Configuration Register (DACFG0 - DACFG3)	8	R	81
1C0	TRDC IDAU Control Register (TRDC_IDAU_CR)	32	RW	0000_0008
1E0	TRDC FLW Control (TRDC_FLW_CTL)	32	RW	0000_0000
1E4	TRDC FLW Physical Base (TRDC_FLW_PBASE)	32	R	0100_0000
1E8	TRDC FLW Array Base (TRDC_FLW_ABASE)	32	RW	0000_0000
1EC	TRDC FLW Block Count (TRDC_FLW_BCNT)	32	RW	0000_0000
1FC	TRDC Fault Domain ID (TRDC_FDID)	32	RW	0000_0000
200 - 23C	TRDC Domain Error Location Register (TRDC_DERRLOC0 - TRDC_DERRLOC15)	32	R	0000_0000
800	DAC Initiator Domain Assignment Register (MDA_W0_0_DFMT1)	32	RW	2000_0000
820	DAC Initiator Domain Assignment Register (MDA_W0_1_DFMT1)	32	RW	2000_0000
840	DAC Initiator Domain Assignment Register (MDA_W0_2_DFMT1)	32	RW	2000_0000
860	DAC Initiator Domain Assignment Register (MDA_W0_3_DFMT1)	32	RW	2000_0000

22.7.3.2 TRDC Register (TRDC_CR)

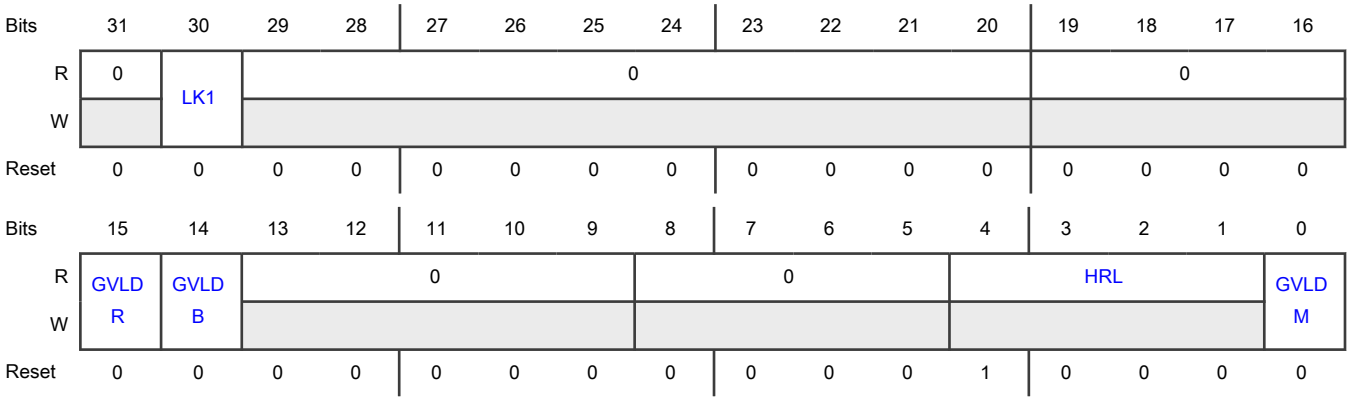
This register provides status about the TRDC and global enable bits for the entire module's operation. A fully operational TRDC requires all global bits GVLD {R,B,M} to be asserted. Undefined behavior results if they are not all asserted. If there are no region checkers (MRCs) present in the configuration, GVLDR need not be asserted for a fully operational TRDC.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_CR	0h

Diagram



Fields

Field	Description
31 —	Reserved
30 LK1	Lock Status This field is the lock status of the TRDC_CR. This field is set when all GVLD bits are set; GVLDR=GVLDB=GVLDM=1b1. Once set, this bit remains asserted until the next reset. 0 - The CR can be written by any secure privileged write. 1 - The CR is locked (read-only) until the next reset.
29-20 —	Reserved
19-16 —	Reserved
15 GVLDR	Global Valid for Memory Region Checkers TRDC global MRC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC MRCs are disabled. 1 - TRDC MRCs are enabled.
14 GVLDB	Global Valid for Memory Block Checkers TRDC global MBC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC MBCs are disabled. 1 - TRDC MBCs are enabled.
13-9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
8-5 —	Reserved
4-1 HRL	Hardware Revision Level This read-only field specifies the TRDC's hardware and definition revision level. It can be read by software to determine the functional definition of the module.
0 GVLDM	Global Valid for Domain Assignment Controllers TRDC global DAC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC DACs are disabled. 1 - TRDC DACs are enabled.

22.7.3.3 TRDC Hardware Configuration Register 0 (TRDC_HWCFG0)

This read-only register contains information on the TRDC's hardware configuration. Specifically, it defines the number of implemented domains and bus initiators along with the number of instances of memory block checkers (MBCs) and memory region checkers (MRCs). The register value at reset is device-specific. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG0	F0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MID				NMRC				0				NMBC			
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NMSTR								0				NDID			
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0

Fields

Field	Description
31-29 MID	Module ID This field defines major version ID of this module.
28-24 NMRC	Number of MRCs This field defines the number of MRCs on the device [1-16].
23-20 —	Reserved
19-16 NMBC	Number of MBCs This field defines the number of MBCs on the device [1-8].
15-8 NMSTR	Number of bus initiators This read-only field defines the number of bus initiators.
7-5 —	Reserved
4-0 NDID	Number of domains This read-only field defines the number of domains on the device [1-16].

22.7.3.4 TRDC Hardware Configuration Register 1 (TRDC_HWCFG1)

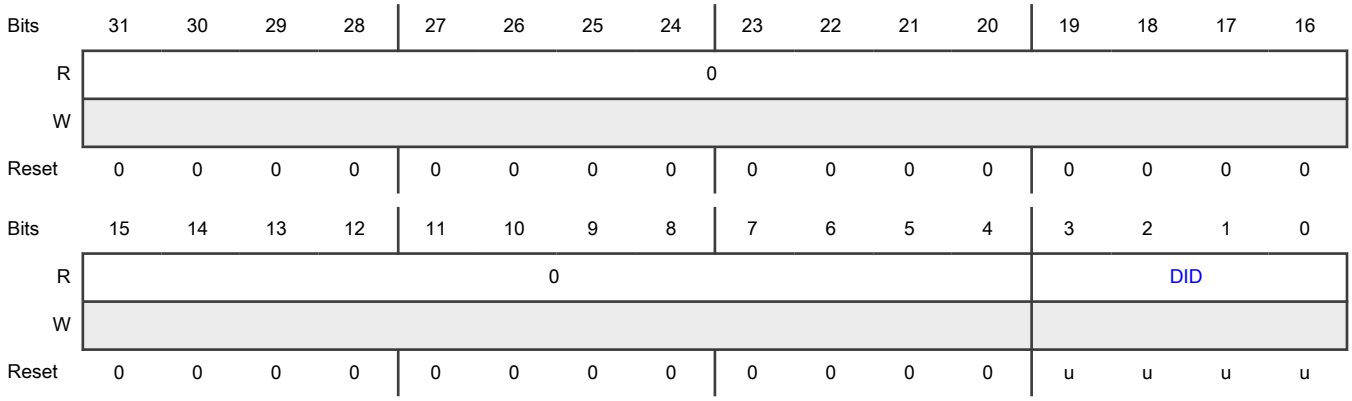
This register contains information on the TRDC's hardware configuration. It provides a mechanism for software to determine its domain number by simply reading the register. See Domain error capture management for more details on typical usage. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG1	F4h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 DID	Domain identifier number This field provides the domain number [0-15] of the requesting bus initiator.

22.7.3.5 Domain Assignment Configuration Register (DACFG0 - DACFG3)

This register defines the number of implemented domain assignment registers for bus initiator m, where m+1 can specify from 1 to 64 bus initiators. These registers are organized as a byte-sized data array and can be read using 8-, 16- or 32-bit accesses. An all-zero value (NCM = 0, NMDAR = 0) indicates a non-existent bus initiator. Attempted writes are error terminated.

Register read will not return transfer error when DACFG for a initiator doesn't exist but it is in the same 32-bit group as DACFG for an existing initiator. For example, when there are only 2 DACFG0-1 registers for 2 initiator, and initiators for DACFG2-3 don't exist, then access to DACFG2-3 won't return transfer error.

Typically, processor bus initiators have one or more domain assignment registers, while non-processor initiators have a single domain assignment register.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
DACFG0	100h
DACFG1	101h
DACFG2	102h
DACFG3	103h

Diagram

Bits	7	6	5	4	3	2	1	0
R	NCM	0			NMDAR			
W								
Reset	1	0	0	0	0	0	0	1

Fields

Field	Description
7 NCM	Non-CPU Initiator This read-only field signals that bus initiator m is a non-CPU initiator. It specifies that the format of the associated MDA_Wr_m register defines a non-processor domain assignment. This field is zero for a non-existent bus initiator. 0 - Bus initiator is a processor. 1 - Bus initiator is a non-processor.
6-4 —	Reserved
3-0 NMDAR	Number of initiator domain assignment registers for bus initiator m This read-only field specifies the number of registers associated with the initiator domain assignment register for a given bus initiator. The value is limited to the range [0-8], where zero indicates a non-existent bus initiator and non-zero values indicate the number of implemented registers associated with this MDAm.

22.7.3.6 TRDC IDAU Control Register (TRDC_IDAU_CR)

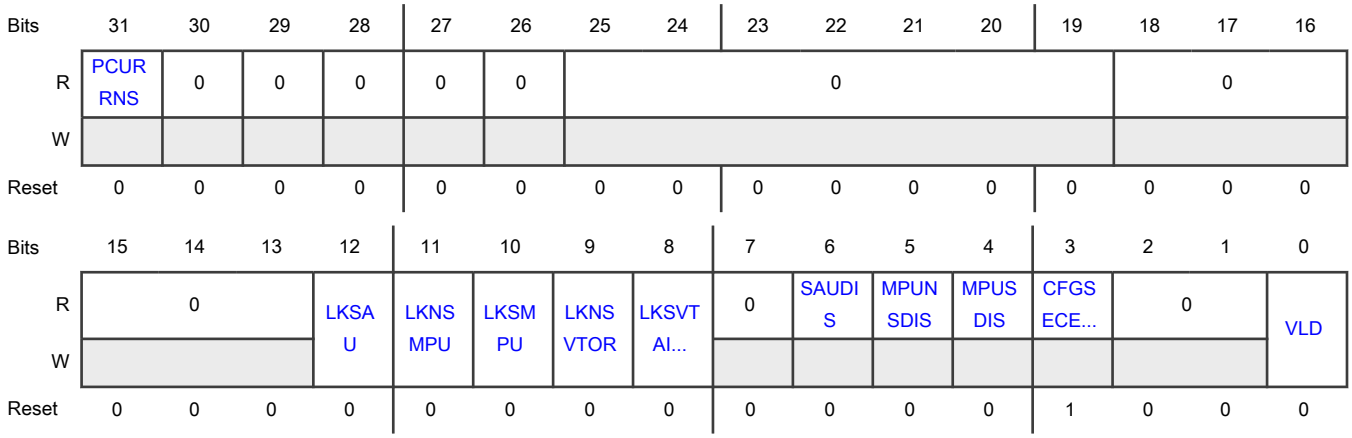
This register defines the configuration for the Implementation-Defined Attribution Unit which is tightly-coupled to the TZ-M extensions in the processor core. Many of the fields in this register explicitly control processor TZ-M functions.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_IDAU_CR	1C0h

Diagram



Fields

Field	Description
31 PCURRNS	Processor current security Current security state of the processor. 0 - Processor is in Secure state 1 - Processor is in Nonsecure state
30 —	Reserved
29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25-19 —	Reserved
18-16 —	Reserved
15-13 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 LKSAU	Lock SAU Asserting this bit prevents changes to the processor's Secure SAU memory regions already programmed. All processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the SAU_CTRL, SAU_RNR, SAU_RBAR and SAU_RLAR registers from software or from a debug agent connected to the processor
11 LKNSMPU	Lock Nonsecure MPU Asserting this bit prevents changes to the processor's Nonsecure MPU memory regions already programmed. All processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the MPU_CTRL_NS, MPU_RNR_NS, MPU_RBAR_NS, MPU_RLAR_NS, MPU_RBAR_A_NSn and MPU_RLAR_A_NSn from software or from a debug agent connected to the processor
10 LKSMPU	Lock Secure MPU Asserting this signal prevents changes to the processor's programmed Secure MPU memory regions and all processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the MPU_CTRL, MPU_RNR, MPU_RBAR, MPU_RLAR, MPU_RBAR_An and MPU_RLAR_An from software or from a debug agent connected to the processor in Secure state
9 LKNSVTOR	Lock Nonsecure Vector Table Offset Register Asserting this signal prevents changes to the processor's Nonsecure vector table base address. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock this register 1 - Disable writes to the VTOR_NS register
8 LKSVTAIRCR	Lock Secure VTOR, Application interrupt and Reset Control Registers This bit is sticky, once set, it can only be cleared with a reset. Asserting this signal prevents processor changes to: • The Secure vector table base address. • Handling of Secure interrupt priority. • BusFault, HardFault, and NMI security target settings in the processor. 0 - Unlock these registers 1 - Disable writes to the VTOR_S, AIRCR[PRIS], and AIRCR[BFHFNMINS] registers
7	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
6 SAUDIS	Security Attribution Unit Disable 0 - SAU is enabled 1 - SAU is disabled
5 MPUNSDIS	NonSecure Memory Protection Unit Disabled 0 - Nonsecure MPU is enabled 1 - Nonsecure MPU is disabled
4 MPUSDIS	Secure Memory Protection Unit Disabled 0 - Secure MPU is enabled 1 - Secure MPU is disabled
3 CFGSECEXT	Configure Security Extension 0 - Armv8M Security Extension is disabled 1 - Armv8-M Security Extension is enabled
2-1 —	Reserved
0 VLD	Valid When VLD =0, all address attribute from IDAU is nonSecure. When VLD=1, values in other fields of this register are valid.

22.7.3.7 TRDC FLW Control (TRDC_FLW_CTL)

This register provides control of the FLW = Flash Logical Window operation.

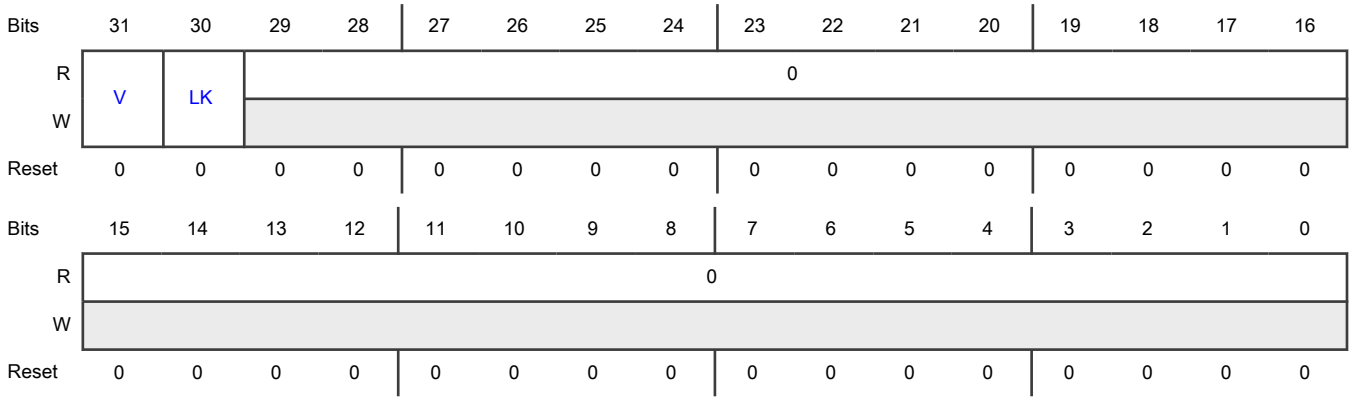
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_FLW_CTL	1E0h

Diagram



Fields

Field	Description
31 V	Valid bit 0 - FLW function is disabled. 1 - FLW function is enabled.
30 LK	Lock bit 0 - FLW registers may be modified. 1 - FLW registers are locked until the next reset.
29-0 —	Reserved

22.7.3.8 TRDC FLW Physical Base (TRDC_FLW_PBASE)

This read-only register gives the physical base address of the Flash Logical Window.

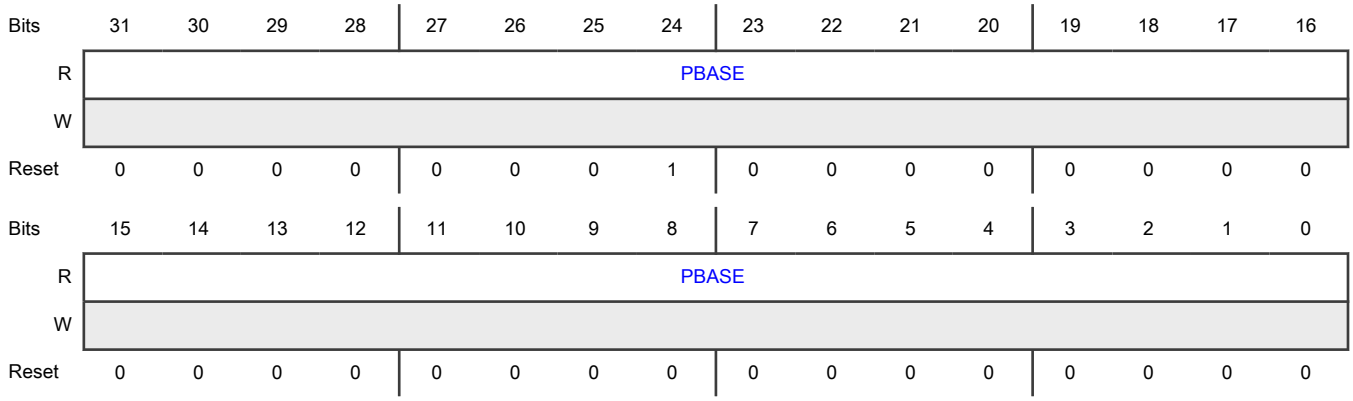
This register exists for this device but has not effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_FLW_PBASE	1E4h

Diagram



Fields

Field	Description
31-0 PBASE	Physical base address Physical address of the base of the FLW (mod 32KBytes).

22.7.3.9 TRDC FLW Array Base (TRDC_FLW_ABASE)

This register gives the flash array base address of the Flash Logical Window.

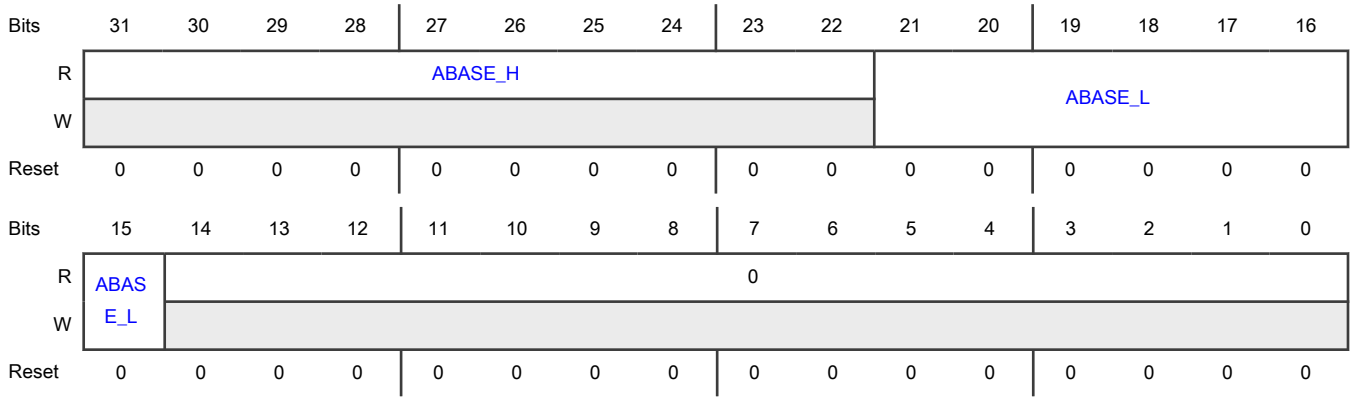
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write. This register is not writable if TRDC_FLW_CTL[LK]=1.

Offset

Register	Offset
TRDC_FLW_ABASE	1E8h

Diagram



Fields

Field	Description
31-22 ABASE_H	Array base address high Flash array address upper bits of the base of the FLW (mod 32KBytes).
21-15 ABASE_L	Array base address low Flash array address lower bits of the base of the FLW (mod 32KBytes).
14-0 —	Reserved

22.7.3.10 TRDC FLW Block Count (TRDC_FLW_BCNT)

This register gives the size of the Flash Logic Window in 32KByte blocks.

This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write. This register is not writable if TRDC_FLW_CTL[LK]=1.

Offset

Register	Offset
TRDC_FLW_BCNT	1ECh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	BCNT														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-15 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
14-0 BCNT	Block Count Size of FLW in number of 32KByte blocks.

22.7.3.11 TRDC Fault Domain ID (TRDC_FDID)

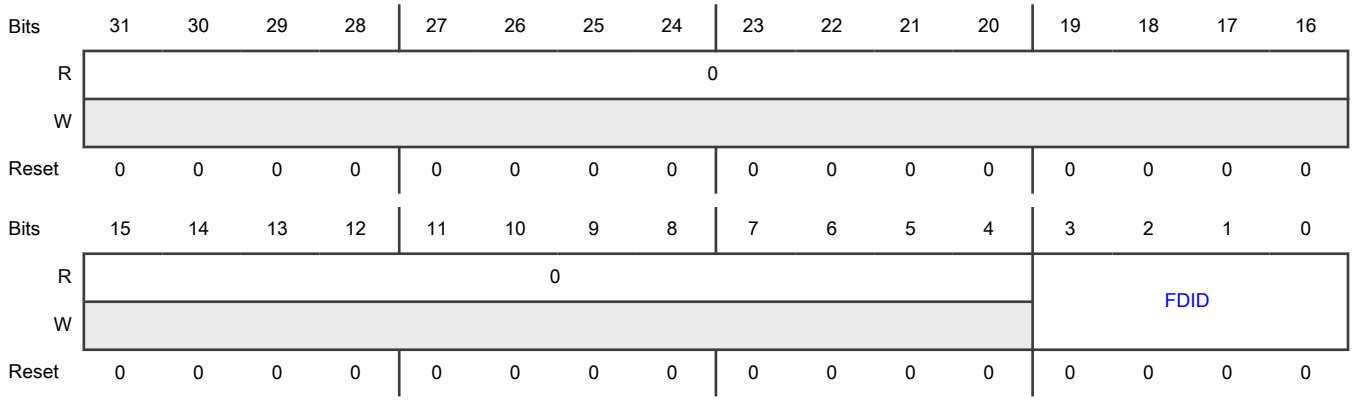
In the event of an access error, this register is used to specify the domainID of the faulting reference before indexing into the Domain Error registers.

If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_FDID	1FCh

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 FDID	Domain ID of Faulted Access This field indicates the domainID of the fault used to index the array of domain error information. The user queries the DERRLOCx registers to find the DomainID of the faulting access and must write this register with the domain ID before reading the Domain Error registers.

22.7.3.12 TRDC Domain Error Location Register (TRDC_DERRLOC0 - TRDC_DERRLOC15)

This array of read-only registers provide the instance number of the submodule where (an) access violation(s) occurred. These registers are organized as a word array, indexed by the faulting domain number, d. The two fields of this register provide a bitmap

of instances associated with all submodules containing captured error information for that domain. These instance numbers are then used as indices into the DERR_W0_*i*, DERR_W1_*i*, and DERR_W3_*i*register arrays. See Domain error capture management for more details.

When an access violation is detected by either a Memory Region Checker (MRC) or a Memory Block Checker (MBC), address and attribute information of the offending access is captured. Using the faulting domainID number as the index, d, this array of read-only registers provide additional information signaling the instance number of the submodule where the access violation(s) occurred. Since the resulting exception handler needs the submodule instance to retrieve the captured address and attribute information from DERR_W0_*i* and DERR_W1_*i*, these registers provide the instance number details.

Attempted writes are error terminated.

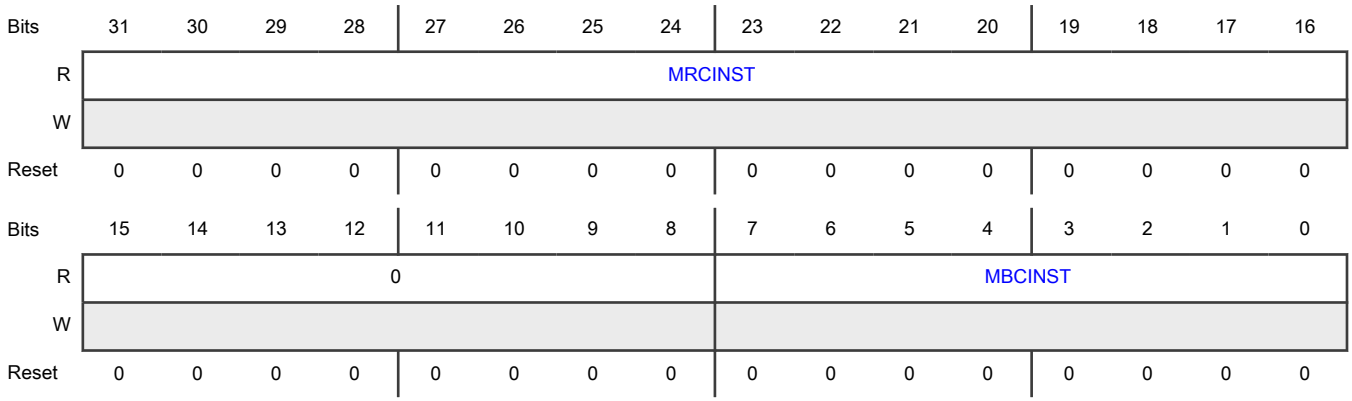
Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

For d = 0 to 15:

Register	Offset
TRDC_DERRLOCd	200h + (d × 4h)

Diagram



Fields

Field	Description
31-16 MRCINST	MRC instance This field is a bitmap indicating the presence of a detected access violation for domain d in the implemented instances of the MRC. The least-significant bit of this field (bit 16) corresponds to MRC instance 0. The most-significant bit of this field (bit 31) corresponds to MRC instance 15 (MRC instance = i-16 where i is the register bit number), and so on. Multiple bits can be set at any time indicating access violations for this domain have been detected across multiple instances of the MRCs. For each bit in this field: • 0 - The memory region checker has not detected an access violation or is not physically present. • 1 - The memory region checker has detected one or more access violations for this domain.
15-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7-0 MBCINST	MBC instance This field is a bitmap indicating the presence of a detected access violation for domain d in the implemented instances of the MBC. The least-significant bit of this field (bit 0) corresponds to MBC instance 0. The most-significant bit of this field (bit 7) corresponds to MBC instance 7 Multiple bits can be set at any time indicating access violations for this domain have been detected across multiple instances of the MBCs. For each bit in this field: • 0 - The memory block checker has not detected an access violation or is not physically present. • 1 - The memory block checker has detected one or more access violations for this domain.

22.7.3.13 DAC Initiator Domain Assignment Register (MDA_W0_0_DFMT1 - MDA_W0_3_DFMT1)

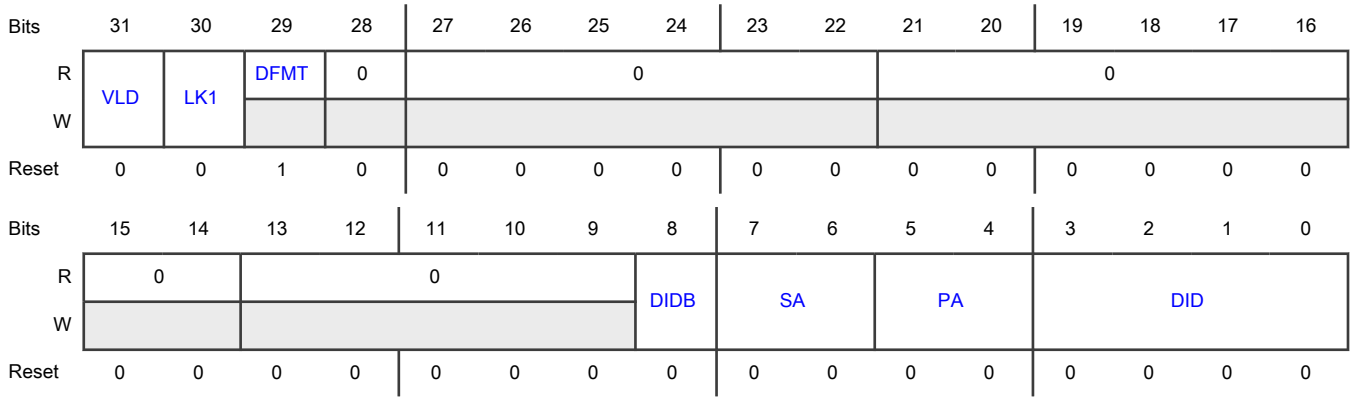
This register is identical to Initiator Domain Assignment (MDA_Wr_m_DFMT0) register except that the domain format field, DFMT, is 1 and the PID field is not used. This format supports two different specifications of the DID for non-core bus initiators.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/write.

Offset

Register	Offset
MDA_W0_0_DFMT1	800h
MDA_W0_1_DFMT1	820h
MDA_W0_2_DFMT1	840h
MDA_W0_3_DFMT1	860h

Diagram



Fields

Field	Description
31 VLD	Valid This field indicates the domain assignment is valid. It is further qualified by TRDC_CR[GVLDM] = 1. If TRDC_CR[GVLDM] is cleared, the DID output is defined by the SoC-specific *default DID* value. See the TRDC chip configuration section for more details. If both MDA_Wr_m[VLD] and TRDC_CR[GVLDM] are asserted, the DID output is defined by the remaining contents of this register. 0 - The Wr domain assignment is invalid. 1 - The Wr domain assignment is valid.
30 LK1	1-bit Lock This field provides a locking mechanism that can be used to limit the ability to write the register. Once set, this bit remains asserted until the next reset. 0 - Register can be written by any secure privileged write. 1 - Register is locked (read-only) until the next reset.
29 DFMT	Domain format Identifies this register's domain assignment format. **NOTE** This bitfield access is ROO 0 - Processor-core domain assignment 1 - Non-processor domain assignment
28 —	Reserved
27-22 —	Reserved
21-16 —	Reserved
15-14 —	Reserved
13-9 —	Reserved
8 DIDB	DID Bypass If asserted, this bit enables the bypassing of an input DID value as the domain identifier for this non-processor bus initiator. This capability allows non-processor bus initiators, for example, a DMA to masquerade as a processor. Once set, this field is “sticky” and remains set until the next reset.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>DAC</th><th>DID Input</th></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>DMA Domain ID</td></tr>
<tr><td>2</td><td>USB Domain ID</td></tr>
</table>
0 - Use MDAn[3:0] as the domain identifier.
1 - Use the DID input as the domain identifier.
</td></tr>
<tr><td>7-6
SA</td><td>Secure attribute
This field defines the secure/nonsecure attribute for non-processor cores.
NOTE
The bus initiator’s input secure/nonsecure attribute is used if SA = 1X, or this VLD = 0. If TZM_ENB = 0, the initiator attribute is forced to nonsecure. A nonsecure write cannot program this field to 2'b00, which is a security level higher than the mode of the process that is writing it.
Reset value of SA:
• if TZM_ENB=0, SA reset value = 2'b01
• if TZM_ENB=1, SA reset value = 2'b00
00 - Force the bus attribute for this initiator to secure.
01 - Force the bus attribute for this initiator to nonsecure.
10 - Use the bus initiator’s secure/nonsecure attribute directly.
11 - Use the bus initiator’s secure/nonsecure attribute directly.</td></tr>
<tr><td>5-4
PA</td><td>Privileged attribute
This field defines the privileged/user attribute for non-processor cores.
NOTE
The bus initiator’s input privileged/user attribute is used if PA = 1X, or this VLD = 0.
00 - Force the bus attribute for this initiator to user.
01 - Force the bus attribute for this initiator to privileged.
10 - Use the bus initiator’s privileged/user attribute directly.
11 - Use the bus initiator’s privileged/user attribute directly.</td></tr>
<tr><td>3-0
DID</td><td>Domain identifier
This 4-bit field is the domain ID attribute that is sent on the accesses from the bus initiator connected to the DAC when DIDB=0.
Write a value greater than the number of DIDs is illegal.</td></tr>
</table>

22.7.4 TRDC register descriptions

22.7.4.1 TRDC memory map

TRDC_MEGAMIX base address: 4281_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	TRDC Register (TRDC_CR)	32	RW	0000_0010
F0	TRDC Hardware Configuration Register 0 (TRDC_HWCFG0)	32	R	2000_0710
F4	TRDC Hardware Configuration Register 1 (TRDC_HWCFG1)	32	R	See section
100 - 106	Domain Assignment Configuration Register (DACFG0 - DACFG6)	8	R	81
1C0	TRDC IDAU Control Register (TRDC_IDAU_CR)	32	RW	0000_0008
1E0	TRDC FLW Control (TRDC_FLW_CTL)	32	RW	0000_0000
1E4	TRDC FLW Physical Base (TRDC_FLW_PBASE)	32	R	0100_0000
1E8	TRDC FLW Array Base (TRDC_FLW_ABASE)	32	RW	0000_0000
1EC	TRDC FLW Block Count (TRDC_FLW_BCNT)	32	RW	0000_0000
1FC	TRDC Fault Domain ID (TRDC_FDID)	32	RW	0000_0000
200 - 23C	TRDC Domain Error Location Register (TRDC_DERRLOC0 - TRDC_DERRLOC15)	32	R	0000_0000
800	DAC Initiator Domain Assignment Register (MDA_W0_0_DFMT1)	32	RW	2000_0000
820	DAC Initiator Domain Assignment Register (MDA_W0_1_DFMT1)	32	RW	2000_0000
840	DAC Initiator Domain Assignment Register (MDA_W0_2_DFMT1)	32	RW	2000_0000
860	DAC Initiator Domain Assignment Register (MDA_W0_3_DFMT1)	32	RW	2000_0000
880	DAC Initiator Domain Assignment Register (MDA_W0_4_DFMT1)	32	RW	2000_0000
8A0	DAC Initiator Domain Assignment Register (MDA_W0_5_DFMT1)	32	RW	2000_0000
8C0	DAC Initiator Domain Assignment Register (MDA_W0_6_DFMT1)	32	RW	2000_0000

22.7.4.2 TRDC Register (TRDC_CR)

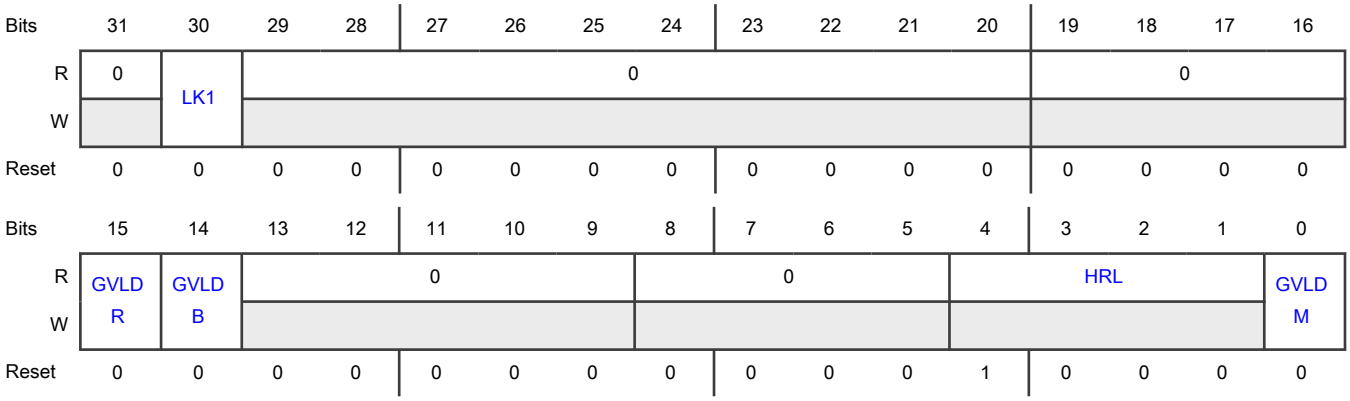
This register provides status about the TRDC and global enable bits for the entire module's operation. A fully operational TRDC requires all global bits GVLD {R,B,M} to be asserted. Undefined behavior results if they are not all asserted. If there are no region checkers (MRCs) present in the configuration, GVLDR need not be asserted for a fully operational TRDC.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_CR	0h

Diagram



Fields

Field	Description
31 —	Reserved
30 LK1	Lock Status This field is the lock status of the TRDC_CR. This field is set when all GVLD bits are set; GVLDR=GVLDB=GVLDM=1b1. Once set, this bit remains asserted until the next reset. 0 - The CR can be written by any secure privileged write. 1 - The CR is locked (read-only) until the next reset.
29-20 —	Reserved
19-16 —	Reserved
15 GVLDR	Global Valid for Memory Region Checkers TRDC global MRC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC MRCs are disabled. 1 - TRDC MRCs are enabled.
14 GVLDB	Global Valid for Memory Block Checkers TRDC global MBC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC MBCs are disabled. 1 - TRDC MBCs are enabled.
13-9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
8-5 —	Reserved
4-1 HRL	Hardware Revision Level This read-only field specifies the TRDC's hardware and definition revision level. It can be read by software to determine the functional definition of the module.
0 GVLDM	Global Valid for Domain Assignment Controllers TRDC global DAC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC DACs are disabled. 1 - TRDC DACs are enabled.

22.7.4.3 TRDC Hardware Configuration Register 0 (TRDC_HWCFG0)

This read-only register contains information on the TRDC's hardware configuration. Specifically, it defines the number of implemented domains and bus initiators along with the number of instances of memory block checkers (MBCs) and memory region checkers (MRCs). The register value at reset is device-specific. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG0	F0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MID				NMRC				0				NMBC			
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NMSTR								0				NDID			
W																
Reset	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0

Fields

Field	Description
31-29 MID	Module ID This field defines major version ID of this module.
28-24 NMRC	Number of MRCs This field defines the number of MRCs on the device [1-16].
23-20 —	Reserved
19-16 NMBC	Number of MBCs This field defines the number of MBCs on the device [1-8].
15-8 NMSTR	Number of bus initiators This read-only field defines the number of bus initiators.
7-5 —	Reserved
4-0 NDID	Number of domains This read-only field defines the number of domains on the device [1-16].

22.7.4.4 TRDC Hardware Configuration Register 1 (TRDC_HWCFG1)

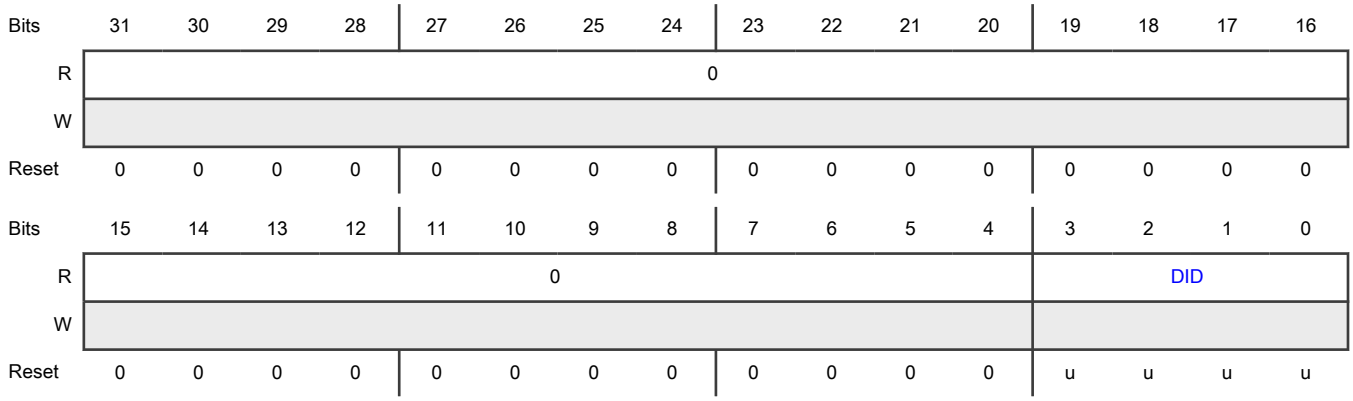
This register contains information on the TRDC's hardware configuration. It provides a mechanism for software to determine its domain number by simply reading the register. See Domain error capture management for more details on typical usage. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG1	F4h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 DID	Domain identifier number This field provides the domain number [0-15] of the requesting bus initiator.

22.7.4.5 Domain Assignment Configuration Register (DACFG0 - DACFG6)

This register defines the number of implemented domain assignment registers for bus initiator m, where m+1 can specify from 1 to 64 bus initiators. These registers are organized as a byte-sized data array and can be read using 8-, 16- or 32-bit accesses. An all-zero value (NCM = 0, NMDAR = 0) indicates a non-existent bus initiator. Attempted writes are error terminated.

Register read will not return transfer error when DACFG for a initiator doesn't exist but it is in the same 32-bit group as DACFG for an existing initiator. For example, when there are only 2 DACFG0-1 registers for 2 initiator, and initiators for DACFG2-3 don't exist, then access to DACFG2-3 won't return transfer error.

Typically, processor bus initiators have one or more domain assignment registers, while non-processor initiators have a single domain assignment register.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
DACFG0	100h
DACFG1	101h
DACFG2	102h
DACFG3	103h
DACFG4	104h
DACFG5	105h
DACFG6	106h

Diagram

Bits	7	6	5	4	3	2	1	0
R	NCM	0			NMDAR			
W								
Reset	1	0	0	0	0	0	0	1

Fields

Field	Description
7 NCM	Non-CPU Initiator This read-only field signals that bus initiator m is a non-CPU initiator. It specifies that the format of the associated MDA_Wr_m register defines a non-processor domain assignment. This field is zero for a non-existent bus initiator. 0 - Bus initiator is a processor. 1 - Bus initiator is a non-processor.
6-4 —	Reserved
3-0 NMDAR	Number of initiator domain assignment registers for bus initiator m This read-only field specifies the number of registers associated with the initiator domain assignment register for a given bus initiator. The value is limited to the range [0-8], where zero indicates a non-existent bus initiator and non-zero values indicate the number of implemented registers associated with this MDAm.

22.7.4.6 TRDC IDAU Control Register (TRDC_IDAU_CR)

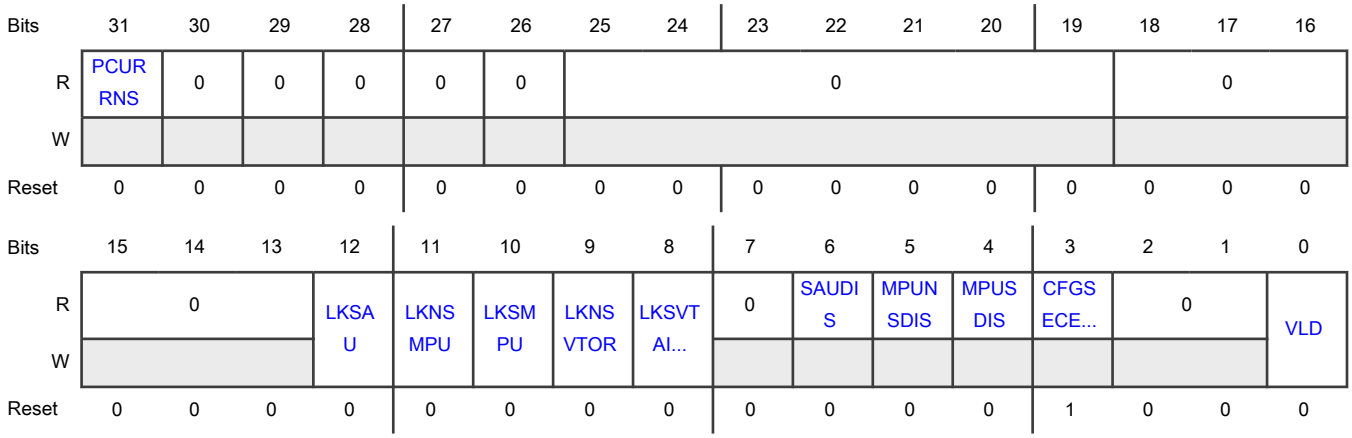
This register defines the configuration for the Implementation-Defined Attribution Unit which is tightly-coupled to the TZ-M extensions in the processor core. Many of the fields in this register explicitly control processor TZ-M functions.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_IDAU_CR	1C0h

Diagram



Fields

Field	Description
31 PCURRNS	Processor current security Current security state of the processor. 0 - Processor is in Secure state 1 - Processor is in Nonsecure state
30 —	Reserved
29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25-19 —	Reserved
18-16 —	Reserved
15-13 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 LKSAU	Lock SAU Asserting this bit prevents changes to the processor's Secure SAU memory regions already programmed. All processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the SAU_CTRL, SAU_RNR, SAU_RBAR and SAU_RLAR registers from software or from a debug agent connected to the processor
11 LKNSMPU	Lock Nonsecure MPU Asserting this bit prevents changes to the processor's Nonsecure MPU memory regions already programmed. All processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the MPU_CTRL_NS, MPU_RNR_NS, MPU_RBAR_NS, MPU_RLAR_NS, MPU_RBAR_A_NSn and MPU_RLAR_A_NSn from software or from a debug agent connected to the processor
10 LKSMPU	Lock Secure MPU Asserting this signal prevents changes to the processor's programmed Secure MPU memory regions and all processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the MPU_CTRL, MPU_RNR, MPU_RBAR, MPU_RLAR, MPU_RBAR_An and MPU_RLAR_An from software or from a debug agent connected to the processor in Secure state
9 LKNSVTOR	Lock Nonsecure Vector Table Offset Register Asserting this signal prevents changes to the processor's Nonsecure vector table base address. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock this register 1 - Disable writes to the VTOR_NS register
8 LKSVTAIRCR	Lock Secure VTOR, Application interrupt and Reset Control Registers This bit is sticky, once set, it can only be cleared with a reset. Asserting this signal prevents processor changes to: • The Secure vector table base address. • Handling of Secure interrupt priority. • BusFault, HardFault, and NMI security target settings in the processor. 0 - Unlock these registers 1 - Disable writes to the VTOR_S, AIRCR[PRIS], and AIRCR[BFHFNMINS] registers
7	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
6 SAUDIS	Security Attribution Unit Disable 0 - SAU is enabled 1 - SAU is disabled
5 MPUNSDIS	NonSecure Memory Protection Unit Disabled 0 - Nonsecure MPU is enabled 1 - Nonsecure MPU is disabled
4 MPUSDIS	Secure Memory Protection Unit Disabled 0 - Secure MPU is enabled 1 - Secure MPU is disabled
3 CFGSECEXT	Configure Security Extension 0 - Armv8M Security Extension is disabled 1 - Armv8-M Security Extension is enabled
2-1 —	Reserved
0 VLD	Valid When VLD =0, all address attribute from IDAU is nonSecure. When VLD=1, values in other fields of this register are valid.

22.7.4.7 TRDC FLW Control (TRDC_FLW_CTL)

This register provides control of the FLW = Flash Logical Window operation.

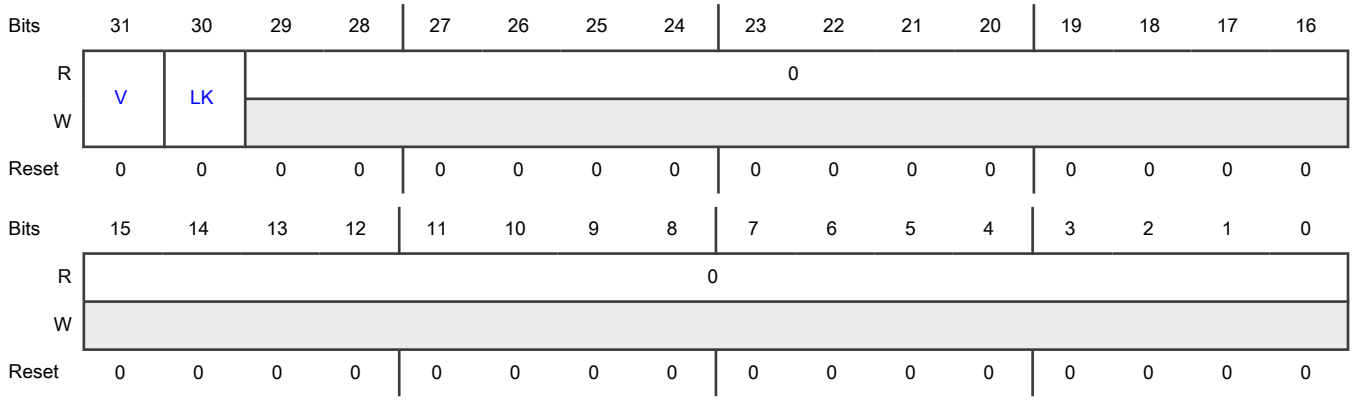
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_FLW_CTL	1E0h

Diagram



Fields

Field	Description
31 V	Valid bit 0 - FLW function is disabled. 1 - FLW function is enabled.
30 LK	Lock bit 0 - FLW registers may be modified. 1 - FLW registers are locked until the next reset.
29-0 —	Reserved

22.7.4.8 TRDC FLW Physical Base (TRDC_FLW_PBASE)

This read-only register gives the physical base address of the Flash Logical Window.

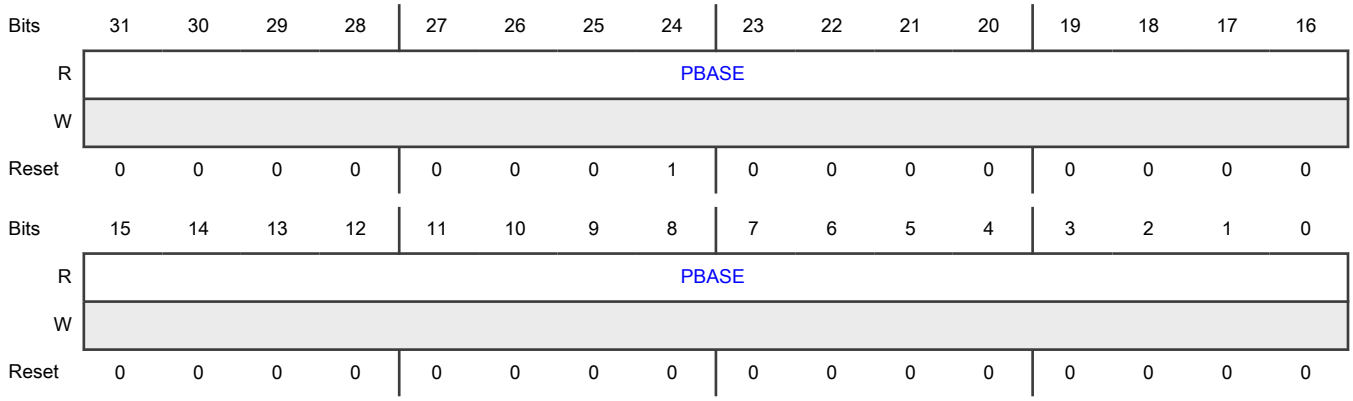
This register exists for this device but has not effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_FLW_PBASE	1E4h

Diagram



Fields

Field	Description
31-0 PBASE	Physical base address Physical address of the base of the FLW (mod 32KBytes).

22.7.4.9 TRDC FLW Array Base (TRDC_FLW_ABASE)

This register gives the flash array base address of the Flash Logical Window.

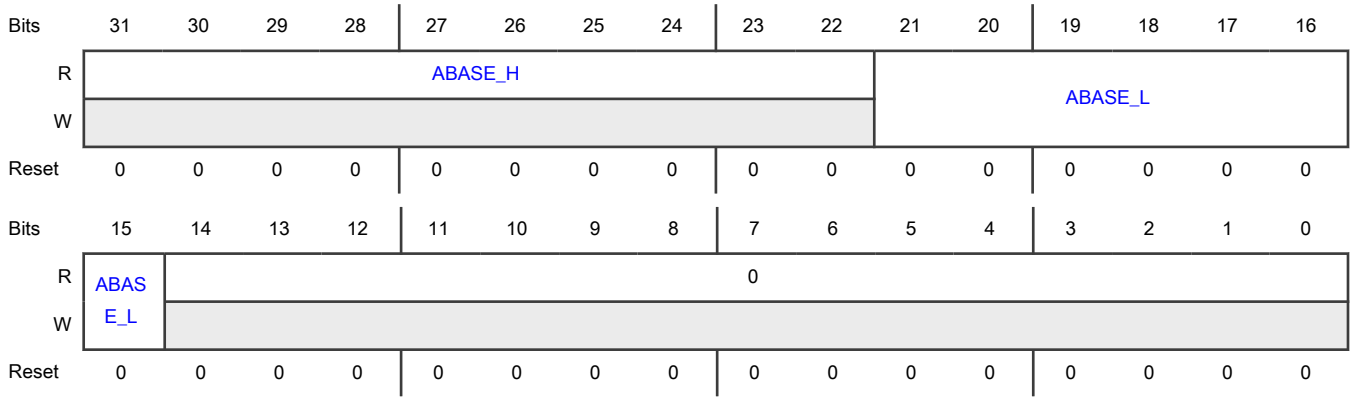
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write. This register is not writable if TRDC_FLW_CTL[LK]=1.

Offset

Register	Offset
TRDC_FLW_ABASE	1E8h

Diagram



Fields

Field	Description
31-22 ABASE_H	Array base address high Flash array address upper bits of the base of the FLW (mod 32KBytes).
21-15 ABASE_L	Array base address low Flash array address lower bits of the base of the FLW (mod 32KBytes).
14-0 —	Reserved

22.7.4.10 TRDC FLW Block Count (TRDC_FLW_BCNT)

This register gives the size of the Flash Logic Window in 32KByte blocks.

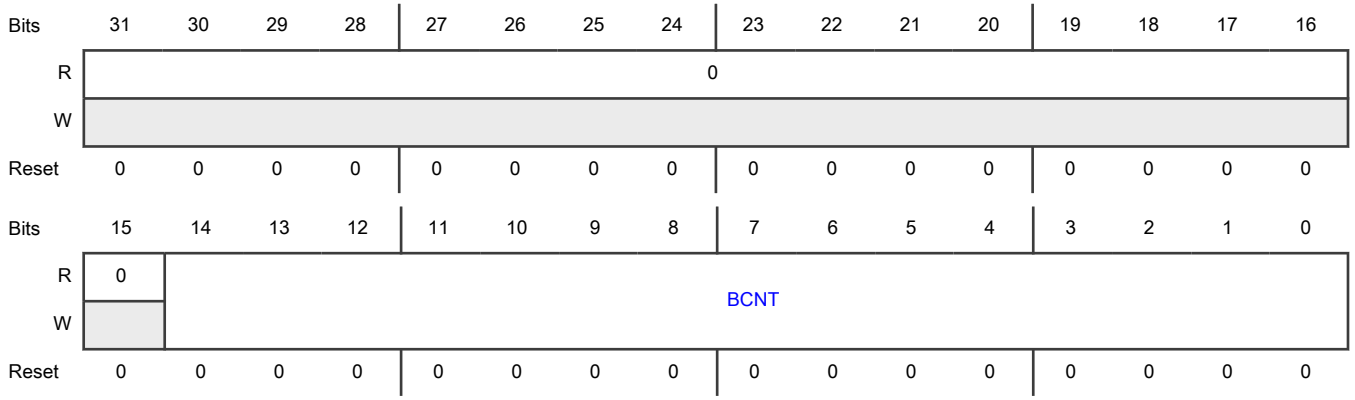
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write. This register is not writable if TRDC_FLW_CTL[LK]=1.

Offset

Register	Offset
TRDC_FLW_BCNT	1ECh

Diagram



Fields

Field	Description
31-15 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
14-0 BCNT	Block Count Size of FLW in number of 32KByte blocks.

22.7.4.11 TRDC Fault Domain ID (TRDC_FDID)

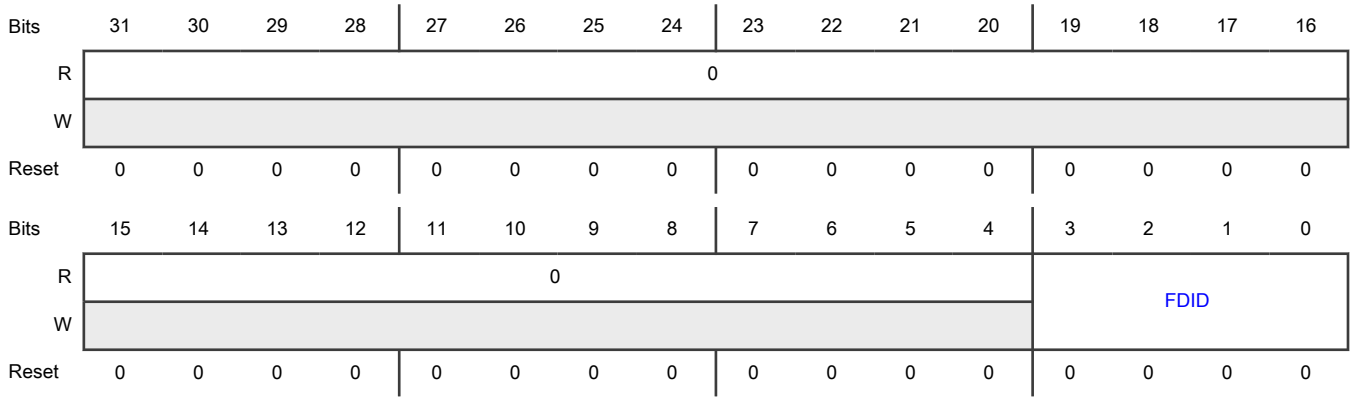
In the event of an access error, this register is used to specify the domainID of the faulting reference before indexing into the Domain Error registers.

If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_FDID	1FCh

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 FDID	Domain ID of Faulted Access This field indicates the domainID of the fault used to index the array of domain error information. The user queries the DERRLOCx registers to find the DomainID of the faulting access and must write this register with the domain ID before reading the Domain Error registers.

22.7.4.12 TRDC Domain Error Location Register (TRDC_DERRLOC0 - TRDC_DERRLOC15)

This array of read-only registers provide the instance number of the submodule where (an) access violation(s) occurred. These registers are organized as a word array, indexed by the faulting domain number, d. The two fields of this register provide a bitmap

of instances associated with all submodules containing captured error information for that domain. These instance numbers are then used as indices into the DERR_W0_*i*, DERR_W1_*i*, and DERR_W3_*i*register arrays. See Domain error capture management for more details.

When an access violation is detected by either a Memory Region Checker (MRC) or a Memory Block Checker (MBC), address and attribute information of the offending access is captured. Using the faulting domainID number as the index, d, this array of read-only registers provide additional information signaling the instance number of the submodule where the access violation(s) occurred. Since the resulting exception handler needs the submodule instance to retrieve the captured address and attribute information from DERR_W0_*i* and DERR_W1_*i*, these registers provide the instance number details.

Attempted writes are error terminated.

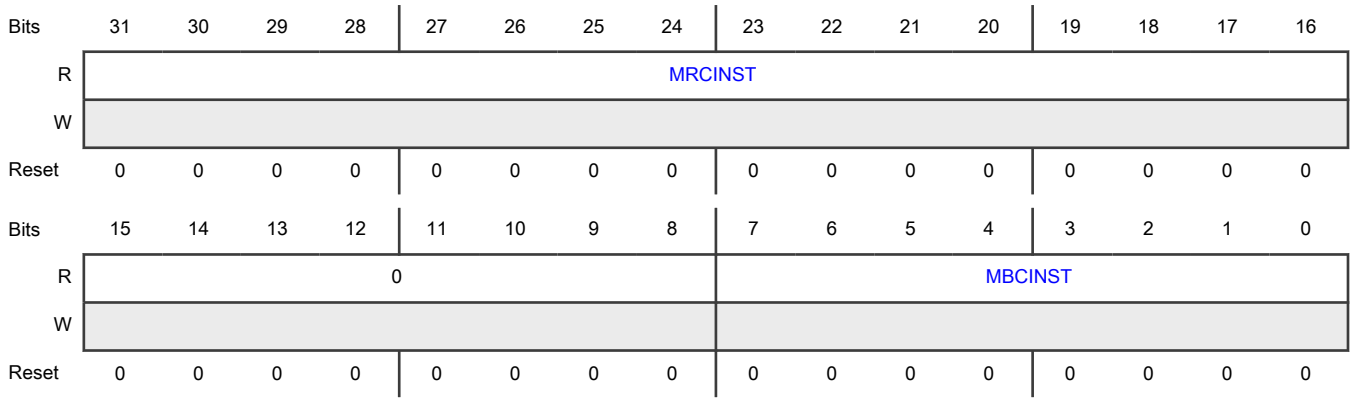
Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

For d = 0 to 15:

Register	Offset
TRDC_DERRLOCd	200h + (d × 4h)

Diagram



Fields

Field	Description
31-16 MRCINST	MRC instance This field is a bitmap indicating the presence of a detected access violation for domain d in the implemented instances of the MRC. The least-significant bit of this field (bit 16) corresponds to MRC instance 0. The most-significant bit of this field (bit 31) corresponds to MRC instance 15 (MRC instance = i-16 where i is the register bit number), and so on. Multiple bits can be set at any time indicating access violations for this domain have been detected across multiple instances of the MRCs. For each bit in this field: • 0 - The memory region checker has not detected an access violation or is not physically present. • 1 - The memory region checker has detected one or more access violations for this domain.
15-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7-0 MBCINST	MBC instance This field is a bitmap indicating the presence of a detected access violation for domain d in the implemented instances of the MBC. The least-significant bit of this field (bit 0) corresponds to MBC instance 0. The most-significant bit of this field (bit 7) corresponds to MBC instance 7 Multiple bits can be set at any time indicating access violations for this domain have been detected across multiple instances of the MBCs. For each bit in this field: • 0 - The memory block checker has not detected an access violation or is not physically present. • 1 - The memory block checker has detected one or more access violations for this domain.

22.7.4.13 DAC Initiator Domain Assignment Register (MDA_W0_0_DFMT1 - MDA_W0_6_DFMT1)

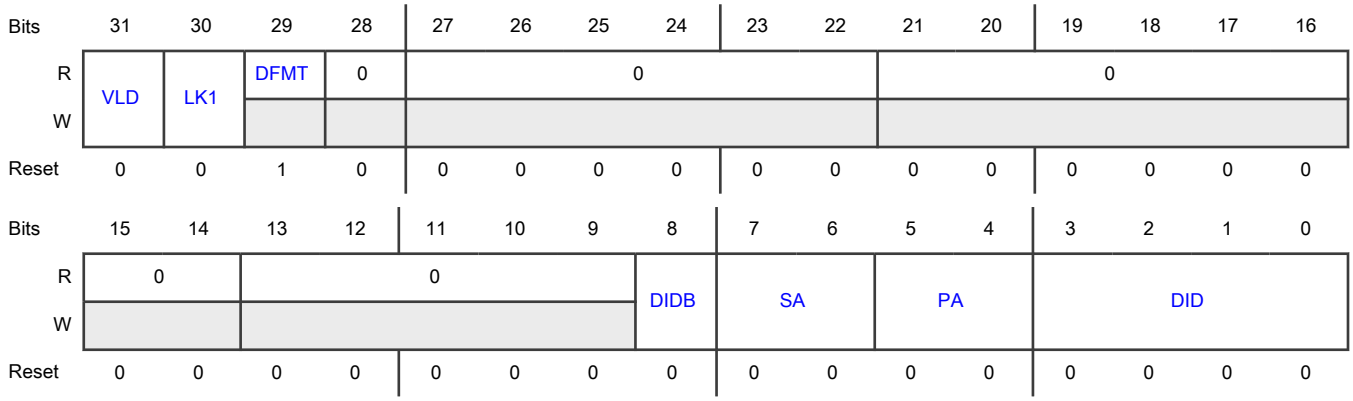
This register is identical to Initiator Domain Assignment (MDA_Wr_m_DFMT0) register except that the domain format field, DFMT, is 1 and the PID field is not used. This format supports two different specifications of the DID for non-core bus initiators.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/write.

Offset

Register	Offset
MDA_W0_0_DFMT1	800h
MDA_W0_1_DFMT1	820h
MDA_W0_2_DFMT1	840h
MDA_W0_3_DFMT1	860h
MDA_W0_4_DFMT1	880h
MDA_W0_5_DFMT1	8A0h
MDA_W0_6_DFMT1	8C0h

Diagram



Fields

Field	Description
31 VLD	Valid This field indicates the domain assignment is valid. It is further qualified by TRDC_CR[GVLDM] = 1. If TRDC_CR[GVLDM] is cleared, the DID output is defined by the SoC-specific *default DID* value. See the TRDC chip configuration section for more details. If both MDA_Wr_m[VLD] and TRDC_CR[GVLDM] are asserted, the DID output is defined by the remaining contents of this register. 0 - The Wr domain assignment is invalid. 1 - The Wr domain assignment is valid.
30 LK1	1-bit Lock This field provides a locking mechanism that can be used to limit the ability to write the register. Once set, this bit remains asserted until the next reset. 0 - Register can be written by any secure privileged write. 1 - Register is locked (read-only) until the next reset.
29 DFMT	Domain format Identifies this register's domain assignment format. **NOTE** This bitfield access is ROO 0 - Processor-core domain assignment 1 - Non-processor domain assignment
28 —	Reserved
27-22 —	Reserved
21-16 —	Reserved
15-14 —	Reserved
13-9 —	Reserved
8 DIDB	DID Bypass If asserted, this bit enables the bypassing of an input DID value as the domain identifier for this non-processor bus initiator. This capability allows non-processor bus initiators, for example, a DMA to masquerade as a processor. Once set, this field is “sticky” and remains set until the next reset.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>DAC</th><th>DID Input</th></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>DMA Domain ID</td></tr>
<tr><td>2</td><td>USB Domain ID</td></tr>
</table>
0 - Use MDAn[3:0] as the domain identifier.
1 - Use the DID input as the domain identifier.
</td></tr>
<tr><td>7-6
SA</td><td>Secure attribute
This field defines the secure/nonsecure attribute for non-processor cores.
NOTE
The bus initiator’s input secure/nonsecure attribute is used if SA = 1X, or this VLD = 0. If TZM_ENB = 0, the initiator attribute is forced to nonsecure. A nonsecure write cannot program this field to 2'b00, which is a security level higher than the mode of the process that is writing it.
Reset value of SA:
• if TZM_ENB=0, SA reset value = 2'b01
• if TZM_ENB=1, SA reset value = 2'b00
00 - Force the bus attribute for this initiator to secure.
01 - Force the bus attribute for this initiator to nonsecure.
10 - Use the bus initiator’s secure/nonsecure attribute directly.
11 - Use the bus initiator’s secure/nonsecure attribute directly.</td></tr>
<tr><td>5-4
PA</td><td>Privileged attribute
This field defines the privileged/user attribute for non-processor cores.
NOTE
The bus initiator’s input privileged/user attribute is used if PA = 1X, or this VLD = 0.
00 - Force the bus attribute for this initiator to user.
01 - Force the bus attribute for this initiator to privileged.
10 - Use the bus initiator’s privileged/user attribute directly.
11 - Use the bus initiator’s privileged/user attribute directly.</td></tr>
<tr><td>3-0
DID</td><td>Domain identifier
This 4-bit field is the domain ID attribute that is sent on the accesses from the bus initiator connected to the DAC when DIDB=0.
Write a value greater than the number of DIDs is illegal.</td></tr>
</table>

22.7.5 TRDC register descriptions

22.7.5.1 TRDC memory map

TRDC_NICMIX base address: 4901_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	TRDC Register (TRDC_CR)	32	RW	0000_0010
F0	TRDC Hardware Configuration Register 0 (TRDC_HWCFG0)	32	R	2104_0410
F4	TRDC Hardware Configuration Register 1 (TRDC_HWCFG1)	32	R	See section
F8	TRDC Hardware Configuration Register 2 (TRDC_HWCFG2)	32	R	0000_0003
FC	TRDC Hardware Configuration Register 3 (TRDC_HWCFG3)	32	R	0000_0000
100 - 103	Domain Assignment Configuration Register (DACFG0 - DACFG3)	8	R	See section
1C0	TRDC IDAU Control Register (TRDC_IDAU_CR)	32	RW	0000_0008
1E0	TRDC FLW Control (TRDC_FLW_CTL)	32	RW	0000_0000
1E4	TRDC FLW Physical Base (TRDC_FLW_PBASE)	32	R	0100_0000
1E8	TRDC FLW Array Base (TRDC_FLW_ABASE)	32	RW	0000_0000
1EC	TRDC FLW Block Count (TRDC_FLW_BCNT)	32	RW	0000_0000
1FC	TRDC Fault Domain ID (TRDC_FDID)	32	RW	0000_0000
200 - 23C	TRDC Domain Error Location Register (TRDC_DERRLOC0 - TRDC_DERRLOC15)	32	R	0000_0000
400	MBC Domain Error Word0 Register (MBC0_DERR_W0)	32	R	0000_0000
404	MBC Domain Error Word1 Register (MBC0_DERR_W1)	32	R	0000_0000
40C	MBC Domain Error Word3 Register (MBC0_DERR_W3)	32	RW	0000_0000
410	MBC Domain Error Word0 Register (MBC1_DERR_W0)	32	R	0000_0000
414	MBC Domain Error Word1 Register (MBC1_DERR_W1)	32	R	0000_0000
41C	MBC Domain Error Word3 Register (MBC1_DERR_W3)	32	RW	0000_0000
420	MBC Domain Error Word0 Register (MBC2_DERR_W0)	32	R	0000_0000
424	MBC Domain Error Word1 Register (MBC2_DERR_W1)	32	R	0000_0000
42C	MBC Domain Error Word3 Register (MBC2_DERR_W3)	32	RW	0000_0000
430	MBC Domain Error Word0 Register (MBC3_DERR_W0)	32	R	0000_0000
434	MBC Domain Error Word1 Register (MBC3_DERR_W1)	32	R	0000_0000
43C	MBC Domain Error Word3 Register (MBC3_DERR_W3)	32	RW	0000_0000
480	MRC Domain Error Word0 Register (MRC0_DERR_W0)	32	R	0000_0000
484	MRC Domain Error Word1 Register (MRC0_DERR_W1)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
48C	MRC Domain Error Word3 Register (MRC0_DERR_W3)	32	RW	0000_0000
800	DAC Initiator Domain Assignment Register (MDA_W0_0_DFMT0)	32	RW	0000_0000
804	DAC Initiator Domain Assignment Register (MDA_W1_0_DFMT0)	32	RW	0000_0000
808	DAC Initiator Domain Assignment Register (MDA_W2_0_DFMT0)	32	RW	0000_0000
80C	DAC Initiator Domain Assignment Register (MDA_W3_0_DFMT0)	32	RW	0000_0000
820	DAC Initiator Domain Assignment Register (MDA_W0_1_DFMT0)	32	RW	0000_0000
824	DAC Initiator Domain Assignment Register (MDA_W1_1_DFMT0)	32	RW	0000_0000
828	DAC Initiator Domain Assignment Register (MDA_W2_1_DFMT0)	32	RW	0000_0000
82C	DAC Initiator Domain Assignment Register (MDA_W3_1_DFMT0)	32	RW	0000_0000
840	DAC Initiator Domain Assignment Register (MDA_W0_2_DFMT1)	32	RW	2000_0000
860	DAC Initiator Domain Assignment Register (MDA_W0_3_DFMT1)	32	RW	2000_0000
1_0000	MBC Global Configuration Register (MBC0_MEM0_GLBCFG)	32	R	0010_0031
1_0004	MBC Global Configuration Register (MBC0_MEM1_GLBCFG)	32	R	0010_000B
1_0008	MBC Global Configuration Register (MBC0_MEM2_GLBCFG)	32	R	0010_0025
1_000C	MBC Global Configuration Register (MBC0_MEM3_GLBCFG)	32	R	0010_0030
1_0010	MBC NonSecure Enable Block Index (MBC0_NSE_BLK_INDEX)	32	RW	0000_0000
1_0014	MBC NonSecure Enable Block Set (MBC0_NSE_BLK_SET)	32	RW	0000_0000
1_0018	MBC NonSecure Enable Block Clear (MBC0_NSE_BLK_CLR)	32	RW	0000_0000
1_001C	MBC NonSecure Enable Block Clear All (MBC0_NSE_BLK_CLR_ALL)	32	RW	0000_0000
1_0020	MBC Global Access Control (MBC0_MEMN_GLBAC0)	32	RW	0000_0000
1_0024	MBC Global Access Control (MBC0_MEMN_GLBAC1)	32	RW	0000_0000
1_0028	MBC Global Access Control (MBC0_MEMN_GLBAC2)	32	RW	0000_0000
1_002C	MBC Global Access Control (MBC0_MEMN_GLBAC3)	32	RW	0000_0000
1_0030	MBC Global Access Control (MBC0_MEMN_GLBAC4)	32	RW	0000_0000
1_0034	MBC Global Access Control (MBC0_MEMN_GLBAC5)	32	RW	0000_0000
1_0038	MBC Global Access Control (MBC0_MEMN_GLBAC6)	32	RW	0000_0000
1_003C	MBC Global Access Control (MBC0_MEMN_GLBAC7)	32	RW	0000_0000
1_0040	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0044	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0048	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_004C	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0050	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0054	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0058	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0140	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0144	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0180	MBC Memory Block Configuration Word (MBC0_DOM0_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0184	MBC Memory Block Configuration Word (MBC0_DOM0_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_01A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_01A8	MBC Memory Block Configuration Word (MBC0_DOM0_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_01AC	MBC Memory Block Configuration Word (MBC0_DOM0_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_01B0	MBC Memory Block Configuration Word (MBC0_DOM0_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_01B4	MBC Memory Block Configuration Word (MBC0_DOM0_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_01B8	MBC Memory Block Configuration Word (MBC0_DOM0_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_01C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_01CC	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_01D0	MBC Memory Block Configuration Word (MBC0_DOM0_MEM3_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_01D4	MBC Memory Block Configuration Word (MBC0_DOM0_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_01D8	MBC Memory Block Configuration Word (MBC0_DOM0_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_01DC	MBC Memory Block Configuration Word (MBC0_DOM0_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_01E0	MBC Memory Block Configuration Word (MBC0_DOM0_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_01E4	MBC Memory Block Configuration Word (MBC0_DOM0_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_01F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_01F4	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_0240	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0244	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0248	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_024C	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0250	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0254	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0258	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0340	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0344	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0380	MBC Memory Block Configuration Word (MBC0_DOM1_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0384	MBC Memory Block Configuration Word (MBC0_DOM1_MEM1_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_03A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_03A8	MBC Memory Block Configuration Word (MBC0_DOM1_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_03AC	MBC Memory Block Configuration Word (MBC0_DOM1_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_03B0	MBC Memory Block Configuration Word (MBC0_DOM1_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_03B4	MBC Memory Block Configuration Word (MBC0_DOM1_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_03B8	MBC Memory Block Configuration Word (MBC0_DOM1_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_03C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_03CC	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_03D0	MBC Memory Block Configuration Word (MBC0_DOM1_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_03D4	MBC Memory Block Configuration Word (MBC0_DOM1_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_03D8	MBC Memory Block Configuration Word (MBC0_DOM1_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_03DC	MBC Memory Block Configuration Word (MBC0_DOM1_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_03E0	MBC Memory Block Configuration Word (MBC0_DOM1_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_03E4	MBC Memory Block Configuration Word (MBC0_DOM1_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_03F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_03F4	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_0440	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0444	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0448	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_044C	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0450	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0454	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0458	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0540	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0544	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0580	MBC Memory Block Configuration Word (MBC0_DOM2_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0584	MBC Memory Block Configuration Word (MBC0_DOM2_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_05A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_05A8	MBC Memory Block Configuration Word (MBC0_DOM2_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_05AC	MBC Memory Block Configuration Word (MBC0_DOM2_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_05B0	MBC Memory Block Configuration Word (MBC0_DOM2_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_05B4	MBC Memory Block Configuration Word (MBC0_DOM2_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_05B8	MBC Memory Block Configuration Word (MBC0_DOM2_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_05C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_05CC	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_05D0	MBC Memory Block Configuration Word (MBC0_DOM2_MEM3_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_05D4	MBC Memory Block Configuration Word (MBC0_DOM2_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_05D8	MBC Memory Block Configuration Word (MBC0_DOM2_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_05DC	MBC Memory Block Configuration Word (MBC0_DOM2_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_05E0	MBC Memory Block Configuration Word (MBC0_DOM2_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_05E4	MBC Memory Block Configuration Word (MBC0_DOM2_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_05F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_05F4	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_0640	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0644	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0648	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_064C	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0650	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0654	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0658	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0740	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0744	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0780	MBC Memory Block Configuration Word (MBC0_DOM3_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0784	MBC Memory Block Configuration Word (MBC0_DOM3_MEM1_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_07A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_07A8	MBC Memory Block Configuration Word (MBC0_DOM3_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_07AC	MBC Memory Block Configuration Word (MBC0_DOM3_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_07B0	MBC Memory Block Configuration Word (MBC0_DOM3_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_07B4	MBC Memory Block Configuration Word (MBC0_DOM3_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_07B8	MBC Memory Block Configuration Word (MBC0_DOM3_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_07C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_07CC	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_07D0	MBC Memory Block Configuration Word (MBC0_DOM3_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_07D4	MBC Memory Block Configuration Word (MBC0_DOM3_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_07D8	MBC Memory Block Configuration Word (MBC0_DOM3_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_07DC	MBC Memory Block Configuration Word (MBC0_DOM3_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_07E0	MBC Memory Block Configuration Word (MBC0_DOM3_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_07E4	MBC Memory Block Configuration Word (MBC0_DOM3_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_07F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_07F4	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_0840	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0844	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0848	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_084C	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0850	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0854	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0858	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0940	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0944	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0980	MBC Memory Block Configuration Word (MBC0_DOM4_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0984	MBC Memory Block Configuration Word (MBC0_DOM4_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_09A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_09A8	MBC Memory Block Configuration Word (MBC0_DOM4_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_09AC	MBC Memory Block Configuration Word (MBC0_DOM4_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_09B0	MBC Memory Block Configuration Word (MBC0_DOM4_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_09B4	MBC Memory Block Configuration Word (MBC0_DOM4_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_09B8	MBC Memory Block Configuration Word (MBC0_DOM4_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_09C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_09CC	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_09D0	MBC Memory Block Configuration Word (MBC0_DOM4_MEM3_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_09D4	MBC Memory Block Configuration Word (MBC0_DOM4_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_09D8	MBC Memory Block Configuration Word (MBC0_DOM4_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_09DC	MBC Memory Block Configuration Word (MBC0_DOM4_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_09E0	MBC Memory Block Configuration Word (MBC0_DOM4_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_09E4	MBC Memory Block Configuration Word (MBC0_DOM4_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_09F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_09F4	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_0A40	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0A44	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0A48	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_0A4C	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0A50	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0A54	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0A58	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0B40	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0B44	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0B80	MBC Memory Block Configuration Word (MBC0_DOM5_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0B84	MBC Memory Block Configuration Word (MBC0_DOM5_MEM1_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0BA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_0BA8	MBC Memory Block Configuration Word (MBC0_DOM5_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_0BAC	MBC Memory Block Configuration Word (MBC0_DOM5_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_0BB0	MBC Memory Block Configuration Word (MBC0_DOM5_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_0BB4	MBC Memory Block Configuration Word (MBC0_DOM5_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_0BB8	MBC Memory Block Configuration Word (MBC0_DOM5_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_0BC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0BCC	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_0BD0	MBC Memory Block Configuration Word (MBC0_DOM5_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_0BD4	MBC Memory Block Configuration Word (MBC0_DOM5_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_0BD8	MBC Memory Block Configuration Word (MBC0_DOM5_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_0BDC	MBC Memory Block Configuration Word (MBC0_DOM5_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_0BE0	MBC Memory Block Configuration Word (MBC0_DOM5_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_0BE4	MBC Memory Block Configuration Word (MBC0_DOM5_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_0BF0	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_0BF4	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_0C40	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0C44	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0C48	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_0C4C	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0C50	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0C54	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0C58	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0D40	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0D44	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0D80	MBC Memory Block Configuration Word (MBC0_DOM6_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0D84	MBC Memory Block Configuration Word (MBC0_DOM6_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_0DA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_0DA8	MBC Memory Block Configuration Word (MBC0_DOM6_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_0DAC	MBC Memory Block Configuration Word (MBC0_DOM6_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_0DB0	MBC Memory Block Configuration Word (MBC0_DOM6_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_0DB4	MBC Memory Block Configuration Word (MBC0_DOM6_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_0DB8	MBC Memory Block Configuration Word (MBC0_DOM6_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_0DC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0DCC	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_0DD0	MBC Memory Block Configuration Word (MBC0_DOM6_MEM3_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0DD4	MBC Memory Block Configuration Word (MBC0_DOM6_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_0DD8	MBC Memory Block Configuration Word (MBC0_DOM6_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_0DDC	MBC Memory Block Configuration Word (MBC0_DOM6_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_0DE0	MBC Memory Block Configuration Word (MBC0_DOM6_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_0DE4	MBC Memory Block Configuration Word (MBC0_DOM6_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_0DF0	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_0DF4	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_0E40	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0E44	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0E48	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_0E4C	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0E50	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0E54	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0E58	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0F40	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0F44	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0F80	MBC Memory Block Configuration Word (MBC0_DOM7_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0F84	MBC Memory Block Configuration Word (MBC0_DOM7_MEM1_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0FA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_0FA8	MBC Memory Block Configuration Word (MBC0_DOM7_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_0FAC	MBC Memory Block Configuration Word (MBC0_DOM7_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_0FB0	MBC Memory Block Configuration Word (MBC0_DOM7_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_0FB4	MBC Memory Block Configuration Word (MBC0_DOM7_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_0FB8	MBC Memory Block Configuration Word (MBC0_DOM7_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_0FC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0FCC	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_0FD0	MBC Memory Block Configuration Word (MBC0_DOM7_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_0FD4	MBC Memory Block Configuration Word (MBC0_DOM7_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_0FD8	MBC Memory Block Configuration Word (MBC0_DOM7_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_0FDC	MBC Memory Block Configuration Word (MBC0_DOM7_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_0FE0	MBC Memory Block Configuration Word (MBC0_DOM7_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_0FE4	MBC Memory Block Configuration Word (MBC0_DOM7_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_0FF0	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_0FF4	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_1040	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1044	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1048	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_104C	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1050	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1054	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1058	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1140	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1144	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1180	MBC Memory Block Configuration Word (MBC0_DOM8_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1184	MBC Memory Block Configuration Word (MBC0_DOM8_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_11A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_11A8	MBC Memory Block Configuration Word (MBC0_DOM8_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_11AC	MBC Memory Block Configuration Word (MBC0_DOM8_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_11B0	MBC Memory Block Configuration Word (MBC0_DOM8_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_11B4	MBC Memory Block Configuration Word (MBC0_DOM8_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_11B8	MBC Memory Block Configuration Word (MBC0_DOM8_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_11C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_11CC	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_11D0	MBC Memory Block Configuration Word (MBC0_DOM8_MEM3_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_11D4	MBC Memory Block Configuration Word (MBC0_DOM8_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_11D8	MBC Memory Block Configuration Word (MBC0_DOM8_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_11DC	MBC Memory Block Configuration Word (MBC0_DOM8_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_11E0	MBC Memory Block Configuration Word (MBC0_DOM8_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_11E4	MBC Memory Block Configuration Word (MBC0_DOM8_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_11F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_11F4	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_1240	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1244	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1248	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_124C	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1250	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1254	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1258	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1340	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1344	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1380	MBC Memory Block Configuration Word (MBC0_DOM9_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1384	MBC Memory Block Configuration Word (MBC0_DOM9_MEM1_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_13A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_13A8	MBC Memory Block Configuration Word (MBC0_DOM9_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_13AC	MBC Memory Block Configuration Word (MBC0_DOM9_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_13B0	MBC Memory Block Configuration Word (MBC0_DOM9_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_13B4	MBC Memory Block Configuration Word (MBC0_DOM9_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_13B8	MBC Memory Block Configuration Word (MBC0_DOM9_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_13C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_13CC	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_13D0	MBC Memory Block Configuration Word (MBC0_DOM9_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_13D4	MBC Memory Block Configuration Word (MBC0_DOM9_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_13D8	MBC Memory Block Configuration Word (MBC0_DOM9_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_13DC	MBC Memory Block Configuration Word (MBC0_DOM9_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_13E0	MBC Memory Block Configuration Word (MBC0_DOM9_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_13E4	MBC Memory Block Configuration Word (MBC0_DOM9_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_13F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_13F4	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_1440	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1444	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1448	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_144C	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1450	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1454	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1458	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1540	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1544	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1580	MBC Memory Block Configuration Word (MBC0_DOM10_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1584	MBC Memory Block Configuration Word (MBC0_DOM10_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_15A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_15A8	MBC Memory Block Configuration Word (MBC0_DOM10_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_15AC	MBC Memory Block Configuration Word (MBC0_DOM10_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_15B0	MBC Memory Block Configuration Word (MBC0_DOM10_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_15B4	MBC Memory Block Configuration Word (MBC0_DOM10_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_15B8	MBC Memory Block Configuration Word (MBC0_DOM10_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_15C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_15CC	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_15D0	MBC Memory Block Configuration Word (MBC0_DOM10_MEM3_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_15D4	MBC Memory Block Configuration Word (MBC0_DOM10_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_15D8	MBC Memory Block Configuration Word (MBC0_DOM10_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_15DC	MBC Memory Block Configuration Word (MBC0_DOM10_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_15E0	MBC Memory Block Configuration Word (MBC0_DOM10_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_15E4	MBC Memory Block Configuration Word (MBC0_DOM10_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_15F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_15F4	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_1640	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1644	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1648	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_164C	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1650	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1654	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1658	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1740	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1744	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1780	MBC Memory Block Configuration Word (MBC0_DOM11_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1784	MBC Memory Block Configuration Word (MBC0_DOM11_MEM1_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_17A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_17A8	MBC Memory Block Configuration Word (MBC0_DOM11_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_17AC	MBC Memory Block Configuration Word (MBC0_DOM11_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_17B0	MBC Memory Block Configuration Word (MBC0_DOM11_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_17B4	MBC Memory Block Configuration Word (MBC0_DOM11_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_17B8	MBC Memory Block Configuration Word (MBC0_DOM11_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_17C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_17CC	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_17D0	MBC Memory Block Configuration Word (MBC0_DOM11_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_17D4	MBC Memory Block Configuration Word (MBC0_DOM11_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_17D8	MBC Memory Block Configuration Word (MBC0_DOM11_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_17DC	MBC Memory Block Configuration Word (MBC0_DOM11_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_17E0	MBC Memory Block Configuration Word (MBC0_DOM11_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_17E4	MBC Memory Block Configuration Word (MBC0_DOM11_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_17F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_17F4	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_1840	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1844	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1848	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_184C	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1850	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1854	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1858	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1940	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1944	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1980	MBC Memory Block Configuration Word (MBC0_DOM12_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1984	MBC Memory Block Configuration Word (MBC0_DOM12_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_19A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_19A8	MBC Memory Block Configuration Word (MBC0_DOM12_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_19AC	MBC Memory Block Configuration Word (MBC0_DOM12_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_19B0	MBC Memory Block Configuration Word (MBC0_DOM12_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_19B4	MBC Memory Block Configuration Word (MBC0_DOM12_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_19B8	MBC Memory Block Configuration Word (MBC0_DOM12_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_19C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_19CC	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_19D0	MBC Memory Block Configuration Word (MBC0_DOM12_MEM3_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_19D4	MBC Memory Block Configuration Word (MBC0_DOM12_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_19D8	MBC Memory Block Configuration Word (MBC0_DOM12_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_19DC	MBC Memory Block Configuration Word (MBC0_DOM12_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_19E0	MBC Memory Block Configuration Word (MBC0_DOM12_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_19E4	MBC Memory Block Configuration Word (MBC0_DOM12_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_19F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_19F4	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_1A40	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1A44	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1A48	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_1A4C	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1A50	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1A54	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1A58	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1B40	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1B44	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1B80	MBC Memory Block Configuration Word (MBC0_DOM13_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1B84	MBC Memory Block Configuration Word (MBC0_DOM13_MEM1_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1BA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_1BA8	MBC Memory Block Configuration Word (MBC0_DOM13_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_1BAC	MBC Memory Block Configuration Word (MBC0_DOM13_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_1BB0	MBC Memory Block Configuration Word (MBC0_DOM13_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_1BB4	MBC Memory Block Configuration Word (MBC0_DOM13_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_1BB8	MBC Memory Block Configuration Word (MBC0_DOM13_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_1BC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1BCC	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_1BD0	MBC Memory Block Configuration Word (MBC0_DOM13_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_1BD4	MBC Memory Block Configuration Word (MBC0_DOM13_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_1BD8	MBC Memory Block Configuration Word (MBC0_DOM13_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_1BDC	MBC Memory Block Configuration Word (MBC0_DOM13_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_1BE0	MBC Memory Block Configuration Word (MBC0_DOM13_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_1BE4	MBC Memory Block Configuration Word (MBC0_DOM13_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_1BF0	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_1BF4	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_1C40	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1C44	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1C48	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_1C4C	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1C50	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1C54	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1C58	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1D40	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1D44	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1D80	MBC Memory Block Configuration Word (MBC0_DOM14_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1D84	MBC Memory Block Configuration Word (MBC0_DOM14_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_1DA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_1DA8	MBC Memory Block Configuration Word (MBC0_DOM14_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_1DAC	MBC Memory Block Configuration Word (MBC0_DOM14_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_1DB0	MBC Memory Block Configuration Word (MBC0_DOM14_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_1DB4	MBC Memory Block Configuration Word (MBC0_DOM14_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_1DB8	MBC Memory Block Configuration Word (MBC0_DOM14_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_1DC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1DCC	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_1DD0	MBC Memory Block Configuration Word (MBC0_DOM14_MEM3_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1DD4	MBC Memory Block Configuration Word (MBC0_DOM14_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_1DD8	MBC Memory Block Configuration Word (MBC0_DOM14_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_1DDC	MBC Memory Block Configuration Word (MBC0_DOM14_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_1DE0	MBC Memory Block Configuration Word (MBC0_DOM14_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_1DE4	MBC Memory Block Configuration Word (MBC0_DOM14_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_1DF0	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_1DF4	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_1E40	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1E44	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1E48	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_1E4C	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1E50	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1E54	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1E58	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1F40	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1F44	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1F80	MBC Memory Block Configuration Word (MBC0_DOM15_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1F84	MBC Memory Block Configuration Word (MBC0_DOM15_MEM1_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1FA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_1FA8	MBC Memory Block Configuration Word (MBC0_DOM15_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_1FAC	MBC Memory Block Configuration Word (MBC0_DOM15_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_1FB0	MBC Memory Block Configuration Word (MBC0_DOM15_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_1FB4	MBC Memory Block Configuration Word (MBC0_DOM15_MEM2_BLK_CFG_W3)	32	RW	0000_0000
1_1FB8	MBC Memory Block Configuration Word (MBC0_DOM15_MEM2_BLK_CFG_W4)	32	RW	0000_0000
1_1FC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1FCC	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM2_BLK_NSE_W1)	32	RW	0000_0000
1_1FD0	MBC Memory Block Configuration Word (MBC0_DOM15_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_1FD4	MBC Memory Block Configuration Word (MBC0_DOM15_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_1FD8	MBC Memory Block Configuration Word (MBC0_DOM15_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_1FDC	MBC Memory Block Configuration Word (MBC0_DOM15_MEM3_BLK_CFG_W3)	32	RW	0000_0000
1_1FE0	MBC Memory Block Configuration Word (MBC0_DOM15_MEM3_BLK_CFG_W4)	32	RW	0000_0000
1_1FE4	MBC Memory Block Configuration Word (MBC0_DOM15_MEM3_BLK_CFG_W5)	32	RW	0000_0000
1_1FF0	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_1FF4	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM3_BLK_NSE_W1)	32	RW	0000_0000
1_2000	MBC Global Configuration Register (MBC1_MEM0_GLBCFG)	32	R	0010_0005
1_2004	MBC Global Configuration Register (MBC1_MEM1_GLBCFG)	32	R	0010_0005
1_2008	MBC Global Configuration Register (MBC1_MEM2_GLBCFG)	32	R	0010_0011
1_200C	MBC Global Configuration Register (MBC1_MEM3_GLBCFG)	32	R	0010_0011

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2010	MBC NonSecure Enable Block Index (MBC1_NSE_BLK_INDEX)	32	RW	0000_0000
1_2014	MBC NonSecure Enable Block Set (MBC1_NSE_BLK_SET)	32	RW	0000_0000
1_2018	MBC NonSecure Enable Block Clear (MBC1_NSE_BLK_CLR)	32	RW	0000_0000
1_201C	MBC NonSecure Enable Block Clear All (MBC1_NSE_BLK_CLR_ALL)	32	RW	0000_0000
1_2020	MBC Global Access Control (MBC1_MEMN_GLBAC0)	32	RW	0000_0000
1_2024	MBC Global Access Control (MBC1_MEMN_GLBAC1)	32	RW	0000_0000
1_2028	MBC Global Access Control (MBC1_MEMN_GLBAC2)	32	RW	0000_0000
1_202C	MBC Global Access Control (MBC1_MEMN_GLBAC3)	32	RW	0000_0000
1_2030	MBC Global Access Control (MBC1_MEMN_GLBAC4)	32	RW	0000_0000
1_2034	MBC Global Access Control (MBC1_MEMN_GLBAC5)	32	RW	0000_0000
1_2038	MBC Global Access Control (MBC1_MEMN_GLBAC6)	32	RW	0000_0000
1_203C	MBC Global Access Control (MBC1_MEMN_GLBAC7)	32	RW	0000_0000
1_2040	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2140	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2180	MBC Memory Block Configuration Word (MBC1_DOM0_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_21A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_21A8	MBC Memory Block Configuration Word (MBC1_DOM0_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_21AC	MBC Memory Block Configuration Word (MBC1_DOM0_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_21B0	MBC Memory Block Configuration Word (MBC1_DOM0_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_21C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_21D0	MBC Memory Block Configuration Word (MBC1_DOM0_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_21D4	MBC Memory Block Configuration Word (MBC1_DOM0_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_21D8	MBC Memory Block Configuration Word (MBC1_DOM0_MEM3_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_21F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2240	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2340	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2380	MBC Memory Block Configuration Word (MBC1_DOM1_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_23A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_23A8	MBC Memory Block Configuration Word (MBC1_DOM1_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_23AC	MBC Memory Block Configuration Word (MBC1_DOM1_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_23B0	MBC Memory Block Configuration Word (MBC1_DOM1_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_23C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_23D0	MBC Memory Block Configuration Word (MBC1_DOM1_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_23D4	MBC Memory Block Configuration Word (MBC1_DOM1_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_23D8	MBC Memory Block Configuration Word (MBC1_DOM1_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_23F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2440	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2540	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2580	MBC Memory Block Configuration Word (MBC1_DOM2_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_25A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_25A8	MBC Memory Block Configuration Word (MBC1_DOM2_MEM2_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_25AC	MBC Memory Block Configuration Word (MBC1_DOM2_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_25B0	MBC Memory Block Configuration Word (MBC1_DOM2_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_25C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_25D0	MBC Memory Block Configuration Word (MBC1_DOM2_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_25D4	MBC Memory Block Configuration Word (MBC1_DOM2_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_25D8	MBC Memory Block Configuration Word (MBC1_DOM2_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_25F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2640	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2740	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2780	MBC Memory Block Configuration Word (MBC1_DOM3_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_27A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_27A8	MBC Memory Block Configuration Word (MBC1_DOM3_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_27AC	MBC Memory Block Configuration Word (MBC1_DOM3_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_27B0	MBC Memory Block Configuration Word (MBC1_DOM3_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_27C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_27D0	MBC Memory Block Configuration Word (MBC1_DOM3_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_27D4	MBC Memory Block Configuration Word (MBC1_DOM3_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_27D8	MBC Memory Block Configuration Word (MBC1_DOM3_MEM3_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_27F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2840	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2940	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2980	MBC Memory Block Configuration Word (MBC1_DOM4_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_29A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_29A8	MBC Memory Block Configuration Word (MBC1_DOM4_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_29AC	MBC Memory Block Configuration Word (MBC1_DOM4_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_29B0	MBC Memory Block Configuration Word (MBC1_DOM4_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_29C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_29D0	MBC Memory Block Configuration Word (MBC1_DOM4_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_29D4	MBC Memory Block Configuration Word (MBC1_DOM4_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_29D8	MBC Memory Block Configuration Word (MBC1_DOM4_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_29F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2A40	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2B40	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2B80	MBC Memory Block Configuration Word (MBC1_DOM5_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2BA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2BA8	MBC Memory Block Configuration Word (MBC1_DOM5_MEM2_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2BAC	MBC Memory Block Configuration Word (MBC1_DOM5_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_2BB0	MBC Memory Block Configuration Word (MBC1_DOM5_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_2BC8	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_2BD0	MBC Memory Block Configuration Word (MBC1_DOM5_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_2BD4	MBC Memory Block Configuration Word (MBC1_DOM5_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_2BD8	MBC Memory Block Configuration Word (MBC1_DOM5_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_2BF0	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2C40	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2D40	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2D80	MBC Memory Block Configuration Word (MBC1_DOM6_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2DA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2DA8	MBC Memory Block Configuration Word (MBC1_DOM6_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_2DAC	MBC Memory Block Configuration Word (MBC1_DOM6_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_2DB0	MBC Memory Block Configuration Word (MBC1_DOM6_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_2DC8	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_2DD0	MBC Memory Block Configuration Word (MBC1_DOM6_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_2DD4	MBC Memory Block Configuration Word (MBC1_DOM6_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_2DD8	MBC Memory Block Configuration Word (MBC1_DOM6_MEM3_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2DF0	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2E40	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2F40	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2F80	MBC Memory Block Configuration Word (MBC1_DOM7_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2FA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2FA8	MBC Memory Block Configuration Word (MBC1_DOM7_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_2FAC	MBC Memory Block Configuration Word (MBC1_DOM7_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_2FB0	MBC Memory Block Configuration Word (MBC1_DOM7_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_2FC8	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_2FD0	MBC Memory Block Configuration Word (MBC1_DOM7_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_2FD4	MBC Memory Block Configuration Word (MBC1_DOM7_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_2FD8	MBC Memory Block Configuration Word (MBC1_DOM7_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_2FF0	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3040	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3140	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3180	MBC Memory Block Configuration Word (MBC1_DOM8_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_31A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_31A8	MBC Memory Block Configuration Word (MBC1_DOM8_MEM2_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_31AC	MBC Memory Block Configuration Word (MBC1_DOM8_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_31B0	MBC Memory Block Configuration Word (MBC1_DOM8_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_31C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_31D0	MBC Memory Block Configuration Word (MBC1_DOM8_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_31D4	MBC Memory Block Configuration Word (MBC1_DOM8_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_31D8	MBC Memory Block Configuration Word (MBC1_DOM8_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_31F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3240	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3340	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3380	MBC Memory Block Configuration Word (MBC1_DOM9_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_33A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_33A8	MBC Memory Block Configuration Word (MBC1_DOM9_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_33AC	MBC Memory Block Configuration Word (MBC1_DOM9_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_33B0	MBC Memory Block Configuration Word (MBC1_DOM9_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_33C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_33D0	MBC Memory Block Configuration Word (MBC1_DOM9_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_33D4	MBC Memory Block Configuration Word (MBC1_DOM9_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_33D8	MBC Memory Block Configuration Word (MBC1_DOM9_MEM3_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_33F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3440	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3540	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3580	MBC Memory Block Configuration Word (MBC1_DOM10_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_35A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_35A8	MBC Memory Block Configuration Word (MBC1_DOM10_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_35AC	MBC Memory Block Configuration Word (MBC1_DOM10_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_35B0	MBC Memory Block Configuration Word (MBC1_DOM10_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_35C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_35D0	MBC Memory Block Configuration Word (MBC1_DOM10_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_35D4	MBC Memory Block Configuration Word (MBC1_DOM10_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_35D8	MBC Memory Block Configuration Word (MBC1_DOM10_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_35F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3640	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3740	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3780	MBC Memory Block Configuration Word (MBC1_DOM11_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_37A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_37A8	MBC Memory Block Configuration Word (MBC1_DOM11_MEM2_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_37AC	MBC Memory Block Configuration Word (MBC1_DOM11_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_37B0	MBC Memory Block Configuration Word (MBC1_DOM11_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_37C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_37D0	MBC Memory Block Configuration Word (MBC1_DOM11_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_37D4	MBC Memory Block Configuration Word (MBC1_DOM11_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_37D8	MBC Memory Block Configuration Word (MBC1_DOM11_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_37F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3840	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3940	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3980	MBC Memory Block Configuration Word (MBC1_DOM12_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_39A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_39A8	MBC Memory Block Configuration Word (MBC1_DOM12_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_39AC	MBC Memory Block Configuration Word (MBC1_DOM12_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_39B0	MBC Memory Block Configuration Word (MBC1_DOM12_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_39C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_39D0	MBC Memory Block Configuration Word (MBC1_DOM12_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_39D4	MBC Memory Block Configuration Word (MBC1_DOM12_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_39D8	MBC Memory Block Configuration Word (MBC1_DOM12_MEM3_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_39F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3A40	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3B40	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3B80	MBC Memory Block Configuration Word (MBC1_DOM13_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3BA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3BA8	MBC Memory Block Configuration Word (MBC1_DOM13_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_3BAC	MBC Memory Block Configuration Word (MBC1_DOM13_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_3BB0	MBC Memory Block Configuration Word (MBC1_DOM13_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_3BC8	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_3BD0	MBC Memory Block Configuration Word (MBC1_DOM13_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_3BD4	MBC Memory Block Configuration Word (MBC1_DOM13_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_3BD8	MBC Memory Block Configuration Word (MBC1_DOM13_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_3BF0	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3C40	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3D40	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3D80	MBC Memory Block Configuration Word (MBC1_DOM14_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3DA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3DA8	MBC Memory Block Configuration Word (MBC1_DOM14_MEM2_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_3DAC	MBC Memory Block Configuration Word (MBC1_DOM14_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_3DB0	MBC Memory Block Configuration Word (MBC1_DOM14_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_3DC8	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_3DD0	MBC Memory Block Configuration Word (MBC1_DOM14_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_3DD4	MBC Memory Block Configuration Word (MBC1_DOM14_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_3DD8	MBC Memory Block Configuration Word (MBC1_DOM14_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_3DF0	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3E40	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3F40	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3F80	MBC Memory Block Configuration Word (MBC1_DOM15_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3FA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3FA8	MBC Memory Block Configuration Word (MBC1_DOM15_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_3FAC	MBC Memory Block Configuration Word (MBC1_DOM15_MEM2_BLK_CFG_W1)	32	RW	0000_0000
1_3FB0	MBC Memory Block Configuration Word (MBC1_DOM15_MEM2_BLK_CFG_W2)	32	RW	0000_0000
1_3FC8	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_3FD0	MBC Memory Block Configuration Word (MBC1_DOM15_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_3FD4	MBC Memory Block Configuration Word (MBC1_DOM15_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_3FD8	MBC Memory Block Configuration Word (MBC1_DOM15_MEM3_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_3FF0	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_4000	MBC Global Configuration Register (MBC2_MEM0_GLBCFG)	32	R	0010_0009
1_4004	MBC Global Configuration Register (MBC2_MEM1_GLBCFG)	32	R	0010_0009
1_4008	MBC Global Configuration Register (MBC2_MEM2_GLBCFG)	32	R	000C_0000
1_400C	MBC Global Configuration Register (MBC2_MEM3_GLBCFG)	32	R	000C_0000
1_4010	MBC NonSecure Enable Block Index (MBC2_NSE_BLK_INDEX)	32	RW	0000_0000
1_4014	MBC NonSecure Enable Block Set (MBC2_NSE_BLK_SET)	32	RW	0000_0000
1_4018	MBC NonSecure Enable Block Clear (MBC2_NSE_BLK_CLR)	32	RW	0000_0000
1_401C	MBC NonSecure Enable Block Clear All (MBC2_NSE_BLK_CLR_ALL)	32	RW	0000_0000
1_4020	MBC Global Access Control (MBC2_MEMN_GLBAC0)	32	RW	0000_0000
1_4024	MBC Global Access Control (MBC2_MEMN_GLBAC1)	32	RW	0000_0000
1_4028	MBC Global Access Control (MBC2_MEMN_GLBAC2)	32	RW	0000_0000
1_402C	MBC Global Access Control (MBC2_MEMN_GLBAC3)	32	RW	0000_0000
1_4030	MBC Global Access Control (MBC2_MEMN_GLBAC4)	32	RW	0000_0000
1_4034	MBC Global Access Control (MBC2_MEMN_GLBAC5)	32	RW	0000_0000
1_4038	MBC Global Access Control (MBC2_MEMN_GLBAC6)	32	RW	0000_0000
1_403C	MBC Global Access Control (MBC2_MEMN_GLBAC7)	32	RW	0000_0000
1_4040	MBC Memory Block Configuration Word (MBC2_DOM0_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_4044	MBC Memory Block Configuration Word (MBC2_DOM0_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_4140	MBC Memory Block NonSecure Enable Word (MBC2_DOM0_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_4180	MBC Memory Block Configuration Word (MBC2_DOM0_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_4184	MBC Memory Block Configuration Word (MBC2_DOM0_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_41A0	MBC Memory Block NonSecure Enable Word (MBC2_DOM0_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_4240	MBC Memory Block Configuration Word (MBC2_DOM1_MEM0_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4244	MBC Memory Block Configuration Word (MBC2_DOM1_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_4340	MBC Memory Block NonSecure Enable Word (MBC2_DOM1_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_4380	MBC Memory Block Configuration Word (MBC2_DOM1_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_4384	MBC Memory Block Configuration Word (MBC2_DOM1_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_43A0	MBC Memory Block NonSecure Enable Word (MBC2_DOM1_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_4440	MBC Memory Block Configuration Word (MBC2_DOM2_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_4444	MBC Memory Block Configuration Word (MBC2_DOM2_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_4540	MBC Memory Block NonSecure Enable Word (MBC2_DOM2_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_4580	MBC Memory Block Configuration Word (MBC2_DOM2_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_4584	MBC Memory Block Configuration Word (MBC2_DOM2_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_45A0	MBC Memory Block NonSecure Enable Word (MBC2_DOM2_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_4640	MBC Memory Block Configuration Word (MBC2_DOM3_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_4644	MBC Memory Block Configuration Word (MBC2_DOM3_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_4740	MBC Memory Block NonSecure Enable Word (MBC2_DOM3_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_4780	MBC Memory Block Configuration Word (MBC2_DOM3_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_4784	MBC Memory Block Configuration Word (MBC2_DOM3_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_47A0	MBC Memory Block NonSecure Enable Word (MBC2_DOM3_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_4840	MBC Memory Block Configuration Word (MBC2_DOM4_MEM0_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4844	MBC Memory Block Configuration Word (MBC2_DOM4_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_4940	MBC Memory Block NonSecure Enable Word (MBC2_DOM4_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_4980	MBC Memory Block Configuration Word (MBC2_DOM4_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_4984	MBC Memory Block Configuration Word (MBC2_DOM4_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_49A0	MBC Memory Block NonSecure Enable Word (MBC2_DOM4_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_4A40	MBC Memory Block Configuration Word (MBC2_DOM5_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_4A44	MBC Memory Block Configuration Word (MBC2_DOM5_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_4B40	MBC Memory Block NonSecure Enable Word (MBC2_DOM5_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_4B80	MBC Memory Block Configuration Word (MBC2_DOM5_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_4B84	MBC Memory Block Configuration Word (MBC2_DOM5_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_4BA0	MBC Memory Block NonSecure Enable Word (MBC2_DOM5_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_4C40	MBC Memory Block Configuration Word (MBC2_DOM6_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_4C44	MBC Memory Block Configuration Word (MBC2_DOM6_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_4D40	MBC Memory Block NonSecure Enable Word (MBC2_DOM6_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_4D80	MBC Memory Block Configuration Word (MBC2_DOM6_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_4D84	MBC Memory Block Configuration Word (MBC2_DOM6_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_4DA0	MBC Memory Block NonSecure Enable Word (MBC2_DOM6_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_4E40	MBC Memory Block Configuration Word (MBC2_DOM7_MEM0_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4E44	MBC Memory Block Configuration Word (MBC2_DOM7_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_4F40	MBC Memory Block NonSecure Enable Word (MBC2_DOM7_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_4F80	MBC Memory Block Configuration Word (MBC2_DOM7_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_4F84	MBC Memory Block Configuration Word (MBC2_DOM7_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_4FA0	MBC Memory Block NonSecure Enable Word (MBC2_DOM7_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_5040	MBC Memory Block Configuration Word (MBC2_DOM8_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_5044	MBC Memory Block Configuration Word (MBC2_DOM8_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_5140	MBC Memory Block NonSecure Enable Word (MBC2_DOM8_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_5180	MBC Memory Block Configuration Word (MBC2_DOM8_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_5184	MBC Memory Block Configuration Word (MBC2_DOM8_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_51A0	MBC Memory Block NonSecure Enable Word (MBC2_DOM8_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_5240	MBC Memory Block Configuration Word (MBC2_DOM9_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_5244	MBC Memory Block Configuration Word (MBC2_DOM9_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_5340	MBC Memory Block NonSecure Enable Word (MBC2_DOM9_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_5380	MBC Memory Block Configuration Word (MBC2_DOM9_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_5384	MBC Memory Block Configuration Word (MBC2_DOM9_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_53A0	MBC Memory Block NonSecure Enable Word (MBC2_DOM9_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_5440	MBC Memory Block Configuration Word (MBC2_DOM10_MEM0_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_5444	MBC Memory Block Configuration Word (MBC2_DOM10_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_5540	MBC Memory Block NonSecure Enable Word (MBC2_DOM10_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_5580	MBC Memory Block Configuration Word (MBC2_DOM10_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_5584	MBC Memory Block Configuration Word (MBC2_DOM10_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_55A0	MBC Memory Block NonSecure Enable Word (MBC2_DOM10_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_5640	MBC Memory Block Configuration Word (MBC2_DOM11_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_5644	MBC Memory Block Configuration Word (MBC2_DOM11_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_5740	MBC Memory Block NonSecure Enable Word (MBC2_DOM11_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_5780	MBC Memory Block Configuration Word (MBC2_DOM11_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_5784	MBC Memory Block Configuration Word (MBC2_DOM11_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_57A0	MBC Memory Block NonSecure Enable Word (MBC2_DOM11_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_5840	MBC Memory Block Configuration Word (MBC2_DOM12_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_5844	MBC Memory Block Configuration Word (MBC2_DOM12_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_5940	MBC Memory Block NonSecure Enable Word (MBC2_DOM12_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_5980	MBC Memory Block Configuration Word (MBC2_DOM12_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_5984	MBC Memory Block Configuration Word (MBC2_DOM12_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_59A0	MBC Memory Block NonSecure Enable Word (MBC2_DOM12_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_5A40	MBC Memory Block Configuration Word (MBC2_DOM13_MEM0_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_5A44	MBC Memory Block Configuration Word (MBC2_DOM13_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_5B40	MBC Memory Block NonSecure Enable Word (MBC2_DOM13_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_5B80	MBC Memory Block Configuration Word (MBC2_DOM13_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_5B84	MBC Memory Block Configuration Word (MBC2_DOM13_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_5BA0	MBC Memory Block NonSecure Enable Word (MBC2_DOM13_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_5C40	MBC Memory Block Configuration Word (MBC2_DOM14_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_5C44	MBC Memory Block Configuration Word (MBC2_DOM14_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_5D40	MBC Memory Block NonSecure Enable Word (MBC2_DOM14_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_5D80	MBC Memory Block Configuration Word (MBC2_DOM14_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_5D84	MBC Memory Block Configuration Word (MBC2_DOM14_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_5DA0	MBC Memory Block NonSecure Enable Word (MBC2_DOM14_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_5E40	MBC Memory Block Configuration Word (MBC2_DOM15_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_5E44	MBC Memory Block Configuration Word (MBC2_DOM15_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_5F40	MBC Memory Block NonSecure Enable Word (MBC2_DOM15_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_5F80	MBC Memory Block Configuration Word (MBC2_DOM15_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_5F84	MBC Memory Block Configuration Word (MBC2_DOM15_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_5FA0	MBC Memory Block NonSecure Enable Word (MBC2_DOM15_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_6000	MBC Global Configuration Register (MBC3_MEM0_GLBCFG)	32	R	000E_0028
1_6004	MBC Global Configuration Register (MBC3_MEM1_GLBCFG)	32	R	000E_0028
1_6008	MBC Global Configuration Register (MBC3_MEM2_GLBCFG)	32	R	000C_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_600C	MBC Global Configuration Register (MBC3_MEM3_GLBCFG)	32	R	000C_0000
1_6010	MBC NonSecure Enable Block Index (MBC3_NSE_BLK_INDEX)	32	RW	0000_0000
1_6014	MBC NonSecure Enable Block Set (MBC3_NSE_BLK_SET)	32	RW	0000_0000
1_6018	MBC NonSecure Enable Block Clear (MBC3_NSE_BLK_CLR)	32	RW	0000_0000
1_601C	MBC NonSecure Enable Block Clear All (MBC3_NSE_BLK_CLR_ALL)	32	RW	0000_0000
1_6020	MBC Global Access Control (MBC3_MEMN_GLBAC0)	32	RW	0000_0000
1_6024	MBC Global Access Control (MBC3_MEMN_GLBAC1)	32	RW	0000_0000
1_6028	MBC Global Access Control (MBC3_MEMN_GLBAC2)	32	RW	0000_0000
1_602C	MBC Global Access Control (MBC3_MEMN_GLBAC3)	32	RW	0000_0000
1_6030	MBC Global Access Control (MBC3_MEMN_GLBAC4)	32	RW	0000_0000
1_6034	MBC Global Access Control (MBC3_MEMN_GLBAC5)	32	RW	0000_0000
1_6038	MBC Global Access Control (MBC3_MEMN_GLBAC6)	32	RW	0000_0000
1_603C	MBC Global Access Control (MBC3_MEMN_GLBAC7)	32	RW	0000_0000
1_6040	MBC Memory Block Configuration Word (MBC3_DOM0_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_6044	MBC Memory Block Configuration Word (MBC3_DOM0_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_6048	MBC Memory Block Configuration Word (MBC3_DOM0_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_604C	MBC Memory Block Configuration Word (MBC3_DOM0_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_6050	MBC Memory Block Configuration Word (MBC3_DOM0_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_6140	MBC Memory Block NonSecure Enable Word (MBC3_DOM0_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_6144	MBC Memory Block NonSecure Enable Word (MBC3_DOM0_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_6180	MBC Memory Block Configuration Word (MBC3_DOM0_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_6184	MBC Memory Block Configuration Word (MBC3_DOM0_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_6188	MBC Memory Block Configuration Word (MBC3_DOM0_MEM1_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_618C	MBC Memory Block Configuration Word (MBC3_DOM0_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_6190	MBC Memory Block Configuration Word (MBC3_DOM0_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_61A0	MBC Memory Block NonSecure Enable Word (MBC3_DOM0_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_61A4	MBC Memory Block NonSecure Enable Word (MBC3_DOM0_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_6240	MBC Memory Block Configuration Word (MBC3_DOM1_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_6244	MBC Memory Block Configuration Word (MBC3_DOM1_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_6248	MBC Memory Block Configuration Word (MBC3_DOM1_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_624C	MBC Memory Block Configuration Word (MBC3_DOM1_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_6250	MBC Memory Block Configuration Word (MBC3_DOM1_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_6340	MBC Memory Block NonSecure Enable Word (MBC3_DOM1_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_6344	MBC Memory Block NonSecure Enable Word (MBC3_DOM1_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_6380	MBC Memory Block Configuration Word (MBC3_DOM1_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_6384	MBC Memory Block Configuration Word (MBC3_DOM1_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_6388	MBC Memory Block Configuration Word (MBC3_DOM1_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_638C	MBC Memory Block Configuration Word (MBC3_DOM1_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_6390	MBC Memory Block Configuration Word (MBC3_DOM1_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_63A0	MBC Memory Block NonSecure Enable Word (MBC3_DOM1_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_63A4	MBC Memory Block NonSecure Enable Word (MBC3_DOM1_MEM1_BLK_NSE_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_6440	MBC Memory Block Configuration Word (MBC3_DOM2_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_6444	MBC Memory Block Configuration Word (MBC3_DOM2_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_6448	MBC Memory Block Configuration Word (MBC3_DOM2_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_644C	MBC Memory Block Configuration Word (MBC3_DOM2_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_6450	MBC Memory Block Configuration Word (MBC3_DOM2_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_6540	MBC Memory Block NonSecure Enable Word (MBC3_DOM2_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_6544	MBC Memory Block NonSecure Enable Word (MBC3_DOM2_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_6580	MBC Memory Block Configuration Word (MBC3_DOM2_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_6584	MBC Memory Block Configuration Word (MBC3_DOM2_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_6588	MBC Memory Block Configuration Word (MBC3_DOM2_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_658C	MBC Memory Block Configuration Word (MBC3_DOM2_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_6590	MBC Memory Block Configuration Word (MBC3_DOM2_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_65A0	MBC Memory Block NonSecure Enable Word (MBC3_DOM2_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_65A4	MBC Memory Block NonSecure Enable Word (MBC3_DOM2_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_6640	MBC Memory Block Configuration Word (MBC3_DOM3_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_6644	MBC Memory Block Configuration Word (MBC3_DOM3_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_6648	MBC Memory Block Configuration Word (MBC3_DOM3_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_664C	MBC Memory Block Configuration Word (MBC3_DOM3_MEM0_BLK_CFG_W3)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_6650	MBC Memory Block Configuration Word (MBC3_DOM3_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_6740	MBC Memory Block NonSecure Enable Word (MBC3_DOM3_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_6744	MBC Memory Block NonSecure Enable Word (MBC3_DOM3_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_6780	MBC Memory Block Configuration Word (MBC3_DOM3_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_6784	MBC Memory Block Configuration Word (MBC3_DOM3_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_6788	MBC Memory Block Configuration Word (MBC3_DOM3_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_678C	MBC Memory Block Configuration Word (MBC3_DOM3_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_6790	MBC Memory Block Configuration Word (MBC3_DOM3_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_67A0	MBC Memory Block NonSecure Enable Word (MBC3_DOM3_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_67A4	MBC Memory Block NonSecure Enable Word (MBC3_DOM3_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_6840	MBC Memory Block Configuration Word (MBC3_DOM4_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_6844	MBC Memory Block Configuration Word (MBC3_DOM4_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_6848	MBC Memory Block Configuration Word (MBC3_DOM4_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_684C	MBC Memory Block Configuration Word (MBC3_DOM4_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_6850	MBC Memory Block Configuration Word (MBC3_DOM4_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_6940	MBC Memory Block NonSecure Enable Word (MBC3_DOM4_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_6944	MBC Memory Block NonSecure Enable Word (MBC3_DOM4_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_6980	MBC Memory Block Configuration Word (MBC3_DOM4_MEM1_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_6984	MBC Memory Block Configuration Word (MBC3_DOM4_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_6988	MBC Memory Block Configuration Word (MBC3_DOM4_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_698C	MBC Memory Block Configuration Word (MBC3_DOM4_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_6990	MBC Memory Block Configuration Word (MBC3_DOM4_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_69A0	MBC Memory Block NonSecure Enable Word (MBC3_DOM4_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_69A4	MBC Memory Block NonSecure Enable Word (MBC3_DOM4_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_6A40	MBC Memory Block Configuration Word (MBC3_DOM5_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_6A44	MBC Memory Block Configuration Word (MBC3_DOM5_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_6A48	MBC Memory Block Configuration Word (MBC3_DOM5_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_6A4C	MBC Memory Block Configuration Word (MBC3_DOM5_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_6A50	MBC Memory Block Configuration Word (MBC3_DOM5_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_6B40	MBC Memory Block NonSecure Enable Word (MBC3_DOM5_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_6B44	MBC Memory Block NonSecure Enable Word (MBC3_DOM5_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_6B80	MBC Memory Block Configuration Word (MBC3_DOM5_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_6B84	MBC Memory Block Configuration Word (MBC3_DOM5_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_6B88	MBC Memory Block Configuration Word (MBC3_DOM5_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_6B8C	MBC Memory Block Configuration Word (MBC3_DOM5_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_6B90	MBC Memory Block Configuration Word (MBC3_DOM5_MEM1_BLK_CFG_W4)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_6BA0	MBC Memory Block NonSecure Enable Word (MBC3_DOM5_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_6BA4	MBC Memory Block NonSecure Enable Word (MBC3_DOM5_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_6C40	MBC Memory Block Configuration Word (MBC3_DOM6_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_6C44	MBC Memory Block Configuration Word (MBC3_DOM6_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_6C48	MBC Memory Block Configuration Word (MBC3_DOM6_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_6C4C	MBC Memory Block Configuration Word (MBC3_DOM6_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_6C50	MBC Memory Block Configuration Word (MBC3_DOM6_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_6D40	MBC Memory Block NonSecure Enable Word (MBC3_DOM6_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_6D44	MBC Memory Block NonSecure Enable Word (MBC3_DOM6_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_6D80	MBC Memory Block Configuration Word (MBC3_DOM6_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_6D84	MBC Memory Block Configuration Word (MBC3_DOM6_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_6D88	MBC Memory Block Configuration Word (MBC3_DOM6_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_6D8C	MBC Memory Block Configuration Word (MBC3_DOM6_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_6D90	MBC Memory Block Configuration Word (MBC3_DOM6_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_6DA0	MBC Memory Block NonSecure Enable Word (MBC3_DOM6_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_6DA4	MBC Memory Block NonSecure Enable Word (MBC3_DOM6_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_6E40	MBC Memory Block Configuration Word (MBC3_DOM7_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_6E44	MBC Memory Block Configuration Word (MBC3_DOM7_MEM0_BLK_CFG_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_6E48	MBC Memory Block Configuration Word (MBC3_DOM7_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_6E4C	MBC Memory Block Configuration Word (MBC3_DOM7_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_6E50	MBC Memory Block Configuration Word (MBC3_DOM7_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_6F40	MBC Memory Block NonSecure Enable Word (MBC3_DOM7_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_6F44	MBC Memory Block NonSecure Enable Word (MBC3_DOM7_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_6F80	MBC Memory Block Configuration Word (MBC3_DOM7_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_6F84	MBC Memory Block Configuration Word (MBC3_DOM7_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_6F88	MBC Memory Block Configuration Word (MBC3_DOM7_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_6F8C	MBC Memory Block Configuration Word (MBC3_DOM7_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_6F90	MBC Memory Block Configuration Word (MBC3_DOM7_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_6FA0	MBC Memory Block NonSecure Enable Word (MBC3_DOM7_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_6FA4	MBC Memory Block NonSecure Enable Word (MBC3_DOM7_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_7040	MBC Memory Block Configuration Word (MBC3_DOM8_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_7044	MBC Memory Block Configuration Word (MBC3_DOM8_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_7048	MBC Memory Block Configuration Word (MBC3_DOM8_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_704C	MBC Memory Block Configuration Word (MBC3_DOM8_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_7050	MBC Memory Block Configuration Word (MBC3_DOM8_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_7140	MBC Memory Block NonSecure Enable Word (MBC3_DOM8_MEM0_BLK_NSE_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_7144	MBC Memory Block NonSecure Enable Word (MBC3_DOM8_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_7180	MBC Memory Block Configuration Word (MBC3_DOM8_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_7184	MBC Memory Block Configuration Word (MBC3_DOM8_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_7188	MBC Memory Block Configuration Word (MBC3_DOM8_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_718C	MBC Memory Block Configuration Word (MBC3_DOM8_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_7190	MBC Memory Block Configuration Word (MBC3_DOM8_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_71A0	MBC Memory Block NonSecure Enable Word (MBC3_DOM8_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_71A4	MBC Memory Block NonSecure Enable Word (MBC3_DOM8_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_7240	MBC Memory Block Configuration Word (MBC3_DOM9_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_7244	MBC Memory Block Configuration Word (MBC3_DOM9_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_7248	MBC Memory Block Configuration Word (MBC3_DOM9_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_724C	MBC Memory Block Configuration Word (MBC3_DOM9_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_7250	MBC Memory Block Configuration Word (MBC3_DOM9_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_7340	MBC Memory Block NonSecure Enable Word (MBC3_DOM9_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_7344	MBC Memory Block NonSecure Enable Word (MBC3_DOM9_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_7380	MBC Memory Block Configuration Word (MBC3_DOM9_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_7384	MBC Memory Block Configuration Word (MBC3_DOM9_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_7388	MBC Memory Block Configuration Word (MBC3_DOM9_MEM1_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_738C	MBC Memory Block Configuration Word (MBC3_DOM9_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_7390	MBC Memory Block Configuration Word (MBC3_DOM9_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_73A0	MBC Memory Block NonSecure Enable Word (MBC3_DOM9_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_73A4	MBC Memory Block NonSecure Enable Word (MBC3_DOM9_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_7440	MBC Memory Block Configuration Word (MBC3_DOM10_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_7444	MBC Memory Block Configuration Word (MBC3_DOM10_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_7448	MBC Memory Block Configuration Word (MBC3_DOM10_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_744C	MBC Memory Block Configuration Word (MBC3_DOM10_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_7450	MBC Memory Block Configuration Word (MBC3_DOM10_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_7540	MBC Memory Block NonSecure Enable Word (MBC3_DOM10_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_7544	MBC Memory Block NonSecure Enable Word (MBC3_DOM10_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_7580	MBC Memory Block Configuration Word (MBC3_DOM10_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_7584	MBC Memory Block Configuration Word (MBC3_DOM10_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_7588	MBC Memory Block Configuration Word (MBC3_DOM10_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_758C	MBC Memory Block Configuration Word (MBC3_DOM10_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_7590	MBC Memory Block Configuration Word (MBC3_DOM10_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_75A0	MBC Memory Block NonSecure Enable Word (MBC3_DOM10_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_75A4	MBC Memory Block NonSecure Enable Word (MBC3_DOM10_MEM1_BLK_NSE_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_7640	MBC Memory Block Configuration Word (MBC3_DOM11_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_7644	MBC Memory Block Configuration Word (MBC3_DOM11_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_7648	MBC Memory Block Configuration Word (MBC3_DOM11_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_764C	MBC Memory Block Configuration Word (MBC3_DOM11_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_7650	MBC Memory Block Configuration Word (MBC3_DOM11_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_7740	MBC Memory Block NonSecure Enable Word (MBC3_DOM11_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_7744	MBC Memory Block NonSecure Enable Word (MBC3_DOM11_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_7780	MBC Memory Block Configuration Word (MBC3_DOM11_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_7784	MBC Memory Block Configuration Word (MBC3_DOM11_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_7788	MBC Memory Block Configuration Word (MBC3_DOM11_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_778C	MBC Memory Block Configuration Word (MBC3_DOM11_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_7790	MBC Memory Block Configuration Word (MBC3_DOM11_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_77A0	MBC Memory Block NonSecure Enable Word (MBC3_DOM11_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_77A4	MBC Memory Block NonSecure Enable Word (MBC3_DOM11_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_7840	MBC Memory Block Configuration Word (MBC3_DOM12_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_7844	MBC Memory Block Configuration Word (MBC3_DOM12_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_7848	MBC Memory Block Configuration Word (MBC3_DOM12_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_784C	MBC Memory Block Configuration Word (MBC3_DOM12_MEM0_BLK_CFG_W3)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_7850	MBC Memory Block Configuration Word (MBC3_DOM12_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_7940	MBC Memory Block NonSecure Enable Word (MBC3_DOM12_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_7944	MBC Memory Block NonSecure Enable Word (MBC3_DOM12_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_7980	MBC Memory Block Configuration Word (MBC3_DOM12_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_7984	MBC Memory Block Configuration Word (MBC3_DOM12_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_7988	MBC Memory Block Configuration Word (MBC3_DOM12_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_798C	MBC Memory Block Configuration Word (MBC3_DOM12_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_7990	MBC Memory Block Configuration Word (MBC3_DOM12_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_79A0	MBC Memory Block NonSecure Enable Word (MBC3_DOM12_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_79A4	MBC Memory Block NonSecure Enable Word (MBC3_DOM12_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_7A40	MBC Memory Block Configuration Word (MBC3_DOM13_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_7A44	MBC Memory Block Configuration Word (MBC3_DOM13_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_7A48	MBC Memory Block Configuration Word (MBC3_DOM13_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_7A4C	MBC Memory Block Configuration Word (MBC3_DOM13_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_7A50	MBC Memory Block Configuration Word (MBC3_DOM13_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_7B40	MBC Memory Block NonSecure Enable Word (MBC3_DOM13_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_7B44	MBC Memory Block NonSecure Enable Word (MBC3_DOM13_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_7B80	MBC Memory Block Configuration Word (MBC3_DOM13_MEM1_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_7B84	MBC Memory Block Configuration Word (MBC3_DOM13_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_7B88	MBC Memory Block Configuration Word (MBC3_DOM13_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_7B8C	MBC Memory Block Configuration Word (MBC3_DOM13_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_7B90	MBC Memory Block Configuration Word (MBC3_DOM13_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_7BA0	MBC Memory Block NonSecure Enable Word (MBC3_DOM13_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_7BA4	MBC Memory Block NonSecure Enable Word (MBC3_DOM13_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_7C40	MBC Memory Block Configuration Word (MBC3_DOM14_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_7C44	MBC Memory Block Configuration Word (MBC3_DOM14_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_7C48	MBC Memory Block Configuration Word (MBC3_DOM14_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_7C4C	MBC Memory Block Configuration Word (MBC3_DOM14_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_7C50	MBC Memory Block Configuration Word (MBC3_DOM14_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_7D40	MBC Memory Block NonSecure Enable Word (MBC3_DOM14_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_7D44	MBC Memory Block NonSecure Enable Word (MBC3_DOM14_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_7D80	MBC Memory Block Configuration Word (MBC3_DOM14_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_7D84	MBC Memory Block Configuration Word (MBC3_DOM14_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_7D88	MBC Memory Block Configuration Word (MBC3_DOM14_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_7D8C	MBC Memory Block Configuration Word (MBC3_DOM14_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_7D90	MBC Memory Block Configuration Word (MBC3_DOM14_MEM1_BLK_CFG_W4)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_7DA0	MBC Memory Block NonSecure Enable Word (MBC3_DOM14_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_7DA4	MBC Memory Block NonSecure Enable Word (MBC3_DOM14_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_7E40	MBC Memory Block Configuration Word (MBC3_DOM15_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_7E44	MBC Memory Block Configuration Word (MBC3_DOM15_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_7E48	MBC Memory Block Configuration Word (MBC3_DOM15_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_7E4C	MBC Memory Block Configuration Word (MBC3_DOM15_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_7E50	MBC Memory Block Configuration Word (MBC3_DOM15_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_7F40	MBC Memory Block NonSecure Enable Word (MBC3_DOM15_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_7F44	MBC Memory Block NonSecure Enable Word (MBC3_DOM15_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_7F80	MBC Memory Block Configuration Word (MBC3_DOM15_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_7F84	MBC Memory Block Configuration Word (MBC3_DOM15_MEM1_BLK_CFG_W1)	32	RW	0000_0000
1_7F88	MBC Memory Block Configuration Word (MBC3_DOM15_MEM1_BLK_CFG_W2)	32	RW	0000_0000
1_7F8C	MBC Memory Block Configuration Word (MBC3_DOM15_MEM1_BLK_CFG_W3)	32	RW	0000_0000
1_7F90	MBC Memory Block Configuration Word (MBC3_DOM15_MEM1_BLK_CFG_W4)	32	RW	0000_0000
1_7FA0	MBC Memory Block NonSecure Enable Word (MBC3_DOM15_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_7FA4	MBC Memory Block NonSecure Enable Word (MBC3_DOM15_MEM1_BLK_NSE_W1)	32	RW	0000_0000
1_8000	MRC Global Configuration Register (MRC0_GLBCFG)	32	R	0000_0010
1_8010	MRC NonSecure Enable Region Indirect (MRC0_NSE_RGN_INDIRECT)	32	RW	0000_0000
1_8014	MRC NonSecure Enable Region Set (MRC0_NSE_RGN_SET)	32	RW	0000_0000
1_8018	MRC NonSecure Enable Region Clear (MRC0_NSE_RGN_CLR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_801C	MRC NonSecure Enable Region Clear All (MRC0_NSE_RGN_CLR_ALL)	32	RW	0000_0000
1_8020	MRC Global Access Control (MRC0_GLBAC0)	32	RW	0000_0000
1_8024	MRC Global Access Control (MRC0_GLBAC1)	32	RW	0000_0000
1_8028	MRC Global Access Control (MRC0_GLBAC2)	32	RW	0000_0000
1_802C	MRC Global Access Control (MRC0_GLBAC3)	32	RW	0000_0000
1_8030	MRC Global Access Control (MRC0_GLBAC4)	32	RW	0000_0000
1_8034	MRC Global Access Control (MRC0_GLBAC5)	32	RW	0000_0000
1_8038	MRC Global Access Control (MRC0_GLBAC6)	32	RW	0000_0000
1_803C	MRC Global Access Control (MRC0_GLBAC7)	32	RW	0000_0000
1_8040	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD0_W0)	32	RW	0000_0000
1_8044	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD0_W1)	32	RW	0000_0000
1_8048	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD1_W0)	32	RW	0000_0000
1_804C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD1_W1)	32	RW	0000_0000
1_8050	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD2_W0)	32	RW	0000_0000
1_8054	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD2_W1)	32	RW	0000_0000
1_8058	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD3_W0)	32	RW	0000_0000
1_805C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD3_W1)	32	RW	0000_0000
1_8060	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD4_W0)	32	RW	0000_0000
1_8064	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD4_W1)	32	RW	0000_0000
1_8068	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD5_W0)	32	RW	0000_0000
1_806C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD5_W1)	32	RW	0000_0000
1_8070	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD6_W0)	32	RW	0000_0000
1_8074	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD6_W1)	32	RW	0000_0000
1_8078	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD7_W0)	32	RW	0000_0000
1_807C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD7_W1)	32	RW	0000_0000
1_8080	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD8_W0)	32	RW	0000_0000
1_8084	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD8_W1)	32	RW	0000_0000
1_8088	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD9_W0)	32	RW	0000_0000
1_808C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD9_W1)	32	RW	0000_0000
1_8090	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD10_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_8094	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD10_W1)	32	RW	0000_0000
1_8098	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD11_W0)	32	RW	0000_0000
1_809C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD11_W1)	32	RW	0000_0000
1_80A0	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD12_W0)	32	RW	0000_0000
1_80A4	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD12_W1)	32	RW	0000_0000
1_80A8	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD13_W0)	32	RW	0000_0000
1_80AC	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD13_W1)	32	RW	0000_0000
1_80B0	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD14_W0)	32	RW	0000_0000
1_80B4	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD14_W1)	32	RW	0000_0000
1_80B8	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD15_W0)	32	RW	0000_0000
1_80BC	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD15_W1)	32	RW	0000_0000
1_80C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM0_RGD_NSE)	32	RW	0000_0000
1_8140	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD0_W0)	32	RW	0000_0000
1_8144	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD0_W1)	32	RW	0000_0000
1_8148	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD1_W0)	32	RW	0000_0000
1_814C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD1_W1)	32	RW	0000_0000
1_8150	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD2_W0)	32	RW	0000_0000
1_8154	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD2_W1)	32	RW	0000_0000
1_8158	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD3_W0)	32	RW	0000_0000
1_815C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD3_W1)	32	RW	0000_0000
1_8160	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD4_W0)	32	RW	0000_0000
1_8164	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD4_W1)	32	RW	0000_0000
1_8168	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD5_W0)	32	RW	0000_0000
1_816C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD5_W1)	32	RW	0000_0000
1_8170	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD6_W0)	32	RW	0000_0000
1_8174	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD6_W1)	32	RW	0000_0000
1_8178	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD7_W0)	32	RW	0000_0000
1_817C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD7_W1)	32	RW	0000_0000
1_8180	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD8_W0)	32	RW	0000_0000
1_8184	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD8_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_8188	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD9_W0)	32	RW	0000_0000
1_818C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD9_W1)	32	RW	0000_0000
1_8190	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD10_W0)	32	RW	0000_0000
1_8194	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD10_W1)	32	RW	0000_0000
1_8198	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD11_W0)	32	RW	0000_0000
1_819C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD11_W1)	32	RW	0000_0000
1_81A0	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD12_W0)	32	RW	0000_0000
1_81A4	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD12_W1)	32	RW	0000_0000
1_81A8	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD13_W0)	32	RW	0000_0000
1_81AC	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD13_W1)	32	RW	0000_0000
1_81B0	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD14_W0)	32	RW	0000_0000
1_81B4	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD14_W1)	32	RW	0000_0000
1_81B8	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD15_W0)	32	RW	0000_0000
1_81BC	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD15_W1)	32	RW	0000_0000
1_81C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM1_RGD_NSE)	32	RW	0000_0000
1_8240	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD0_W0)	32	RW	0000_0000
1_8244	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD0_W1)	32	RW	0000_0000
1_8248	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD1_W0)	32	RW	0000_0000
1_824C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD1_W1)	32	RW	0000_0000
1_8250	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD2_W0)	32	RW	0000_0000
1_8254	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD2_W1)	32	RW	0000_0000
1_8258	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD3_W0)	32	RW	0000_0000
1_825C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD3_W1)	32	RW	0000_0000
1_8260	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD4_W0)	32	RW	0000_0000
1_8264	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD4_W1)	32	RW	0000_0000
1_8268	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD5_W0)	32	RW	0000_0000
1_826C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD5_W1)	32	RW	0000_0000
1_8270	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD6_W0)	32	RW	0000_0000
1_8274	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD6_W1)	32	RW	0000_0000
1_8278	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD7_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_827C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD7_W1)	32	RW	0000_0000
1_8280	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD8_W0)	32	RW	0000_0000
1_8284	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD8_W1)	32	RW	0000_0000
1_8288	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD9_W0)	32	RW	0000_0000
1_828C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD9_W1)	32	RW	0000_0000
1_8290	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD10_W0)	32	RW	0000_0000
1_8294	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD10_W1)	32	RW	0000_0000
1_8298	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD11_W0)	32	RW	0000_0000
1_829C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD11_W1)	32	RW	0000_0000
1_82A0	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD12_W0)	32	RW	0000_0000
1_82A4	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD12_W1)	32	RW	0000_0000
1_82A8	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD13_W0)	32	RW	0000_0000
1_82AC	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD13_W1)	32	RW	0000_0000
1_82B0	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD14_W0)	32	RW	0000_0000
1_82B4	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD14_W1)	32	RW	0000_0000
1_82B8	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD15_W0)	32	RW	0000_0000
1_82BC	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD15_W1)	32	RW	0000_0000
1_82C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM2_RGD_NSE)	32	RW	0000_0000
1_8340	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD0_W0)	32	RW	0000_0000
1_8344	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD0_W1)	32	RW	0000_0000
1_8348	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD1_W0)	32	RW	0000_0000
1_834C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD1_W1)	32	RW	0000_0000
1_8350	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD2_W0)	32	RW	0000_0000
1_8354	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD2_W1)	32	RW	0000_0000
1_8358	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD3_W0)	32	RW	0000_0000
1_835C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD3_W1)	32	RW	0000_0000
1_8360	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD4_W0)	32	RW	0000_0000
1_8364	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD4_W1)	32	RW	0000_0000
1_8368	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD5_W0)	32	RW	0000_0000
1_836C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD5_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_8370	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD6_W0)	32	RW	0000_0000
1_8374	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD6_W1)	32	RW	0000_0000
1_8378	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD7_W0)	32	RW	0000_0000
1_837C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD7_W1)	32	RW	0000_0000
1_8380	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD8_W0)	32	RW	0000_0000
1_8384	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD8_W1)	32	RW	0000_0000
1_8388	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD9_W0)	32	RW	0000_0000
1_838C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD9_W1)	32	RW	0000_0000
1_8390	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD10_W0)	32	RW	0000_0000
1_8394	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD10_W1)	32	RW	0000_0000
1_8398	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD11_W0)	32	RW	0000_0000
1_839C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD11_W1)	32	RW	0000_0000
1_83A0	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD12_W0)	32	RW	0000_0000
1_83A4	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD12_W1)	32	RW	0000_0000
1_83A8	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD13_W0)	32	RW	0000_0000
1_83AC	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD13_W1)	32	RW	0000_0000
1_83B0	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD14_W0)	32	RW	0000_0000
1_83B4	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD14_W1)	32	RW	0000_0000
1_83B8	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD15_W0)	32	RW	0000_0000
1_83BC	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD15_W1)	32	RW	0000_0000
1_83C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM3_RGD_NSE)	32	RW	0000_0000
1_8440	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD0_W0)	32	RW	0000_0000
1_8444	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD0_W1)	32	RW	0000_0000
1_8448	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD1_W0)	32	RW	0000_0000
1_844C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD1_W1)	32	RW	0000_0000
1_8450	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD2_W0)	32	RW	0000_0000
1_8454	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD2_W1)	32	RW	0000_0000
1_8458	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD3_W0)	32	RW	0000_0000
1_845C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD3_W1)	32	RW	0000_0000
1_8460	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD4_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_8464	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD4_W1)	32	RW	0000_0000
1_8468	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD5_W0)	32	RW	0000_0000
1_846C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD5_W1)	32	RW	0000_0000
1_8470	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD6_W0)	32	RW	0000_0000
1_8474	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD6_W1)	32	RW	0000_0000
1_8478	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD7_W0)	32	RW	0000_0000
1_847C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD7_W1)	32	RW	0000_0000
1_8480	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD8_W0)	32	RW	0000_0000
1_8484	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD8_W1)	32	RW	0000_0000
1_8488	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD9_W0)	32	RW	0000_0000
1_848C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD9_W1)	32	RW	0000_0000
1_8490	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD10_W0)	32	RW	0000_0000
1_8494	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD10_W1)	32	RW	0000_0000
1_8498	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD11_W0)	32	RW	0000_0000
1_849C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD11_W1)	32	RW	0000_0000
1_84A0	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD12_W0)	32	RW	0000_0000
1_84A4	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD12_W1)	32	RW	0000_0000
1_84A8	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD13_W0)	32	RW	0000_0000
1_84AC	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD13_W1)	32	RW	0000_0000
1_84B0	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD14_W0)	32	RW	0000_0000
1_84B4	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD14_W1)	32	RW	0000_0000
1_84B8	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD15_W0)	32	RW	0000_0000
1_84BC	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD15_W1)	32	RW	0000_0000
1_84C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM4_RGD_NSE)	32	RW	0000_0000
1_8540	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD0_W0)	32	RW	0000_0000
1_8544	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD0_W1)	32	RW	0000_0000
1_8548	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD1_W0)	32	RW	0000_0000
1_854C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD1_W1)	32	RW	0000_0000
1_8550	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD2_W0)	32	RW	0000_0000
1_8554	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD2_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_8558	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD3_W0)	32	RW	0000_0000
1_855C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD3_W1)	32	RW	0000_0000
1_8560	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD4_W0)	32	RW	0000_0000
1_8564	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD4_W1)	32	RW	0000_0000
1_8568	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD5_W0)	32	RW	0000_0000
1_856C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD5_W1)	32	RW	0000_0000
1_8570	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD6_W0)	32	RW	0000_0000
1_8574	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD6_W1)	32	RW	0000_0000
1_8578	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD7_W0)	32	RW	0000_0000
1_857C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD7_W1)	32	RW	0000_0000
1_8580	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD8_W0)	32	RW	0000_0000
1_8584	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD8_W1)	32	RW	0000_0000
1_8588	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD9_W0)	32	RW	0000_0000
1_858C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD9_W1)	32	RW	0000_0000
1_8590	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD10_W0)	32	RW	0000_0000
1_8594	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD10_W1)	32	RW	0000_0000
1_8598	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD11_W0)	32	RW	0000_0000
1_859C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD11_W1)	32	RW	0000_0000
1_85A0	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD12_W0)	32	RW	0000_0000
1_85A4	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD12_W1)	32	RW	0000_0000
1_85A8	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD13_W0)	32	RW	0000_0000
1_85AC	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD13_W1)	32	RW	0000_0000
1_85B0	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD14_W0)	32	RW	0000_0000
1_85B4	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD14_W1)	32	RW	0000_0000
1_85B8	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD15_W0)	32	RW	0000_0000
1_85BC	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD15_W1)	32	RW	0000_0000
1_85C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM5_RGD_NSE)	32	RW	0000_0000
1_8640	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD0_W0)	32	RW	0000_0000
1_8644	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD0_W1)	32	RW	0000_0000
1_8648	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD1_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_864C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD1_W1)	32	RW	0000_0000
1_8650	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD2_W0)	32	RW	0000_0000
1_8654	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD2_W1)	32	RW	0000_0000
1_8658	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD3_W0)	32	RW	0000_0000
1_865C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD3_W1)	32	RW	0000_0000
1_8660	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD4_W0)	32	RW	0000_0000
1_8664	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD4_W1)	32	RW	0000_0000
1_8668	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD5_W0)	32	RW	0000_0000
1_866C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD5_W1)	32	RW	0000_0000
1_8670	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD6_W0)	32	RW	0000_0000
1_8674	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD6_W1)	32	RW	0000_0000
1_8678	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD7_W0)	32	RW	0000_0000
1_867C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD7_W1)	32	RW	0000_0000
1_8680	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD8_W0)	32	RW	0000_0000
1_8684	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD8_W1)	32	RW	0000_0000
1_8688	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD9_W0)	32	RW	0000_0000
1_868C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD9_W1)	32	RW	0000_0000
1_8690	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD10_W0)	32	RW	0000_0000
1_8694	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD10_W1)	32	RW	0000_0000
1_8698	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD11_W0)	32	RW	0000_0000
1_869C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD11_W1)	32	RW	0000_0000
1_86A0	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD12_W0)	32	RW	0000_0000
1_86A4	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD12_W1)	32	RW	0000_0000
1_86A8	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD13_W0)	32	RW	0000_0000
1_86AC	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD13_W1)	32	RW	0000_0000
1_86B0	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD14_W0)	32	RW	0000_0000
1_86B4	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD14_W1)	32	RW	0000_0000
1_86B8	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD15_W0)	32	RW	0000_0000
1_86BC	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD15_W1)	32	RW	0000_0000
1_86C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM6_RGD_NSE)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_8740	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD0_W0)	32	RW	0000_0000
1_8744	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD0_W1)	32	RW	0000_0000
1_8748	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD1_W0)	32	RW	0000_0000
1_874C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD1_W1)	32	RW	0000_0000
1_8750	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD2_W0)	32	RW	0000_0000
1_8754	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD2_W1)	32	RW	0000_0000
1_8758	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD3_W0)	32	RW	0000_0000
1_875C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD3_W1)	32	RW	0000_0000
1_8760	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD4_W0)	32	RW	0000_0000
1_8764	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD4_W1)	32	RW	0000_0000
1_8768	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD5_W0)	32	RW	0000_0000
1_876C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD5_W1)	32	RW	0000_0000
1_8770	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD6_W0)	32	RW	0000_0000
1_8774	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD6_W1)	32	RW	0000_0000
1_8778	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD7_W0)	32	RW	0000_0000
1_877C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD7_W1)	32	RW	0000_0000
1_8780	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD8_W0)	32	RW	0000_0000
1_8784	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD8_W1)	32	RW	0000_0000
1_8788	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD9_W0)	32	RW	0000_0000
1_878C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD9_W1)	32	RW	0000_0000
1_8790	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD10_W0)	32	RW	0000_0000
1_8794	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD10_W1)	32	RW	0000_0000
1_8798	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD11_W0)	32	RW	0000_0000
1_879C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD11_W1)	32	RW	0000_0000
1_87A0	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD12_W0)	32	RW	0000_0000
1_87A4	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD12_W1)	32	RW	0000_0000
1_87A8	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD13_W0)	32	RW	0000_0000
1_87AC	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD13_W1)	32	RW	0000_0000
1_87B0	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD14_W0)	32	RW	0000_0000
1_87B4	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD14_W1)	32	RW	0000_0000
1_87B8	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD15_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_87BC	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD15_W1)	32	RW	0000_0000
1_87C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM7_RGD_NSE)	32	RW	0000_0000
1_8840	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD0_W0)	32	RW	0000_0000
1_8844	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD0_W1)	32	RW	0000_0000
1_8848	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD1_W0)	32	RW	0000_0000
1_884C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD1_W1)	32	RW	0000_0000
1_8850	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD2_W0)	32	RW	0000_0000
1_8854	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD2_W1)	32	RW	0000_0000
1_8858	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD3_W0)	32	RW	0000_0000
1_885C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD3_W1)	32	RW	0000_0000
1_8860	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD4_W0)	32	RW	0000_0000
1_8864	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD4_W1)	32	RW	0000_0000
1_8868	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD5_W0)	32	RW	0000_0000
1_886C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD5_W1)	32	RW	0000_0000
1_8870	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD6_W0)	32	RW	0000_0000
1_8874	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD6_W1)	32	RW	0000_0000
1_8878	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD7_W0)	32	RW	0000_0000
1_887C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD7_W1)	32	RW	0000_0000
1_8880	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD8_W0)	32	RW	0000_0000
1_8884	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD8_W1)	32	RW	0000_0000
1_8888	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD9_W0)	32	RW	0000_0000
1_888C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD9_W1)	32	RW	0000_0000
1_8890	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD10_W0)	32	RW	0000_0000
1_8894	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD10_W1)	32	RW	0000_0000
1_8898	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD11_W0)	32	RW	0000_0000
1_889C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD11_W1)	32	RW	0000_0000
1_88A0	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD12_W0)	32	RW	0000_0000
1_88A4	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD12_W1)	32	RW	0000_0000
1_88A8	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD13_W0)	32	RW	0000_0000
1_88AC	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD13_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_88B0	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD14_W0)	32	RW	0000_0000
1_88B4	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD14_W1)	32	RW	0000_0000
1_88B8	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD15_W0)	32	RW	0000_0000
1_88BC	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD15_W1)	32	RW	0000_0000
1_88C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM8_RGD_NSE)	32	RW	0000_0000
1_8940	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD0_W0)	32	RW	0000_0000
1_8944	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD0_W1)	32	RW	0000_0000
1_8948	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD1_W0)	32	RW	0000_0000
1_894C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD1_W1)	32	RW	0000_0000
1_8950	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD2_W0)	32	RW	0000_0000
1_8954	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD2_W1)	32	RW	0000_0000
1_8958	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD3_W0)	32	RW	0000_0000
1_895C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD3_W1)	32	RW	0000_0000
1_8960	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD4_W0)	32	RW	0000_0000
1_8964	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD4_W1)	32	RW	0000_0000
1_8968	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD5_W0)	32	RW	0000_0000
1_896C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD5_W1)	32	RW	0000_0000
1_8970	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD6_W0)	32	RW	0000_0000
1_8974	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD6_W1)	32	RW	0000_0000
1_8978	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD7_W0)	32	RW	0000_0000
1_897C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD7_W1)	32	RW	0000_0000
1_8980	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD8_W0)	32	RW	0000_0000
1_8984	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD8_W1)	32	RW	0000_0000
1_8988	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD9_W0)	32	RW	0000_0000
1_898C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD9_W1)	32	RW	0000_0000
1_8990	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD10_W0)	32	RW	0000_0000
1_8994	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD10_W1)	32	RW	0000_0000
1_8998	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD11_W0)	32	RW	0000_0000
1_899C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD11_W1)	32	RW	0000_0000
1_89A0	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD12_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_89A4	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD12_W1)	32	RW	0000_0000
1_89A8	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD13_W0)	32	RW	0000_0000
1_89AC	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD13_W1)	32	RW	0000_0000
1_89B0	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD14_W0)	32	RW	0000_0000
1_89B4	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD14_W1)	32	RW	0000_0000
1_89B8	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD15_W0)	32	RW	0000_0000
1_89BC	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD15_W1)	32	RW	0000_0000
1_89C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM9_RGD_NSE)	32	RW	0000_0000
1_8A40	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD0_W0)	32	RW	0000_0000
1_8A44	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD0_W1)	32	RW	0000_0000
1_8A48	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD1_W0)	32	RW	0000_0000
1_8A4C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD1_W1)	32	RW	0000_0000
1_8A50	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD2_W0)	32	RW	0000_0000
1_8A54	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD2_W1)	32	RW	0000_0000
1_8A58	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD3_W0)	32	RW	0000_0000
1_8A5C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD3_W1)	32	RW	0000_0000
1_8A60	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD4_W0)	32	RW	0000_0000
1_8A64	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD4_W1)	32	RW	0000_0000
1_8A68	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD5_W0)	32	RW	0000_0000
1_8A6C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD5_W1)	32	RW	0000_0000
1_8A70	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD6_W0)	32	RW	0000_0000
1_8A74	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD6_W1)	32	RW	0000_0000
1_8A78	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD7_W0)	32	RW	0000_0000
1_8A7C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD7_W1)	32	RW	0000_0000
1_8A80	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD8_W0)	32	RW	0000_0000
1_8A84	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD8_W1)	32	RW	0000_0000
1_8A88	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD9_W0)	32	RW	0000_0000
1_8A8C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD9_W1)	32	RW	0000_0000
1_8A90	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD10_W0)	32	RW	0000_0000
1_8A94	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD10_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_8A98	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD11_W0)	32	RW	0000_0000
1_8A9C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD11_W1)	32	RW	0000_0000
1_8AA0	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD12_W0)	32	RW	0000_0000
1_8AA4	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD12_W1)	32	RW	0000_0000
1_8AA8	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD13_W0)	32	RW	0000_0000
1_8AAC	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD13_W1)	32	RW	0000_0000
1_8AB0	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD14_W0)	32	RW	0000_0000
1_8AB4	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD14_W1)	32	RW	0000_0000
1_8AB8	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD15_W0)	32	RW	0000_0000
1_8ABC	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD15_W1)	32	RW	0000_0000
1_8AC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM10_RGD_NSE)	32	RW	0000_0000
1_8B40	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD0_W0)	32	RW	0000_0000
1_8B44	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD0_W1)	32	RW	0000_0000
1_8B48	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD1_W0)	32	RW	0000_0000
1_8B4C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD1_W1)	32	RW	0000_0000
1_8B50	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD2_W0)	32	RW	0000_0000
1_8B54	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD2_W1)	32	RW	0000_0000
1_8B58	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD3_W0)	32	RW	0000_0000
1_8B5C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD3_W1)	32	RW	0000_0000
1_8B60	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD4_W0)	32	RW	0000_0000
1_8B64	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD4_W1)	32	RW	0000_0000
1_8B68	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD5_W0)	32	RW	0000_0000
1_8B6C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD5_W1)	32	RW	0000_0000
1_8B70	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD6_W0)	32	RW	0000_0000
1_8B74	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD6_W1)	32	RW	0000_0000
1_8B78	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD7_W0)	32	RW	0000_0000
1_8B7C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD7_W1)	32	RW	0000_0000
1_8B80	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD8_W0)	32	RW	0000_0000
1_8B84	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD8_W1)	32	RW	0000_0000
1_8B88	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD9_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_8B8C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD9_W1)	32	RW	0000_0000
1_8B90	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD10_W0)	32	RW	0000_0000
1_8B94	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD10_W1)	32	RW	0000_0000
1_8B98	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD11_W0)	32	RW	0000_0000
1_8B9C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD11_W1)	32	RW	0000_0000
1_8BA0	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD12_W0)	32	RW	0000_0000
1_8BA4	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD12_W1)	32	RW	0000_0000
1_8BA8	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD13_W0)	32	RW	0000_0000
1_8BAC	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD13_W1)	32	RW	0000_0000
1_8BB0	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD14_W0)	32	RW	0000_0000
1_8BB4	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD14_W1)	32	RW	0000_0000
1_8BB8	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD15_W0)	32	RW	0000_0000
1_8BBC	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD15_W1)	32	RW	0000_0000
1_8BC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM11_RGD_NSE)	32	RW	0000_0000
1_8C40	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD0_W0)	32	RW	0000_0000
1_8C44	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD0_W1)	32	RW	0000_0000
1_8C48	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD1_W0)	32	RW	0000_0000
1_8C4C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD1_W1)	32	RW	0000_0000
1_8C50	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD2_W0)	32	RW	0000_0000
1_8C54	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD2_W1)	32	RW	0000_0000
1_8C58	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD3_W0)	32	RW	0000_0000
1_8C5C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD3_W1)	32	RW	0000_0000
1_8C60	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD4_W0)	32	RW	0000_0000
1_8C64	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD4_W1)	32	RW	0000_0000
1_8C68	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD5_W0)	32	RW	0000_0000
1_8C6C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD5_W1)	32	RW	0000_0000
1_8C70	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD6_W0)	32	RW	0000_0000
1_8C74	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD6_W1)	32	RW	0000_0000
1_8C78	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD7_W0)	32	RW	0000_0000
1_8C7C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD7_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_8C80	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD8_W0)	32	RW	0000_0000
1_8C84	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD8_W1)	32	RW	0000_0000
1_8C88	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD9_W0)	32	RW	0000_0000
1_8C8C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD9_W1)	32	RW	0000_0000
1_8C90	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD10_W0)	32	RW	0000_0000
1_8C94	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD10_W1)	32	RW	0000_0000
1_8C98	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD11_W0)	32	RW	0000_0000
1_8C9C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD11_W1)	32	RW	0000_0000
1_8CA0	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD12_W0)	32	RW	0000_0000
1_8CA4	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD12_W1)	32	RW	0000_0000
1_8CA8	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD13_W0)	32	RW	0000_0000
1_8CAC	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD13_W1)	32	RW	0000_0000
1_8CB0	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD14_W0)	32	RW	0000_0000
1_8CB4	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD14_W1)	32	RW	0000_0000
1_8CB8	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD15_W0)	32	RW	0000_0000
1_8CBC	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD15_W1)	32	RW	0000_0000
1_8CC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM12_RGD_NSE)	32	RW	0000_0000
1_8D40	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD0_W0)	32	RW	0000_0000
1_8D44	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD0_W1)	32	RW	0000_0000
1_8D48	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD1_W0)	32	RW	0000_0000
1_8D4C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD1_W1)	32	RW	0000_0000
1_8D50	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD2_W0)	32	RW	0000_0000
1_8D54	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD2_W1)	32	RW	0000_0000
1_8D58	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD3_W0)	32	RW	0000_0000
1_8D5C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD3_W1)	32	RW	0000_0000
1_8D60	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD4_W0)	32	RW	0000_0000
1_8D64	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD4_W1)	32	RW	0000_0000
1_8D68	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD5_W0)	32	RW	0000_0000
1_8D6C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD5_W1)	32	RW	0000_0000
1_8D70	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD6_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_8D74	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD6_W1)	32	RW	0000_0000
1_8D78	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD7_W0)	32	RW	0000_0000
1_8D7C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD7_W1)	32	RW	0000_0000
1_8D80	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD8_W0)	32	RW	0000_0000
1_8D84	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD8_W1)	32	RW	0000_0000
1_8D88	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD9_W0)	32	RW	0000_0000
1_8D8C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD9_W1)	32	RW	0000_0000
1_8D90	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD10_W0)	32	RW	0000_0000
1_8D94	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD10_W1)	32	RW	0000_0000
1_8D98	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD11_W0)	32	RW	0000_0000
1_8D9C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD11_W1)	32	RW	0000_0000
1_8DA0	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD12_W0)	32	RW	0000_0000
1_8DA4	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD12_W1)	32	RW	0000_0000
1_8DA8	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD13_W0)	32	RW	0000_0000
1_8DAC	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD13_W1)	32	RW	0000_0000
1_8DB0	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD14_W0)	32	RW	0000_0000
1_8DB4	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD14_W1)	32	RW	0000_0000
1_8DB8	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD15_W0)	32	RW	0000_0000
1_8DBC	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD15_W1)	32	RW	0000_0000
1_8DC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM13_RGD_NSE)	32	RW	0000_0000
1_8E40	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD0_W0)	32	RW	0000_0000
1_8E44	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD0_W1)	32	RW	0000_0000
1_8E48	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD1_W0)	32	RW	0000_0000
1_8E4C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD1_W1)	32	RW	0000_0000
1_8E50	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD2_W0)	32	RW	0000_0000
1_8E54	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD2_W1)	32	RW	0000_0000
1_8E58	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD3_W0)	32	RW	0000_0000
1_8E5C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD3_W1)	32	RW	0000_0000
1_8E60	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD4_W0)	32	RW	0000_0000
1_8E64	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD4_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_8E68	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD5_W0)	32	RW	0000_0000
1_8E6C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD5_W1)	32	RW	0000_0000
1_8E70	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD6_W0)	32	RW	0000_0000
1_8E74	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD6_W1)	32	RW	0000_0000
1_8E78	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD7_W0)	32	RW	0000_0000
1_8E7C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD7_W1)	32	RW	0000_0000
1_8E80	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD8_W0)	32	RW	0000_0000
1_8E84	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD8_W1)	32	RW	0000_0000
1_8E88	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD9_W0)	32	RW	0000_0000
1_8E8C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD9_W1)	32	RW	0000_0000
1_8E90	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD10_W0)	32	RW	0000_0000
1_8E94	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD10_W1)	32	RW	0000_0000
1_8E98	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD11_W0)	32	RW	0000_0000
1_8E9C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD11_W1)	32	RW	0000_0000
1_8EA0	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD12_W0)	32	RW	0000_0000
1_8EA4	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD12_W1)	32	RW	0000_0000
1_8EA8	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD13_W0)	32	RW	0000_0000
1_8EAC	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD13_W1)	32	RW	0000_0000
1_8EB0	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD14_W0)	32	RW	0000_0000
1_8EB4	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD14_W1)	32	RW	0000_0000
1_8EB8	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD15_W0)	32	RW	0000_0000
1_8EBC	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD15_W1)	32	RW	0000_0000
1_8EC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM14_RGD_NSE)	32	RW	0000_0000
1_8F40	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD0_W0)	32	RW	0000_0000
1_8F44	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD0_W1)	32	RW	0000_0000
1_8F48	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD1_W0)	32	RW	0000_0000
1_8F4C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD1_W1)	32	RW	0000_0000
1_8F50	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD2_W0)	32	RW	0000_0000
1_8F54	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD2_W1)	32	RW	0000_0000
1_8F58	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD3_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_8F5C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD3_W1)	32	RW	0000_0000
1_8F60	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD4_W0)	32	RW	0000_0000
1_8F64	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD4_W1)	32	RW	0000_0000
1_8F68	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD5_W0)	32	RW	0000_0000
1_8F6C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD5_W1)	32	RW	0000_0000
1_8F70	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD6_W0)	32	RW	0000_0000
1_8F74	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD6_W1)	32	RW	0000_0000
1_8F78	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD7_W0)	32	RW	0000_0000
1_8F7C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD7_W1)	32	RW	0000_0000
1_8F80	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD8_W0)	32	RW	0000_0000
1_8F84	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD8_W1)	32	RW	0000_0000
1_8F88	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD9_W0)	32	RW	0000_0000
1_8F8C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD9_W1)	32	RW	0000_0000
1_8F90	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD10_W0)	32	RW	0000_0000
1_8F94	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD10_W1)	32	RW	0000_0000
1_8F98	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD11_W0)	32	RW	0000_0000
1_8F9C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD11_W1)	32	RW	0000_0000
1_8FA0	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD12_W0)	32	RW	0000_0000
1_8FA4	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD12_W1)	32	RW	0000_0000
1_8FA8	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD13_W0)	32	RW	0000_0000
1_8FAC	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD13_W1)	32	RW	0000_0000
1_8FB0	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD14_W0)	32	RW	0000_0000
1_8FB4	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD14_W1)	32	RW	0000_0000
1_8FB8	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD15_W0)	32	RW	0000_0000
1_8FBC	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD15_W1)	32	RW	0000_0000
1_8FC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM15_RGD_NSE)	32	RW	0000_0000

22.7.5.2 TRDC Register (TRDC_CR)

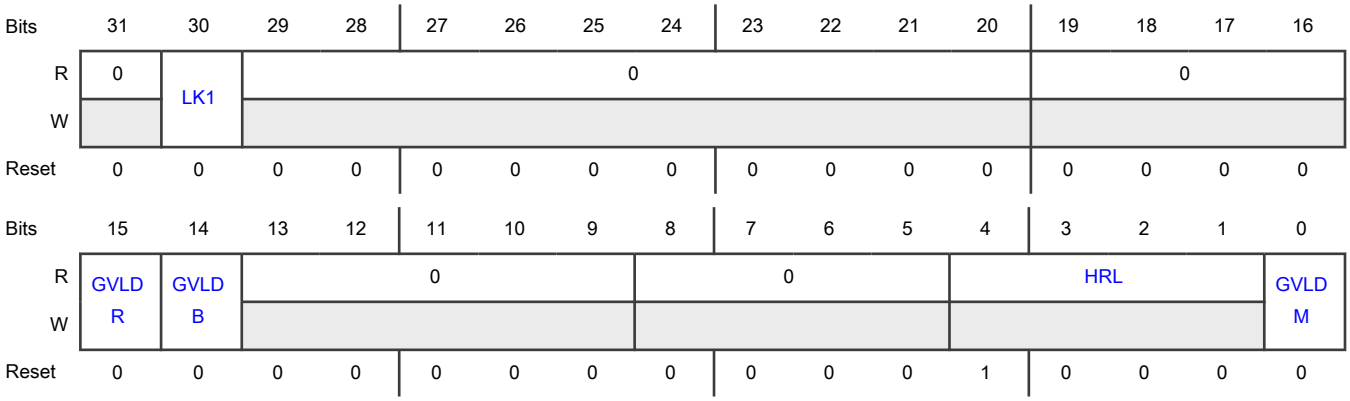
This register provides status about the TRDC and global enable bits for the entire module's operation. A fully operational TRDC requires all global bits GVLD {R,B,M} to be asserted. Undefined behavior results if they are not all asserted. If there are no region checkers (MRCs) present in the configuration, GVLDR need not be asserted for a fully operational TRDC.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_CR	0h

Diagram



Fields

Field	Description
31 —	Reserved
30 LK1	Lock Status This field is the lock status of the TRDC_CR. This field is set when all GVLD bits are set; GVLDR=GVLDB=GVLDM=1b1. Once set, this bit remains asserted until the next reset. 0 - The CR can be written by any secure privileged write. 1 - The CR is locked (read-only) until the next reset.
29-20 —	Reserved
19-16 —	Reserved
15 GVLDR	Global Valid for Memory Region Checkers TRDC global MRC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC MRCs are disabled. 1 - TRDC MRCs are enabled.
14 GVLDB	Global Valid for Memory Block Checkers TRDC global MBC enable/disable. Once set, this bit remains set until the next reset.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - TRDC MBCs are disabled. 1 - TRDC MBCs are enabled.
13-9 —	Reserved
8-5 —	Reserved
4-1 HRL	Hardware Revision Level This read-only field specifies the TRDC's hardware and definition revision level. It can be read by software to determine the functional definition of the module.
0 GVLDM	Global Valid for Domain Assignment Controllers TRDC global DAC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC DACs are disabled. 1 - TRDC DACs are enabled.

22.7.5.3 TRDC Hardware Configuration Register 0 (TRDC_HWCFG0)

This read-only register contains information on the TRDC's hardware configuration. Specifically, it defines the number of implemented domains and bus initiators along with the number of instances of memory block checkers (MBCs) and memory region checkers (MRCs). The register value at reset is device-specific. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG0	F0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MID				NMRC				0				NMBC			
W																
Reset	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NMSTR								0				NDID			
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0

Fields

Field	Description
31-29 MID	Module ID This field defines major version ID of this module.
28-24 NMRC	Number of MRCs This field defines the number of MRCs on the device [1-16].
23-20 —	Reserved
19-16 NMBC	Number of MBCs This field defines the number of MBCs on the device [1-8].
15-8 NMSTR	Number of bus initiators This read-only field defines the number of bus initiators.
7-5 —	Reserved
4-0 NDID	Number of domains This read-only field defines the number of domains on the device [1-16].

22.7.5.4 TRDC Hardware Configuration Register 1 (TRDC_HWCFG1)

This register contains information on the TRDC's hardware configuration. It provides a mechanism for software to determine its domain number by simply reading the register. See Domain error capture management for more details on typical usage. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG1	F4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												DID			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	u	u	u	u

Fields

Field	Description
31-4 —	Reserved
3-0 DID	Domain identifier number This field provides the domain number [0-15] of the requesting bus initiator.

22.7.5.5 TRDC Hardware Configuration Register 2 (TRDC_HWCFG2)

This register contains information on the TRDC's hardware configuration. It provides a bitmap signaling the presence of a process identifier (PID) register sourced from the given bus initiator. The HWCFG2 register is associated with bus initiators [31-0]. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG2	F8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PIDPn															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIDPn															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Fields

Field	Description
31-0 PIDPn	Process identifier present Process identifier present from bus initiator n, where n = [31-0]. This field provides a bitmap to signal that bus initiator (where n is defined as bit number) sources a process identifier register to the TRDC_DAC logic. For each bit in this field: • 0 - Bus initiator n does not source a process identifier register. The TRDC_DAC logic provides the needed PID for processor cores. • 1 - Bus initiator n sources a process identifier register.

22.7.5.6 TRDC Hardware Configuration Register 3 (TRDC_HWCFG3)

This register contains information on the TRDC's hardware configuration. Specifically, it provides a bitmap signaling the presence of a process identifier (PID) register sourced from the given bus initiator. The HWCFG3 register is associated with bus initiators [63-32], while the HWCFG2 register covers bus initiators [31-0]. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG3	FCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PIDPn															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIDPn															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PIDPn	Process identifier present Process identifier present from bus initiator n, where n = [63-32]. This field provides a bitmap to signal that bus initiator n (where n is defined as 32 + bit number) sources a process identifier register to the TRDC_DAC logic.

Table continues on the next page...

Field	Description
	Process identifier present from bus initiator n, where n = [31-0]. This field provides a bitmap to signal that bus initiator (where n is defined as bit number) sources a process identifier register to the TRDC_DAC logic. For each bit in this field: • 0 - Bus initiator n does not source a process identifier register. The TRDC_DAC logic provides the needed PID for processor cores. • 1 - Bus initiator n sources a process identifier register.

22.7.5.7 Domain Assignment Configuration Register (DACFG0 - DACFG3)

This register defines the number of implemented domain assignment registers for bus initiator m, where m+1 can specify from 1 to 64 bus initiators. These registers are organized as a byte-sized data array and can be read using 8-, 16- or 32-bit accesses. An all-zero value (NCM = 0, NMDAR = 0) indicates a non-existent bus initiator. Attempted writes are error terminated.

Register read will not return transfer error when DACFG for a initiator doesn't exist but it is in the same 32-bit group as DACFG for an existing initiator. For example, when there are only 2 DACFG0-1 registers for 2 initiator, and initiators for DACFG2-3 don't exist, then access to DACFG2-3 won't return transfer error.

Typically, processor bus initiators have one or more domain assignment registers, while non-processor initiators have a single domain assignment register.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
DACFG0	100h
DACFG1	101h
DACFG2	102h
DACFG3	103h

Diagram

Bits	7	6	5	4	3	2	1	0
R	NCM	0			NMDAR			
W								
Reset	See Register reset values.							

Register reset values

Register	Reset value
DACFG0–DACFG1	04h
DACFG2–DACFG3	81h

Fields

Field	Description
7 NCM	Non-CPU Initiator This read-only field signals that bus initiator m is a non-CPU initiator. It specifies that the format of the associated MDA_Wr_m register defines a non-processor domain assignment. This field is zero for a non-existent bus initiator. 0 - Bus initiator is a processor. 1 - Bus initiator is a non-processor.
6-4 —	Reserved
3-0 NMDAR	Number of initiator domain assignment registers for bus initiator m This read-only field specifies the number of registers associated with the initiator domain assignment register for a given bus initiator. The value is limited to the range [0-8], where zero indicates a non-existent bus initiator and non-zero values indicate the number of implemented registers associated with this MDAm.

22.7.5.8 TRDC IDAU Control Register (TRDC_IDAU_CR)

This register defines the configuration for the Implementation-Defined Attribution Unit which is tightly-coupled to the TZ-M extensions in the processor core. Many of the fields in this register explicitly control processor TZ-M functions.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_IDAU_CR	1C0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PCURRNS	0	0	0	0	0	0						0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			LKSAU	LKNSMPU	LKSMPU	LKNSVTOR	LKSVTAI...	0	SAUDIS	MPUNSDIS	MPUSDIS	CFGSECE...	0		VLD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Fields

Field	Description
31 PCURRNS	Processor current security Current security state of the processor. 0 - Processor is in Secure state 1 - Processor is in Nonsecure state
30 —	Reserved
29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25-19 —	Reserved
18-16 —	Reserved
15-13 —	Reserved
12 LKSAU	Lock SAU Asserting this bit prevents changes to the processor's Secure SAU memory regions already programmed. All processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the SAU_CTRL, SAU_RNR, SAU_RBAR and SAU_RLAR registers from software or from a debug agent connected to the processor
11 LKNSMPU	Lock Nonsecure MPU Asserting this bit prevents changes to the processor's Nonsecure MPU memory regions already programmed. All processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Disable writes to the MPU_CTRL_NS, MPU_RNR_NS, MPU_RBAR_NS, MPU_RLAR_NS, MPU_RBAR_A_NSn and MPU_RLAR_A_NSn from software or from a debug agent connected to the processor
10 LKSMPU	Lock Secure MPU Asserting this signal prevents changes to the processor's programmed Secure MPU memory regions and all processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the MPU_CTRL, MPU_RNR, MPU_RBAR, MPU_RLAR, MPU_RBAR_An and MPU_RLAR_An from software or from a debug agent connected to the processor in Secure state
9 LKNSVTOR	Lock Nonsecure Vector Table Offset Register Asserting this signal prevents changes to the processor's Nonsecure vector table base address. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock this register 1 - Disable writes to the VTOR_NS register
8 LKSVTAIRCR	Lock Secure VTOR, Application interrupt and Reset Control Registers This bit is sticky, once set, it can only be cleared with a reset. Asserting this signal prevents processor changes to: • The Secure vector table base address. • Handling of Secure interrupt priority. • BusFault, HardFault, and NMI security target settings in the processor. 0 - Unlock these registers 1 - Disable writes to the VTOR_S, AIRCR[PRIS], and AIRCR[BFHFNMINS] registers
7 —	Reserved
6 SAUDIS	Security Attribution Unit Disable 0 - SAU is enabled 1 - SAU is disabled
5 MPUNSDIS	NonSecure Memory Protection Unit Disabled 0 - Nonsecure MPU is enabled 1 - Nonsecure MPU is disabled
4 MPUSDIS	Secure Memory Protection Unit Disabled 0 - Secure MPU is enabled 1 - Secure MPU is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
3 CFGSECEXT	Configure Security Extension 0 - Armv8M Security Extension is disabled 1 - Armv8-M Security Extension is enabled
2-1 —	Reserved
0 VLD	Valid When VLD =0, all address attribute from IDAU is nonSecure. When VLD=1, values in other fields of this register are valid.

22.7.5.9 TRDC FLW Control (TRDC_FLW_CTL)

This register provides control of the FLW = Flash Logical Window operation.

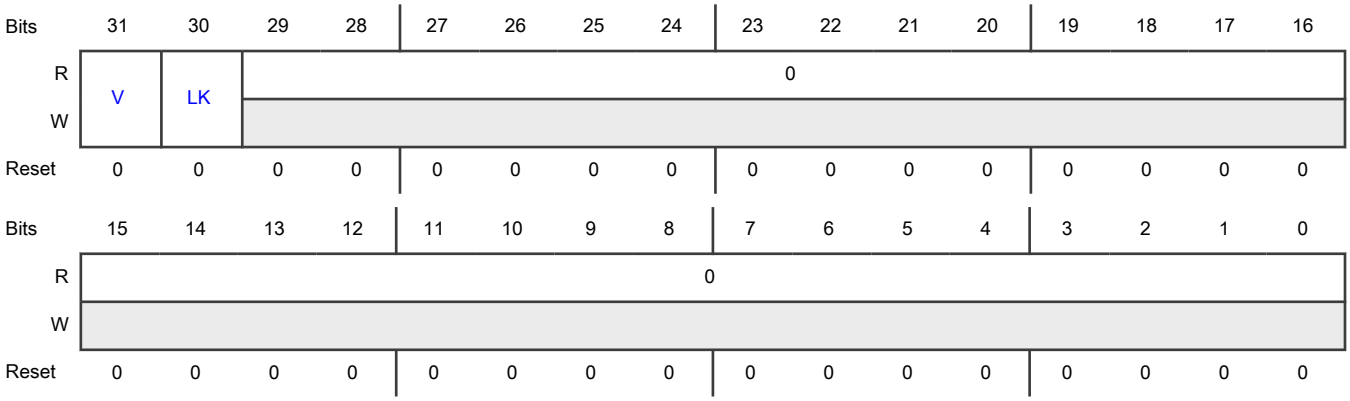
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_FLW_CTL	1E0h

Diagram



Fields

Field	Description
31 V	Valid bit 0 - FLW function is disabled.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - FLW function is enabled.
30 LK	Lock bit 0 - FLW registers may be modified. 1 - FLW registers are locked until the next reset.
29-0 —	Reserved

22.7.5.10 TRDC FLW Physical Base (TRDC_FLW_PBASE)

This read-only register gives the physical base address of the Flash Logical Window.

This register exists for this device but has not effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_FLW_PBASE	1E4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PBASE															
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PBASE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PBASE	Physical base address Physical address of the base of the FLW (mod 32KBytes).

22.7.5.11 TRDC FLW Array Base (TRDC_FLW_ABASE)

This register gives the flash array base address of the Flash Logical Window.

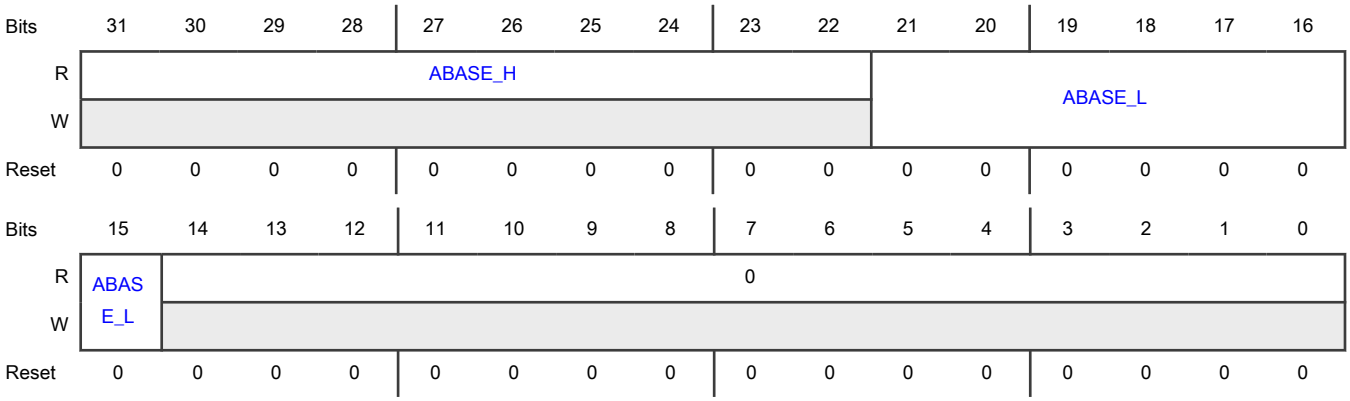
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write. This register is not writable if TRDC_FLW_CTL[LK]=1.

Offset

Register	Offset
TRDC_FLW_ABASE	1E8h

Diagram



Fields

Field	Description
31-22 ABASE_H	Array base address high Flash array address upper bits of the base of the FLW (mod 32KBytes).
21-15 ABASE_L	Array base address low Flash array address lower bits of the base of the FLW (mod 32KBytes).
14-0 —	Reserved

22.7.5.12 TRDC FLW Block Count (TRDC_FLW_BCNT)

This register gives the size of the Flash Logic Window in 32KByte blocks.

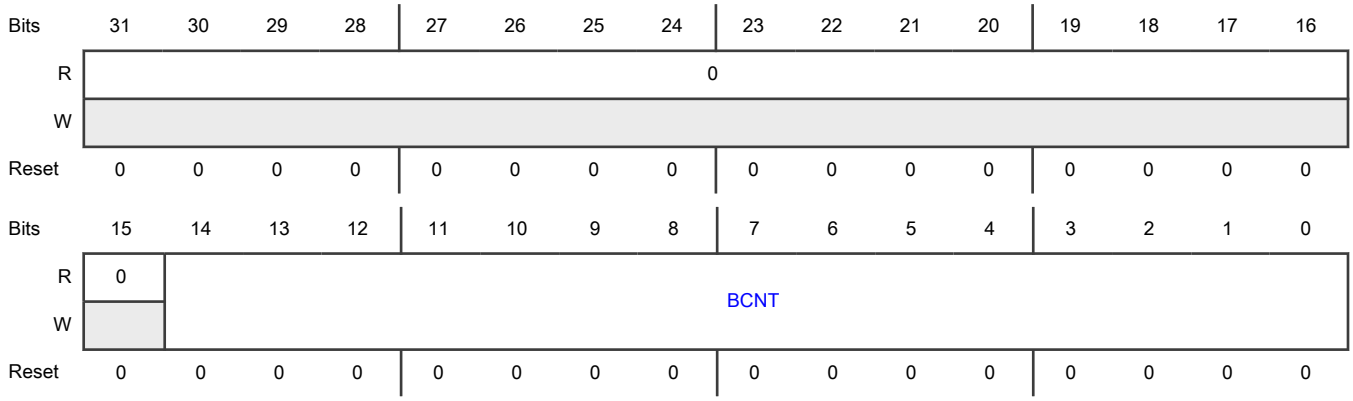
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write. This register is not writable if TRDC_FLW_CTL[LK]=1.

Offset

Register	Offset
TRDC_FLW_BCNT	1ECh

Diagram



Fields

Field	Description
31-15 —	Reserved
14-0 BCNT	Block Count Size of FLW in number of 32KByte blocks.

22.7.5.13 TRDC Fault Domain ID (TRDC_FDID)

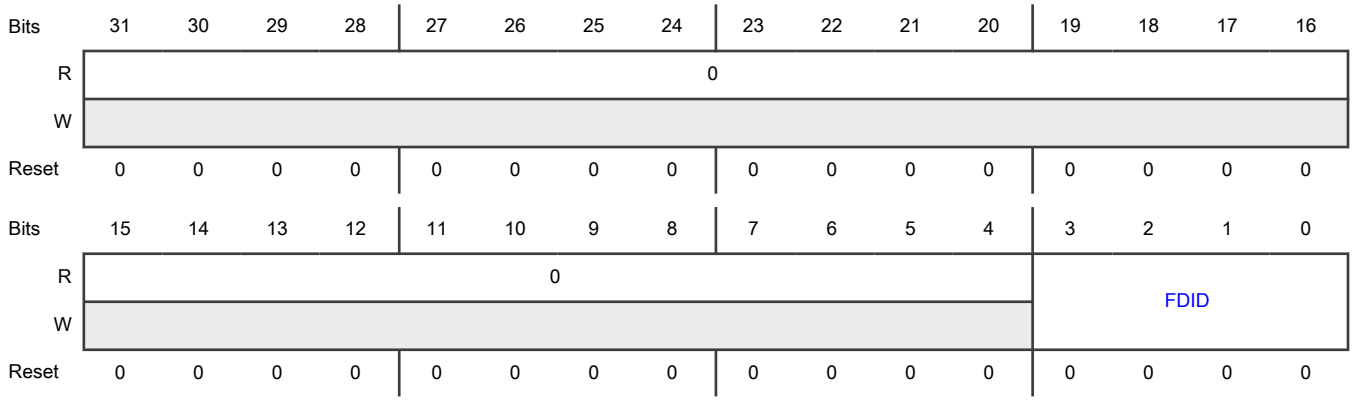
In the event of an access error, this register is used to specify the domainID of the faulting reference before indexing into the Domain Error registers.

If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_FDID	1FCh

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 FDID	Domain ID of Faulted Access This field indicates the domainID of the fault used to index the array of domain error information. The user queries the DERRLOCx registers to find the DomainID of the faulting access and must write this register with the domain ID before reading the Domain Error registers.

22.7.5.14 TRDC Domain Error Location Register (TRDC_DERRLOC0 - TRDC_DERRLOC15)

This array of read-only registers provide the instance number of the submodule where (an) access violation(s) occurred. These registers are organized as a word array, indexed by the faulting domain number, d. The two fields of this register provide a bitmap of instances associated with all submodules containing captured error information for that domain. These instance numbers are then used as indices into the DERR_W0_*i*, DERR_W1_*i*, and DERR_W3_*i*register arrays. See Domain error capture management for more details.

When an access violation is detected by either a Memory Region Checker (MRC) or a Memory Block Checker (MBC), address and attribute information of the offending access is captured. Using the faulting domainID number as the index, d, this array of read-only registers provide additional information signaling the instance number of the submodule where the access violation(s) occurred. Since the resulting exception handler needs the submodule instance to retrieve the captured address and attribute information from DERR_W0_*i* and DERR_W1_*i*, these registers provide the instance number details.

Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

For d = 0 to 15:

Register	Offset
TRDC_DERRLOCd	200h + (d × 4h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MRCINST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								MBCINST							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 MRCINST	MRC instance This field is a bitmap indicating the presence of a detected access violation for domain d in the implemented instances of the MRC. The least-significant bit of this field (bit 16) corresponds to MRC instance 0. The most-significant bit of this field (bit 31) corresponds to MRC instance 15 (MRC instance = i-16 where i is the register bit number), and so on. Multiple bits can be set at any time indicating access violations for this domain have been detected across multiple instances of the MRCs. For each bit in this field: • 0 - The memory region checker has not detected an access violation or is not physically present. • 1 - The memory region checker has detected one or more access violations for this domain.
15-8 —	Reserved
7-0 MBCINST	MBC instance This field is a bitmap indicating the presence of a detected access violation for domain d in the implemented instances of the MBC. The least-significant bit of this field (bit 0) corresponds to MBC instance 0. The most-significant bit of this field (bit 7) corresponds to MBC instance 7 Multiple bits can be set at any time indicating access violations for this domain have been detected across multiple instances of the MBCs. For each bit in this field: • 0 - The memory block checker has not detected an access violation or is not physically present. • 1 - The memory block checker has detected one or more access violations for this domain.

22.7.5.15 MBC Domain Error Word0 Register (MBC0_DERR_W0 - MBC3_DERR_W0)

This read-only register array provides the address of an access violation detected by memory block checker (MBC). These registers are organized as a word array, which is indexed by the MBC instance number. That is, the index, i, of this array is the instance number of MBC with the access violation. The submodule instance numbers are provided by the DERRLOC registers. The memory mapped error capture detail registers are organized as 24 sequential 16 byte entries.

When an access violation is detected and the offending information captured, subsequent updates to this register are disabled until the required data pattern is written to the DERR_W3_i register. At that time, this register is cleared and re-enabled to capture the next access violation.

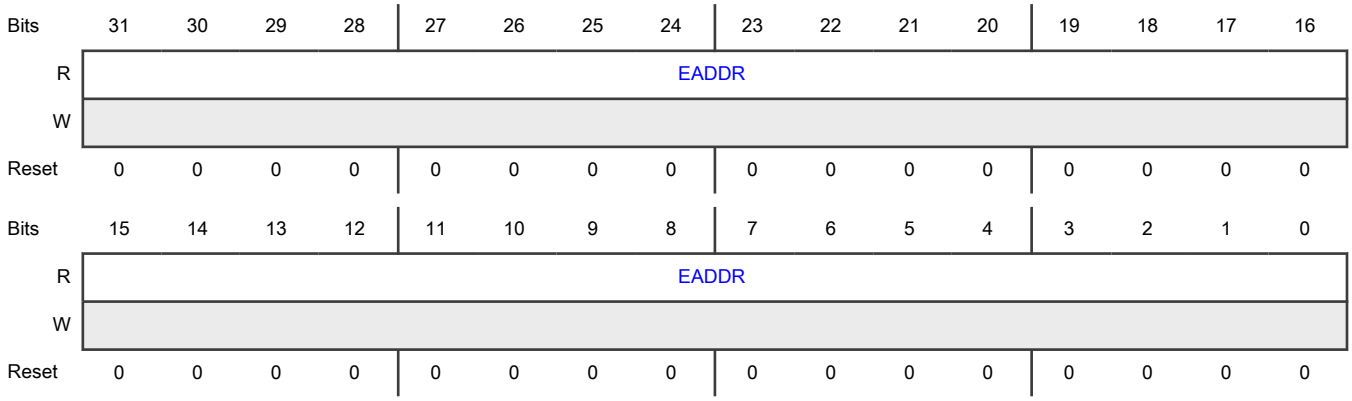
Attempted writes are error terminated as are attempted reads of an MBC instance that is not physically present.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
MBC0_DERR_W0	400h
MBC1_DERR_W0	410h
MBC2_DERR_W0	420h
MBC3_DERR_W0	430h

Diagram



Fields

Field	Description
31-0 EADDR	Error address This is the unaliased virtual address of the access that generated the access violation.

22.7.5.16 MBC Domain Error Word1 Register (MBC0_DERR_W1 - MBC3_DERR_W1)

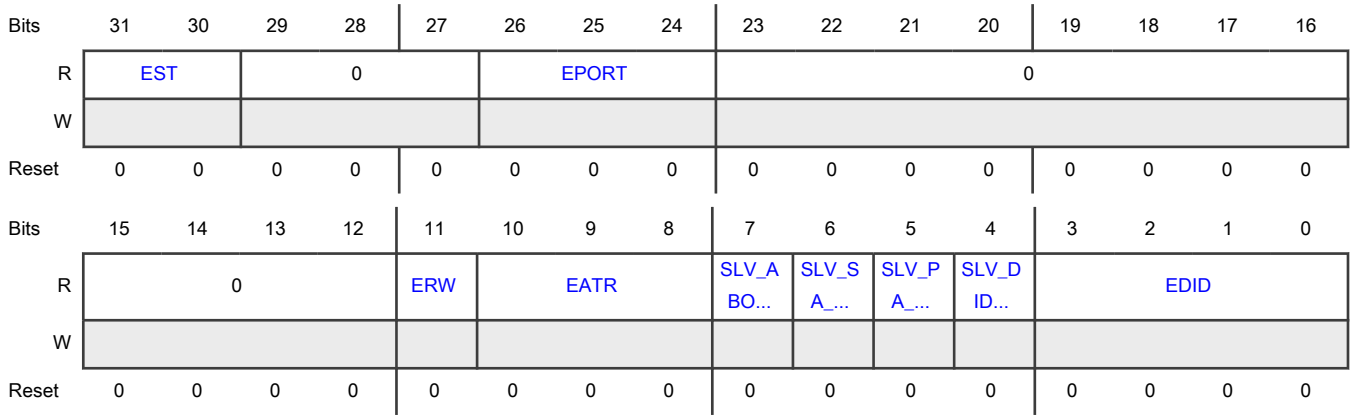
This read-only register array provides the attributes of an access violation detected by Memory Block Checker (MBC). These registers are organized as a word array, which is indexed by the violating submodule instance number. Refer to register Domain Error Location(DERRLOCd), Domain Error Word0 Register(DERR_W0_i), and Domain error capture management for more information.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
MBC0_DERR_W1	404h
MBC1_DERR_W1	414h
MBC2_DERR_W1	424h
MBC3_DERR_W1	434h

Diagram



Fields

Field	Description
31-30 EST	Error state This field signals the state of access violations for this domain in this instance of the block checker or region checker. Once an access violation has been detected and the faulting address and attribute information stored, subsequent errors are simply recorded as an overrun condition without any data captured. After retrieving the faulting address and attributes, the error capture mechanism must be rearmed by performing a write to DERR_W3_i. 00 - No access violation has been detected. 01 - No access violation has been detected. 10 - A single access violation has been detected. 11 - Multiple access violations for this domain have been detected by this submodule instance. Only the address and attribute information for the first error have been captured in DERR_W0_i and DERR_W1_i.
29-27 —	Reserved
26-24 EPORT	Error port This field identifies the encoded port number of the MBC that detected the access violation. The MBC port number connection is device-specific. See the chip configuration details for more information. This is the MBC target that has the violation. 000 - mbcxslv0 001 - mbcxslv1 010 - mbcxslv2 011 - mbcxslv3
23-12 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
11 ERW	Error read/write This field signals whether the captured access violation occurred on a read or write reference. 0 - Read access 1 - Write access
10-8 EATR	Error attributes This field captures attributes of the access violation. 000 - Secure user mode, instruction fetch access. 001 - Secure user mode, data access. 010 - Secure privileged mode, instruction fetch access. 011 - Secure privileged mode, data access. 100 - Nonsecure user mode, instruction fetch access. 101 - Nonsecure user mode, data access. 110 - Nonsecure privileged mode, instruction fetch access. 111 - Nonsecure privileged mode, data access.
7 SLV_ABORT	Bus protect error This field captures the bus protect error from target which cause access abort.
6 SLV_SA_ERR	Secure attribute check error This field captures the secure attribute check error from target.
5 SLV_PA_ERR	Privilege attribute check error This field captures the privilege attribute check error from target.
4 SLV_DID_ERR	DID check error This field captures the DID check error from target.
3-0 EDID	Error domain identifier This field captures the domain identifier of the access violation.

22.7.5.17 MBC Domain Error Word3 Register (MBC0_DERR_W3 - MBC3_DERR_W3)

This register is used to rearm the error capture logic and clear the DERR_W0_i and DERR_W1_i registers. After the domain access violation error details have been read, typically in an exception service routine, a 32-bit word write to this register is required to rearm the error capture logic.

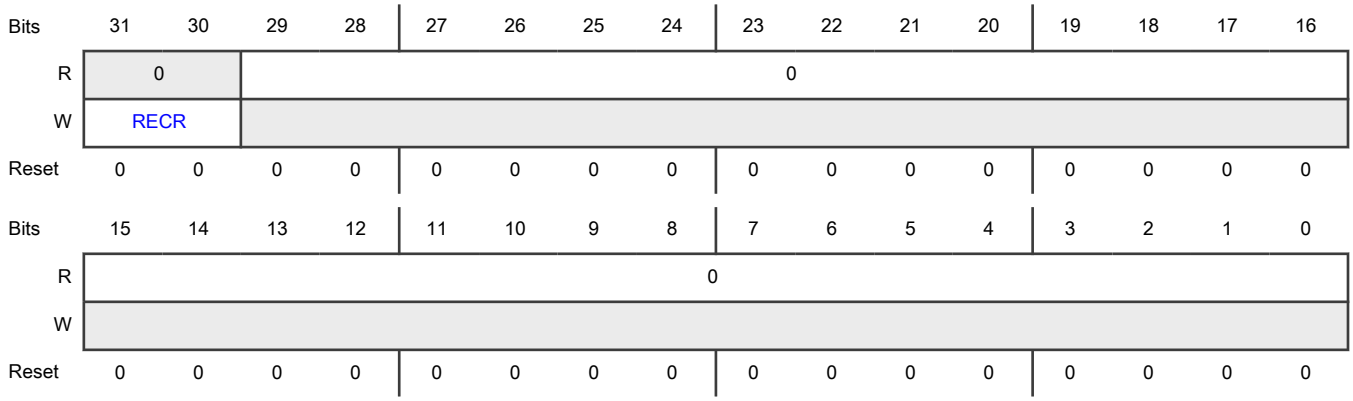
A read of this location returns zeroes. Attempted reads of a Memory Region Checker (MRC) or Memory Block Checker (MBC) instance that is not physically present are error terminated.

See Domain error capture management for more details.

Offset

Register	Offset
MBC0_DERR_W3	40Ch
MBC1_DERR_W3	41Ch
MBC2_DERR_W3	42Ch
MBC3_DERR_W3	43Ch

Diagram



Fields

Field	Description
31-30 RECR	Rearm Error Capture Registers This 2-bit, write-only field controls the rearming of the domain error capture registers. Once an access violation has been detected with the faulting address and attribute information stored, subsequent errors are simply recorded as an overrun condition without any data captured. Writing 01b to this field rearms the error capture mechanism and clears the DERR_W0_i and DERR_W1_i registers. A write of any value other than 01b has no effect.
29-0 —	Reserved

22.7.5.18 MRC Domain Error Word0 Register (MRC0_DERR_W0)

This read-only register array provides the address of an access violation detected by memory region checker (MRC). These registers are organized as a word array, which is indexed by the MRC instance number. That is, the index, i, of this array is the instance number of MRC with the access violation. The submodule instance numbers are provided by the DERRLOC registers. The memory mapped error capture detail registers are organized as 24 sequential 16 byte entries.

When an access violation is detected and the offending information captured, subsequent updates to this register are disabled until the required data pattern is written to the DERR_W3_i register. At that time, this register is cleared and re-enabled to capture the next access violation.

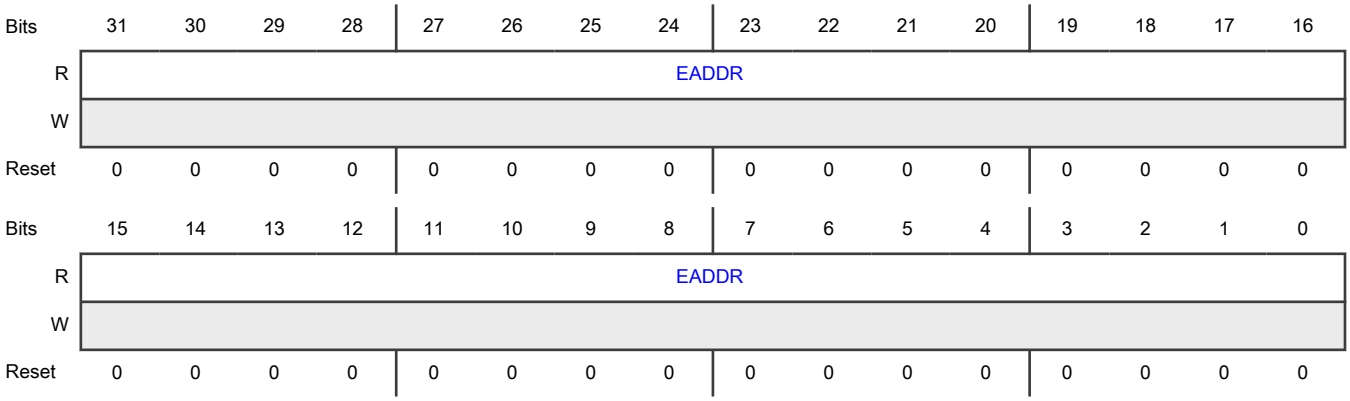
Attempted writes are error terminated as are attempted reads of an MRC instance that is not physically present.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
MRC0_DERR_W0	480h

Diagram



Fields

Field	Description
31-0 EADDR	Error address This is the unaliased virtual address of the access that generated the access violation.

22.7.5.19 MRC Domain Error Word1 Register (MRC0_DERR_W1)

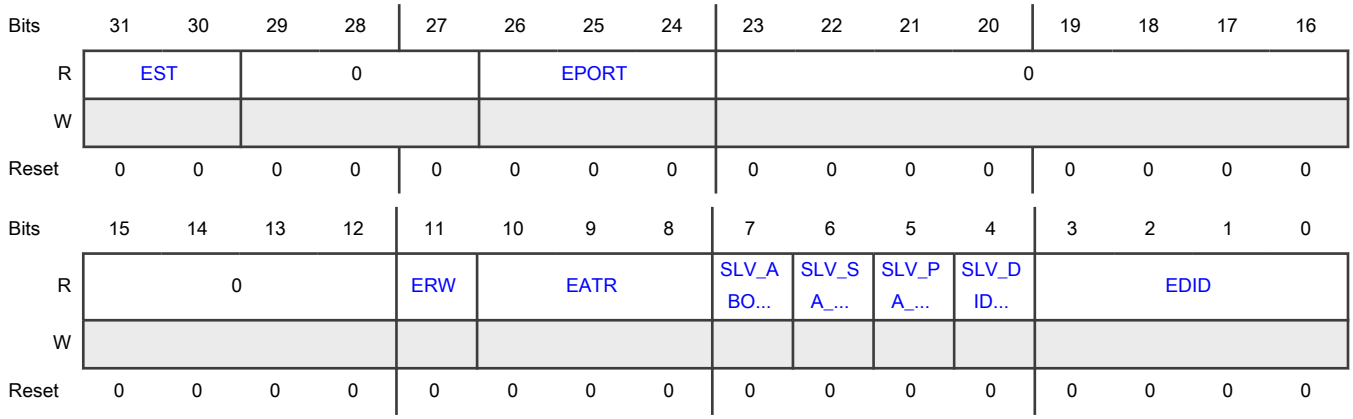
This read-only register array provides the attributes of an access violation detected by memory region checker (MRC). These registers are organized as a word array, which is indexed by the violating submodule instance number. Refer to register Domain Error Location(DERRLOCd), Domain Error Word0 Register(DERR_W0_i), and Domain error capture management for more information.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
MRC0_DERR_W1	484h

Diagram



Fields

Field	Description
31-30 EST	Error state This field signals the state of access violations for this domain in this instance of the block checker or region checker. Once an access violation has been detected and the faulting address and attribute information stored, subsequent errors are simply recorded as an overrun condition without any data captured. After retrieving the faulting address and attributes, the error capture mechanism must be rearmed by performing a write to DERR_W3_i. 00 - No access violation has been detected. 01 - No access violation has been detected. 10 - A single access violation has been detected. 11 - Multiple access violations for this domain have been detected by this submodule instance. Only the address and attribute information for the first error have been captured in DERR_W0_i and DERR_W1_i.
29-27 —	Reserved
26-24 EPORT	Error port This field identifies the encoded port number of the MRC that detected the access violation. The MRC port number connection is device-specific. See the chip configuration details for more information.
23-12 —	Reserved
11 ERW	Error read/write This field signals whether the captured access violation occurred on a read or write reference. 0 - Read access 1 - Write access
10-8	Error attributes

Table continues on the next page...

Table continued from the previous page...

Field	Description
EATR	This field captures certain attributes of the access violation. 000 - Secure user mode, instruction fetch access. 001 - Secure user mode, data access. 010 - Secure privileged mode, instruction fetch access. 011 - Secure privileged mode, data access. 100 - Nonsecure user mode, instruction fetch access. 101 - Nonsecure user mode, data access. 110 - Nonsecure privileged mode, instruction fetch access. 111 - Nonsecure privileged mode, data access.
7 SLV_ABORT	Bus protect error This field captures the bus protect error from target which cause access abort.
6 SLV_SA_ERR	Secure attribute check error This field captures the secure attribute check error from target.
5 SLV_PA_ERR	Privilege attribute check error This field captures the privilege attribute check error from target.
4 SLV_DID_ERR	DID check error This field captures the DID check error from target.
3-0 EDID	Error domain identifier This field captures the domain identifier of the access violation.

22.7.5.20 MRC Domain Error Word3 Register (MRC0_DERR_W3)

This register is used to rearm the error capture logic and clear the DERR_W0_i and DERR_W1_i registers. After the domain access violation error details have been read, typically in an exception service routine, a 32-bit word write to this register is required to rearm the error capture logic.

A read of this location returns zeroes. Attempted reads of a memory region checker (MRC) or Memory Block Checker (MBC) instance that is not physically present are error terminated.

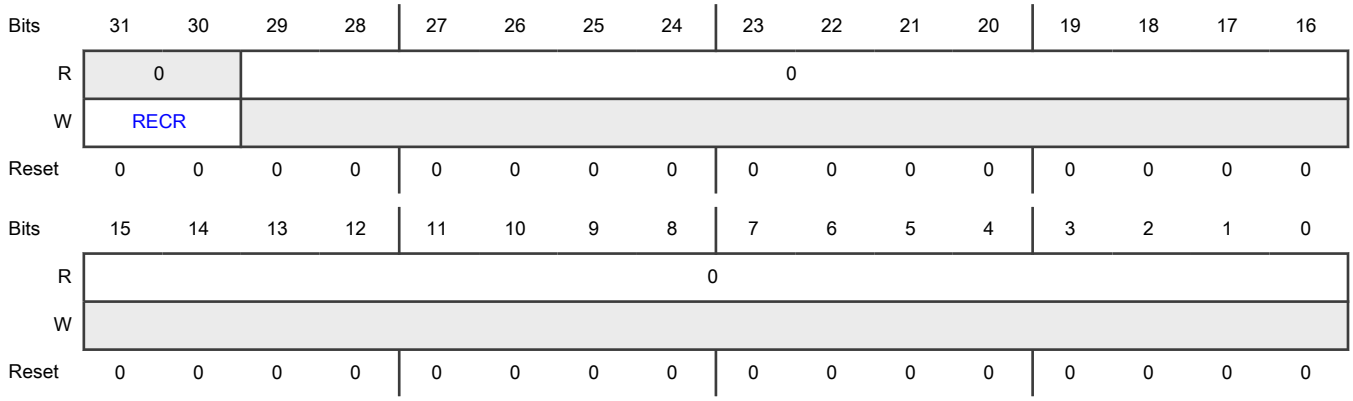
See Domain error capture management for more details.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
MRC0_DERR_W3	48Ch

Diagram



Fields

Field	Description
31-30 RECR	Rearm Error Capture Registers This 2-bit, write-only field controls the rearming of the domain error capture registers. Once an access violation has been detected with the faulting address and attribute information stored, subsequent errors are simply recorded as an overrun condition without any data captured. Writing 01b to this field rearms the error capture mechanism and clears the DERR_W0_i and DERR_W1_i registers. A write of any value other than 01b has no effect.
29-0 —	Reserved

22.7.5.21 DAC Initiator Domain Assignment Register (MDA_W0_0_DFMT0 - MDA_W3_1_DFMT0)

The MDA_Wr_m registers provide a 2-dimensional data structure for assigning bus initiators to domains. The number of implemented registers is defined by DACFGm[NMDAR]. This per-initiator domain assignment is then repeated for each bus initiator (MDAm). Thus, m specifies the initiator number and r refers to the specific MDA register for a given bus initiator.

Each Wr within the MDAm structure is a word-sized definition; there are two formats supported, one for processor cores and another for non-processors. Processor initiators typically support one or more Wr domain definitions, while non-processor initiators support a single Wr. Processor initiators Domain Assignment register names use a _DFMT0 suffix. Non-processor initiators Domain Assignment register names use a _DFMT1 suffix.

The DAC submodule is responsible for the generation of domain identifiers for every transaction from every bus initiator. If there is a single Wr for a given initiator, then the specified domain identifier is used directly. If there are multiple Wr values for a given initiator, then the DAC evaluates the conditional terms to determine a “hit”. For all Wr hits, their corresponding domain identifiers are simply logically summed together (boolean OR). Use cases are typically expected to *hit in a single Wr* for a processor initiator. Special care is needed if *none* of the conditional terms hit in any Wr evaluation; for this case, the generated DID = 0 and software needs to be aware of any potential access rights granted for this DID.

Each MDA_Wr_m register has one of two programming models depending on the state of the domain format field, DFMT. The model described in this section is for DFMT = 0. This definition allows three different specifications of the DID for processors. The DFMT = 1 model is described in Initiator Domain Assignment (MDA_Wr_m_DFMT1) register.

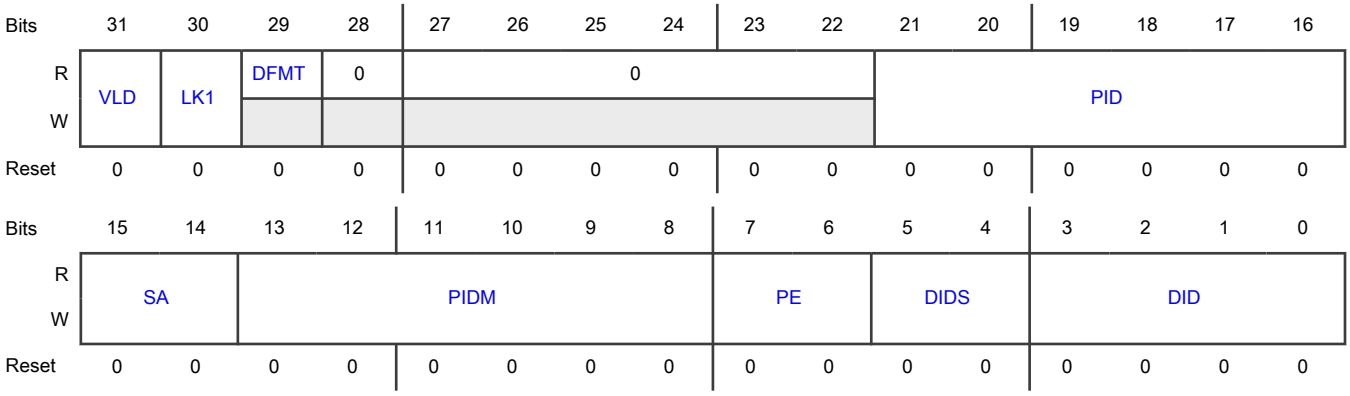
Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/write.

Offset

For m = 0 to 1; r = 0 to 3:

Register	Offset
MDA_Wr_m_DFMT0	800h + (m × 20h) + (r × 4h)

Diagram



Fields

Field	Description
31 VLD	Valid This field indicates the domain assignment is valid. It is further qualified by TRDC_CR[GVLDM] = 1. If TRDC_CR[GVLDM] is cleared, the DID output is defined by the SoC-specific *default DID* value. See the TRDC chip configuration section for more details. If both MDA_Wr_m[VLD] and TRDC_CR[GVLDM] are asserted, the DID output is defined by the remaining contents of this register. 0 - The Wr domain assignment is invalid. 1 - The Wr domain assignment is valid.
30 LK1	1-bit Lock This field provides a locking mechanism that can be used to limit the ability to write the register. Once set, this bit remains asserted until the next reset. 0 - Register can be written by any secure privileged write. 1 - Register is locked (read-only) until the next reset.
29 DFMT	Domain format Identifies this register's domain assignment format. **NOTE** This bitfield access is ROZ 0 - Processor-core domain assignment 1 - Non-processor domain assignment

Table continues on the next page...

Table continued from the previous page...

Field	Description
28 —	Reserved
27-22 —	Reserved
21-16 PID	Process Identifier This field specifies that the process identifier is to be combined with the PIDM field and included in the domain hit determination. The optional inclusion of the PID and PIDM is controlled by the MDA_Wr_m[PE] field.
15-14 SA	Secure attribute This field defines the secure/nonsecure attribute for processor cores. **NOTE** The bus initiator's input secure/nonsecure attribute is used if SA = 1X, or this VLD = 0. If TZM_ENB = 0, the initiator attribute is forced to nonsecure. A nonsecure write cannot program this field to 2'b00, which is a security level higher than the mode of the process that is writing it. Reset value of SA: • if TZM_ENB=0, SA reset value = 2'b01 • if TZM_ENB=1, SA reset value = 2'b00 00 - Force the bus attribute for this initiator to secure. 01 - Force the bus attribute for this initiator to nonsecure. 10 - Use the bus initiator's secure/nonsecure attribute directly. 11 - Use the bus initiator's secure/nonsecure attribute directly.
13-8 PIDM	Process Identifier Mask This field provides a masking capability so that multiple process identifiers can be included as part of the domain hit determination. If a bit in the PIDM is set, then the corresponding bit of the PID is ignored in the comparison. The optional inclusion of the PID and PIDM is controlled by the MDA_Wr_m[PE] field.

7-6
PE
	Process identifier enable
	Controls the optional inclusion of the PID, qualified by PIDM, into the domain hit evaluation. It provides the ability to include inclusive or exclusive sets of masked PID values.
	00 - No process identifier is included in the domain hit evaluation.
	01 - No process identifier is included in the domain hit evaluation.
	10 - PE = 2. The process identifier is included in the domain hit evaluation as defined by the expression:
	`partial_domain_hit = (PE == 2) && ((PID & ~PIDM) == (TRDC_PIDn[PID] & ~PIDM))`
	11 - PE = 3. The process identifier is included in the domain hit evaluation as defined by the expression:
	`partial_domain_hit = (PE == 3) && ~((PID & ~PIDM) == (TRDC_PIDn[PID] & ~PIDM))`

Table continues on the next page...

Table continued from the previous page...

Field	Description
5-4 DIDS	DID Select This field selects the source of the domain identifier. 00 - Use MDAm[3:0] as the domain identifier. 01 - Use the input DID as the domain identifier. 10 - Use MDAm[3:2] concatenated with the low-order 2 bits of the input DID (DID_in[1:0]) as the domain identifier. 11 - Reserved for future use.
3-0 DID	Domain identifier This 4-bit field is the domain ID attribute that is sent on the accesses from the bus initiator connected to the DAC when DIDS=00b. DID[3:2] is used when DIDS=10b. Write a value greater than the number of DIDs is illegal.

22.7.5.22 DAC Initiator Domain Assignment Register (MDA_W0_2_DFMT1 - MDA_W0_3_DFMT1)

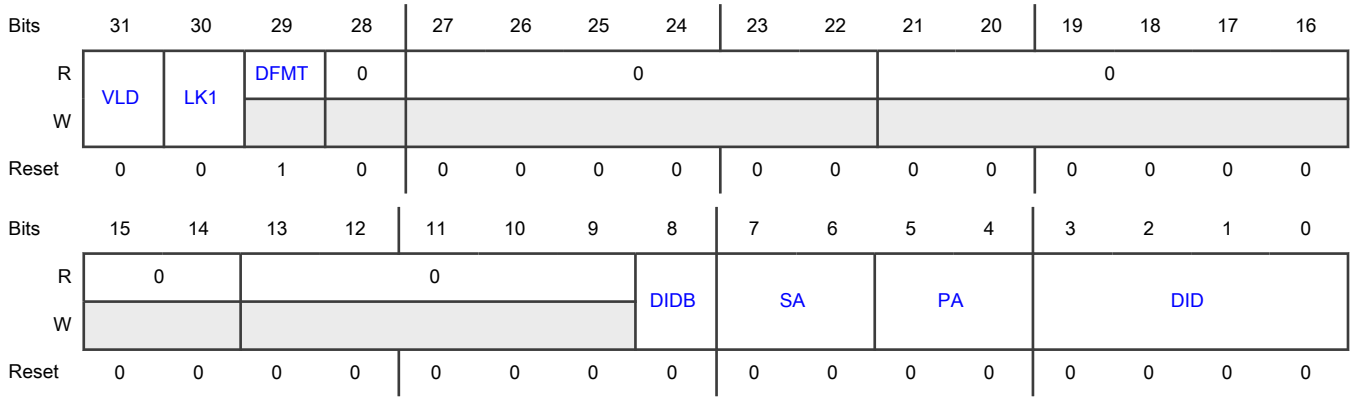
This register is identical to Initiator Domain Assignment (MDA_Wr_m_DFMT0) register except that the domain format field, DFMT, is 1 and the PID field is not used. This format supports two different specifications of the DID for non-core bus initiators.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/write.

Offset

Register	Offset
MDA_W0_2_DFMT1	840h
MDA_W0_3_DFMT1	860h

Diagram



Fields

Field	Description
31 VLD	Valid This field indicates the domain assignment is valid. It is further qualified by TRDC_CR[GVLDM] = 1. If TRDC_CR[GVLDM] is cleared, the DID output is defined by the SoC-specific *default DID* value. See the TRDC chip configuration section for more details. If both MDA_Wr_m[VLD] and TRDC_CR[GVLDM] are asserted, the DID output is defined by the remaining contents of this register. 0 - The Wr domain assignment is invalid. 1 - The Wr domain assignment is valid.
30 LK1	1-bit Lock This field provides a locking mechanism that can be used to limit the ability to write the register. Once set, this bit remains asserted until the next reset. 0 - Register can be written by any secure privileged write. 1 - Register is locked (read-only) until the next reset.
29 DFMT	Domain format Identifies this register's domain assignment format. **NOTE** This bitfield access is ROO 0 - Processor-core domain assignment 1 - Non-processor domain assignment
28 —	Reserved
27-22 —	Reserved
21-16 —	Reserved
15-14 —	Reserved
13-9 —	Reserved
8 DIDB	DID Bypass If asserted, this bit enables the bypassing of an input DID value as the domain identifier for this non-processor bus initiator. This capability allows non-processor bus initiators, for example, a DMA to masquerade as a processor. Once set, this field is “sticky” and remains set until the next reset.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>DAC</th><th>DID Input</th></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>DMA Domain ID</td></tr>
<tr><td>2</td><td>USB Domain ID</td></tr>
</table>
0 - Use MDAn[3:0] as the domain identifier.
1 - Use the DID input as the domain identifier.
</td></tr>
<tr><td>7-6
SA</td><td>Secure attribute
This field defines the secure/nonsecure attribute for non-processor cores.
NOTE
The bus initiator’s input secure/nonsecure attribute is used if SA = 1X, or this VLD = 0. If TZM_ENB = 0, the initiator attribute is forced to nonsecure. A nonsecure write cannot program this field to 2'b00, which is a security level higher than the mode of the process that is writing it.
Reset value of SA:
• if TZM_ENB=0, SA reset value = 2'b01
• if TZM_ENB=1, SA reset value = 2'b00
00 - Force the bus attribute for this initiator to secure.
01 - Force the bus attribute for this initiator to nonsecure.
10 - Use the bus initiator’s secure/nonsecure attribute directly.
11 - Use the bus initiator’s secure/nonsecure attribute directly.</td></tr>
<tr><td>5-4
PA</td><td>Privileged attribute
This field defines the privileged/user attribute for non-processor cores.
NOTE
The bus initiator’s input privileged/user attribute is used if PA = 1X, or this VLD = 0.
00 - Force the bus attribute for this initiator to user.
01 - Force the bus attribute for this initiator to privileged.
10 - Use the bus initiator’s privileged/user attribute directly.
11 - Use the bus initiator’s privileged/user attribute directly.</td></tr>
<tr><td>3-0
DID</td><td>Domain identifier
This 4-bit field is the domain ID attribute that is sent on the accesses from the bus initiator connected to the DAC when DIDB=0.
Write a value greater than the number of DIDs is illegal.</td></tr>
</table>

22.7.5.23 MBC Global Configuration Register (MBC0_MEM0_GLBCFG - MBC3_MEM3_GLBCFG)

These MBC global configuration read-only registers contain information on the MBC's hardware configuration. Specifically, it defines the number of memory blocks and the size of each block in each MBC mem (r).

NOTE

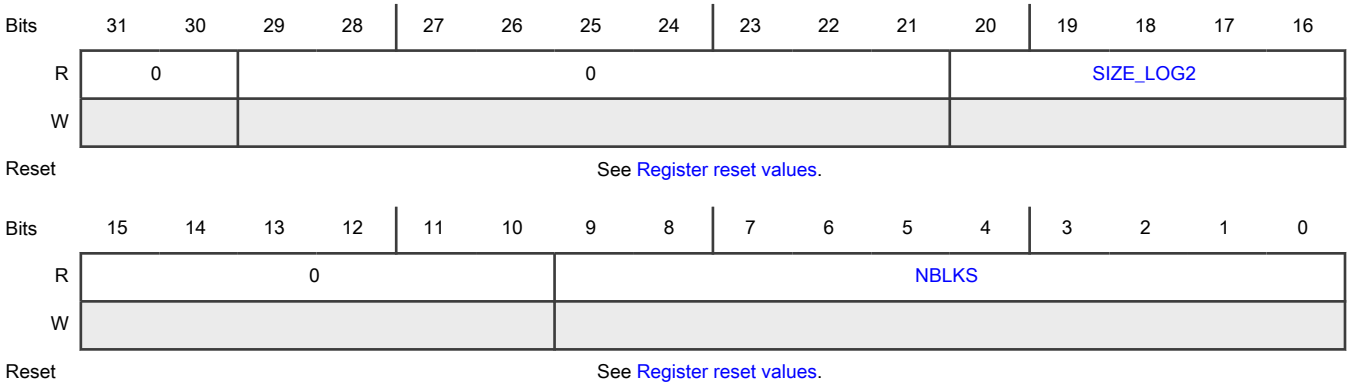
An all zero value (SIZE_LOG2 = 0, NBLKS = 0) indicates a non-existent memory. For the number of memories implemented in the MBCs, refer to chip-specific information

Offset

For m = 0 to 3; r = 0 to 3:

Register	Offset
MBCm_MEMr_GLBCFG	1_0000h + (m × 2000h) + (r × 4h)

Diagram



Register reset values

Register	Reset value
MBC0_MEM0_GLBCFG	0010_0031h
MBC0_MEM1_GLBCFG	0010_000Bh
MBC0_MEM2_GLBCFG	0010_0025h
MBC0_MEM3_GLBCFG	0010_0030h
MBC1_MEM0_GLBCFG–MBC1_MEM1_GLBCFG	0010_0005h
MBC1_MEM2_GLBCFG–MBC1_MEM3_GLBCFG	0010_0011h
MBC2_MEM0_GLBCFG–MBC2_MEM1_GLBCFG	0010_0009h
MBC2_MEM2_GLBCFG–MBC2_MEM3_GLBCFG	000C_0000h
MBC3_MEM0_GLBCFG–MBC3_MEM1_GLBCFG	000E_0028h
MBC3_MEM2_GLBCFG–MBC3_MEM3_GLBCFG	000C_0000h

Fields

Field	Description
31-30 —	Reserved
29-21 —	Reserved
20-16 SIZE_LOG2	Log2 size per block For example SIZE_LOG2=0x0C is 2^12=4 KB blocks.
15-10 —	Reserved
9-0 NBLKS	Number of blocks in this memory

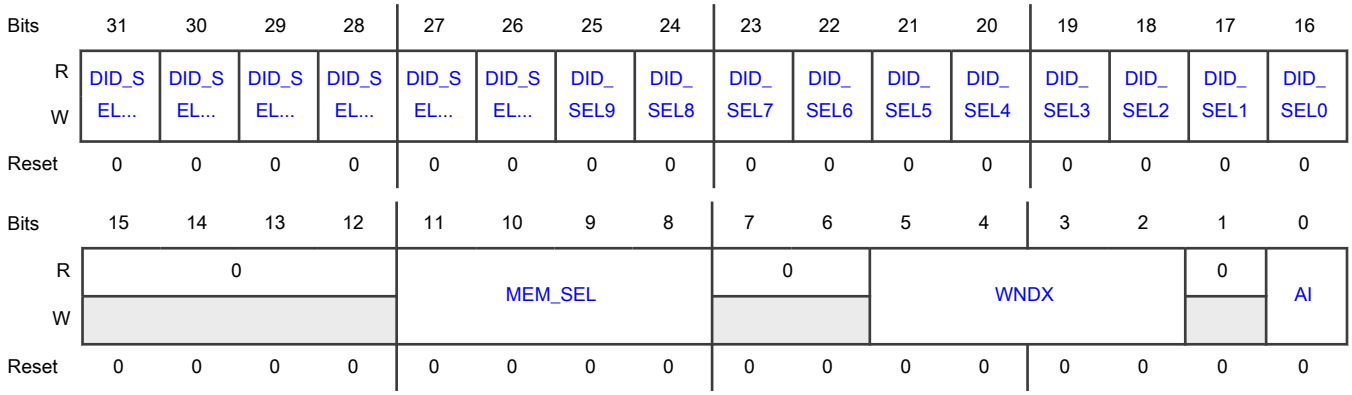
22.7.5.24 MBC NonSecure Enable Block Index (MBC0_NSE_BLK_INDEX - MBC3_NSE_BLK_INDEX)

This R/W register defines the selected memories that are affected by the writes to the NSE_BLK_SET and NSE_BLK_CLR registers.

Offset

Register	Offset
MBC0_NSE_BLK_INDEX	1_0010h
MBC1_NSE_BLK_INDEX	1_2010h
MBC2_NSE_BLK_INDEX	1_4010h
MBC3_NSE_BLK_INDEX	1_6010h

Diagram



Fields

Field	Description
31-16 DID_SELn	DID Select Destination domain bitmap select. 0 - No effect. 1 - Selects NSE bits for this domain.
15-12 —	Reserved
11-8 MEM_SEL	Memory Select Destination memory bitmap select. • Bit [11] - MBC MEM 3 • Bit [10] - MBC MEM 2 • Bit [9] - MBC MEM 1 • Bit [8] - MBC MEM 0
7-6 —	Reserved
5-2 WNDX	Word index into the block NSE bitmap. It selects the BLK_NSE_Wn register, where WNDX determines the value of n.
1 —	Reserved
0 AI	Auto Increment 0 - No effect. 1 - Add 1 to the WNDX field after the register write.

22.7.5.25 MBC NonSecure Enable Block Set (MBC0_NSE_BLK_SET - MBC3_NSE_BLK_SET)

A write to this register sets the appropriate NSE Bits for the selected domains and memories defined at the word location NSE_BLK_INDEX[WDNX]. If NSE_BLK_INDEX[AI] = 1, then the NSE_BLK_INDEX[WDNX] field is incremented by 1 (modulo-16) after the write is completed. This register reads as zero.

Offset

Register	Offset
MBC0_NSE_BLK_SET	1_0014h
MBC1_NSE_BLK_SET	1_2014h
MBC2_NSE_BLK_SET	1_4014h
MBC3_NSE_BLK_SET	1_6014h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	W1SET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	W1SET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 W1SET	Write-1 Set If set to 1, sets appropriate NSE bit for the selected domains and memories defined at the word location NSE_BLK_INDEX[WDNX]. Write with value 0 has no effect.

22.7.5.26 MBC NonSecure Enable Block Clear (MBC0_NSE_BLK_CLR - MBC3_NSE_BLK_CLR)

A write to this location clears the appropriate NSE bits as defined by the W1CLR[n]=1 for the selected domains and memories Defined at the word location NSE_BLK_INDEX[WDNX]. If NSE_BLK_INDEX[AI] = 1, then the NSE_BLK_INDEX[WDNX] field is incremented by 1 (modulo-16) after the write is completed. This register reads as zero.

Offset

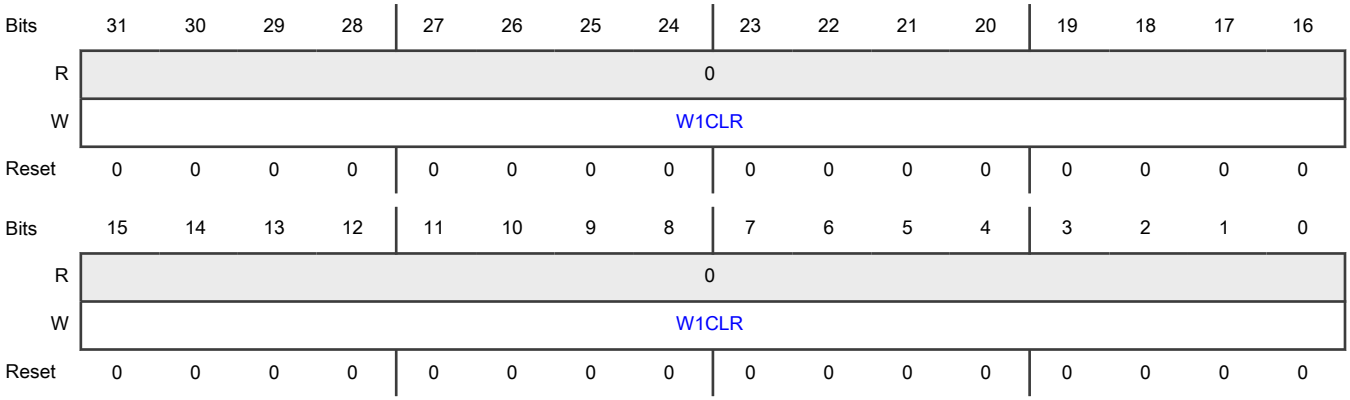
Register	Offset
MBC0_NSE_BLK_CLR	1_0018h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_NSE_BLK_CLR	1_2018h
MBC2_NSE_BLK_CLR	1_4018h
MBC3_NSE_BLK_CLR	1_6018h

Diagram



Fields

Field	Description
31-0 W1CLR	Write-1 Clear If set to 1, Clear the appropriate NSE bit for the selected domains and memories defined at the word location NSE_BLK_INDEX[WDNX]. Write with value 0 has no effect.

22.7.5.27 MBC NonSecure Enable Block Clear All (MBC0_NSE_BLK_CLR_ALL - MBC3_NSE_BLK_CLR_ALL)

A write to this location clears all the block NSE bits for the selected domains and memories defined in the write data. This register reads as zero.

Offset

Register	Offset
MBC0_NSE_BLK_CLR_ALL	1_001Ch
MBC1_NSE_BLK_CLR_ALL	1_201Ch
MBC2_NSE_BLK_CLR_ALL	1_401Ch
MBC3_NSE_BLK_CLR_ALL	1_601Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL9	DID_SEL8	DID_SEL7	DID_SEL6	DID_SEL5	DID_SEL4	DID_SEL3	DID_SEL2	DID_SEL1	DID_SEL0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0				0							
W					MEMSEL											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 DID_SELn	DID Select Destination domain bitmap select. 0 - No effect. 1 - Clear all NSE bits for this domain.
15-12 —	Reserved
11-8 MEMSEL	Memory Select Destination memory bitmap select. • Bit [11] - clear all MEM 3 NSE bits • Bit [10] - clear all MEM 2 NSE bits • Bit [9] - clear all MEM 1 NSE bits • Bit [8] - clear all MEM 0 NSE bits
7-0 —	Reserved

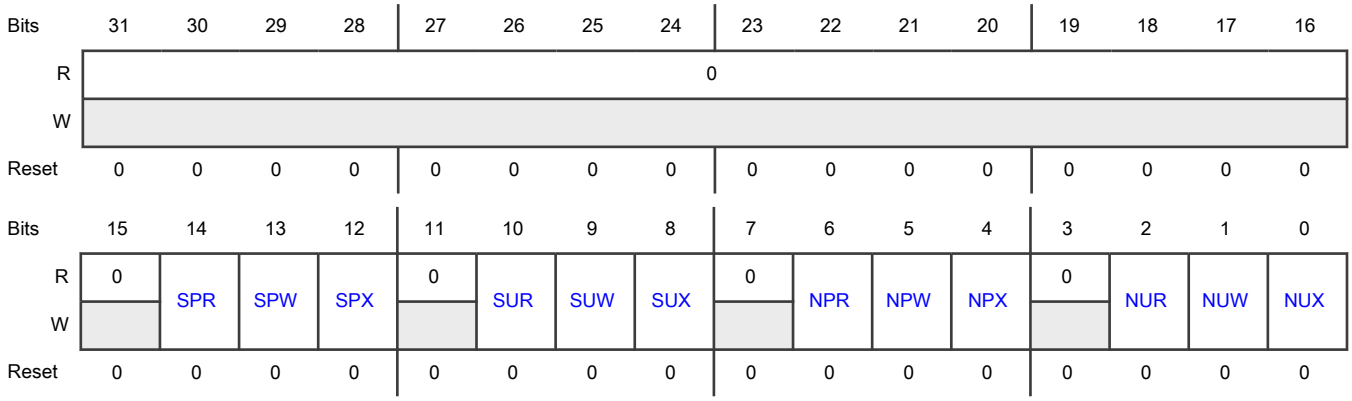
22.7.5.28 MBC Global Access Control (MBC0_MEMN_GLBAC0)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User.

Offset

Register	Offset
MBC0_MEMN_GLBAC0	1_0020h

Diagram



Fields

Field	Description
31-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9	SecureUser Write

Table continues on the next page...

Table continued from the previous page...

Field	Description
SUW	SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

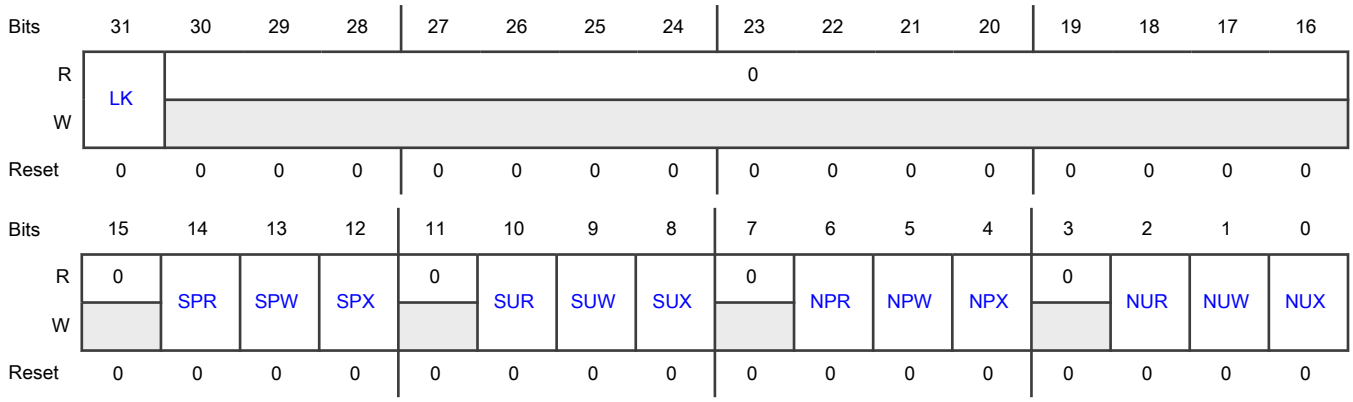
22.7.5.29 MBC Global Access Control (MBC0_MEMN_GLBAC1 - MBC0_MEMN_GLBAC7)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

Offset

Register	Offset
MBC0_MEMN_GLBAC1	1_0024h
MBC0_MEMN_GLBAC2	1_0028h
MBC0_MEMN_GLBAC3	1_002Ch
MBC0_MEMN_GLBAC4	1_0030h
MBC0_MEMN_GLBAC5	1_0034h
MBC0_MEMN_GLBAC6	1_0038h
MBC0_MEMN_GLBAC7	1_003Ch

Diagram



Fields

Field	Description
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0 - This register is not locked and can be altered. 1 - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

22.7.5.30 MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W0 - MBC1_DOM15_MEM3_BLK_CFG_W2)

MBC[m]_DOM[d]_MEM[s]_BLK_CFG_W[w], where,

- m - mbc index
- d - domain index
- s - memory target index
- w - word index

These registers are read/write for both the alternate view of the NSE bit and the 3-bit MBACSEL field. This is an array of 4 bit fields defining the block configuration for the given submemory. Each 4-bit field includes an alternative view of the associated NSE bit plus a 3-bit access control select that selects the global access control value that applies to the referenced memory block.

For a given memory block, B, if the value programmed in the MBACSELn field selects a locked MBC_MEMN_GLBACr register, the MBACSEL field is locked until the next reset. Depending on BLK_CFG_W, the NSEn/MBASELn fields correspond to different blocks. The following table describes the relationship between W (word index) and B (block number).

MEM0 supports up to 512 blocks ; W0 - W63

MEM1-3 supports up to 64 block ; W0 - W7

NOTE

This register is shown with fields such that it covers entire 32 bit register. However, actual fields may be less if number of blocks are such that fields don't align to 32 bits. These absent fields may be shown but user can refer to table below to deduce actual fields and treat absent fields as reserved.

Table 112. Block Config Word to Block Number Relationship

Block Config Word	NSE7/ MBACSEL7	NSE6/ MBACSEL6	NSE5/ MBACSEL5	NSE4/ MBACSEL4	NSE3/ MBACSEL3	NSE2/ MBACSEL2	NSE1/ MBACSEL1	NSE0/ MBACSEL0
W0	block 7	block 6	block 5	block 4	block 3	block 2	block 1	block 0
W1	block 15	block 14	block 13	block 12	block 11	block 10	block 9	block 8
W2	block 23	block 22	block 21	block 20	block 19	block 18	block 17	block 16
W3	block 31	block 30	block 29	block 28	block 27	block 26	block 25	block 24
W4	block 39	block 38	block 37	block 36	block 35	block 34	block 33	block 32
W5	block 47	block 46	block 45	block 44	block 43	block 42	block 41	block 40
W6	block 55	block 54	block 53	block 52	block 51	block 50	block 49	block 48
W7	block 63	block 62	block 61	block 60	block 59	block 58	block 57	block 56
For MEM0, block 64 - block 511								
W8	block 71	block 70	block 69	block 68	block 67	block 66	block 65	block 64
. .								

Table continues on the next page...

Table 112. Block Config Word to Block Number Relationship (continued)

Block Config Word	NSE7/ MBACSEL7	NSE6/ MBACSEL6	NSE5/ MBACSEL5	NSE4/ MBACSEL4	NSE3/ MBACSEL3	NSE2/ MBACSEL2	NSE1/ MBACSEL1	NSE0/ MBACSEL0
.								
W63	block 511	block 510	block 509	block 508	block 507	block 506	block 505	block 504

Offset

Register	Offset
MBC0_DOM0_MEM0_BLK_CFG_W0	1_0040h
MBC0_DOM0_MEM0_BLK_CFG_W1	1_0044h
MBC0_DOM0_MEM0_BLK_CFG_W2	1_0048h
MBC0_DOM0_MEM0_BLK_CFG_W3	1_004Ch
MBC0_DOM0_MEM0_BLK_CFG_W4	1_0050h
MBC0_DOM0_MEM0_BLK_CFG_W5	1_0054h
MBC0_DOM0_MEM0_BLK_CFG_W6	1_0058h
MBC0_DOM0_MEM1_BLK_CFG_W0	1_0180h
MBC0_DOM0_MEM1_BLK_CFG_W1	1_0184h
MBC0_DOM0_MEM2_BLK_CFG_W0	1_01A8h
MBC0_DOM0_MEM2_BLK_CFG_W1	1_01ACh
MBC0_DOM0_MEM2_BLK_CFG_W2	1_01B0h
MBC0_DOM0_MEM2_BLK_CFG_W3	1_01B4h
MBC0_DOM0_MEM2_BLK_CFG_W4	1_01B8h
MBC0_DOM0_MEM3_BLK_CFG_W0	1_01D0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM0_MEM3_BLK_CFG_W1	1_01D4h
MBC0_DOM0_MEM3_BLK_CFG_W2	1_01D8h
MBC0_DOM0_MEM3_BLK_CFG_W3	1_01DCh
MBC0_DOM0_MEM3_BLK_CFG_W4	1_01E0h
MBC0_DOM0_MEM3_BLK_CFG_W5	1_01E4h
MBC0_DOM1_MEM0_BLK_CFG_W0	1_0240h
MBC0_DOM1_MEM0_BLK_CFG_W1	1_0244h
MBC0_DOM1_MEM0_BLK_CFG_W2	1_0248h
MBC0_DOM1_MEM0_BLK_CFG_W3	1_024Ch
MBC0_DOM1_MEM0_BLK_CFG_W4	1_0250h
MBC0_DOM1_MEM0_BLK_CFG_W5	1_0254h
MBC0_DOM1_MEM0_BLK_CFG_W6	1_0258h
MBC0_DOM1_MEM1_BLK_CFG_W0	1_0380h
MBC0_DOM1_MEM1_BLK_CFG_W1	1_0384h
MBC0_DOM1_MEM2_BLK_CFG_W0	1_03A8h
MBC0_DOM1_MEM2_BLK_CFG_W1	1_03ACh
MBC0_DOM1_MEM2_BLK_CFG_W2	1_03B0h
MBC0_DOM1_MEM2_BLK_CFG_W3	1_03B4h
MBC0_DOM1_MEM2_BLK_CFG_W4	1_03B8h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM1_MEM3_BLK_CFG_W0	1_03D0h
MBC0_DOM1_MEM3_BLK_CFG_W1	1_03D4h
MBC0_DOM1_MEM3_BLK_CFG_W2	1_03D8h
MBC0_DOM1_MEM3_BLK_CFG_W3	1_03DCh
MBC0_DOM1_MEM3_BLK_CFG_W4	1_03E0h
MBC0_DOM1_MEM3_BLK_CFG_W5	1_03E4h
MBC0_DOM2_MEM0_BLK_CFG_W0	1_0440h
MBC0_DOM2_MEM0_BLK_CFG_W1	1_0444h
MBC0_DOM2_MEM0_BLK_CFG_W2	1_0448h
MBC0_DOM2_MEM0_BLK_CFG_W3	1_044Ch
MBC0_DOM2_MEM0_BLK_CFG_W4	1_0450h
MBC0_DOM2_MEM0_BLK_CFG_W5	1_0454h
MBC0_DOM2_MEM0_BLK_CFG_W6	1_0458h
MBC0_DOM2_MEM1_BLK_CFG_W0	1_0580h
MBC0_DOM2_MEM1_BLK_CFG_W1	1_0584h
MBC0_DOM2_MEM2_BLK_CFG_W0	1_05A8h
MBC0_DOM2_MEM2_BLK_CFG_W1	1_05ACh
MBC0_DOM2_MEM2_BLK_CFG_W2	1_05B0h
MBC0_DOM2_MEM2_BLK_CFG_W3	1_05B4h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM2_MEM2_BLK_CFG_W4	1_05B8h
MBC0_DOM2_MEM3_BLK_CFG_W0	1_05D0h
MBC0_DOM2_MEM3_BLK_CFG_W1	1_05D4h
MBC0_DOM2_MEM3_BLK_CFG_W2	1_05D8h
MBC0_DOM2_MEM3_BLK_CFG_W3	1_05DCh
MBC0_DOM2_MEM3_BLK_CFG_W4	1_05E0h
MBC0_DOM2_MEM3_BLK_CFG_W5	1_05E4h
MBC0_DOM3_MEM0_BLK_CFG_W0	1_0640h
MBC0_DOM3_MEM0_BLK_CFG_W1	1_0644h
MBC0_DOM3_MEM0_BLK_CFG_W2	1_0648h
MBC0_DOM3_MEM0_BLK_CFG_W3	1_064Ch
MBC0_DOM3_MEM0_BLK_CFG_W4	1_0650h
MBC0_DOM3_MEM0_BLK_CFG_W5	1_0654h
MBC0_DOM3_MEM0_BLK_CFG_W6	1_0658h
MBC0_DOM3_MEM1_BLK_CFG_W0	1_0780h
MBC0_DOM3_MEM1_BLK_CFG_W1	1_0784h
MBC0_DOM3_MEM2_BLK_CFG_W0	1_07A8h
MBC0_DOM3_MEM2_BLK_CFG_W1	1_07ACh
MBC0_DOM3_MEM2_BLK_CFG_W2	1_07B0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM3_MEM2_BLK_CFG_W3	1_07B4h
MBC0_DOM3_MEM2_BLK_CFG_W4	1_07B8h
MBC0_DOM3_MEM3_BLK_CFG_W0	1_07D0h
MBC0_DOM3_MEM3_BLK_CFG_W1	1_07D4h
MBC0_DOM3_MEM3_BLK_CFG_W2	1_07D8h
MBC0_DOM3_MEM3_BLK_CFG_W3	1_07DCh
MBC0_DOM3_MEM3_BLK_CFG_W4	1_07E0h
MBC0_DOM3_MEM3_BLK_CFG_W5	1_07E4h
MBC0_DOM4_MEM0_BLK_CFG_W0	1_0840h
MBC0_DOM4_MEM0_BLK_CFG_W1	1_0844h
MBC0_DOM4_MEM0_BLK_CFG_W2	1_0848h
MBC0_DOM4_MEM0_BLK_CFG_W3	1_084Ch
MBC0_DOM4_MEM0_BLK_CFG_W4	1_0850h
MBC0_DOM4_MEM0_BLK_CFG_W5	1_0854h
MBC0_DOM4_MEM0_BLK_CFG_W6	1_0858h
MBC0_DOM4_MEM1_BLK_CFG_W0	1_0980h
MBC0_DOM4_MEM1_BLK_CFG_W1	1_0984h
MBC0_DOM4_MEM2_BLK_CFG_W0	1_09A8h
MBC0_DOM4_MEM2_BLK_CFG_W1	1_09ACh

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM4_MEM2_BLK_CFG_W2	1_09B0h
MBC0_DOM4_MEM2_BLK_CFG_W3	1_09B4h
MBC0_DOM4_MEM2_BLK_CFG_W4	1_09B8h
MBC0_DOM4_MEM3_BLK_CFG_W0	1_09D0h
MBC0_DOM4_MEM3_BLK_CFG_W1	1_09D4h
MBC0_DOM4_MEM3_BLK_CFG_W2	1_09D8h
MBC0_DOM4_MEM3_BLK_CFG_W3	1_09DCh
MBC0_DOM4_MEM3_BLK_CFG_W4	1_09E0h
MBC0_DOM4_MEM3_BLK_CFG_W5	1_09E4h
MBC0_DOM5_MEM0_BLK_CFG_W0	1_0A40h
MBC0_DOM5_MEM0_BLK_CFG_W1	1_0A44h
MBC0_DOM5_MEM0_BLK_CFG_W2	1_0A48h
MBC0_DOM5_MEM0_BLK_CFG_W3	1_0A4Ch
MBC0_DOM5_MEM0_BLK_CFG_W4	1_0A50h
MBC0_DOM5_MEM0_BLK_CFG_W5	1_0A54h
MBC0_DOM5_MEM0_BLK_CFG_W6	1_0A58h
MBC0_DOM5_MEM1_BLK_CFG_W0	1_0B80h
MBC0_DOM5_MEM1_BLK_CFG_W1	1_0B84h
MBC0_DOM5_MEM2_BLK_CFG_W0	1_0BA8h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM5_MEM2_BLK_CFG_W1	1_0BACh
MBC0_DOM5_MEM2_BLK_CFG_W2	1_0BB0h
MBC0_DOM5_MEM2_BLK_CFG_W3	1_0BB4h
MBC0_DOM5_MEM2_BLK_CFG_W4	1_0BB8h
MBC0_DOM5_MEM3_BLK_CFG_W0	1_0BD0h
MBC0_DOM5_MEM3_BLK_CFG_W1	1_0BD4h
MBC0_DOM5_MEM3_BLK_CFG_W2	1_0BD8h
MBC0_DOM5_MEM3_BLK_CFG_W3	1_0BDCh
MBC0_DOM5_MEM3_BLK_CFG_W4	1_0BE0h
MBC0_DOM5_MEM3_BLK_CFG_W5	1_0BE4h
MBC0_DOM6_MEM0_BLK_CFG_W0	1_0C40h
MBC0_DOM6_MEM0_BLK_CFG_W1	1_0C44h
MBC0_DOM6_MEM0_BLK_CFG_W2	1_0C48h
MBC0_DOM6_MEM0_BLK_CFG_W3	1_0C4Ch
MBC0_DOM6_MEM0_BLK_CFG_W4	1_0C50h
MBC0_DOM6_MEM0_BLK_CFG_W5	1_0C54h
MBC0_DOM6_MEM0_BLK_CFG_W6	1_0C58h
MBC0_DOM6_MEM1_BLK_CFG_W0	1_0D80h
MBC0_DOM6_MEM1_BLK_CFG_W1	1_0D84h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM6_MEM2_BLK_CFG_W0	1_0DA8h
MBC0_DOM6_MEM2_BLK_CFG_W1	1_0DACh
MBC0_DOM6_MEM2_BLK_CFG_W2	1_0DB0h
MBC0_DOM6_MEM2_BLK_CFG_W3	1_0DB4h
MBC0_DOM6_MEM2_BLK_CFG_W4	1_0DB8h
MBC0_DOM6_MEM3_BLK_CFG_W0	1_0DD0h
MBC0_DOM6_MEM3_BLK_CFG_W1	1_0DD4h
MBC0_DOM6_MEM3_BLK_CFG_W2	1_0DD8h
MBC0_DOM6_MEM3_BLK_CFG_W3	1_0DDCh
MBC0_DOM6_MEM3_BLK_CFG_W4	1_0DE0h
MBC0_DOM6_MEM3_BLK_CFG_W5	1_0DE4h
MBC0_DOM7_MEM0_BLK_CFG_W0	1_0E40h
MBC0_DOM7_MEM0_BLK_CFG_W1	1_0E44h
MBC0_DOM7_MEM0_BLK_CFG_W2	1_0E48h
MBC0_DOM7_MEM0_BLK_CFG_W3	1_0E4Ch
MBC0_DOM7_MEM0_BLK_CFG_W4	1_0E50h
MBC0_DOM7_MEM0_BLK_CFG_W5	1_0E54h
MBC0_DOM7_MEM0_BLK_CFG_W6	1_0E58h
MBC0_DOM7_MEM1_BLK_CFG_W0	1_0F80h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM7_MEM1_BLK_CFG_W1	1_0F84h
MBC0_DOM7_MEM2_BLK_CFG_W0	1_0FA8h
MBC0_DOM7_MEM2_BLK_CFG_W1	1_0FACh
MBC0_DOM7_MEM2_BLK_CFG_W2	1_0FB0h
MBC0_DOM7_MEM2_BLK_CFG_W3	1_0FB4h
MBC0_DOM7_MEM2_BLK_CFG_W4	1_0FB8h
MBC0_DOM7_MEM3_BLK_CFG_W0	1_0FD0h
MBC0_DOM7_MEM3_BLK_CFG_W1	1_0FD4h
MBC0_DOM7_MEM3_BLK_CFG_W2	1_0FD8h
MBC0_DOM7_MEM3_BLK_CFG_W3	1_0FDCh
MBC0_DOM7_MEM3_BLK_CFG_W4	1_0FE0h
MBC0_DOM7_MEM3_BLK_CFG_W5	1_0FE4h
MBC0_DOM8_MEM0_BLK_CFG_W0	1_1040h
MBC0_DOM8_MEM0_BLK_CFG_W1	1_1044h
MBC0_DOM8_MEM0_BLK_CFG_W2	1_1048h
MBC0_DOM8_MEM0_BLK_CFG_W3	1_104Ch
MBC0_DOM8_MEM0_BLK_CFG_W4	1_1050h
MBC0_DOM8_MEM0_BLK_CFG_W5	1_1054h
MBC0_DOM8_MEM0_BLK_CFG_W6	1_1058h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM8_MEM1_BLK_CFG_W0	1_1180h
MBC0_DOM8_MEM1_BLK_CFG_W1	1_1184h
MBC0_DOM8_MEM2_BLK_CFG_W0	1_11A8h
MBC0_DOM8_MEM2_BLK_CFG_W1	1_11ACh
MBC0_DOM8_MEM2_BLK_CFG_W2	1_11B0h
MBC0_DOM8_MEM2_BLK_CFG_W3	1_11B4h
MBC0_DOM8_MEM2_BLK_CFG_W4	1_11B8h
MBC0_DOM8_MEM3_BLK_CFG_W0	1_11D0h
MBC0_DOM8_MEM3_BLK_CFG_W1	1_11D4h
MBC0_DOM8_MEM3_BLK_CFG_W2	1_11D8h
MBC0_DOM8_MEM3_BLK_CFG_W3	1_11DCh
MBC0_DOM8_MEM3_BLK_CFG_W4	1_11E0h
MBC0_DOM8_MEM3_BLK_CFG_W5	1_11E4h
MBC0_DOM9_MEM0_BLK_CFG_W0	1_1240h
MBC0_DOM9_MEM0_BLK_CFG_W1	1_1244h
MBC0_DOM9_MEM0_BLK_CFG_W2	1_1248h
MBC0_DOM9_MEM0_BLK_CFG_W3	1_124Ch
MBC0_DOM9_MEM0_BLK_CFG_W4	1_1250h
MBC0_DOM9_MEM0_BLK_CFG_W5	1_1254h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM9_MEM0_BLK_CFG_W6	1_1258h
MBC0_DOM9_MEM1_BLK_CFG_W0	1_1380h
MBC0_DOM9_MEM1_BLK_CFG_W1	1_1384h
MBC0_DOM9_MEM2_BLK_CFG_W0	1_13A8h
MBC0_DOM9_MEM2_BLK_CFG_W1	1_13ACh
MBC0_DOM9_MEM2_BLK_CFG_W2	1_13B0h
MBC0_DOM9_MEM2_BLK_CFG_W3	1_13B4h
MBC0_DOM9_MEM2_BLK_CFG_W4	1_13B8h
MBC0_DOM9_MEM3_BLK_CFG_W0	1_13D0h
MBC0_DOM9_MEM3_BLK_CFG_W1	1_13D4h
MBC0_DOM9_MEM3_BLK_CFG_W2	1_13D8h
MBC0_DOM9_MEM3_BLK_CFG_W3	1_13DCh
MBC0_DOM9_MEM3_BLK_CFG_W4	1_13E0h
MBC0_DOM9_MEM3_BLK_CFG_W5	1_13E4h
MBC0_DOM10_MEM0_BLK_CFG_W0	1_1440h
MBC0_DOM10_MEM0_BLK_CFG_W1	1_1444h
MBC0_DOM10_MEM0_BLK_CFG_W2	1_1448h
MBC0_DOM10_MEM0_BLK_CFG_W3	1_144Ch
MBC0_DOM10_MEM0_BLK_CFG_W4	1_1450h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM10_MEM0_BLK_CFG_W5	1_1454h
MBC0_DOM10_MEM0_BLK_CFG_W6	1_1458h
MBC0_DOM10_MEM1_BLK_CFG_W0	1_1580h
MBC0_DOM10_MEM1_BLK_CFG_W1	1_1584h
MBC0_DOM10_MEM2_BLK_CFG_W0	1_15A8h
MBC0_DOM10_MEM2_BLK_CFG_W1	1_15ACh
MBC0_DOM10_MEM2_BLK_CFG_W2	1_15B0h
MBC0_DOM10_MEM2_BLK_CFG_W3	1_15B4h
MBC0_DOM10_MEM2_BLK_CFG_W4	1_15B8h
MBC0_DOM10_MEM3_BLK_CFG_W0	1_15D0h
MBC0_DOM10_MEM3_BLK_CFG_W1	1_15D4h
MBC0_DOM10_MEM3_BLK_CFG_W2	1_15D8h
MBC0_DOM10_MEM3_BLK_CFG_W3	1_15DCh
MBC0_DOM10_MEM3_BLK_CFG_W4	1_15E0h
MBC0_DOM10_MEM3_BLK_CFG_W5	1_15E4h
MBC0_DOM11_MEM0_BLK_CFG_W0	1_1640h
MBC0_DOM11_MEM0_BLK_CFG_W1	1_1644h
MBC0_DOM11_MEM0_BLK_CFG_W2	1_1648h
MBC0_DOM11_MEM0_BLK_CFG_W3	1_164Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM11_MEM0_BLK_CFG_W4	1_1650h
MBC0_DOM11_MEM0_BLK_CFG_W5	1_1654h
MBC0_DOM11_MEM0_BLK_CFG_W6	1_1658h
MBC0_DOM11_MEM1_BLK_CFG_W0	1_1780h
MBC0_DOM11_MEM1_BLK_CFG_W1	1_1784h
MBC0_DOM11_MEM2_BLK_CFG_W0	1_17A8h
MBC0_DOM11_MEM2_BLK_CFG_W1	1_17ACh
MBC0_DOM11_MEM2_BLK_CFG_W2	1_17B0h
MBC0_DOM11_MEM2_BLK_CFG_W3	1_17B4h
MBC0_DOM11_MEM2_BLK_CFG_W4	1_17B8h
MBC0_DOM11_MEM3_BLK_CFG_W0	1_17D0h
MBC0_DOM11_MEM3_BLK_CFG_W1	1_17D4h
MBC0_DOM11_MEM3_BLK_CFG_W2	1_17D8h
MBC0_DOM11_MEM3_BLK_CFG_W3	1_17DCh
MBC0_DOM11_MEM3_BLK_CFG_W4	1_17E0h
MBC0_DOM11_MEM3_BLK_CFG_W5	1_17E4h
MBC0_DOM12_MEM0_BLK_CFG_W0	1_1840h
MBC0_DOM12_MEM0_BLK_CFG_W1	1_1844h
MBC0_DOM12_MEM0_BLK_CFG_W2	1_1848h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM12_MEM0_BLK_CFG_W3	1_184Ch
MBC0_DOM12_MEM0_BLK_CFG_W4	1_1850h
MBC0_DOM12_MEM0_BLK_CFG_W5	1_1854h
MBC0_DOM12_MEM0_BLK_CFG_W6	1_1858h
MBC0_DOM12_MEM1_BLK_CFG_W0	1_1980h
MBC0_DOM12_MEM1_BLK_CFG_W1	1_1984h
MBC0_DOM12_MEM2_BLK_CFG_W0	1_19A8h
MBC0_DOM12_MEM2_BLK_CFG_W1	1_19ACh
MBC0_DOM12_MEM2_BLK_CFG_W2	1_19B0h
MBC0_DOM12_MEM2_BLK_CFG_W3	1_19B4h
MBC0_DOM12_MEM2_BLK_CFG_W4	1_19B8h
MBC0_DOM12_MEM3_BLK_CFG_W0	1_19D0h
MBC0_DOM12_MEM3_BLK_CFG_W1	1_19D4h
MBC0_DOM12_MEM3_BLK_CFG_W2	1_19D8h
MBC0_DOM12_MEM3_BLK_CFG_W3	1_19DCh
MBC0_DOM12_MEM3_BLK_CFG_W4	1_19E0h
MBC0_DOM12_MEM3_BLK_CFG_W5	1_19E4h
MBC0_DOM13_MEM0_BLK_CFG_W0	1_1A40h
MBC0_DOM13_MEM0_BLK_CFG_W1	1_1A44h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM13_MEM0_BLK_CFG_W2	1_1A48h
MBC0_DOM13_MEM0_BLK_CFG_W3	1_1A4Ch
MBC0_DOM13_MEM0_BLK_CFG_W4	1_1A50h
MBC0_DOM13_MEM0_BLK_CFG_W5	1_1A54h
MBC0_DOM13_MEM0_BLK_CFG_W6	1_1A58h
MBC0_DOM13_MEM1_BLK_CFG_W0	1_1B80h
MBC0_DOM13_MEM1_BLK_CFG_W1	1_1B84h
MBC0_DOM13_MEM2_BLK_CFG_W0	1_1BA8h
MBC0_DOM13_MEM2_BLK_CFG_W1	1_1BACh
MBC0_DOM13_MEM2_BLK_CFG_W2	1_1BB0h
MBC0_DOM13_MEM2_BLK_CFG_W3	1_1BB4h
MBC0_DOM13_MEM2_BLK_CFG_W4	1_1BB8h
MBC0_DOM13_MEM3_BLK_CFG_W0	1_1BD0h
MBC0_DOM13_MEM3_BLK_CFG_W1	1_1BD4h
MBC0_DOM13_MEM3_BLK_CFG_W2	1_1BD8h
MBC0_DOM13_MEM3_BLK_CFG_W3	1_1BDCh
MBC0_DOM13_MEM3_BLK_CFG_W4	1_1BE0h
MBC0_DOM13_MEM3_BLK_CFG_W5	1_1BE4h
MBC0_DOM14_MEM0_BLK_CFG_W0	1_1C40h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM14_MEM0_BLK_CFG_W1	1_1C44h
MBC0_DOM14_MEM0_BLK_CFG_W2	1_1C48h
MBC0_DOM14_MEM0_BLK_CFG_W3	1_1C4Ch
MBC0_DOM14_MEM0_BLK_CFG_W4	1_1C50h
MBC0_DOM14_MEM0_BLK_CFG_W5	1_1C54h
MBC0_DOM14_MEM0_BLK_CFG_W6	1_1C58h
MBC0_DOM14_MEM1_BLK_CFG_W0	1_1D80h
MBC0_DOM14_MEM1_BLK_CFG_W1	1_1D84h
MBC0_DOM14_MEM2_BLK_CFG_W0	1_1DA8h
MBC0_DOM14_MEM2_BLK_CFG_W1	1_1DACh
MBC0_DOM14_MEM2_BLK_CFG_W2	1_1DB0h
MBC0_DOM14_MEM2_BLK_CFG_W3	1_1DB4h
MBC0_DOM14_MEM2_BLK_CFG_W4	1_1DB8h
MBC0_DOM14_MEM3_BLK_CFG_W0	1_1DD0h
MBC0_DOM14_MEM3_BLK_CFG_W1	1_1DD4h
MBC0_DOM14_MEM3_BLK_CFG_W2	1_1DD8h
MBC0_DOM14_MEM3_BLK_CFG_W3	1_1DDCh
MBC0_DOM14_MEM3_BLK_CFG_W4	1_1DE0h
MBC0_DOM14_MEM3_BLK_CFG_W5	1_1DE4h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM15_MEM0_BLK_CFG_W0	1_1E40h
MBC0_DOM15_MEM0_BLK_CFG_W1	1_1E44h
MBC0_DOM15_MEM0_BLK_CFG_W2	1_1E48h
MBC0_DOM15_MEM0_BLK_CFG_W3	1_1E4Ch
MBC0_DOM15_MEM0_BLK_CFG_W4	1_1E50h
MBC0_DOM15_MEM0_BLK_CFG_W5	1_1E54h
MBC0_DOM15_MEM0_BLK_CFG_W6	1_1E58h
MBC0_DOM15_MEM1_BLK_CFG_W0	1_1F80h
MBC0_DOM15_MEM1_BLK_CFG_W1	1_1F84h
MBC0_DOM15_MEM2_BLK_CFG_W0	1_1FA8h
MBC0_DOM15_MEM2_BLK_CFG_W1	1_1FACh
MBC0_DOM15_MEM2_BLK_CFG_W2	1_1FB0h
MBC0_DOM15_MEM2_BLK_CFG_W3	1_1FB4h
MBC0_DOM15_MEM2_BLK_CFG_W4	1_1FB8h
MBC0_DOM15_MEM3_BLK_CFG_W0	1_1FD0h
MBC0_DOM15_MEM3_BLK_CFG_W1	1_1FD4h
MBC0_DOM15_MEM3_BLK_CFG_W2	1_1FD8h
MBC0_DOM15_MEM3_BLK_CFG_W3	1_1FDCh
MBC0_DOM15_MEM3_BLK_CFG_W4	1_1FE0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM15_MEM3_BLK_CFG_W5	1_1FE4h
MBC1_DOM0_MEM0_BLK_CFG_W0	1_2040h
MBC1_DOM0_MEM1_BLK_CFG_W0	1_2180h
MBC1_DOM0_MEM2_BLK_CFG_W0	1_21A8h
MBC1_DOM0_MEM2_BLK_CFG_W1	1_21ACh
MBC1_DOM0_MEM2_BLK_CFG_W2	1_21B0h
MBC1_DOM0_MEM3_BLK_CFG_W0	1_21D0h
MBC1_DOM0_MEM3_BLK_CFG_W1	1_21D4h
MBC1_DOM0_MEM3_BLK_CFG_W2	1_21D8h
MBC1_DOM1_MEM0_BLK_CFG_W0	1_2240h
MBC1_DOM1_MEM1_BLK_CFG_W0	1_2380h
MBC1_DOM1_MEM2_BLK_CFG_W0	1_23A8h
MBC1_DOM1_MEM2_BLK_CFG_W1	1_23ACh
MBC1_DOM1_MEM2_BLK_CFG_W2	1_23B0h
MBC1_DOM1_MEM3_BLK_CFG_W0	1_23D0h
MBC1_DOM1_MEM3_BLK_CFG_W1	1_23D4h
MBC1_DOM1_MEM3_BLK_CFG_W2	1_23D8h
MBC1_DOM2_MEM0_BLK_CFG_W0	1_2440h
MBC1_DOM2_MEM1_BLK_CFG_W0	1_2580h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM2_MEM2_BLK_CFG_W0	1_25A8h
MBC1_DOM2_MEM2_BLK_CFG_W1	1_25ACh
MBC1_DOM2_MEM2_BLK_CFG_W2	1_25B0h
MBC1_DOM2_MEM3_BLK_CFG_W0	1_25D0h
MBC1_DOM2_MEM3_BLK_CFG_W1	1_25D4h
MBC1_DOM2_MEM3_BLK_CFG_W2	1_25D8h
MBC1_DOM3_MEM0_BLK_CFG_W0	1_2640h
MBC1_DOM3_MEM1_BLK_CFG_W0	1_2780h
MBC1_DOM3_MEM2_BLK_CFG_W0	1_27A8h
MBC1_DOM3_MEM2_BLK_CFG_W1	1_27ACh
MBC1_DOM3_MEM2_BLK_CFG_W2	1_27B0h
MBC1_DOM3_MEM3_BLK_CFG_W0	1_27D0h
MBC1_DOM3_MEM3_BLK_CFG_W1	1_27D4h
MBC1_DOM3_MEM3_BLK_CFG_W2	1_27D8h
MBC1_DOM4_MEM0_BLK_CFG_W0	1_2840h
MBC1_DOM4_MEM1_BLK_CFG_W0	1_2980h
MBC1_DOM4_MEM2_BLK_CFG_W0	1_29A8h
MBC1_DOM4_MEM2_BLK_CFG_W1	1_29ACh
MBC1_DOM4_MEM2_BLK_CFG_W2	1_29B0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM4_MEM3_BLK_CFG_W0	1_29D0h
MBC1_DOM4_MEM3_BLK_CFG_W1	1_29D4h
MBC1_DOM4_MEM3_BLK_CFG_W2	1_29D8h
MBC1_DOM5_MEM0_BLK_CFG_W0	1_2A40h
MBC1_DOM5_MEM1_BLK_CFG_W0	1_2B80h
MBC1_DOM5_MEM2_BLK_CFG_W0	1_2BA8h
MBC1_DOM5_MEM2_BLK_CFG_W1	1_2BACh
MBC1_DOM5_MEM2_BLK_CFG_W2	1_2BB0h
MBC1_DOM5_MEM3_BLK_CFG_W0	1_2BD0h
MBC1_DOM5_MEM3_BLK_CFG_W1	1_2BD4h
MBC1_DOM5_MEM3_BLK_CFG_W2	1_2BD8h
MBC1_DOM6_MEM0_BLK_CFG_W0	1_2C40h
MBC1_DOM6_MEM1_BLK_CFG_W0	1_2D80h
MBC1_DOM6_MEM2_BLK_CFG_W0	1_2DA8h
MBC1_DOM6_MEM2_BLK_CFG_W1	1_2DACh
MBC1_DOM6_MEM2_BLK_CFG_W2	1_2DB0h
MBC1_DOM6_MEM3_BLK_CFG_W0	1_2DD0h
MBC1_DOM6_MEM3_BLK_CFG_W1	1_2DD4h
MBC1_DOM6_MEM3_BLK_CFG_W2	1_2DD8h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM7_MEM0_BLK_CFG_W0	1_2E40h
MBC1_DOM7_MEM1_BLK_CFG_W0	1_2F80h
MBC1_DOM7_MEM2_BLK_CFG_W0	1_2FA8h
MBC1_DOM7_MEM2_BLK_CFG_W1	1_2FACh
MBC1_DOM7_MEM2_BLK_CFG_W2	1_2FB0h
MBC1_DOM7_MEM3_BLK_CFG_W0	1_2FD0h
MBC1_DOM7_MEM3_BLK_CFG_W1	1_2FD4h
MBC1_DOM7_MEM3_BLK_CFG_W2	1_2FD8h
MBC1_DOM8_MEM0_BLK_CFG_W0	1_3040h
MBC1_DOM8_MEM1_BLK_CFG_W0	1_3180h
MBC1_DOM8_MEM2_BLK_CFG_W0	1_31A8h
MBC1_DOM8_MEM2_BLK_CFG_W1	1_31ACh
MBC1_DOM8_MEM2_BLK_CFG_W2	1_31B0h
MBC1_DOM8_MEM3_BLK_CFG_W0	1_31D0h
MBC1_DOM8_MEM3_BLK_CFG_W1	1_31D4h
MBC1_DOM8_MEM3_BLK_CFG_W2	1_31D8h
MBC1_DOM9_MEM0_BLK_CFG_W0	1_3240h
MBC1_DOM9_MEM1_BLK_CFG_W0	1_3380h
MBC1_DOM9_MEM2_BLK_CFG_W0	1_33A8h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM9_MEM2_BLK_CFG_W1	1_33ACh
MBC1_DOM9_MEM2_BLK_CFG_W2	1_33B0h
MBC1_DOM9_MEM3_BLK_CFG_W0	1_33D0h
MBC1_DOM9_MEM3_BLK_CFG_W1	1_33D4h
MBC1_DOM9_MEM3_BLK_CFG_W2	1_33D8h
MBC1_DOM10_MEM0_BLK_CFG_W0	1_3440h
MBC1_DOM10_MEM1_BLK_CFG_W0	1_3580h
MBC1_DOM10_MEM2_BLK_CFG_W0	1_35A8h
MBC1_DOM10_MEM2_BLK_CFG_W1	1_35ACh
MBC1_DOM10_MEM2_BLK_CFG_W2	1_35B0h
MBC1_DOM10_MEM3_BLK_CFG_W0	1_35D0h
MBC1_DOM10_MEM3_BLK_CFG_W1	1_35D4h
MBC1_DOM10_MEM3_BLK_CFG_W2	1_35D8h
MBC1_DOM11_MEM0_BLK_CFG_W0	1_3640h
MBC1_DOM11_MEM1_BLK_CFG_W0	1_3780h
MBC1_DOM11_MEM2_BLK_CFG_W0	1_37A8h
MBC1_DOM11_MEM2_BLK_CFG_W1	1_37ACh
MBC1_DOM11_MEM2_BLK_CFG_W2	1_37B0h
MBC1_DOM11_MEM3_BLK_CFG_W0	1_37D0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM11_MEM3_BLK_CFG_W1	1_37D4h
MBC1_DOM11_MEM3_BLK_CFG_W2	1_37D8h
MBC1_DOM12_MEM0_BLK_CFG_W0	1_3840h
MBC1_DOM12_MEM1_BLK_CFG_W0	1_3980h
MBC1_DOM12_MEM2_BLK_CFG_W0	1_39A8h
MBC1_DOM12_MEM2_BLK_CFG_W1	1_39ACh
MBC1_DOM12_MEM2_BLK_CFG_W2	1_39B0h
MBC1_DOM12_MEM3_BLK_CFG_W0	1_39D0h
MBC1_DOM12_MEM3_BLK_CFG_W1	1_39D4h
MBC1_DOM12_MEM3_BLK_CFG_W2	1_39D8h
MBC1_DOM13_MEM0_BLK_CFG_W0	1_3A40h
MBC1_DOM13_MEM1_BLK_CFG_W0	1_3B80h
MBC1_DOM13_MEM2_BLK_CFG_W0	1_3BA8h
MBC1_DOM13_MEM2_BLK_CFG_W1	1_3BACh
MBC1_DOM13_MEM2_BLK_CFG_W2	1_3BB0h
MBC1_DOM13_MEM3_BLK_CFG_W0	1_3BD0h
MBC1_DOM13_MEM3_BLK_CFG_W1	1_3BD4h
MBC1_DOM13_MEM3_BLK_CFG_W2	1_3BD8h
MBC1_DOM14_MEM0_BLK_CFG_W0	1_3C40h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM14_MEM1_BLK_CFG_W0	1_3D80h
MBC1_DOM14_MEM2_BLK_CFG_W0	1_3DA8h
MBC1_DOM14_MEM2_BLK_CFG_W1	1_3DACh
MBC1_DOM14_MEM2_BLK_CFG_W2	1_3DB0h
MBC1_DOM14_MEM3_BLK_CFG_W0	1_3DD0h
MBC1_DOM14_MEM3_BLK_CFG_W1	1_3DD4h
MBC1_DOM14_MEM3_BLK_CFG_W2	1_3DD8h
MBC1_DOM15_MEM0_BLK_CFG_W0	1_3E40h
MBC1_DOM15_MEM1_BLK_CFG_W0	1_3F80h
MBC1_DOM15_MEM2_BLK_CFG_W0	1_3FA8h
MBC1_DOM15_MEM2_BLK_CFG_W1	1_3FACh
MBC1_DOM15_MEM2_BLK_CFG_W2	1_3FB0h
MBC1_DOM15_MEM3_BLK_CFG_W0	1_3FD0h
MBC1_DOM15_MEM3_BLK_CFG_W1	1_3FD4h
MBC1_DOM15_MEM3_BLK_CFG_W2	1_3FD8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NSE7	MBACSEL7			NSE6	MBACSEL6			NSE5	MBACSEL5			NSE4	MBACSEL4		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NSE3	MBACSEL3			NSE2	MBACSEL2			NSE1	MBACSEL1			NSE0	MBACSEL0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 NSE7	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
30-28 MBACSEL7	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL7 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
27 NSE6	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
26-24	Memory Block Access Control Select for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
MBACSEL6	This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL6 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
23 NSE5	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
22-20 MBACSEL5	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL5 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
19 NSE4	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
18-16 MBACSEL4	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL4 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
15 NSE3	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
14-12 MBACSEL3	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL3 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
11	NonSecure Enable for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
NSE2	0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
10-8 MBACSEL2	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL2 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
7 NSE1	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
6-4 MBACSEL1	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL1 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
	111 - select MBC_MEMN_GLBAC7 access control policy for block B
3 NSE0	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
2-0 MBACSEL0	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL0 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B

22.7.5.31 MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W0 - MBC1_DOM15_MEM3_BLK_NSE_W0)

MBC[m]_DOM[d]_MEM[s]_BLK_NSE_W[w], where,

- m - mbc index
- d - domain index
- s - memory target index
- w - word index

This is the bitmap of the individual NSE bits for the memory block.

NOTE

This register is shown with fields such that it covers entire 32 bit register. However, actual fields may be less if number of blocks are such that fields don't align to 32 bits. These absent fields may be shown but user can refer to table below to deduce actual fields and treat absent fields as reserved.

The following table describes the relationship between W (word index) and B (block number).

Table 113. Block Config Word to Block Number Relationship

Block NSE Word	BIT31	BIT30-BIT1	BIT0
W0	block 31	block 30 - block 1	block 0
W1	block 63	block 62 - block 33	block 32
For MEM0, block 64 - block 511			
W2	block 95	block 94 - block 65	block 64
. . .			
W15	block 511	block 510 - block 481	block 480

The NSE bitmap can be accessed directly by register reads and writes to these locations, or "indirectly" through writes to NSE_SET, NSE_CLR, NSE_CLR_ALL, and BLK_CFG[NSEn] registers.

Offset

Register	Offset
MBC0_DOM0_MEM0_BLK_NSE_W0	1_0140h
MBC0_DOM0_MEM0_BLK_NSE_W1	1_0144h
MBC0_DOM0_MEM1_BLK_NSE_W0	1_01A0h
MBC0_DOM0_MEM2_BLK_NSE_W0	1_01C8h
MBC0_DOM0_MEM2_BLK_NSE_W1	1_01CCh
MBC0_DOM0_MEM3_BLK_NSE_W0	1_01F0h
MBC0_DOM0_MEM3_BLK_NSE_W1	1_01F4h
MBC0_DOM1_MEM0_BLK_NSE_W0	1_0340h
MBC0_DOM1_MEM0_BLK_NSE_W1	1_0344h
MBC0_DOM1_MEM1_BLK_NSE_W0	1_03A0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM1_MEM2_BLK_NSE_W0	1_03C8h
MBC0_DOM1_MEM2_BLK_NSE_W1	1_03CCh
MBC0_DOM1_MEM3_BLK_NSE_W0	1_03F0h
MBC0_DOM1_MEM3_BLK_NSE_W1	1_03F4h
MBC0_DOM2_MEM0_BLK_NSE_W0	1_0540h
MBC0_DOM2_MEM0_BLK_NSE_W1	1_0544h
MBC0_DOM2_MEM1_BLK_NSE_W0	1_05A0h
MBC0_DOM2_MEM2_BLK_NSE_W0	1_05C8h
MBC0_DOM2_MEM2_BLK_NSE_W1	1_05CCh
MBC0_DOM2_MEM3_BLK_NSE_W0	1_05F0h
MBC0_DOM2_MEM3_BLK_NSE_W1	1_05F4h
MBC0_DOM3_MEM0_BLK_NSE_W0	1_0740h
MBC0_DOM3_MEM0_BLK_NSE_W1	1_0744h
MBC0_DOM3_MEM1_BLK_NSE_W0	1_07A0h
MBC0_DOM3_MEM2_BLK_NSE_W0	1_07C8h
MBC0_DOM3_MEM2_BLK_NSE_W1	1_07CCh
MBC0_DOM3_MEM3_BLK_NSE_W0	1_07F0h
MBC0_DOM3_MEM3_BLK_NSE_W1	1_07F4h
MBC0_DOM4_MEM0_BLK_NSE_W0	1_0940h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM4_MEM0_BLK_NSE_W1	1_0944h
MBC0_DOM4_MEM1_BLK_NSE_W0	1_09A0h
MBC0_DOM4_MEM2_BLK_NSE_W0	1_09C8h
MBC0_DOM4_MEM2_BLK_NSE_W1	1_09CCh
MBC0_DOM4_MEM3_BLK_NSE_W0	1_09F0h
MBC0_DOM4_MEM3_BLK_NSE_W1	1_09F4h
MBC0_DOM5_MEM0_BLK_NSE_W0	1_0B40h
MBC0_DOM5_MEM0_BLK_NSE_W1	1_0B44h
MBC0_DOM5_MEM1_BLK_NSE_W0	1_0BA0h
MBC0_DOM5_MEM2_BLK_NSE_W0	1_0BC8h
MBC0_DOM5_MEM2_BLK_NSE_W1	1_0BCCh
MBC0_DOM5_MEM3_BLK_NSE_W0	1_0BF0h
MBC0_DOM5_MEM3_BLK_NSE_W1	1_0BF4h
MBC0_DOM6_MEM0_BLK_NSE_W0	1_0D40h
MBC0_DOM6_MEM0_BLK_NSE_W1	1_0D44h
MBC0_DOM6_MEM1_BLK_NSE_W0	1_0DA0h
MBC0_DOM6_MEM2_BLK_NSE_W0	1_0DC8h
MBC0_DOM6_MEM2_BLK_NSE_W1	1_0DCCh
MBC0_DOM6_MEM3_BLK_NSE_W0	1_0DF0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM6_MEM3_BLK_NSE_W1	1_0DF4h
MBC0_DOM7_MEM0_BLK_NSE_W0	1_0F40h
MBC0_DOM7_MEM0_BLK_NSE_W1	1_0F44h
MBC0_DOM7_MEM1_BLK_NSE_W0	1_0FA0h
MBC0_DOM7_MEM2_BLK_NSE_W0	1_0FC8h
MBC0_DOM7_MEM2_BLK_NSE_W1	1_0FCCh
MBC0_DOM7_MEM3_BLK_NSE_W0	1_0FF0h
MBC0_DOM7_MEM3_BLK_NSE_W1	1_0FF4h
MBC0_DOM8_MEM0_BLK_NSE_W0	1_1140h
MBC0_DOM8_MEM0_BLK_NSE_W1	1_1144h
MBC0_DOM8_MEM1_BLK_NSE_W0	1_11A0h
MBC0_DOM8_MEM2_BLK_NSE_W0	1_11C8h
MBC0_DOM8_MEM2_BLK_NSE_W1	1_11CCh
MBC0_DOM8_MEM3_BLK_NSE_W0	1_11F0h
MBC0_DOM8_MEM3_BLK_NSE_W1	1_11F4h
MBC0_DOM9_MEM0_BLK_NSE_W0	1_1340h
MBC0_DOM9_MEM0_BLK_NSE_W1	1_1344h
MBC0_DOM9_MEM1_BLK_NSE_W0	1_13A0h
MBC0_DOM9_MEM2_BLK_NSE_W0	1_13C8h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM9_MEM2_BLK_NSE_W1	1_13CCh
MBC0_DOM9_MEM3_BLK_NSE_W0	1_13F0h
MBC0_DOM9_MEM3_BLK_NSE_W1	1_13F4h
MBC0_DOM10_MEM0_BLK_NSE_W0	1_1540h
MBC0_DOM10_MEM0_BLK_NSE_W1	1_1544h
MBC0_DOM10_MEM1_BLK_NSE_W0	1_15A0h
MBC0_DOM10_MEM2_BLK_NSE_W0	1_15C8h
MBC0_DOM10_MEM2_BLK_NSE_W1	1_15CCh
MBC0_DOM10_MEM3_BLK_NSE_W0	1_15F0h
MBC0_DOM10_MEM3_BLK_NSE_W1	1_15F4h
MBC0_DOM11_MEM0_BLK_NSE_W0	1_1740h
MBC0_DOM11_MEM0_BLK_NSE_W1	1_1744h
MBC0_DOM11_MEM1_BLK_NSE_W0	1_17A0h
MBC0_DOM11_MEM2_BLK_NSE_W0	1_17C8h
MBC0_DOM11_MEM2_BLK_NSE_W1	1_17CCh
MBC0_DOM11_MEM3_BLK_NSE_W0	1_17F0h
MBC0_DOM11_MEM3_BLK_NSE_W1	1_17F4h
MBC0_DOM12_MEM0_BLK_NSE_W0	1_1940h
MBC0_DOM12_MEM0_BLK_NSE_W1	1_1944h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM12_MEM1_BLK_NSE_W0	1_19A0h
MBC0_DOM12_MEM2_BLK_NSE_W0	1_19C8h
MBC0_DOM12_MEM2_BLK_NSE_W1	1_19CCh
MBC0_DOM12_MEM3_BLK_NSE_W0	1_19F0h
MBC0_DOM12_MEM3_BLK_NSE_W1	1_19F4h
MBC0_DOM13_MEM0_BLK_NSE_W0	1_1B40h
MBC0_DOM13_MEM0_BLK_NSE_W1	1_1B44h
MBC0_DOM13_MEM1_BLK_NSE_W0	1_1BA0h
MBC0_DOM13_MEM2_BLK_NSE_W0	1_1BC8h
MBC0_DOM13_MEM2_BLK_NSE_W1	1_1BCCh
MBC0_DOM13_MEM3_BLK_NSE_W0	1_1BF0h
MBC0_DOM13_MEM3_BLK_NSE_W1	1_1BF4h
MBC0_DOM14_MEM0_BLK_NSE_W0	1_1D40h
MBC0_DOM14_MEM0_BLK_NSE_W1	1_1D44h
MBC0_DOM14_MEM1_BLK_NSE_W0	1_1DA0h
MBC0_DOM14_MEM2_BLK_NSE_W0	1_1DC8h
MBC0_DOM14_MEM2_BLK_NSE_W1	1_1DCCh
MBC0_DOM14_MEM3_BLK_NSE_W0	1_1DF0h
MBC0_DOM14_MEM3_BLK_NSE_W1	1_1DF4h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM15_MEM0_BLK_NSE_W0	1_1F40h
MBC0_DOM15_MEM0_BLK_NSE_W1	1_1F44h
MBC0_DOM15_MEM1_BLK_NSE_W0	1_1FA0h
MBC0_DOM15_MEM2_BLK_NSE_W0	1_1FC8h
MBC0_DOM15_MEM2_BLK_NSE_W1	1_1FCCh
MBC0_DOM15_MEM3_BLK_NSE_W0	1_1FF0h
MBC0_DOM15_MEM3_BLK_NSE_W1	1_1FF4h
MBC1_DOM0_MEM0_BLK_NSE_W0	1_2140h
MBC1_DOM0_MEM1_BLK_NSE_W0	1_21A0h
MBC1_DOM0_MEM2_BLK_NSE_W0	1_21C8h
MBC1_DOM0_MEM3_BLK_NSE_W0	1_21F0h
MBC1_DOM1_MEM0_BLK_NSE_W0	1_2340h
MBC1_DOM1_MEM1_BLK_NSE_W0	1_23A0h
MBC1_DOM1_MEM2_BLK_NSE_W0	1_23C8h
MBC1_DOM1_MEM3_BLK_NSE_W0	1_23F0h
MBC1_DOM2_MEM0_BLK_NSE_W0	1_2540h
MBC1_DOM2_MEM1_BLK_NSE_W0	1_25A0h
MBC1_DOM2_MEM2_BLK_NSE_W0	1_25C8h
MBC1_DOM2_MEM3_BLK_NSE_W0	1_25F0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM3_MEM0_BLK_NSE_W0	1_2740h
MBC1_DOM3_MEM1_BLK_NSE_W0	1_27A0h
MBC1_DOM3_MEM2_BLK_NSE_W0	1_27C8h
MBC1_DOM3_MEM3_BLK_NSE_W0	1_27F0h
MBC1_DOM4_MEM0_BLK_NSE_W0	1_2940h
MBC1_DOM4_MEM1_BLK_NSE_W0	1_29A0h
MBC1_DOM4_MEM2_BLK_NSE_W0	1_29C8h
MBC1_DOM4_MEM3_BLK_NSE_W0	1_29F0h
MBC1_DOM5_MEM0_BLK_NSE_W0	1_2B40h
MBC1_DOM5_MEM1_BLK_NSE_W0	1_2BA0h
MBC1_DOM5_MEM2_BLK_NSE_W0	1_2BC8h
MBC1_DOM5_MEM3_BLK_NSE_W0	1_2BF0h
MBC1_DOM6_MEM0_BLK_NSE_W0	1_2D40h
MBC1_DOM6_MEM1_BLK_NSE_W0	1_2DA0h
MBC1_DOM6_MEM2_BLK_NSE_W0	1_2DC8h
MBC1_DOM6_MEM3_BLK_NSE_W0	1_2DF0h
MBC1_DOM7_MEM0_BLK_NSE_W0	1_2F40h
MBC1_DOM7_MEM1_BLK_NSE_W0	1_2FA0h
MBC1_DOM7_MEM2_BLK_NSE_W0	1_2FC8h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM7_MEM3_BLK_NSE_W0	1_2FF0h
MBC1_DOM8_MEM0_BLK_NSE_W0	1_3140h
MBC1_DOM8_MEM1_BLK_NSE_W0	1_31A0h
MBC1_DOM8_MEM2_BLK_NSE_W0	1_31C8h
MBC1_DOM8_MEM3_BLK_NSE_W0	1_31F0h
MBC1_DOM9_MEM0_BLK_NSE_W0	1_3340h
MBC1_DOM9_MEM1_BLK_NSE_W0	1_33A0h
MBC1_DOM9_MEM2_BLK_NSE_W0	1_33C8h
MBC1_DOM9_MEM3_BLK_NSE_W0	1_33F0h
MBC1_DOM10_MEM0_BLK_NSE_W0	1_3540h
MBC1_DOM10_MEM1_BLK_NSE_W0	1_35A0h
MBC1_DOM10_MEM2_BLK_NSE_W0	1_35C8h
MBC1_DOM10_MEM3_BLK_NSE_W0	1_35F0h
MBC1_DOM11_MEM0_BLK_NSE_W0	1_3740h
MBC1_DOM11_MEM1_BLK_NSE_W0	1_37A0h
MBC1_DOM11_MEM2_BLK_NSE_W0	1_37C8h
MBC1_DOM11_MEM3_BLK_NSE_W0	1_37F0h
MBC1_DOM12_MEM0_BLK_NSE_W0	1_3940h
MBC1_DOM12_MEM1_BLK_NSE_W0	1_39A0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM12_MEM2_BLK_NSE_W0	1_39C8h
MBC1_DOM12_MEM3_BLK_NSE_W0	1_39F0h
MBC1_DOM13_MEM0_BLK_NSE_W0	1_3B40h
MBC1_DOM13_MEM1_BLK_NSE_W0	1_3BA0h
MBC1_DOM13_MEM2_BLK_NSE_W0	1_3BC8h
MBC1_DOM13_MEM3_BLK_NSE_W0	1_3BF0h
MBC1_DOM14_MEM0_BLK_NSE_W0	1_3D40h
MBC1_DOM14_MEM1_BLK_NSE_W0	1_3DA0h
MBC1_DOM14_MEM2_BLK_NSE_W0	1_3DC8h
MBC1_DOM14_MEM3_BLK_NSE_W0	1_3DF0h
MBC1_DOM15_MEM0_BLK_NSE_W0	1_3F40h
MBC1_DOM15_MEM1_BLK_NSE_W0	1_3FA0h
MBC1_DOM15_MEM2_BLK_NSE_W0	1_3FC8h
MBC1_DOM15_MEM3_BLK_NSE_W0	1_3FF0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BIT31	BIT30	BIT29	BIT28	BIT27	BIT26	BIT25	BIT24	BIT23	BIT22	BIT21	BIT20	BIT19	BIT18	BIT17	BIT16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BITb	Bit b NonSecure Enable [b = 0 - 31] 0 - Secure read and write to block B are based on corresponding MBACSEL field in register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).

22.7.5.32 MBC Global Access Control (MBC1_MEMN_GLBAC0)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User.

Offset

Register	Offset
MBC1_MEMN_GLBAC0	1_2020h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SPR	SPW	SPX	0	SUR	SUW	SUX	0	NPR	NPW	NPX	0	NUR	NUW	NUX
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

22.7.5.33 MBC Global Access Control (MBC1_MEMN_GLBAC1 - MBC1_MEMN_GLBAC7)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

Offset

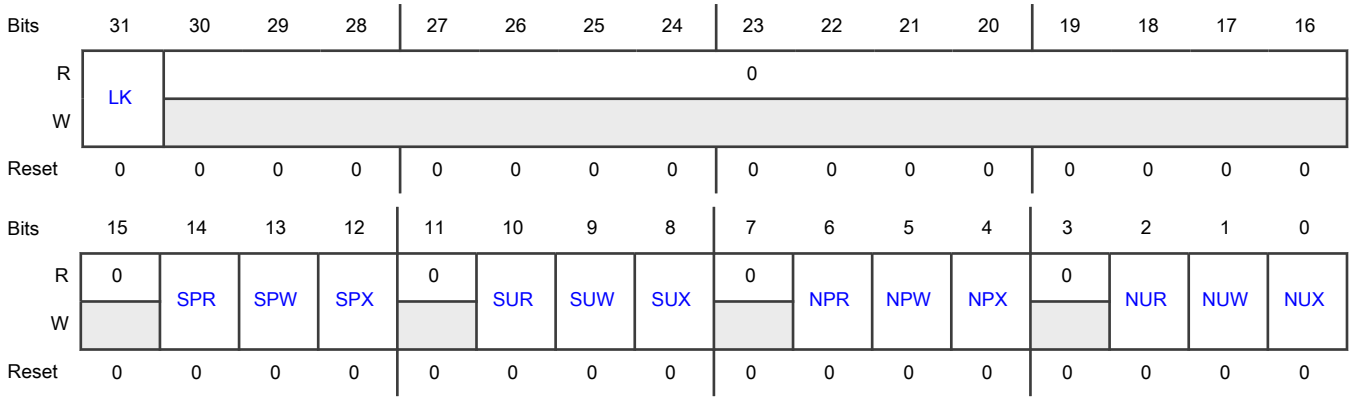
Register	Offset
MBC1_MEMN_GLBAC1	1_2024h
MBC1_MEMN_GLBAC2	1_2028h
MBC1_MEMN_GLBAC3	1_202Ch
MBC1_MEMN_GLBAC4	1_2030h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_MEMN_GLBAC5	1_2034h
MBC1_MEMN_GLBAC6	1_2038h
MBC1_MEMN_GLBAC7	1_203Ch

Diagram



Fields

Field	Description
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0 - This register is not locked and can be altered. 1 - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

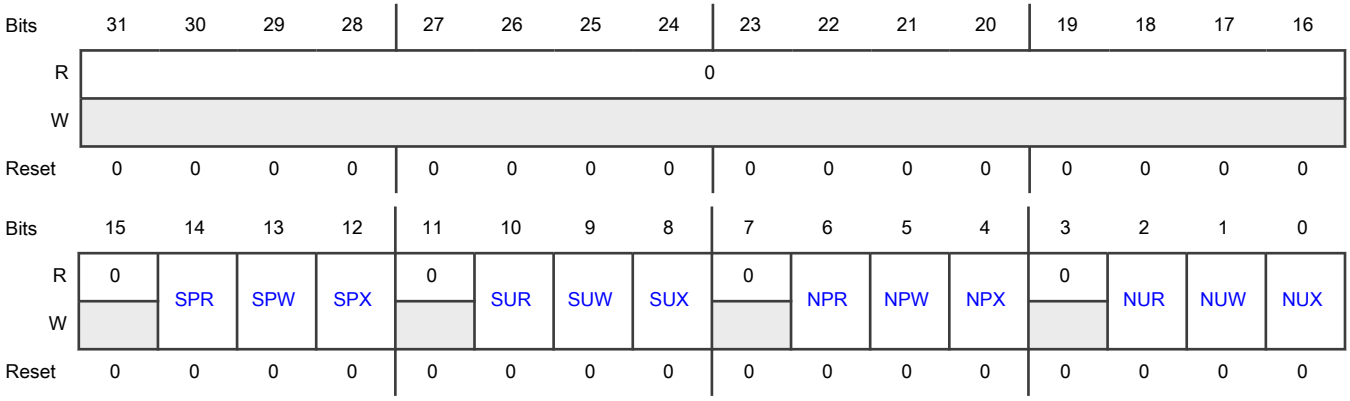
22.7.5.34 MBC Global Access Control (MBC2_MEMN_GLBAC0)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User.

Offset

Register	Offset
MBC2_MEMN_GLBAC0	1_4020h

Diagram



Fields

Field	Description
31-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6	NonsecurePriv Read

Table continues on the next page...

Table continued from the previous page...

Field	Description
NPR	NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

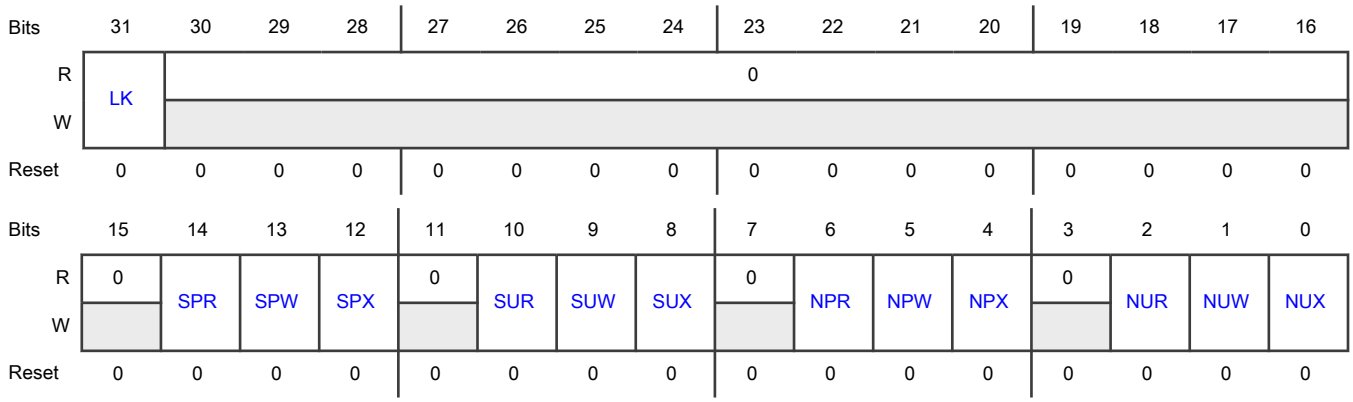
22.7.5.35 MBC Global Access Control (MBC2_MEMN_GLBAC1 - MBC2_MEMN_GLBAC7)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

Offset

Register	Offset
MBC2_MEMN_GLBAC1	1_4024h
MBC2_MEMN_GLBAC2	1_4028h
MBC2_MEMN_GLBAC3	1_402Ch
MBC2_MEMN_GLBAC4	1_4030h
MBC2_MEMN_GLBAC5	1_4034h
MBC2_MEMN_GLBAC6	1_4038h
MBC2_MEMN_GLBAC7	1_403Ch

Diagram



Fields

Field	Description
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0 - This register is not locked and can be altered. 1 - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13	SecurePriv Write

Table continues on the next page...

Table continued from the previous page...

Field	Description
SPW	SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

22.7.5.36 MBC Memory Block Configuration Word (MBC2_DOM0_MEM0_BLK_CFG_W0 - MBC3_DOM15_MEM1_BLK_CFG_W4)

MBC[m]_DOM[d]_MEM[s]_BLK_CFG_W[w], where,

- m - mbc index
- d - domain index
- s - memory target index
- w - word index

These registers are read/write for both the alternate view of the NSE bit and the 3-bit MBACSEL field. This is an array of 4 bit fields defining the block configuration for the given submemory. Each 4-bit field includes an alternative view of the associated NSE bit plus a 3-bit access control select that selects the global access control value that applies to the referenced memory block.

For a given memory block, B, if the value programmed in the MBACSELn field selects a locked MBC_MEMN_GLBACr register, the MBACSEL field is locked until the next reset. Depending on BLK_CFG_W, the NSEn/MBASELn fields correspond to different blocks. The following table describes the relationship between W (word index) and B (block number).

MEM0 supports up to 512 blocks ; W0 - W63

MEM1-3 supports up to 64 block ; W0 - W7

NOTE

This register is shown with fields such that it covers entire 32 bit register. However, actual fields may be less if number of blocks are such that fields don't align to 32 bits. These absent fields may be shown but user can refer to table below to deduce actual fields and treat absent fields as reserved.

Table 114. Block Config Word to Block Number Relationship

Block Config Word	NSE7/ MBACSEL7	NSE6/ MBACSEL6	NSE5/ MBACSEL5	NSE4/ MBACSEL4	NSE3/ MBACSEL3	NSE2/ MBACSEL2	NSE1/ MBACSEL1	NSE0/ MBACSEL0
W0	block 7	block 6	block 5	block 4	block 3	block 2	block 1	block 0
W1	block 15	block 14	block 13	block 12	block 11	block 10	block 9	block 8
W2	block 23	block 22	block 21	block 20	block 19	block 18	block 17	block 16
W3	block 31	block 30	block 29	block 28	block 27	block 26	block 25	block 24
W4	block 39	block 38	block 37	block 36	block 35	block 34	block 33	block 32
W5	block 47	block 46	block 45	block 44	block 43	block 42	block 41	block 40
W6	block 55	block 54	block 53	block 52	block 51	block 50	block 49	block 48
W7	block 63	block 62	block 61	block 60	block 59	block 58	block 57	block 56
For MEM0, block 64 - block 511								
W8	block 71	block 70	block 69	block 68	block 67	block 66	block 65	block 64
. . .								
W63	block 511	block 510	block 509	block 508	block 507	block 506	block 505	block 504

Offset

Register	Offset
MBC2_DOM0_MEM0_BLK_CFG_W0	1_4040h
MBC2_DOM0_MEM0_BLK_CFG_W1	1_4044h
MBC2_DOM0_MEM1_BLK_CFG_W0	1_4180h
MBC2_DOM0_MEM1_BLK_CFG_W1	1_4184h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC2_DOM1_MEM0_BLK_CFG_W0	1_4240h
MBC2_DOM1_MEM0_BLK_CFG_W1	1_4244h
MBC2_DOM1_MEM1_BLK_CFG_W0	1_4380h
MBC2_DOM1_MEM1_BLK_CFG_W1	1_4384h
MBC2_DOM2_MEM0_BLK_CFG_W0	1_4440h
MBC2_DOM2_MEM0_BLK_CFG_W1	1_4444h
MBC2_DOM2_MEM1_BLK_CFG_W0	1_4580h
MBC2_DOM2_MEM1_BLK_CFG_W1	1_4584h
MBC2_DOM3_MEM0_BLK_CFG_W0	1_4640h
MBC2_DOM3_MEM0_BLK_CFG_W1	1_4644h
MBC2_DOM3_MEM1_BLK_CFG_W0	1_4780h
MBC2_DOM3_MEM1_BLK_CFG_W1	1_4784h
MBC2_DOM4_MEM0_BLK_CFG_W0	1_4840h
MBC2_DOM4_MEM0_BLK_CFG_W1	1_4844h
MBC2_DOM4_MEM1_BLK_CFG_W0	1_4980h
MBC2_DOM4_MEM1_BLK_CFG_W1	1_4984h
MBC2_DOM5_MEM0_BLK_CFG_W0	1_4A40h
MBC2_DOM5_MEM0_BLK_CFG_W1	1_4A44h
MBC2_DOM5_MEM1_BLK_CFG_W0	1_4B80h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC2_DOM5_MEM1_BLK_CFG_W1	1_4B84h
MBC2_DOM6_MEM0_BLK_CFG_W0	1_4C40h
MBC2_DOM6_MEM0_BLK_CFG_W1	1_4C44h
MBC2_DOM6_MEM1_BLK_CFG_W0	1_4D80h
MBC2_DOM6_MEM1_BLK_CFG_W1	1_4D84h
MBC2_DOM7_MEM0_BLK_CFG_W0	1_4E40h
MBC2_DOM7_MEM0_BLK_CFG_W1	1_4E44h
MBC2_DOM7_MEM1_BLK_CFG_W0	1_4F80h
MBC2_DOM7_MEM1_BLK_CFG_W1	1_4F84h
MBC2_DOM8_MEM0_BLK_CFG_W0	1_5040h
MBC2_DOM8_MEM0_BLK_CFG_W1	1_5044h
MBC2_DOM8_MEM1_BLK_CFG_W0	1_5180h
MBC2_DOM8_MEM1_BLK_CFG_W1	1_5184h
MBC2_DOM9_MEM0_BLK_CFG_W0	1_5240h
MBC2_DOM9_MEM0_BLK_CFG_W1	1_5244h
MBC2_DOM9_MEM1_BLK_CFG_W0	1_5380h
MBC2_DOM9_MEM1_BLK_CFG_W1	1_5384h
MBC2_DOM10_MEM0_BLK_CFG_W0	1_5440h
MBC2_DOM10_MEM0_BLK_CFG_W1	1_5444h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC2_DOM10_MEM1_BLK_CFG_W0	1_5580h
MBC2_DOM10_MEM1_BLK_CFG_W1	1_5584h
MBC2_DOM11_MEM0_BLK_CFG_W0	1_5640h
MBC2_DOM11_MEM0_BLK_CFG_W1	1_5644h
MBC2_DOM11_MEM1_BLK_CFG_W0	1_5780h
MBC2_DOM11_MEM1_BLK_CFG_W1	1_5784h
MBC2_DOM12_MEM0_BLK_CFG_W0	1_5840h
MBC2_DOM12_MEM0_BLK_CFG_W1	1_5844h
MBC2_DOM12_MEM1_BLK_CFG_W0	1_5980h
MBC2_DOM12_MEM1_BLK_CFG_W1	1_5984h
MBC2_DOM13_MEM0_BLK_CFG_W0	1_5A40h
MBC2_DOM13_MEM0_BLK_CFG_W1	1_5A44h
MBC2_DOM13_MEM1_BLK_CFG_W0	1_5B80h
MBC2_DOM13_MEM1_BLK_CFG_W1	1_5B84h
MBC2_DOM14_MEM0_BLK_CFG_W0	1_5C40h
MBC2_DOM14_MEM0_BLK_CFG_W1	1_5C44h
MBC2_DOM14_MEM1_BLK_CFG_W0	1_5D80h
MBC2_DOM14_MEM1_BLK_CFG_W1	1_5D84h
MBC2_DOM15_MEM0_BLK_CFG_W0	1_5E40h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC2_DOM15_MEM0_BLK_CFG_W1	1_5E44h
MBC2_DOM15_MEM1_BLK_CFG_W0	1_5F80h
MBC2_DOM15_MEM1_BLK_CFG_W1	1_5F84h
MBC3_DOM0_MEM0_BLK_CFG_W0	1_6040h
MBC3_DOM0_MEM0_BLK_CFG_W1	1_6044h
MBC3_DOM0_MEM0_BLK_CFG_W2	1_6048h
MBC3_DOM0_MEM0_BLK_CFG_W3	1_604Ch
MBC3_DOM0_MEM0_BLK_CFG_W4	1_6050h
MBC3_DOM0_MEM1_BLK_CFG_W0	1_6180h
MBC3_DOM0_MEM1_BLK_CFG_W1	1_6184h
MBC3_DOM0_MEM1_BLK_CFG_W2	1_6188h
MBC3_DOM0_MEM1_BLK_CFG_W3	1_618Ch
MBC3_DOM0_MEM1_BLK_CFG_W4	1_6190h
MBC3_DOM1_MEM0_BLK_CFG_W0	1_6240h
MBC3_DOM1_MEM0_BLK_CFG_W1	1_6244h
MBC3_DOM1_MEM0_BLK_CFG_W2	1_6248h
MBC3_DOM1_MEM0_BLK_CFG_W3	1_624Ch
MBC3_DOM1_MEM0_BLK_CFG_W4	1_6250h
MBC3_DOM1_MEM1_BLK_CFG_W0	1_6380h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC3_DOM1_MEM1_BLK_CFG_W1	1_6384h
MBC3_DOM1_MEM1_BLK_CFG_W2	1_6388h
MBC3_DOM1_MEM1_BLK_CFG_W3	1_638Ch
MBC3_DOM1_MEM1_BLK_CFG_W4	1_6390h
MBC3_DOM2_MEM0_BLK_CFG_W0	1_6440h
MBC3_DOM2_MEM0_BLK_CFG_W1	1_6444h
MBC3_DOM2_MEM0_BLK_CFG_W2	1_6448h
MBC3_DOM2_MEM0_BLK_CFG_W3	1_644Ch
MBC3_DOM2_MEM0_BLK_CFG_W4	1_6450h
MBC3_DOM2_MEM1_BLK_CFG_W0	1_6580h
MBC3_DOM2_MEM1_BLK_CFG_W1	1_6584h
MBC3_DOM2_MEM1_BLK_CFG_W2	1_6588h
MBC3_DOM2_MEM1_BLK_CFG_W3	1_658Ch
MBC3_DOM2_MEM1_BLK_CFG_W4	1_6590h
MBC3_DOM3_MEM0_BLK_CFG_W0	1_6640h
MBC3_DOM3_MEM0_BLK_CFG_W1	1_6644h
MBC3_DOM3_MEM0_BLK_CFG_W2	1_6648h
MBC3_DOM3_MEM0_BLK_CFG_W3	1_664Ch
MBC3_DOM3_MEM0_BLK_CFG_W4	1_6650h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC3_DOM3_MEM1_BLK_CFG_W0	1_6780h
MBC3_DOM3_MEM1_BLK_CFG_W1	1_6784h
MBC3_DOM3_MEM1_BLK_CFG_W2	1_6788h
MBC3_DOM3_MEM1_BLK_CFG_W3	1_678Ch
MBC3_DOM3_MEM1_BLK_CFG_W4	1_6790h
MBC3_DOM4_MEM0_BLK_CFG_W0	1_6840h
MBC3_DOM4_MEM0_BLK_CFG_W1	1_6844h
MBC3_DOM4_MEM0_BLK_CFG_W2	1_6848h
MBC3_DOM4_MEM0_BLK_CFG_W3	1_684Ch
MBC3_DOM4_MEM0_BLK_CFG_W4	1_6850h
MBC3_DOM4_MEM1_BLK_CFG_W0	1_6980h
MBC3_DOM4_MEM1_BLK_CFG_W1	1_6984h
MBC3_DOM4_MEM1_BLK_CFG_W2	1_6988h
MBC3_DOM4_MEM1_BLK_CFG_W3	1_698Ch
MBC3_DOM4_MEM1_BLK_CFG_W4	1_6990h
MBC3_DOM5_MEM0_BLK_CFG_W0	1_6A40h
MBC3_DOM5_MEM0_BLK_CFG_W1	1_6A44h
MBC3_DOM5_MEM0_BLK_CFG_W2	1_6A48h
MBC3_DOM5_MEM0_BLK_CFG_W3	1_6A4Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC3_DOM5_MEM0_BLK_CFG_W4	1_6A50h
MBC3_DOM5_MEM1_BLK_CFG_W0	1_6B80h
MBC3_DOM5_MEM1_BLK_CFG_W1	1_6B84h
MBC3_DOM5_MEM1_BLK_CFG_W2	1_6B88h
MBC3_DOM5_MEM1_BLK_CFG_W3	1_6B8Ch
MBC3_DOM5_MEM1_BLK_CFG_W4	1_6B90h
MBC3_DOM6_MEM0_BLK_CFG_W0	1_6C40h
MBC3_DOM6_MEM0_BLK_CFG_W1	1_6C44h
MBC3_DOM6_MEM0_BLK_CFG_W2	1_6C48h
MBC3_DOM6_MEM0_BLK_CFG_W3	1_6C4Ch
MBC3_DOM6_MEM0_BLK_CFG_W4	1_6C50h
MBC3_DOM6_MEM1_BLK_CFG_W0	1_6D80h
MBC3_DOM6_MEM1_BLK_CFG_W1	1_6D84h
MBC3_DOM6_MEM1_BLK_CFG_W2	1_6D88h
MBC3_DOM6_MEM1_BLK_CFG_W3	1_6D8Ch
MBC3_DOM6_MEM1_BLK_CFG_W4	1_6D90h
MBC3_DOM7_MEM0_BLK_CFG_W0	1_6E40h
MBC3_DOM7_MEM0_BLK_CFG_W1	1_6E44h
MBC3_DOM7_MEM0_BLK_CFG_W2	1_6E48h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC3_DOM7_MEM0_BLK_CFG_W3	1_6E4Ch
MBC3_DOM7_MEM0_BLK_CFG_W4	1_6E50h
MBC3_DOM7_MEM1_BLK_CFG_W0	1_6F80h
MBC3_DOM7_MEM1_BLK_CFG_W1	1_6F84h
MBC3_DOM7_MEM1_BLK_CFG_W2	1_6F88h
MBC3_DOM7_MEM1_BLK_CFG_W3	1_6F8Ch
MBC3_DOM7_MEM1_BLK_CFG_W4	1_6F90h
MBC3_DOM8_MEM0_BLK_CFG_W0	1_7040h
MBC3_DOM8_MEM0_BLK_CFG_W1	1_7044h
MBC3_DOM8_MEM0_BLK_CFG_W2	1_7048h
MBC3_DOM8_MEM0_BLK_CFG_W3	1_704Ch
MBC3_DOM8_MEM0_BLK_CFG_W4	1_7050h
MBC3_DOM8_MEM1_BLK_CFG_W0	1_7180h
MBC3_DOM8_MEM1_BLK_CFG_W1	1_7184h
MBC3_DOM8_MEM1_BLK_CFG_W2	1_7188h
MBC3_DOM8_MEM1_BLK_CFG_W3	1_718Ch
MBC3_DOM8_MEM1_BLK_CFG_W4	1_7190h
MBC3_DOM9_MEM0_BLK_CFG_W0	1_7240h
MBC3_DOM9_MEM0_BLK_CFG_W1	1_7244h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC3_DOM9_MEM0_BLK_CFG_W2	1_7248h
MBC3_DOM9_MEM0_BLK_CFG_W3	1_724Ch
MBC3_DOM9_MEM0_BLK_CFG_W4	1_7250h
MBC3_DOM9_MEM1_BLK_CFG_W0	1_7380h
MBC3_DOM9_MEM1_BLK_CFG_W1	1_7384h
MBC3_DOM9_MEM1_BLK_CFG_W2	1_7388h
MBC3_DOM9_MEM1_BLK_CFG_W3	1_738Ch
MBC3_DOM9_MEM1_BLK_CFG_W4	1_7390h
MBC3_DOM10_MEM0_BLK_CFG_W0	1_7440h
MBC3_DOM10_MEM0_BLK_CFG_W1	1_7444h
MBC3_DOM10_MEM0_BLK_CFG_W2	1_7448h
MBC3_DOM10_MEM0_BLK_CFG_W3	1_744Ch
MBC3_DOM10_MEM0_BLK_CFG_W4	1_7450h
MBC3_DOM10_MEM1_BLK_CFG_W0	1_7580h
MBC3_DOM10_MEM1_BLK_CFG_W1	1_7584h
MBC3_DOM10_MEM1_BLK_CFG_W2	1_7588h
MBC3_DOM10_MEM1_BLK_CFG_W3	1_758Ch
MBC3_DOM10_MEM1_BLK_CFG_W4	1_7590h
MBC3_DOM11_MEM0_BLK_CFG_W0	1_7640h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC3_DOM11_MEM0_BLK_CFG_W1	1_7644h
MBC3_DOM11_MEM0_BLK_CFG_W2	1_7648h
MBC3_DOM11_MEM0_BLK_CFG_W3	1_764Ch
MBC3_DOM11_MEM0_BLK_CFG_W4	1_7650h
MBC3_DOM11_MEM1_BLK_CFG_W0	1_7780h
MBC3_DOM11_MEM1_BLK_CFG_W1	1_7784h
MBC3_DOM11_MEM1_BLK_CFG_W2	1_7788h
MBC3_DOM11_MEM1_BLK_CFG_W3	1_778Ch
MBC3_DOM11_MEM1_BLK_CFG_W4	1_7790h
MBC3_DOM12_MEM0_BLK_CFG_W0	1_7840h
MBC3_DOM12_MEM0_BLK_CFG_W1	1_7844h
MBC3_DOM12_MEM0_BLK_CFG_W2	1_7848h
MBC3_DOM12_MEM0_BLK_CFG_W3	1_784Ch
MBC3_DOM12_MEM0_BLK_CFG_W4	1_7850h
MBC3_DOM12_MEM1_BLK_CFG_W0	1_7980h
MBC3_DOM12_MEM1_BLK_CFG_W1	1_7984h
MBC3_DOM12_MEM1_BLK_CFG_W2	1_7988h
MBC3_DOM12_MEM1_BLK_CFG_W3	1_798Ch
MBC3_DOM12_MEM1_BLK_CFG_W4	1_7990h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC3_DOM13_MEM0_BLK_CFG_W0	1_7A40h
MBC3_DOM13_MEM0_BLK_CFG_W1	1_7A44h
MBC3_DOM13_MEM0_BLK_CFG_W2	1_7A48h
MBC3_DOM13_MEM0_BLK_CFG_W3	1_7A4Ch
MBC3_DOM13_MEM0_BLK_CFG_W4	1_7A50h
MBC3_DOM13_MEM1_BLK_CFG_W0	1_7B80h
MBC3_DOM13_MEM1_BLK_CFG_W1	1_7B84h
MBC3_DOM13_MEM1_BLK_CFG_W2	1_7B88h
MBC3_DOM13_MEM1_BLK_CFG_W3	1_7B8Ch
MBC3_DOM13_MEM1_BLK_CFG_W4	1_7B90h
MBC3_DOM14_MEM0_BLK_CFG_W0	1_7C40h
MBC3_DOM14_MEM0_BLK_CFG_W1	1_7C44h
MBC3_DOM14_MEM0_BLK_CFG_W2	1_7C48h
MBC3_DOM14_MEM0_BLK_CFG_W3	1_7C4Ch
MBC3_DOM14_MEM0_BLK_CFG_W4	1_7C50h
MBC3_DOM14_MEM1_BLK_CFG_W0	1_7D80h
MBC3_DOM14_MEM1_BLK_CFG_W1	1_7D84h
MBC3_DOM14_MEM1_BLK_CFG_W2	1_7D88h
MBC3_DOM14_MEM1_BLK_CFG_W3	1_7D8Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC3_DOM14_MEM1_BLK_CFG_W4	1_7D90h
MBC3_DOM15_MEM0_BLK_CFG_W0	1_7E40h
MBC3_DOM15_MEM0_BLK_CFG_W1	1_7E44h
MBC3_DOM15_MEM0_BLK_CFG_W2	1_7E48h
MBC3_DOM15_MEM0_BLK_CFG_W3	1_7E4Ch
MBC3_DOM15_MEM0_BLK_CFG_W4	1_7E50h
MBC3_DOM15_MEM1_BLK_CFG_W0	1_7F80h
MBC3_DOM15_MEM1_BLK_CFG_W1	1_7F84h
MBC3_DOM15_MEM1_BLK_CFG_W2	1_7F88h
MBC3_DOM15_MEM1_BLK_CFG_W3	1_7F8Ch
MBC3_DOM15_MEM1_BLK_CFG_W4	1_7F90h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NSE7	MBACSEL7			NSE6	MBACSEL6			NSE5	MBACSEL5			NSE4	MBACSEL4		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NSE3	MBACSEL3			NSE2	MBACSEL2			NSE1	MBACSEL1			NSE0	MBACSEL0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31	NonSecure Enable for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
NSE7	0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
30-28 MBACSEL7	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL7 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
27 NSE6	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
26-24 MBACSEL6	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL6 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
	111 - select MBC_MEMN_GLBAC7 access control policy for block B
23 NSE5	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
22-20 MBACSEL5	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL5 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
19 NSE4	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
18-16 MBACSEL4	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL4 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
15 NSE3	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
14-12 MBACSEL3	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL3 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
11 NSE2	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
10-8 MBACSEL2	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL2 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
	011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
7 NSE1	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
6-4 MBACSEL1	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL1 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
3 NSE0	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
2-0 MBACSEL0	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL0 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
	001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B

22.7.5.37 MBC Memory Block NonSecure Enable Word (MBC2_DOM0_MEM0_BLK_NSE_W0 - MBC3_DOM15_MEM1_BLK_NSE_W1)

MBC[m]_DOM[d]_MEM[s]_BLK_NSE_W[w], where,

- m - mbc index
- d - domain index
- s - memory target index
- w - word index

This is the bitmap of the individual NSE bits for the memory block.

NOTE

This register is shown with fields such that it covers entire 32 bit register. However, actual fields may be less if number of blocks are such that fields don't align to 32 bits. These absent fields may be shown but user can refer to table below to deduce actual fields and treat absent fields as reserved.

The following table describes the relationship between W (word index) and B (block number).

Table 115. Block Config Word to Block Number Relationship

Block NSE Word	BIT31	BIT30-BIT1	BIT0
W0	block 31	block 30 - block 1	block 0
W1	block 63	block 62 - block 33	block 32
For MEM0, block 64 - block 511			
W2	block 95	block 94 - block 65	block 64
		.	
		.	
		.	
W15	block 511	block 510 - block 481	block 480

The NSE bitmap can be accessed directly by register reads and writes to these locations, or "indirectly" through writes to NSE_SET, NSE_CLR, NSE_CLR_ALL, and BLK_CFG[NSEn] registers.

Offset

Register	Offset
MBC2_DOM0_MEM0_BLK_NSE_W0	1_4140h
MBC2_DOM0_MEM1_BLK_NSE_W0	1_41A0h
MBC2_DOM1_MEM0_BLK_NSE_W0	1_4340h
MBC2_DOM1_MEM1_BLK_NSE_W0	1_43A0h
MBC2_DOM2_MEM0_BLK_NSE_W0	1_4540h
MBC2_DOM2_MEM1_BLK_NSE_W0	1_45A0h
MBC2_DOM3_MEM0_BLK_NSE_W0	1_4740h
MBC2_DOM3_MEM1_BLK_NSE_W0	1_47A0h
MBC2_DOM4_MEM0_BLK_NSE_W0	1_4940h
MBC2_DOM4_MEM1_BLK_NSE_W0	1_49A0h
MBC2_DOM5_MEM0_BLK_NSE_W0	1_4B40h
MBC2_DOM5_MEM1_BLK_NSE_W0	1_4BA0h
MBC2_DOM6_MEM0_BLK_NSE_W0	1_4D40h
MBC2_DOM6_MEM1_BLK_NSE_W0	1_4DA0h
MBC2_DOM7_MEM0_BLK_NSE_W0	1_4F40h
MBC2_DOM7_MEM1_BLK_NSE_W0	1_4FA0h
MBC2_DOM8_MEM0_BLK_NSE_W0	1_5140h
MBC2_DOM8_MEM1_BLK_NSE_W0	1_51A0h
MBC2_DOM9_MEM0_BLK_NSE_W0	1_5340h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC2_DOM9_MEM1_BLK_NSE_W0	1_53A0h
MBC2_DOM10_MEM0_BLK_NSE_W0	1_5540h
MBC2_DOM10_MEM1_BLK_NSE_W0	1_55A0h
MBC2_DOM11_MEM0_BLK_NSE_W0	1_5740h
MBC2_DOM11_MEM1_BLK_NSE_W0	1_57A0h
MBC2_DOM12_MEM0_BLK_NSE_W0	1_5940h
MBC2_DOM12_MEM1_BLK_NSE_W0	1_59A0h
MBC2_DOM13_MEM0_BLK_NSE_W0	1_5B40h
MBC2_DOM13_MEM1_BLK_NSE_W0	1_5BA0h
MBC2_DOM14_MEM0_BLK_NSE_W0	1_5D40h
MBC2_DOM14_MEM1_BLK_NSE_W0	1_5DA0h
MBC2_DOM15_MEM0_BLK_NSE_W0	1_5F40h
MBC2_DOM15_MEM1_BLK_NSE_W0	1_5FA0h
MBC3_DOM0_MEM0_BLK_NSE_W0	1_6140h
MBC3_DOM0_MEM0_BLK_NSE_W1	1_6144h
MBC3_DOM0_MEM1_BLK_NSE_W0	1_61A0h
MBC3_DOM0_MEM1_BLK_NSE_W1	1_61A4h
MBC3_DOM1_MEM0_BLK_NSE_W0	1_6340h
MBC3_DOM1_MEM0_BLK_NSE_W1	1_6344h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC3_DOM1_MEM1_BLK_NSE_W0	1_63A0h
MBC3_DOM1_MEM1_BLK_NSE_W1	1_63A4h
MBC3_DOM2_MEM0_BLK_NSE_W0	1_6540h
MBC3_DOM2_MEM0_BLK_NSE_W1	1_6544h
MBC3_DOM2_MEM1_BLK_NSE_W0	1_65A0h
MBC3_DOM2_MEM1_BLK_NSE_W1	1_65A4h
MBC3_DOM3_MEM0_BLK_NSE_W0	1_6740h
MBC3_DOM3_MEM0_BLK_NSE_W1	1_6744h
MBC3_DOM3_MEM1_BLK_NSE_W0	1_67A0h
MBC3_DOM3_MEM1_BLK_NSE_W1	1_67A4h
MBC3_DOM4_MEM0_BLK_NSE_W0	1_6940h
MBC3_DOM4_MEM0_BLK_NSE_W1	1_6944h
MBC3_DOM4_MEM1_BLK_NSE_W0	1_69A0h
MBC3_DOM4_MEM1_BLK_NSE_W1	1_69A4h
MBC3_DOM5_MEM0_BLK_NSE_W0	1_6B40h
MBC3_DOM5_MEM0_BLK_NSE_W1	1_6B44h
MBC3_DOM5_MEM1_BLK_NSE_W0	1_6BA0h
MBC3_DOM5_MEM1_BLK_NSE_W1	1_6BA4h
MBC3_DOM6_MEM0_BLK_NSE_W0	1_6D40h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC3_DOM6_MEM0_BLK_NSE_W1	1_6D44h
MBC3_DOM6_MEM1_BLK_NSE_W0	1_6DA0h
MBC3_DOM6_MEM1_BLK_NSE_W1	1_6DA4h
MBC3_DOM7_MEM0_BLK_NSE_W0	1_6F40h
MBC3_DOM7_MEM0_BLK_NSE_W1	1_6F44h
MBC3_DOM7_MEM1_BLK_NSE_W0	1_6FA0h
MBC3_DOM7_MEM1_BLK_NSE_W1	1_6FA4h
MBC3_DOM8_MEM0_BLK_NSE_W0	1_7140h
MBC3_DOM8_MEM0_BLK_NSE_W1	1_7144h
MBC3_DOM8_MEM1_BLK_NSE_W0	1_71A0h
MBC3_DOM8_MEM1_BLK_NSE_W1	1_71A4h
MBC3_DOM9_MEM0_BLK_NSE_W0	1_7340h
MBC3_DOM9_MEM0_BLK_NSE_W1	1_7344h
MBC3_DOM9_MEM1_BLK_NSE_W0	1_73A0h
MBC3_DOM9_MEM1_BLK_NSE_W1	1_73A4h
MBC3_DOM10_MEM0_BLK_NSE_W0	1_7540h
MBC3_DOM10_MEM0_BLK_NSE_W1	1_7544h
MBC3_DOM10_MEM1_BLK_NSE_W0	1_75A0h
MBC3_DOM10_MEM1_BLK_NSE_W1	1_75A4h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC3_DOM11_MEM0_BLK_NSE_W0	1_7740h
MBC3_DOM11_MEM0_BLK_NSE_W1	1_7744h
MBC3_DOM11_MEM1_BLK_NSE_W0	1_77A0h
MBC3_DOM11_MEM1_BLK_NSE_W1	1_77A4h
MBC3_DOM12_MEM0_BLK_NSE_W0	1_7940h
MBC3_DOM12_MEM0_BLK_NSE_W1	1_7944h
MBC3_DOM12_MEM1_BLK_NSE_W0	1_79A0h
MBC3_DOM12_MEM1_BLK_NSE_W1	1_79A4h
MBC3_DOM13_MEM0_BLK_NSE_W0	1_7B40h
MBC3_DOM13_MEM0_BLK_NSE_W1	1_7B44h
MBC3_DOM13_MEM1_BLK_NSE_W0	1_7BA0h
MBC3_DOM13_MEM1_BLK_NSE_W1	1_7BA4h
MBC3_DOM14_MEM0_BLK_NSE_W0	1_7D40h
MBC3_DOM14_MEM0_BLK_NSE_W1	1_7D44h
MBC3_DOM14_MEM1_BLK_NSE_W0	1_7DA0h
MBC3_DOM14_MEM1_BLK_NSE_W1	1_7DA4h
MBC3_DOM15_MEM0_BLK_NSE_W0	1_7F40h
MBC3_DOM15_MEM0_BLK_NSE_W1	1_7F44h
MBC3_DOM15_MEM1_BLK_NSE_W0	1_7FA0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC3_DOM15_MEM1_BLK_NSE_W1	1_7FA4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BIT31	BIT30	BIT29	BIT28	BIT27	BIT26	BIT25	BIT24	BIT23	BIT22	BIT21	BIT20	BIT19	BIT18	BIT17	BIT16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BITb	Bit b NonSecure Enable [b = 0 - 31] 0 - Secure read and write to block B are based on corresponding MBACSEL field in register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure and nonsecure accesses to block B are based on corresponding MBACSEL field in register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).

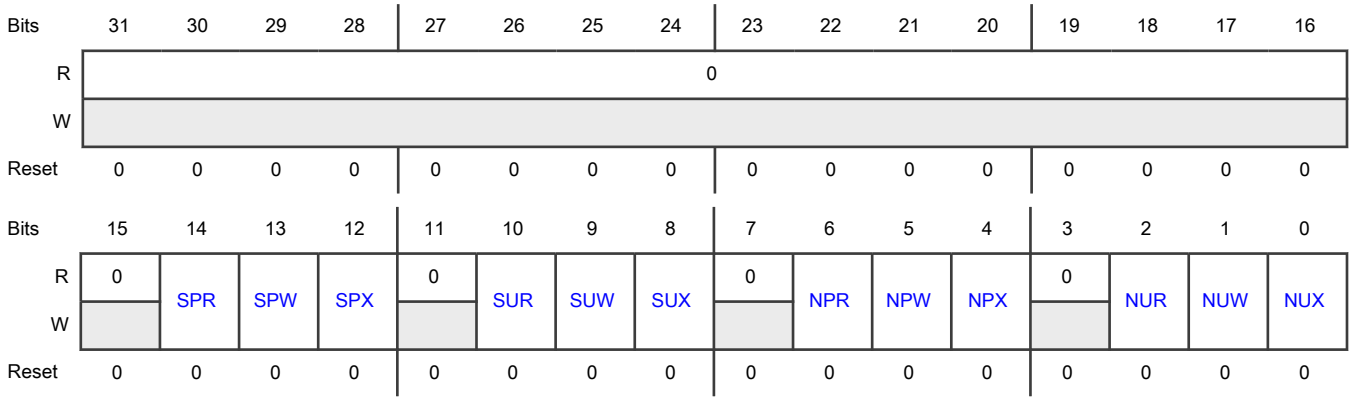
22.7.5.38 MBC Global Access Control (MBC3_MEMN_GLBAC0)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User.

Offset

Register	Offset
MBC3_MEMN_GLBAC0	1_6020h

Diagram



Fields

Field	Description
31-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9	SecureUser Write

Table continues on the next page...

Table continued from the previous page...

Field	Description
SUW	SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

22.7.5.39 MBC Global Access Control (MBC3_MEMN_GLBAC1 - MBC3_MEMN_GLBAC7)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

Offset

Register	Offset
MBC3_MEMN_GLBAC1	1_6024h
MBC3_MEMN_GLBAC2	1_6028h
MBC3_MEMN_GLBAC3	1_602Ch
MBC3_MEMN_GLBAC4	1_6030h
MBC3_MEMN_GLBAC5	1_6034h
MBC3_MEMN_GLBAC6	1_6038h
MBC3_MEMN_GLBAC7	1_603Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	0														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SPR	SPW	SPX	0	SUR	SUW	SUX	0	NPR	NPW	NPX	0	NUR	NUW	NUX
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0 - This register is not locked and can be altered. 1 - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

22.7.5.40 MRC Global Configuration Register (MRC0_GLBCFG)

This read-only register describes the number of region descriptors for this memory.

Offset

Register	Offset
MRC0_GLBCFG	1_8000h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											NRGNS				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Fields

Field	Description
31-5 —	Reserved
4-0 NRGNS	Number of regions [1-16]

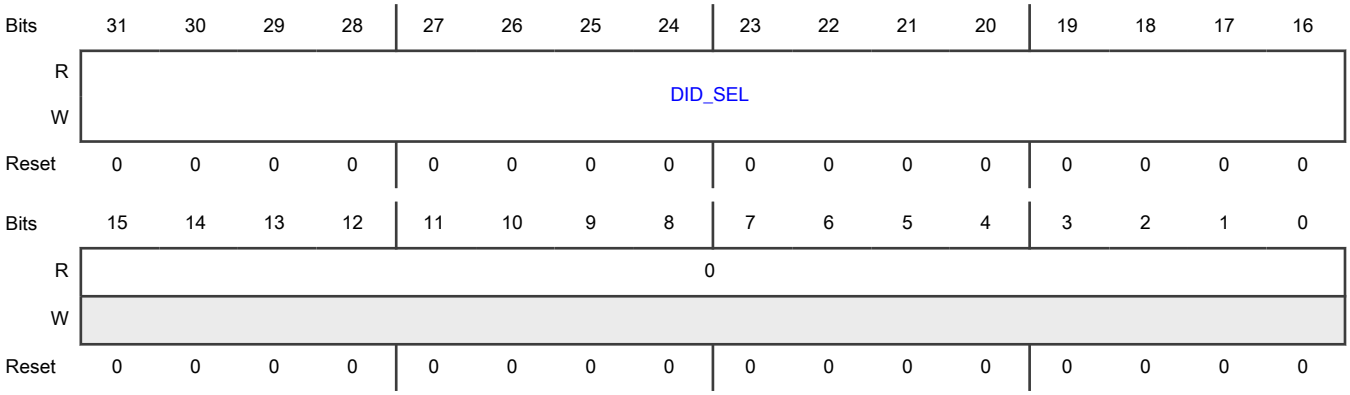
22.7.5.41 MRC NonSecure Enable Region Indirect (MRC0_NSE_RGN_INDIRECT)

This R/W register defines the selected domains that are affected by writes to the NSE_RGN_SET and NSE_RGN_CLR registers.

Offset

Register	Offset
MRC0_NSE_RGN_INDIRECT	1_8010h

Diagram



Fields

Field	Description
31-16 DID_SEL	DID Select Destination domain bitmap select. • Bit [31] = 1, update domain 15 • Bit [30] = 1, update domain 14 • Bit [29] = 1, update domain 13 • Bit [28] = 1, update domain 12 • Bit [27] = 1, update domain 11 • Bit [26] = 1, update domain 10 • Bit [25] = 1, update domain 9 • Bit [24] = 1, update domain 8 • Bit [23] = 1, update domain 7 • Bit [22] = 1, update domain 6 • Bit [21] = 1, update domain 5 • Bit [20] = 1, update domain 4 • Bit [19] = 1, update domain 3 • Bit [18] = 1, update domain 2 • Bit [17] = 1, update domain 1 • Bit [16] = 1, update domain 0
15-0 —	Reserved

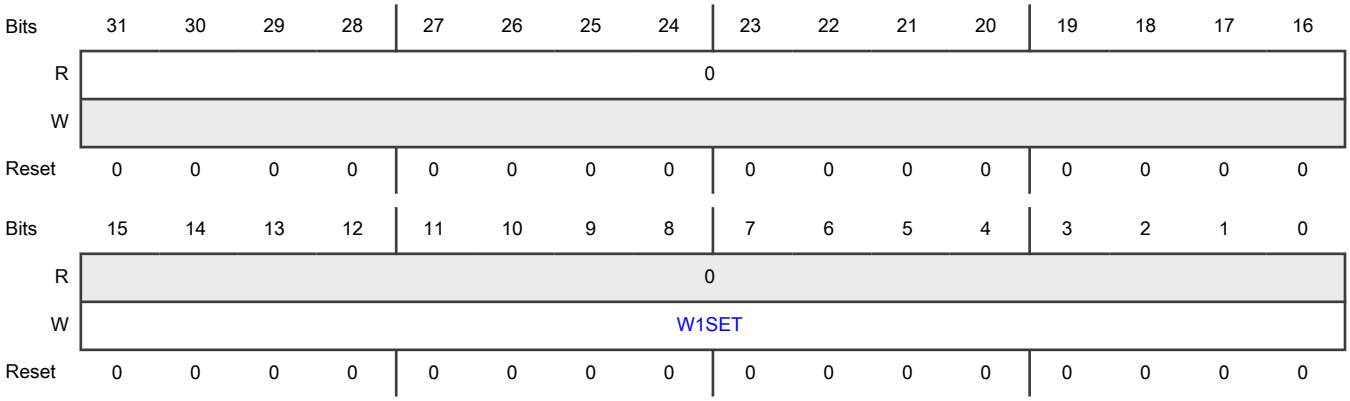
22.7.5.42 MRC NonSecure Enable Region Set (MRC0_NSE_RGN_SET)

A write to this location sets the appropriate NSE bits in region descriptors for the selected domain in MRC NonSecure Enable Region Indirect register. This register reads as 0.

Offset

Register	Offset
MRC0_NSE_RGN_SET	1_8014h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 W1SET	Write-1 Set • Bit [15] = Set the NSE bits for RGD15 • Bit [14] = Set the NSE bits for RGD14 • Bit [13] = Set the NSE bits for RGD13 • Bit [12] = Set the NSE bits for RGD12 • Bit [11] = Set the NSE bits for RGD11 • Bit [10] = Set the NSE bits for RGD10 • Bit [9] = Set the NSE bits for RGD9 • Bit [8] = Set the NSE bits for RGD8 • Bit [7] = Set the NSE bits for RGD7 • Bit [6] = Set the NSE bits for RGD6 • Bit [5] = Set the NSE bits for RGD5 • Bit [4] = Set the NSE bits for RGD4 • Bit [3] = Set the NSE bits for RGD3 • Bit [2] = Set the NSE bits for RGD2 • Bit [1] = Set the NSE bits for RGD1 • Bit [0] = Set the NSE bits for RGD0

22.7.5.43 MRC NonSecure Enable Region Clear (MRC0_NSE_RGN_CLR)

A write to this location clears the appropriate NSE bits in region descriptors for the selected domain in MRC NonSecure Enable Region Indirect register. This register reads as 0.

Offset

Register	Offset
MRC0_NSE_RGN_CLR	1_8018h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	W1CLR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 W1CLR	Write-1 Clear • Bit [15] = Clear the NSE bits for RGD15 • Bit [14] = Clear the NSE bits for RGD14 • Bit [13] = Clear the NSE bits for RGD13 • Bit [12] = Clear the NSE bits for RGD12 • Bit [11] = Clear the NSE bits for RGD11 • Bit [10] = Clear the NSE bits for RGD10 • Bit [9] = Clear the NSE bits for RGD9 • Bit [8] = Clear the NSE bits for RGD8 • Bit [7] = Clear the NSE bits for RGD7 • Bit [6] = Clear the NSE bits for RGD6 • Bit [5] = Clear the NSE bits for RGD5

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Bit [4] = Clear the NSE bits for RGD4 • Bit [3] = Clear the NSE bits for RGD3 • Bit [2] = Clear the NSE bits for RGD2 • Bit [1] = Clear the NSE bits for RGD1 • Bit [0] = Clear the NSE bits for RGD0

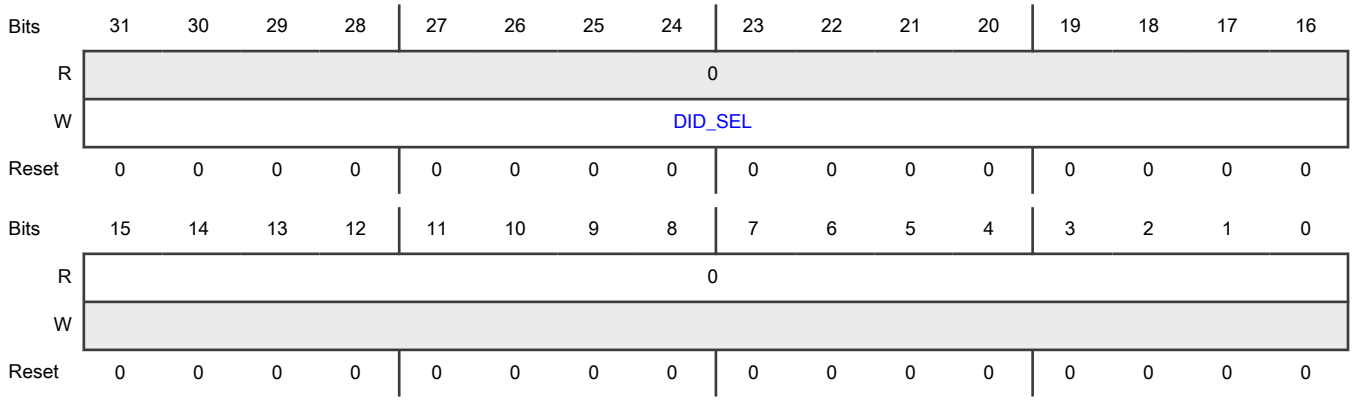
22.7.5.44 MRC NonSecure Enable Region Clear All (MRC0_NSE_RGN_CLR_ALL)

A write to this register clears all the region NSE bits for the selected domains defined in the write data.

Offset

Register	Offset
MRC0_NSE_RGN_CLR_ALL	1_801Ch

Diagram



Fields

Field	Description
31-16 DID_SEL	DID Select Destination domain bitmap select. • Bit [31] = 1 clear NSE bits for domain 15 • Bit [30] = 1 clear NSE bits for domain 14 • Bit [29] = 1 clear NSE bits for domain 13

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Bit [28] = 1 clear NSE bits for domain 12 • Bit [27] = 1 clear NSE bits for domain 11 • Bit [26] = 1 clear NSE bits for domain 10 • Bit [25] = 1 clear NSE bits for domain 9 • Bit [24] = 1 clear NSE bits for domain 8 • Bit [23] = 1 clear NSE bits for domain 7 • Bit [22] = 1 clear NSE bits for domain 6 • Bit [21] = 1 clear NSE bits for domain 5 • Bit [20] = 1 clear NSE bits for domain 4 • Bit [19] = 1 clear NSE bits for domain 3 • Bit [18] = 1 clear NSE bits for domain 2 • Bit [17] = 1 clear NSE bits for domain 1 • Bit [16] = 1 clear NSE bits for domain 0
15-0 —	Reserved

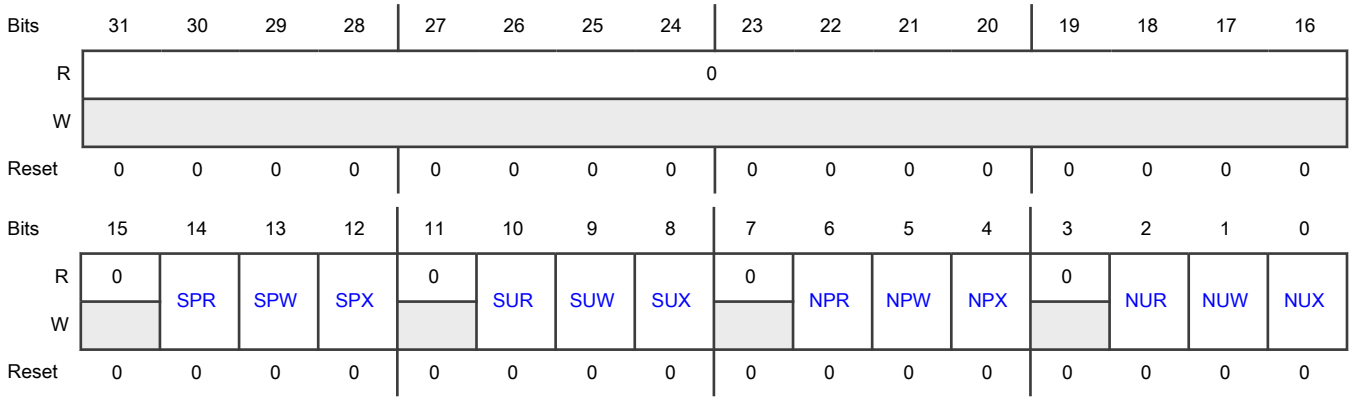
22.7.5.45 MRC Global Access Control (MRC0_GLBAC0)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, NonsecureUser. The lock bit provides the ability to lock these control access flags, to lock the MRC_BLK_CFG[MRACSEL] field, and to lock the MRC_RGD[STRT_ADDR, END_ADDR, VLD] fields when a locked MRC_GLBAC register is selected.

Offset

Register	Offset
MRC0_GLBAC0	1_8020h

Diagram



Fields

Field	Description
31-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9	SecureUser Write

Table continues on the next page...

Table continued from the previous page...

Field	Description
SUW	SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

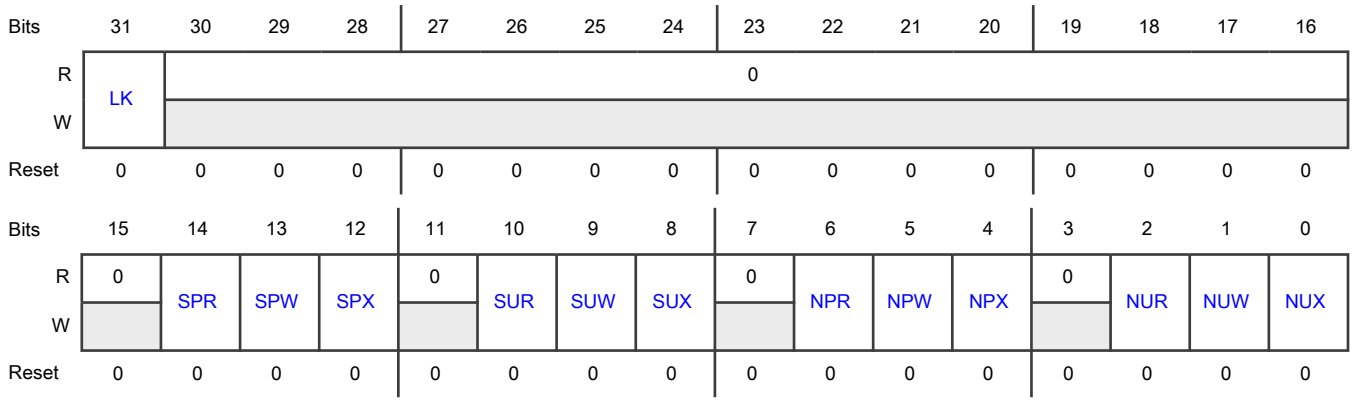
22.7.5.46 MRC Global Access Control (MRC0_GLBAC1 - MRC0_GLBAC7)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, NonsecureUser. The lock bit provides the ability to lock these control access flags, to lock the MRC_BLK_CFG[MRACSEL] field, and to lock the MRC_RGD[STRT_ADDR, END_ADDR, VLD] fields when a locked MRC_GLBAC register is selected.

Offset

Register	Offset
MRC0_GLBAC1	1_8024h
MRC0_GLBAC2	1_8028h
MRC0_GLBAC3	1_802Ch
MRC0_GLBAC4	1_8030h
MRC0_GLBAC5	1_8034h
MRC0_GLBAC6	1_8038h
MRC0_GLBAC7	1_803Ch

Diagram



Fields

Field	Description
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0 - This register is not locked and can be altered. 1 - This register is locked (read-only) and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

22.7.5.47 MRC Region Descriptor Word 0 (MRC0_DOM0_RGD0_W0 - MRC0_DOM15_RGD15_W0)

MRC[m]_DOM[d]_RGD[r]_W[w], where,

- m - mrc index
- d - domain index
- r - region descriptor index
- w - word index

The MRC[m]_DOM[d]_RGD[r]_W[w] registers provide a 3-dimensional data structure for defining access control policies per domain for each supported memory region.

Memory map/register definition:

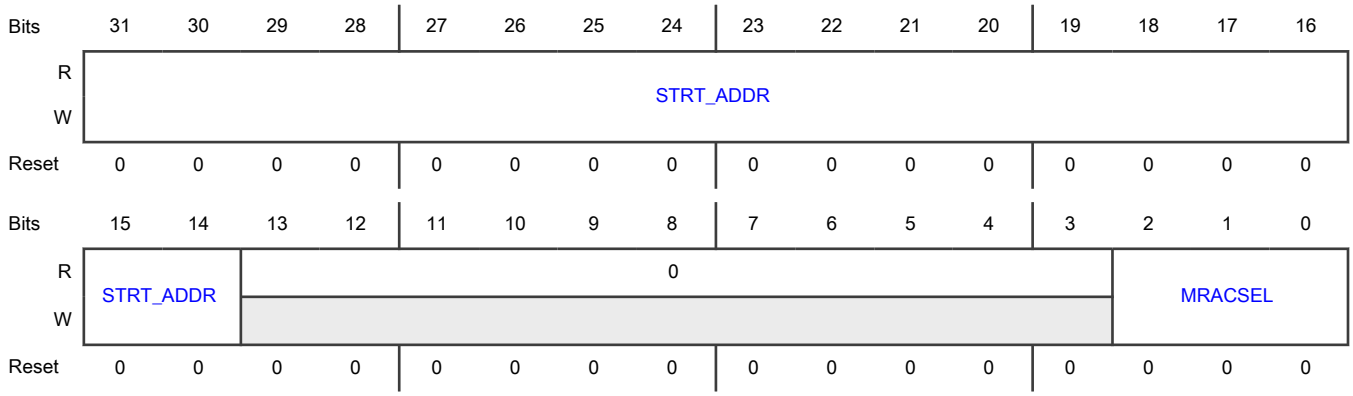
There two word-size registers (W[w]) defined in each memory region. There are up to 16 memory region checkers (MRCs), m, each supporting up to 16 memory region descriptors (RGD), r. The number of MRCs is defined by HWCFG0[NMRC].

Offset

For d = 0 to 15; r = 0 to 15:

Register	Offset
MRC0_DOMd_RGDr_W0	1_8040h + (d × 100h) + (r × 8h)

Diagram



Fields

Field	Description
31-14 STRT_ADDR	Start Address 0-mod-16K physical start address [31:14] of region r. **NOTE** This field is locked (read-only) if MRACSEL = r and MRC_GLBACr[LK] = 1.
13-3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
2-0 MRACSEL	Memory Region Access Control Select This field selects the global access control associated with region r. **NOTE** This field is locked (read-only) if MRACSEL = r and MRC_GLBACr[LK] = 1. 000 - Select MRC_GLBAC0 access control policy 001 - Select MRC_GLBAC1 access control policy 010 - Select MRC_GLBAC2 access control policy 011 - Select MRC_GLBAC3 access control policy 100 - Select MRC_GLBAC4 access control policy 101 - Select MRC_GLBAC5 access control policy 110 - Select MRC_GLBAC6 access control policy 111 - Select MRC_GLBAC7 access control policy

22.7.5.48 MRC Region Descriptor Word 1 (MRC0_DOM0_RGD0_W1 - MRC0_DOM15_RGD15_W1)

MRC[m]_DOM[d]_RGD[r]_W[w], where,

- m - mrc index
- d - domain index
- r - region descriptor index
- w - word index

The MRC[m]_[d]_rgd[r]_w[w] registers provide a 3-dimensional data structure for defining access control policies per domain for each supported memory region.

Memory map/register definition:

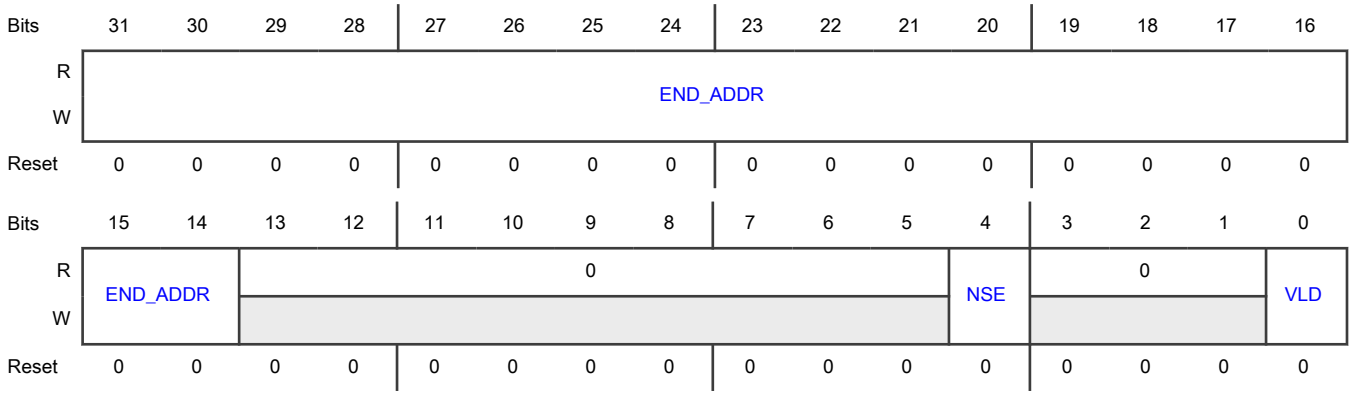
There two word-size registers (W[w]) defined in each memory region. There are up to 16 memory region checkers (MRCs), m, each supporting up to 16 region descriptors (RGD), r. The number of MRCs is defined by HWCFG0[NMRC].

Offset

For d = 0 to 15; r = 0 to 15:

Register	Offset
MRC0_DOMd_RGDr_W1	1_8044h + (d × 100h) + (r × 8h)

Diagram



Fields

Field	Description
31-14 END_ADDR	End Address (16K-1)-mod-16K physical end address [31:14] of region r. **NOTE** This field is locked (read-only) if MRACSEL = r and MRC_GLBACr[LK] = 1.
13-5 —	Reserved
4 NSE	NonSecure Enable Alternate view of NSE flag for Region r. 0 - Secure accesses to region r are based on corresponding MRACSEL field in this register (MRCm_DOMd_RGDr_Ww[MRACSEL]), nonsecure accesses to region r are not allowed. 1 - Secure and nonsecure accesses to region r are based on corresponding MRACSEL field in this register (MRCm_DOMd_RGDr_Ww[MRACSEL]).
3-1 —	Reserved
0 VLD	Valid If set, this region is valid and accesses are checked for this region. **NOTE** This field is locked (read-only) if MRACSEL = r and MRC_GLBACr[LK] = 1.

22.7.5.49 MRC Region Descriptor NonSecure Enable (MRC0_DOM0_RGD_NSE - MRC0_DOM15_RGD_NSE)

This register is the bitmap of the individual NSE bits for the region descriptors. Bit n corresponds to region n.

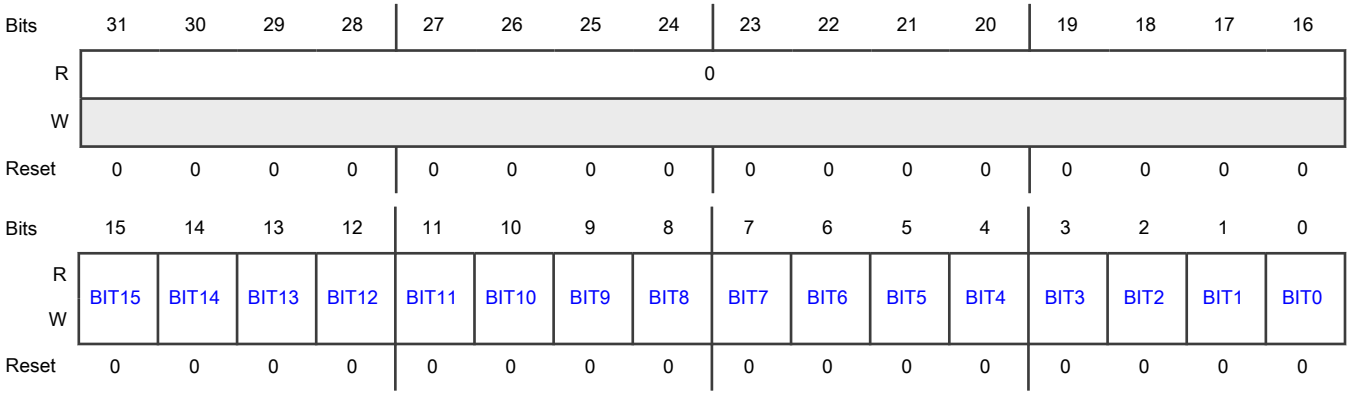
These bits can also be written or "indirectly" through writes to NSE_SET, NSE_CLR, NSE_CLR_ALL, and RGD_W0[NSE] registers.

Offset

For d = 0 to 15:

Register	Offset
MRC0_DOMd_RGD_NSE	1_80C0h + (d × 100h)

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 BITn	Bit n NonSecure Enable [n = 0 - 15] 0 - Secure accesses to region r are based on corresponding MRACSEL field in this register (MRCm_DOMd_RGDr_Ww[MRACSEL]), nonsecure accesses to region r are not allowed. 1 - Secure and nonsecure accesses to region r are based on corresponding MRACSEL field in register (MRCm_DOMd_RGDr_Ww[MRACSEL]).

22.7.6 TRDC register descriptions

22.7.6.1 TRDC memory map

TRDC_WAKEUPMIX base address: 4246_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	TRDC Register (TRDC_CR)	32	RW	0000_0010
F0	TRDC Hardware Configuration Register 0 (TRDC_HWCFG0)	32	R	2202_0510
F4	TRDC Hardware Configuration Register 1 (TRDC_HWCFG1)	32	R	See section

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
102 - 104	Domain Assignment Configuration Register (DACFG2 - DACFG4)	8	R	81
1C0	TRDC IDAU Control Register (TRDC_IDAU_CR)	32	RW	0000_0008
1E0	TRDC FLW Control (TRDC_FLW_CTL)	32	RW	0000_0000
1E4	TRDC FLW Physical Base (TRDC_FLW_PBASE)	32	R	0100_0000
1E8	TRDC FLW Array Base (TRDC_FLW_ABASE)	32	RW	0000_0000
1EC	TRDC FLW Block Count (TRDC_FLW_BCNT)	32	RW	0000_0000
1FC	TRDC Fault Domain ID (TRDC_FDID)	32	RW	0000_0000
200 - 23C	TRDC Domain Error Location Register (TRDC_DERRLOC0 - TRDC_DERRLOC15)	32	R	0000_0000
400	MBC Domain Error Word0 Register (MBC0_DERR_W0)	32	R	0000_0000
404	MBC Domain Error Word1 Register (MBC0_DERR_W1)	32	R	0000_0000
40C	MBC Domain Error Word3 Register (MBC0_DERR_W3)	32	RW	0000_0000
410	MBC Domain Error Word0 Register (MBC1_DERR_W0)	32	R	0000_0000
414	MBC Domain Error Word1 Register (MBC1_DERR_W1)	32	R	0000_0000
41C	MBC Domain Error Word3 Register (MBC1_DERR_W3)	32	RW	0000_0000
480	MRC Domain Error Word0 Register (MRC0_DERR_W0)	32	R	0000_0000
484	MRC Domain Error Word1 Register (MRC0_DERR_W1)	32	R	0000_0000
48C	MRC Domain Error Word3 Register (MRC0_DERR_W3)	32	RW	0000_0000
490	MRC Domain Error Word0 Register (MRC1_DERR_W0)	32	R	0000_0000
494	MRC Domain Error Word1 Register (MRC1_DERR_W1)	32	R	0000_0000
49C	MRC Domain Error Word3 Register (MRC1_DERR_W3)	32	RW	0000_0000
840	DAC Initiator Domain Assignment Register (MDA_W0_2_DFMT1)	32	RW	2000_0000
860	DAC Initiator Domain Assignment Register (MDA_W0_3_DFMT1)	32	RW	2000_0000
880	DAC Initiator Domain Assignment Register (MDA_W0_4_DFMT1)	32	RW	2000_0000
1_0000	MBC Global Configuration Register (MBC0_MEM0_GLBCFG)	32	R	000C_0080
1_0004	MBC Global Configuration Register (MBC0_MEM1_GLBCFG)	32	R	000C_0001
1_0008	MBC Global Configuration Register (MBC0_MEM2_GLBCFG)	32	R	000C_0001
1_000C	MBC Global Configuration Register (MBC0_MEM3_GLBCFG)	32	R	000C_0011
1_0010	MBC NonSecure Enable Block Index (MBC0_NSE_BLK_INDEX)	32	RW	0000_0000
1_0014	MBC NonSecure Enable Block Set (MBC0_NSE_BLK_SET)	32	RW	0000_0000
1_0018	MBC NonSecure Enable Block Clear (MBC0_NSE_BLK_CLR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_001C	MBC NonSecure Enable Block Clear All (MBC0_NSE_BLK_CLR_ALL)	32	RW	0000_0000
1_0020	MBC Global Access Control (MBC0_MEMN_GLBAC0)	32	RW	0000_0000
1_0024	MBC Global Access Control (MBC0_MEMN_GLBAC1)	32	RW	0000_0000
1_0028	MBC Global Access Control (MBC0_MEMN_GLBAC2)	32	RW	0000_0000
1_002C	MBC Global Access Control (MBC0_MEMN_GLBAC3)	32	RW	0000_0000
1_0030	MBC Global Access Control (MBC0_MEMN_GLBAC4)	32	RW	0000_0000
1_0034	MBC Global Access Control (MBC0_MEMN_GLBAC5)	32	RW	0000_0000
1_0038	MBC Global Access Control (MBC0_MEMN_GLBAC6)	32	RW	0000_0000
1_003C	MBC Global Access Control (MBC0_MEMN_GLBAC7)	32	RW	0000_0000
1_0040	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0044	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0048	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_004C	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0050	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0054	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0058	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_005C	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0060	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0064	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0068	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_006C	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0070	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W12)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0074	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0078	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_007C	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0140	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0144	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0148	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_014C	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0180	MBC Memory Block Configuration Word (MBC0_DOM0_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_01A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_01A8	MBC Memory Block Configuration Word (MBC0_DOM0_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_01C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_01D0	MBC Memory Block Configuration Word (MBC0_DOM0_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_01D4	MBC Memory Block Configuration Word (MBC0_DOM0_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_01D8	MBC Memory Block Configuration Word (MBC0_DOM0_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_01F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_0240	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0244	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0248	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_024C	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0250	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0254	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0258	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_025C	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0260	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0264	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0268	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_026C	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0270	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0274	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0278	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_027C	MBC Memory Block Configuration Word (MBC0_DOM1_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0340	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0344	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0348	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_034C	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0380	MBC Memory Block Configuration Word (MBC0_DOM1_MEM1_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_03A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_03A8	MBC Memory Block Configuration Word (MBC0_DOM1_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_03C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_03D0	MBC Memory Block Configuration Word (MBC0_DOM1_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_03D4	MBC Memory Block Configuration Word (MBC0_DOM1_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_03D8	MBC Memory Block Configuration Word (MBC0_DOM1_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_03F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM1_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_0440	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0444	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0448	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_044C	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0450	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0454	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0458	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_045C	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0460	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0464	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0468	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W10)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_046C	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0470	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0474	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0478	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_047C	MBC Memory Block Configuration Word (MBC0_DOM2_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0540	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0544	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0548	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_054C	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0580	MBC Memory Block Configuration Word (MBC0_DOM2_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_05A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_05A8	MBC Memory Block Configuration Word (MBC0_DOM2_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_05C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_05D0	MBC Memory Block Configuration Word (MBC0_DOM2_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_05D4	MBC Memory Block Configuration Word (MBC0_DOM2_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_05D8	MBC Memory Block Configuration Word (MBC0_DOM2_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_05F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM2_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_0640	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0644	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0648	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_064C	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0650	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0654	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0658	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_065C	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0660	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0664	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0668	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_066C	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0670	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0674	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0678	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_067C	MBC Memory Block Configuration Word (MBC0_DOM3_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0740	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0744	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0748	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM0_BLK_NSE_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_074C	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0780	MBC Memory Block Configuration Word (MBC0_DOM3_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_07A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_07A8	MBC Memory Block Configuration Word (MBC0_DOM3_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_07C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_07D0	MBC Memory Block Configuration Word (MBC0_DOM3_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_07D4	MBC Memory Block Configuration Word (MBC0_DOM3_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_07D8	MBC Memory Block Configuration Word (MBC0_DOM3_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_07F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM3_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_0840	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0844	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0848	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_084C	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0850	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0854	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0858	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_085C	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0860	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W8)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0864	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0868	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_086C	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0870	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0874	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0878	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_087C	MBC Memory Block Configuration Word (MBC0_DOM4_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0940	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0944	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0948	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_094C	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0980	MBC Memory Block Configuration Word (MBC0_DOM4_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_09A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_09A8	MBC Memory Block Configuration Word (MBC0_DOM4_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_09C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_09D0	MBC Memory Block Configuration Word (MBC0_DOM4_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_09D4	MBC Memory Block Configuration Word (MBC0_DOM4_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_09D8	MBC Memory Block Configuration Word (MBC0_DOM4_MEM3_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_09F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM4_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_0A40	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0A44	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0A48	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_0A4C	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0A50	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0A54	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0A58	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0A5C	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0A60	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0A64	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0A68	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_0A6C	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0A70	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0A74	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0A78	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_0A7C	MBC Memory Block Configuration Word (MBC0_DOM5_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0B40	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM0_BLK_NSE_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0B44	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0B48	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_0B4C	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0B80	MBC Memory Block Configuration Word (MBC0_DOM5_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0BA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_0BA8	MBC Memory Block Configuration Word (MBC0_DOM5_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_0BC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0BD0	MBC Memory Block Configuration Word (MBC0_DOM5_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_0BD4	MBC Memory Block Configuration Word (MBC0_DOM5_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_0BD8	MBC Memory Block Configuration Word (MBC0_DOM5_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_0BF0	MBC Memory Block NonSecure Enable Word (MBC0_DOM5_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_0C40	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0C44	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0C48	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_0C4C	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0C50	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0C54	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0C58	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W6)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0C5C	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0C60	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0C64	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0C68	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_0C6C	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0C70	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0C74	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0C78	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_0C7C	MBC Memory Block Configuration Word (MBC0_DOM6_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0D40	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0D44	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0D48	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_0D4C	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0D80	MBC Memory Block Configuration Word (MBC0_DOM6_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0DA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_0DA8	MBC Memory Block Configuration Word (MBC0_DOM6_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_0DC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0DD0	MBC Memory Block Configuration Word (MBC0_DOM6_MEM3_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0DD4	MBC Memory Block Configuration Word (MBC0_DOM6_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_0DD8	MBC Memory Block Configuration Word (MBC0_DOM6_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_0DF0	MBC Memory Block NonSecure Enable Word (MBC0_DOM6_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_0E40	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_0E44	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_0E48	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_0E4C	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_0E50	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_0E54	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_0E58	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_0E5C	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_0E60	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_0E64	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_0E68	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_0E6C	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_0E70	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_0E74	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_0E78	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W14)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_0E7C	MBC Memory Block Configuration Word (MBC0_DOM7_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_0F40	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_0F44	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_0F48	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_0F4C	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_0F80	MBC Memory Block Configuration Word (MBC0_DOM7_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_0FA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_0FA8	MBC Memory Block Configuration Word (MBC0_DOM7_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_0FC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_0FD0	MBC Memory Block Configuration Word (MBC0_DOM7_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_0FD4	MBC Memory Block Configuration Word (MBC0_DOM7_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_0FD8	MBC Memory Block Configuration Word (MBC0_DOM7_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_0FF0	MBC Memory Block NonSecure Enable Word (MBC0_DOM7_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_1040	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1044	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1048	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_104C	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1050	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W4)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1054	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1058	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_105C	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1060	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1064	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1068	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_106C	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1070	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1074	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1078	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_107C	MBC Memory Block Configuration Word (MBC0_DOM8_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1140	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1144	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1148	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_114C	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1180	MBC Memory Block Configuration Word (MBC0_DOM8_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_11A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_11A8	MBC Memory Block Configuration Word (MBC0_DOM8_MEM2_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_11C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_11D0	MBC Memory Block Configuration Word (MBC0_DOM8_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_11D4	MBC Memory Block Configuration Word (MBC0_DOM8_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_11D8	MBC Memory Block Configuration Word (MBC0_DOM8_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_11F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM8_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_1240	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1244	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1248	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_124C	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1250	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1254	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1258	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_125C	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1260	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1264	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1268	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_126C	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1270	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W12)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1274	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1278	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_127C	MBC Memory Block Configuration Word (MBC0_DOM9_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1340	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1344	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1348	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_134C	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1380	MBC Memory Block Configuration Word (MBC0_DOM9_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_13A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_13A8	MBC Memory Block Configuration Word (MBC0_DOM9_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_13C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_13D0	MBC Memory Block Configuration Word (MBC0_DOM9_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_13D4	MBC Memory Block Configuration Word (MBC0_DOM9_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_13D8	MBC Memory Block Configuration Word (MBC0_DOM9_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_13F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM9_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_1440	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1444	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1448	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_144C	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1450	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1454	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1458	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_145C	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1460	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1464	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1468	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_146C	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1470	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1474	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1478	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_147C	MBC Memory Block Configuration Word (MBC0_DOM10_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1540	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1544	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1548	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_154C	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1580	MBC Memory Block Configuration Word (MBC0_DOM10_MEM1_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_15A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_15A8	MBC Memory Block Configuration Word (MBC0_DOM10_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_15C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_15D0	MBC Memory Block Configuration Word (MBC0_DOM10_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_15D4	MBC Memory Block Configuration Word (MBC0_DOM10_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_15D8	MBC Memory Block Configuration Word (MBC0_DOM10_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_15F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM10_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_1640	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1644	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1648	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_164C	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1650	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1654	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1658	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_165C	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1660	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1664	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1668	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W10)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_166C	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1670	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1674	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1678	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_167C	MBC Memory Block Configuration Word (MBC0_DOM11_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1740	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1744	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1748	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_174C	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1780	MBC Memory Block Configuration Word (MBC0_DOM11_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_17A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_17A8	MBC Memory Block Configuration Word (MBC0_DOM11_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_17C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_17D0	MBC Memory Block Configuration Word (MBC0_DOM11_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_17D4	MBC Memory Block Configuration Word (MBC0_DOM11_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_17D8	MBC Memory Block Configuration Word (MBC0_DOM11_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_17F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM11_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_1840	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1844	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1848	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_184C	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1850	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1854	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1858	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_185C	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1860	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1864	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1868	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_186C	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1870	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1874	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1878	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_187C	MBC Memory Block Configuration Word (MBC0_DOM12_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1940	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1944	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1948	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM0_BLK_NSE_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_194C	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1980	MBC Memory Block Configuration Word (MBC0_DOM12_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_19A0	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_19A8	MBC Memory Block Configuration Word (MBC0_DOM12_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_19C8	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_19D0	MBC Memory Block Configuration Word (MBC0_DOM12_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_19D4	MBC Memory Block Configuration Word (MBC0_DOM12_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_19D8	MBC Memory Block Configuration Word (MBC0_DOM12_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_19F0	MBC Memory Block NonSecure Enable Word (MBC0_DOM12_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_1A40	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1A44	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1A48	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_1A4C	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1A50	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1A54	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1A58	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1A5C	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1A60	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W8)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1A64	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1A68	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_1A6C	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1A70	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1A74	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1A78	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_1A7C	MBC Memory Block Configuration Word (MBC0_DOM13_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1B40	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1B44	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1B48	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_1B4C	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1B80	MBC Memory Block Configuration Word (MBC0_DOM13_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1BA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_1BA8	MBC Memory Block Configuration Word (MBC0_DOM13_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_1BC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1BD0	MBC Memory Block Configuration Word (MBC0_DOM13_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_1BD4	MBC Memory Block Configuration Word (MBC0_DOM13_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_1BD8	MBC Memory Block Configuration Word (MBC0_DOM13_MEM3_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1BF0	MBC Memory Block NonSecure Enable Word (MBC0_DOM13_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_1C40	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1C44	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1C48	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_1C4C	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1C50	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1C54	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1C58	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_1C5C	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1C60	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1C64	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1C68	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_1C6C	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1C70	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1C74	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1C78	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_1C7C	MBC Memory Block Configuration Word (MBC0_DOM14_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1D40	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM0_BLK_NSE_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1D44	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1D48	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_1D4C	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1D80	MBC Memory Block Configuration Word (MBC0_DOM14_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1DA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_1DA8	MBC Memory Block Configuration Word (MBC0_DOM14_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_1DC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1DD0	MBC Memory Block Configuration Word (MBC0_DOM14_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_1DD4	MBC Memory Block Configuration Word (MBC0_DOM14_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_1DD8	MBC Memory Block Configuration Word (MBC0_DOM14_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_1DF0	MBC Memory Block NonSecure Enable Word (MBC0_DOM14_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_1E40	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_1E44	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_1E48	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_1E4C	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_1E50	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_1E54	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_1E58	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W6)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1E5C	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_1E60	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_1E64	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_1E68	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_1E6C	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_1E70	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_1E74	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_1E78	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_1E7C	MBC Memory Block Configuration Word (MBC0_DOM15_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_1F40	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_1F44	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_1F48	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_1F4C	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_1F80	MBC Memory Block Configuration Word (MBC0_DOM15_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_1FA0	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_1FA8	MBC Memory Block Configuration Word (MBC0_DOM15_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_1FC8	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_1FD0	MBC Memory Block Configuration Word (MBC0_DOM15_MEM3_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_1FD4	MBC Memory Block Configuration Word (MBC0_DOM15_MEM3_BLK_CFG_W1)	32	RW	0000_0000
1_1FD8	MBC Memory Block Configuration Word (MBC0_DOM15_MEM3_BLK_CFG_W2)	32	RW	0000_0000
1_1FF0	MBC Memory Block NonSecure Enable Word (MBC0_DOM15_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2000	MBC Global Configuration Register (MBC1_MEM0_GLBCFG)	32	R	000C_0080
1_2004	MBC Global Configuration Register (MBC1_MEM1_GLBCFG)	32	R	000C_0001
1_2008	MBC Global Configuration Register (MBC1_MEM2_GLBCFG)	32	R	000C_0001
1_200C	MBC Global Configuration Register (MBC1_MEM3_GLBCFG)	32	R	000C_0001
1_2010	MBC NonSecure Enable Block Index (MBC1_NSE_BLK_INDEX)	32	RW	0000_0000
1_2014	MBC NonSecure Enable Block Set (MBC1_NSE_BLK_SET)	32	RW	0000_0000
1_2018	MBC NonSecure Enable Block Clear (MBC1_NSE_BLK_CLR)	32	RW	0000_0000
1_201C	MBC NonSecure Enable Block Clear All (MBC1_NSE_BLK_CLR_ALL)	32	RW	0000_0000
1_2020	MBC Global Access Control (MBC1_MEMN_GLBAC0)	32	RW	0000_0000
1_2024	MBC Global Access Control (MBC1_MEMN_GLBAC1)	32	RW	0000_0000
1_2028	MBC Global Access Control (MBC1_MEMN_GLBAC2)	32	RW	0000_0000
1_202C	MBC Global Access Control (MBC1_MEMN_GLBAC3)	32	RW	0000_0000
1_2030	MBC Global Access Control (MBC1_MEMN_GLBAC4)	32	RW	0000_0000
1_2034	MBC Global Access Control (MBC1_MEMN_GLBAC5)	32	RW	0000_0000
1_2038	MBC Global Access Control (MBC1_MEMN_GLBAC6)	32	RW	0000_0000
1_203C	MBC Global Access Control (MBC1_MEMN_GLBAC7)	32	RW	0000_0000
1_2040	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2044	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2048	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_204C	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2050	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W4)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2054	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_2058	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_205C	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_2060	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_2064	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_2068	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_206C	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_2070	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_2074	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_2078	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_207C	MBC Memory Block Configuration Word (MBC1_DOM0_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_2140	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2144	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_2148	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_214C	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_2180	MBC Memory Block Configuration Word (MBC1_DOM0_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_21A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_21A8	MBC Memory Block Configuration Word (MBC1_DOM0_MEM2_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_21C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_21D0	MBC Memory Block Configuration Word (MBC1_DOM0_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_21F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM0_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2240	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2244	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2248	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_224C	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2250	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_2254	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_2258	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_225C	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_2260	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_2264	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_2268	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_226C	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_2270	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_2274	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_2278	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W14)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_227C	MBC Memory Block Configuration Word (MBC1_DOM1_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_2340	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2344	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_2348	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_234C	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_2380	MBC Memory Block Configuration Word (MBC1_DOM1_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_23A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_23A8	MBC Memory Block Configuration Word (MBC1_DOM1_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_23C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_23D0	MBC Memory Block Configuration Word (MBC1_DOM1_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_23F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM1_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2440	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2444	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2448	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_244C	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2450	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_2454	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_2458	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W6)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_245C	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_2460	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_2464	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_2468	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_246C	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_2470	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_2474	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_2478	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_247C	MBC Memory Block Configuration Word (MBC1_DOM2_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_2540	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2544	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_2548	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_254C	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_2580	MBC Memory Block Configuration Word (MBC1_DOM2_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_25A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_25A8	MBC Memory Block Configuration Word (MBC1_DOM2_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_25C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_25D0	MBC Memory Block Configuration Word (MBC1_DOM2_MEM3_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_25F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM2_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2640	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2644	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2648	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_264C	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2650	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_2654	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_2658	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_265C	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_2660	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_2664	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_2668	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_266C	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_2670	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_2674	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_2678	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_267C	MBC Memory Block Configuration Word (MBC1_DOM3_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_2740	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM0_BLK_NSE_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2744	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_2748	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_274C	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_2780	MBC Memory Block Configuration Word (MBC1_DOM3_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_27A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_27A8	MBC Memory Block Configuration Word (MBC1_DOM3_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_27C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_27D0	MBC Memory Block Configuration Word (MBC1_DOM3_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_27F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM3_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2840	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2844	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2848	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_284C	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2850	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_2854	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_2858	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_285C	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_2860	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W8)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2864	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_2868	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_286C	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_2870	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_2874	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_2878	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_287C	MBC Memory Block Configuration Word (MBC1_DOM4_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_2940	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2944	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_2948	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_294C	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_2980	MBC Memory Block Configuration Word (MBC1_DOM4_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_29A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_29A8	MBC Memory Block Configuration Word (MBC1_DOM4_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_29C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_29D0	MBC Memory Block Configuration Word (MBC1_DOM4_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_29F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM4_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2A40	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2A44	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2A48	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_2A4C	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2A50	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_2A54	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_2A58	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_2A5C	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_2A60	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_2A64	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_2A68	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_2A6C	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_2A70	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_2A74	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_2A78	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_2A7C	MBC Memory Block Configuration Word (MBC1_DOM5_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_2B40	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2B44	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_2B48	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM0_BLK_NSE_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2B4C	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_2B80	MBC Memory Block Configuration Word (MBC1_DOM5_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2BA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2BA8	MBC Memory Block Configuration Word (MBC1_DOM5_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_2BC8	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_2BD0	MBC Memory Block Configuration Word (MBC1_DOM5_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_2BF0	MBC Memory Block NonSecure Enable Word (MBC1_DOM5_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2C40	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2C44	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2C48	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_2C4C	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2C50	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_2C54	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_2C58	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_2C5C	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_2C60	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_2C64	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_2C68	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W10)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2C6C	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_2C70	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_2C74	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_2C78	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_2C7C	MBC Memory Block Configuration Word (MBC1_DOM6_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_2D40	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2D44	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_2D48	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_2D4C	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_2D80	MBC Memory Block Configuration Word (MBC1_DOM6_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_2DA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2DA8	MBC Memory Block Configuration Word (MBC1_DOM6_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_2DC8	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_2DD0	MBC Memory Block Configuration Word (MBC1_DOM6_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_2DF0	MBC Memory Block NonSecure Enable Word (MBC1_DOM6_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_2E40	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_2E44	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_2E48	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2E4C	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_2E50	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_2E54	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_2E58	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_2E5C	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_2E60	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_2E64	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_2E68	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_2E6C	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_2E70	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_2E74	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_2E78	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_2E7C	MBC Memory Block Configuration Word (MBC1_DOM7_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_2F40	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_2F44	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_2F48	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_2F4C	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_2F80	MBC Memory Block Configuration Word (MBC1_DOM7_MEM1_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_2FA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_2FA8	MBC Memory Block Configuration Word (MBC1_DOM7_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_2FC8	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_2FD0	MBC Memory Block Configuration Word (MBC1_DOM7_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_2FF0	MBC Memory Block NonSecure Enable Word (MBC1_DOM7_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3040	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3044	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3048	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_304C	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3050	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_3054	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_3058	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_305C	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_3060	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_3064	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_3068	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_306C	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_3070	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W12)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_3074	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_3078	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_307C	MBC Memory Block Configuration Word (MBC1_DOM8_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_3140	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3144	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_3148	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_314C	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_3180	MBC Memory Block Configuration Word (MBC1_DOM8_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_31A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_31A8	MBC Memory Block Configuration Word (MBC1_DOM8_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_31C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_31D0	MBC Memory Block Configuration Word (MBC1_DOM8_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_31F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM8_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3240	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3244	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3248	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_324C	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3250	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W4)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_3254	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_3258	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_325C	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_3260	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_3264	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_3268	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_326C	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_3270	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_3274	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_3278	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_327C	MBC Memory Block Configuration Word (MBC1_DOM9_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_3340	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3344	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_3348	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_334C	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_3380	MBC Memory Block Configuration Word (MBC1_DOM9_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_33A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_33A8	MBC Memory Block Configuration Word (MBC1_DOM9_MEM2_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_33C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_33D0	MBC Memory Block Configuration Word (MBC1_DOM9_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_33F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM9_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3440	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3444	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3448	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_344C	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3450	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_3454	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_3458	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_345C	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_3460	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_3464	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_3468	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_346C	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_3470	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_3474	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_3478	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W14)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_347C	MBC Memory Block Configuration Word (MBC1_DOM10_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_3540	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3544	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_3548	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_354C	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_3580	MBC Memory Block Configuration Word (MBC1_DOM10_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_35A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_35A8	MBC Memory Block Configuration Word (MBC1_DOM10_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_35C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_35D0	MBC Memory Block Configuration Word (MBC1_DOM10_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_35F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM10_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3640	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3644	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3648	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_364C	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3650	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_3654	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_3658	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W6)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_365C	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_3660	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_3664	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_3668	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_366C	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_3670	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_3674	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_3678	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_367C	MBC Memory Block Configuration Word (MBC1_DOM11_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_3740	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3744	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_3748	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_374C	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_3780	MBC Memory Block Configuration Word (MBC1_DOM11_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_37A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_37A8	MBC Memory Block Configuration Word (MBC1_DOM11_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_37C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_37D0	MBC Memory Block Configuration Word (MBC1_DOM11_MEM3_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_37F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM11_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3840	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3844	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3848	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_384C	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3850	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_3854	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_3858	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_385C	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_3860	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_3864	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_3868	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_386C	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_3870	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_3874	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_3878	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_387C	MBC Memory Block Configuration Word (MBC1_DOM12_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_3940	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM0_BLK_NSE_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_3944	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_3948	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_394C	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_3980	MBC Memory Block Configuration Word (MBC1_DOM12_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_39A0	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_39A8	MBC Memory Block Configuration Word (MBC1_DOM12_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_39C8	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_39D0	MBC Memory Block Configuration Word (MBC1_DOM12_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_39F0	MBC Memory Block NonSecure Enable Word (MBC1_DOM12_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3A40	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3A44	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3A48	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_3A4C	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3A50	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_3A54	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_3A58	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_3A5C	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_3A60	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W8)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_3A64	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_3A68	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_3A6C	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_3A70	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_3A74	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_3A78	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_3A7C	MBC Memory Block Configuration Word (MBC1_DOM13_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_3B40	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3B44	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_3B48	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_3B4C	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_3B80	MBC Memory Block Configuration Word (MBC1_DOM13_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3BA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3BA8	MBC Memory Block Configuration Word (MBC1_DOM13_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_3BC8	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_3BD0	MBC Memory Block Configuration Word (MBC1_DOM13_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_3BF0	MBC Memory Block NonSecure Enable Word (MBC1_DOM13_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3C40	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_3C44	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3C48	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_3C4C	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3C50	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_3C54	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_3C58	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_3C5C	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_3C60	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_3C64	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_3C68	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W10)	32	RW	0000_0000
1_3C6C	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_3C70	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_3C74	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_3C78	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_3C7C	MBC Memory Block Configuration Word (MBC1_DOM14_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_3D40	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3D44	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_3D48	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM0_BLK_NSE_W2)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_3D4C	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_3D80	MBC Memory Block Configuration Word (MBC1_DOM14_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3DA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3DA8	MBC Memory Block Configuration Word (MBC1_DOM14_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_3DC8	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_3DD0	MBC Memory Block Configuration Word (MBC1_DOM14_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_3DF0	MBC Memory Block NonSecure Enable Word (MBC1_DOM14_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_3E40	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W0)	32	RW	0000_0000
1_3E44	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W1)	32	RW	0000_0000
1_3E48	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W2)	32	RW	0000_0000
1_3E4C	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W3)	32	RW	0000_0000
1_3E50	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W4)	32	RW	0000_0000
1_3E54	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W5)	32	RW	0000_0000
1_3E58	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W6)	32	RW	0000_0000
1_3E5C	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W7)	32	RW	0000_0000
1_3E60	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W8)	32	RW	0000_0000
1_3E64	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W9)	32	RW	0000_0000
1_3E68	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W10)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_3E6C	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W11)	32	RW	0000_0000
1_3E70	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W12)	32	RW	0000_0000
1_3E74	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W13)	32	RW	0000_0000
1_3E78	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W14)	32	RW	0000_0000
1_3E7C	MBC Memory Block Configuration Word (MBC1_DOM15_MEM0_BLK_CFG_W15)	32	RW	0000_0000
1_3F40	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM0_BLK_NSE_W0)	32	RW	0000_0000
1_3F44	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM0_BLK_NSE_W1)	32	RW	0000_0000
1_3F48	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM0_BLK_NSE_W2)	32	RW	0000_0000
1_3F4C	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM0_BLK_NSE_W3)	32	RW	0000_0000
1_3F80	MBC Memory Block Configuration Word (MBC1_DOM15_MEM1_BLK_CFG_W0)	32	RW	0000_0000
1_3FA0	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM1_BLK_NSE_W0)	32	RW	0000_0000
1_3FA8	MBC Memory Block Configuration Word (MBC1_DOM15_MEM2_BLK_CFG_W0)	32	RW	0000_0000
1_3FC8	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM2_BLK_NSE_W0)	32	RW	0000_0000
1_3FD0	MBC Memory Block Configuration Word (MBC1_DOM15_MEM3_BLK_CFG_W0)	32	RW	0000_0000
1_3FF0	MBC Memory Block NonSecure Enable Word (MBC1_DOM15_MEM3_BLK_NSE_W0)	32	RW	0000_0000
1_4000	MRC Global Configuration Register (MRC0_GLBCFG)	32	R	0000_0008
1_4010	MRC NonSecure Enable Region Indirect (MRC0_NSE_RGN_INDIRECT)	32	RW	0000_0000
1_4014	MRC NonSecure Enable Region Set (MRC0_NSE_RGN_SET)	32	RW	0000_0000
1_4018	MRC NonSecure Enable Region Clear (MRC0_NSE_RGN_CLR)	32	RW	0000_0000
1_401C	MRC NonSecure Enable Region Clear All (MRC0_NSE_RGN_CLR_ALL)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4020	MRC Global Access Control (MRC0_GLBAC0)	32	RW	0000_0000
1_4024	MRC Global Access Control (MRC0_GLBAC1)	32	RW	0000_0000
1_4028	MRC Global Access Control (MRC0_GLBAC2)	32	RW	0000_0000
1_402C	MRC Global Access Control (MRC0_GLBAC3)	32	RW	0000_0000
1_4030	MRC Global Access Control (MRC0_GLBAC4)	32	RW	0000_0000
1_4034	MRC Global Access Control (MRC0_GLBAC5)	32	RW	0000_0000
1_4038	MRC Global Access Control (MRC0_GLBAC6)	32	RW	0000_0000
1_403C	MRC Global Access Control (MRC0_GLBAC7)	32	RW	0000_0000
1_4040	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD0_W0)	32	RW	0000_0000
1_4044	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD0_W1)	32	RW	0000_0000
1_4048	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD1_W0)	32	RW	0000_0000
1_404C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD1_W1)	32	RW	0000_0000
1_4050	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD2_W0)	32	RW	0000_0000
1_4054	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD2_W1)	32	RW	0000_0000
1_4058	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD3_W0)	32	RW	0000_0000
1_405C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD3_W1)	32	RW	0000_0000
1_4060	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD4_W0)	32	RW	0000_0000
1_4064	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD4_W1)	32	RW	0000_0000
1_4068	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD5_W0)	32	RW	0000_0000
1_406C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD5_W1)	32	RW	0000_0000
1_4070	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD6_W0)	32	RW	0000_0000
1_4074	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD6_W1)	32	RW	0000_0000
1_4078	MRC Region Descriptor Word 0 (MRC0_DOM0_RGD7_W0)	32	RW	0000_0000
1_407C	MRC Region Descriptor Word 1 (MRC0_DOM0_RGD7_W1)	32	RW	0000_0000
1_40C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM0_RGD_NSE)	32	RW	0000_0000
1_4140	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD0_W0)	32	RW	0000_0000
1_4144	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD0_W1)	32	RW	0000_0000
1_4148	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD1_W0)	32	RW	0000_0000
1_414C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD1_W1)	32	RW	0000_0000
1_4150	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD2_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4154	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD2_W1)	32	RW	0000_0000
1_4158	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD3_W0)	32	RW	0000_0000
1_415C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD3_W1)	32	RW	0000_0000
1_4160	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD4_W0)	32	RW	0000_0000
1_4164	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD4_W1)	32	RW	0000_0000
1_4168	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD5_W0)	32	RW	0000_0000
1_416C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD5_W1)	32	RW	0000_0000
1_4170	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD6_W0)	32	RW	0000_0000
1_4174	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD6_W1)	32	RW	0000_0000
1_4178	MRC Region Descriptor Word 0 (MRC0_DOM1_RGD7_W0)	32	RW	0000_0000
1_417C	MRC Region Descriptor Word 1 (MRC0_DOM1_RGD7_W1)	32	RW	0000_0000
1_41C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM1_RGD_NSE)	32	RW	0000_0000
1_4240	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD0_W0)	32	RW	0000_0000
1_4244	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD0_W1)	32	RW	0000_0000
1_4248	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD1_W0)	32	RW	0000_0000
1_424C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD1_W1)	32	RW	0000_0000
1_4250	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD2_W0)	32	RW	0000_0000
1_4254	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD2_W1)	32	RW	0000_0000
1_4258	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD3_W0)	32	RW	0000_0000
1_425C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD3_W1)	32	RW	0000_0000
1_4260	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD4_W0)	32	RW	0000_0000
1_4264	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD4_W1)	32	RW	0000_0000
1_4268	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD5_W0)	32	RW	0000_0000
1_426C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD5_W1)	32	RW	0000_0000
1_4270	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD6_W0)	32	RW	0000_0000
1_4274	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD6_W1)	32	RW	0000_0000
1_4278	MRC Region Descriptor Word 0 (MRC0_DOM2_RGD7_W0)	32	RW	0000_0000
1_427C	MRC Region Descriptor Word 1 (MRC0_DOM2_RGD7_W1)	32	RW	0000_0000
1_42C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM2_RGD_NSE)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4340	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD0_W0)	32	RW	0000_0000
1_4344	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD0_W1)	32	RW	0000_0000
1_4348	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD1_W0)	32	RW	0000_0000
1_434C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD1_W1)	32	RW	0000_0000
1_4350	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD2_W0)	32	RW	0000_0000
1_4354	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD2_W1)	32	RW	0000_0000
1_4358	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD3_W0)	32	RW	0000_0000
1_435C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD3_W1)	32	RW	0000_0000
1_4360	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD4_W0)	32	RW	0000_0000
1_4364	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD4_W1)	32	RW	0000_0000
1_4368	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD5_W0)	32	RW	0000_0000
1_436C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD5_W1)	32	RW	0000_0000
1_4370	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD6_W0)	32	RW	0000_0000
1_4374	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD6_W1)	32	RW	0000_0000
1_4378	MRC Region Descriptor Word 0 (MRC0_DOM3_RGD7_W0)	32	RW	0000_0000
1_437C	MRC Region Descriptor Word 1 (MRC0_DOM3_RGD7_W1)	32	RW	0000_0000
1_43C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM3_RGD_NSE)	32	RW	0000_0000
1_4440	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD0_W0)	32	RW	0000_0000
1_4444	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD0_W1)	32	RW	0000_0000
1_4448	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD1_W0)	32	RW	0000_0000
1_444C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD1_W1)	32	RW	0000_0000
1_4450	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD2_W0)	32	RW	0000_0000
1_4454	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD2_W1)	32	RW	0000_0000
1_4458	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD3_W0)	32	RW	0000_0000
1_445C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD3_W1)	32	RW	0000_0000
1_4460	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD4_W0)	32	RW	0000_0000
1_4464	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD4_W1)	32	RW	0000_0000
1_4468	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD5_W0)	32	RW	0000_0000
1_446C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD5_W1)	32	RW	0000_0000
1_4470	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD6_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4474	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD6_W1)	32	RW	0000_0000
1_4478	MRC Region Descriptor Word 0 (MRC0_DOM4_RGD7_W0)	32	RW	0000_0000
1_447C	MRC Region Descriptor Word 1 (MRC0_DOM4_RGD7_W1)	32	RW	0000_0000
1_44C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM4_RGD_NSE)	32	RW	0000_0000
1_4540	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD0_W0)	32	RW	0000_0000
1_4544	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD0_W1)	32	RW	0000_0000
1_4548	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD1_W0)	32	RW	0000_0000
1_454C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD1_W1)	32	RW	0000_0000
1_4550	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD2_W0)	32	RW	0000_0000
1_4554	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD2_W1)	32	RW	0000_0000
1_4558	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD3_W0)	32	RW	0000_0000
1_455C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD3_W1)	32	RW	0000_0000
1_4560	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD4_W0)	32	RW	0000_0000
1_4564	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD4_W1)	32	RW	0000_0000
1_4568	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD5_W0)	32	RW	0000_0000
1_456C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD5_W1)	32	RW	0000_0000
1_4570	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD6_W0)	32	RW	0000_0000
1_4574	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD6_W1)	32	RW	0000_0000
1_4578	MRC Region Descriptor Word 0 (MRC0_DOM5_RGD7_W0)	32	RW	0000_0000
1_457C	MRC Region Descriptor Word 1 (MRC0_DOM5_RGD7_W1)	32	RW	0000_0000
1_45C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM5_RGD_NSE)	32	RW	0000_0000
1_4640	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD0_W0)	32	RW	0000_0000
1_4644	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD0_W1)	32	RW	0000_0000
1_4648	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD1_W0)	32	RW	0000_0000
1_464C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD1_W1)	32	RW	0000_0000
1_4650	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD2_W0)	32	RW	0000_0000
1_4654	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD2_W1)	32	RW	0000_0000
1_4658	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD3_W0)	32	RW	0000_0000
1_465C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD3_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4660	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD4_W0)	32	RW	0000_0000
1_4664	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD4_W1)	32	RW	0000_0000
1_4668	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD5_W0)	32	RW	0000_0000
1_466C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD5_W1)	32	RW	0000_0000
1_4670	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD6_W0)	32	RW	0000_0000
1_4674	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD6_W1)	32	RW	0000_0000
1_4678	MRC Region Descriptor Word 0 (MRC0_DOM6_RGD7_W0)	32	RW	0000_0000
1_467C	MRC Region Descriptor Word 1 (MRC0_DOM6_RGD7_W1)	32	RW	0000_0000
1_46C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM6_RGD_NSE)	32	RW	0000_0000
1_4740	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD0_W0)	32	RW	0000_0000
1_4744	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD0_W1)	32	RW	0000_0000
1_4748	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD1_W0)	32	RW	0000_0000
1_474C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD1_W1)	32	RW	0000_0000
1_4750	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD2_W0)	32	RW	0000_0000
1_4754	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD2_W1)	32	RW	0000_0000
1_4758	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD3_W0)	32	RW	0000_0000
1_475C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD3_W1)	32	RW	0000_0000
1_4760	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD4_W0)	32	RW	0000_0000
1_4764	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD4_W1)	32	RW	0000_0000
1_4768	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD5_W0)	32	RW	0000_0000
1_476C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD5_W1)	32	RW	0000_0000
1_4770	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD6_W0)	32	RW	0000_0000
1_4774	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD6_W1)	32	RW	0000_0000
1_4778	MRC Region Descriptor Word 0 (MRC0_DOM7_RGD7_W0)	32	RW	0000_0000
1_477C	MRC Region Descriptor Word 1 (MRC0_DOM7_RGD7_W1)	32	RW	0000_0000
1_47C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM7_RGD_NSE)	32	RW	0000_0000
1_4840	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD0_W0)	32	RW	0000_0000
1_4844	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD0_W1)	32	RW	0000_0000
1_4848	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD1_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_484C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD1_W1)	32	RW	0000_0000
1_4850	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD2_W0)	32	RW	0000_0000
1_4854	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD2_W1)	32	RW	0000_0000
1_4858	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD3_W0)	32	RW	0000_0000
1_485C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD3_W1)	32	RW	0000_0000
1_4860	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD4_W0)	32	RW	0000_0000
1_4864	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD4_W1)	32	RW	0000_0000
1_4868	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD5_W0)	32	RW	0000_0000
1_486C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD5_W1)	32	RW	0000_0000
1_4870	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD6_W0)	32	RW	0000_0000
1_4874	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD6_W1)	32	RW	0000_0000
1_4878	MRC Region Descriptor Word 0 (MRC0_DOM8_RGD7_W0)	32	RW	0000_0000
1_487C	MRC Region Descriptor Word 1 (MRC0_DOM8_RGD7_W1)	32	RW	0000_0000
1_48C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM8_RGD_NSE)	32	RW	0000_0000
1_4940	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD0_W0)	32	RW	0000_0000
1_4944	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD0_W1)	32	RW	0000_0000
1_4948	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD1_W0)	32	RW	0000_0000
1_494C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD1_W1)	32	RW	0000_0000
1_4950	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD2_W0)	32	RW	0000_0000
1_4954	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD2_W1)	32	RW	0000_0000
1_4958	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD3_W0)	32	RW	0000_0000
1_495C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD3_W1)	32	RW	0000_0000
1_4960	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD4_W0)	32	RW	0000_0000
1_4964	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD4_W1)	32	RW	0000_0000
1_4968	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD5_W0)	32	RW	0000_0000
1_496C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD5_W1)	32	RW	0000_0000
1_4970	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD6_W0)	32	RW	0000_0000
1_4974	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD6_W1)	32	RW	0000_0000
1_4978	MRC Region Descriptor Word 0 (MRC0_DOM9_RGD7_W0)	32	RW	0000_0000
1_497C	MRC Region Descriptor Word 1 (MRC0_DOM9_RGD7_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_49C0	MRC Region Descriptor NonSecure Enable (MRC0_DOM9_RGD_NSE)	32	RW	0000_0000
1_4A40	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD0_W0)	32	RW	0000_0000
1_4A44	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD0_W1)	32	RW	0000_0000
1_4A48	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD1_W0)	32	RW	0000_0000
1_4A4C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD1_W1)	32	RW	0000_0000
1_4A50	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD2_W0)	32	RW	0000_0000
1_4A54	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD2_W1)	32	RW	0000_0000
1_4A58	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD3_W0)	32	RW	0000_0000
1_4A5C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD3_W1)	32	RW	0000_0000
1_4A60	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD4_W0)	32	RW	0000_0000
1_4A64	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD4_W1)	32	RW	0000_0000
1_4A68	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD5_W0)	32	RW	0000_0000
1_4A6C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD5_W1)	32	RW	0000_0000
1_4A70	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD6_W0)	32	RW	0000_0000
1_4A74	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD6_W1)	32	RW	0000_0000
1_4A78	MRC Region Descriptor Word 0 (MRC0_DOM10_RGD7_W0)	32	RW	0000_0000
1_4A7C	MRC Region Descriptor Word 1 (MRC0_DOM10_RGD7_W1)	32	RW	0000_0000
1_4AC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM10_RGD_NSE)	32	RW	0000_0000
1_4B40	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD0_W0)	32	RW	0000_0000
1_4B44	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD0_W1)	32	RW	0000_0000
1_4B48	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD1_W0)	32	RW	0000_0000
1_4B4C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD1_W1)	32	RW	0000_0000
1_4B50	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD2_W0)	32	RW	0000_0000
1_4B54	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD2_W1)	32	RW	0000_0000
1_4B58	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD3_W0)	32	RW	0000_0000
1_4B5C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD3_W1)	32	RW	0000_0000
1_4B60	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD4_W0)	32	RW	0000_0000
1_4B64	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD4_W1)	32	RW	0000_0000
1_4B68	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD5_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4B6C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD5_W1)	32	RW	0000_0000
1_4B70	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD6_W0)	32	RW	0000_0000
1_4B74	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD6_W1)	32	RW	0000_0000
1_4B78	MRC Region Descriptor Word 0 (MRC0_DOM11_RGD7_W0)	32	RW	0000_0000
1_4B7C	MRC Region Descriptor Word 1 (MRC0_DOM11_RGD7_W1)	32	RW	0000_0000
1_4BC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM11_RGD_NSE)	32	RW	0000_0000
1_4C40	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD0_W0)	32	RW	0000_0000
1_4C44	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD0_W1)	32	RW	0000_0000
1_4C48	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD1_W0)	32	RW	0000_0000
1_4C4C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD1_W1)	32	RW	0000_0000
1_4C50	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD2_W0)	32	RW	0000_0000
1_4C54	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD2_W1)	32	RW	0000_0000
1_4C58	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD3_W0)	32	RW	0000_0000
1_4C5C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD3_W1)	32	RW	0000_0000
1_4C60	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD4_W0)	32	RW	0000_0000
1_4C64	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD4_W1)	32	RW	0000_0000
1_4C68	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD5_W0)	32	RW	0000_0000
1_4C6C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD5_W1)	32	RW	0000_0000
1_4C70	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD6_W0)	32	RW	0000_0000
1_4C74	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD6_W1)	32	RW	0000_0000
1_4C78	MRC Region Descriptor Word 0 (MRC0_DOM12_RGD7_W0)	32	RW	0000_0000
1_4C7C	MRC Region Descriptor Word 1 (MRC0_DOM12_RGD7_W1)	32	RW	0000_0000
1_4CC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM12_RGD_NSE)	32	RW	0000_0000
1_4D40	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD0_W0)	32	RW	0000_0000
1_4D44	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD0_W1)	32	RW	0000_0000
1_4D48	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD1_W0)	32	RW	0000_0000
1_4D4C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD1_W1)	32	RW	0000_0000
1_4D50	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD2_W0)	32	RW	0000_0000
1_4D54	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD2_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4D58	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD3_W0)	32	RW	0000_0000
1_4D5C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD3_W1)	32	RW	0000_0000
1_4D60	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD4_W0)	32	RW	0000_0000
1_4D64	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD4_W1)	32	RW	0000_0000
1_4D68	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD5_W0)	32	RW	0000_0000
1_4D6C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD5_W1)	32	RW	0000_0000
1_4D70	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD6_W0)	32	RW	0000_0000
1_4D74	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD6_W1)	32	RW	0000_0000
1_4D78	MRC Region Descriptor Word 0 (MRC0_DOM13_RGD7_W0)	32	RW	0000_0000
1_4D7C	MRC Region Descriptor Word 1 (MRC0_DOM13_RGD7_W1)	32	RW	0000_0000
1_4DC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM13_RGD_NSE)	32	RW	0000_0000
1_4E40	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD0_W0)	32	RW	0000_0000
1_4E44	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD0_W1)	32	RW	0000_0000
1_4E48	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD1_W0)	32	RW	0000_0000
1_4E4C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD1_W1)	32	RW	0000_0000
1_4E50	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD2_W0)	32	RW	0000_0000
1_4E54	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD2_W1)	32	RW	0000_0000
1_4E58	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD3_W0)	32	RW	0000_0000
1_4E5C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD3_W1)	32	RW	0000_0000
1_4E60	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD4_W0)	32	RW	0000_0000
1_4E64	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD4_W1)	32	RW	0000_0000
1_4E68	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD5_W0)	32	RW	0000_0000
1_4E6C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD5_W1)	32	RW	0000_0000
1_4E70	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD6_W0)	32	RW	0000_0000
1_4E74	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD6_W1)	32	RW	0000_0000
1_4E78	MRC Region Descriptor Word 0 (MRC0_DOM14_RGD7_W0)	32	RW	0000_0000
1_4E7C	MRC Region Descriptor Word 1 (MRC0_DOM14_RGD7_W1)	32	RW	0000_0000
1_4EC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM14_RGD_NSE)	32	RW	0000_0000
1_4F40	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD0_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_4F44	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD0_W1)	32	RW	0000_0000
1_4F48	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD1_W0)	32	RW	0000_0000
1_4F4C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD1_W1)	32	RW	0000_0000
1_4F50	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD2_W0)	32	RW	0000_0000
1_4F54	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD2_W1)	32	RW	0000_0000
1_4F58	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD3_W0)	32	RW	0000_0000
1_4F5C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD3_W1)	32	RW	0000_0000
1_4F60	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD4_W0)	32	RW	0000_0000
1_4F64	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD4_W1)	32	RW	0000_0000
1_4F68	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD5_W0)	32	RW	0000_0000
1_4F6C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD5_W1)	32	RW	0000_0000
1_4F70	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD6_W0)	32	RW	0000_0000
1_4F74	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD6_W1)	32	RW	0000_0000
1_4F78	MRC Region Descriptor Word 0 (MRC0_DOM15_RGD7_W0)	32	RW	0000_0000
1_4F7C	MRC Region Descriptor Word 1 (MRC0_DOM15_RGD7_W1)	32	RW	0000_0000
1_4FC0	MRC Region Descriptor NonSecure Enable (MRC0_DOM15_RGD_NSE)	32	RW	0000_0000
1_5000	MRC Global Configuration Register (MRC1_GLBCFG)	32	R	0000_0008
1_5010	MRC NonSecure Enable Region Indirect (MRC1_NSE_RGN_INDIRECT)	32	RW	0000_0000
1_5014	MRC NonSecure Enable Region Set (MRC1_NSE_RGN_SET)	32	RW	0000_0000
1_5018	MRC NonSecure Enable Region Clear (MRC1_NSE_RGN_CLR)	32	RW	0000_0000
1_501C	MRC NonSecure Enable Region Clear All (MRC1_NSE_RGN_CLR_ALL)	32	RW	0000_0000
1_5020	MRC Global Access Control (MRC1_GLBAC0)	32	RW	0000_0000
1_5024	MRC Global Access Control (MRC1_GLBAC1)	32	RW	0000_0000
1_5028	MRC Global Access Control (MRC1_GLBAC2)	32	RW	0000_0000
1_502C	MRC Global Access Control (MRC1_GLBAC3)	32	RW	0000_0000
1_5030	MRC Global Access Control (MRC1_GLBAC4)	32	RW	0000_0000
1_5034	MRC Global Access Control (MRC1_GLBAC5)	32	RW	0000_0000
1_5038	MRC Global Access Control (MRC1_GLBAC6)	32	RW	0000_0000
1_503C	MRC Global Access Control (MRC1_GLBAC7)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_5040	MRC Region Descriptor Word 0 (MRC1_DOM0_RGD0_W0)	32	RW	0000_0000
1_5044	MRC Region Descriptor Word 1 (MRC1_DOM0_RGD0_W1)	32	RW	0000_0000
1_5048	MRC Region Descriptor Word 0 (MRC1_DOM0_RGD1_W0)	32	RW	0000_0000
1_504C	MRC Region Descriptor Word 1 (MRC1_DOM0_RGD1_W1)	32	RW	0000_0000
1_5050	MRC Region Descriptor Word 0 (MRC1_DOM0_RGD2_W0)	32	RW	0000_0000
1_5054	MRC Region Descriptor Word 1 (MRC1_DOM0_RGD2_W1)	32	RW	0000_0000
1_5058	MRC Region Descriptor Word 0 (MRC1_DOM0_RGD3_W0)	32	RW	0000_0000
1_505C	MRC Region Descriptor Word 1 (MRC1_DOM0_RGD3_W1)	32	RW	0000_0000
1_5060	MRC Region Descriptor Word 0 (MRC1_DOM0_RGD4_W0)	32	RW	0000_0000
1_5064	MRC Region Descriptor Word 1 (MRC1_DOM0_RGD4_W1)	32	RW	0000_0000
1_5068	MRC Region Descriptor Word 0 (MRC1_DOM0_RGD5_W0)	32	RW	0000_0000
1_506C	MRC Region Descriptor Word 1 (MRC1_DOM0_RGD5_W1)	32	RW	0000_0000
1_5070	MRC Region Descriptor Word 0 (MRC1_DOM0_RGD6_W0)	32	RW	0000_0000
1_5074	MRC Region Descriptor Word 1 (MRC1_DOM0_RGD6_W1)	32	RW	0000_0000
1_5078	MRC Region Descriptor Word 0 (MRC1_DOM0_RGD7_W0)	32	RW	0000_0000
1_507C	MRC Region Descriptor Word 1 (MRC1_DOM0_RGD7_W1)	32	RW	0000_0000
1_50C0	MRC Region Descriptor NonSecure Enable (MRC1_DOM0_RGD_NSE)	32	RW	0000_0000
1_5140	MRC Region Descriptor Word 0 (MRC1_DOM1_RGD0_W0)	32	RW	0000_0000
1_5144	MRC Region Descriptor Word 1 (MRC1_DOM1_RGD0_W1)	32	RW	0000_0000
1_5148	MRC Region Descriptor Word 0 (MRC1_DOM1_RGD1_W0)	32	RW	0000_0000
1_514C	MRC Region Descriptor Word 1 (MRC1_DOM1_RGD1_W1)	32	RW	0000_0000
1_5150	MRC Region Descriptor Word 0 (MRC1_DOM1_RGD2_W0)	32	RW	0000_0000
1_5154	MRC Region Descriptor Word 1 (MRC1_DOM1_RGD2_W1)	32	RW	0000_0000
1_5158	MRC Region Descriptor Word 0 (MRC1_DOM1_RGD3_W0)	32	RW	0000_0000
1_515C	MRC Region Descriptor Word 1 (MRC1_DOM1_RGD3_W1)	32	RW	0000_0000
1_5160	MRC Region Descriptor Word 0 (MRC1_DOM1_RGD4_W0)	32	RW	0000_0000
1_5164	MRC Region Descriptor Word 1 (MRC1_DOM1_RGD4_W1)	32	RW	0000_0000
1_5168	MRC Region Descriptor Word 0 (MRC1_DOM1_RGD5_W0)	32	RW	0000_0000
1_516C	MRC Region Descriptor Word 1 (MRC1_DOM1_RGD5_W1)	32	RW	0000_0000
1_5170	MRC Region Descriptor Word 0 (MRC1_DOM1_RGD6_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_5174	MRC Region Descriptor Word 1 (MRC1_DOM1_RGD6_W1)	32	RW	0000_0000
1_5178	MRC Region Descriptor Word 0 (MRC1_DOM1_RGD7_W0)	32	RW	0000_0000
1_517C	MRC Region Descriptor Word 1 (MRC1_DOM1_RGD7_W1)	32	RW	0000_0000
1_51C0	MRC Region Descriptor NonSecure Enable (MRC1_DOM1_RGD_NSE)	32	RW	0000_0000
1_5240	MRC Region Descriptor Word 0 (MRC1_DOM2_RGD0_W0)	32	RW	0000_0000
1_5244	MRC Region Descriptor Word 1 (MRC1_DOM2_RGD0_W1)	32	RW	0000_0000
1_5248	MRC Region Descriptor Word 0 (MRC1_DOM2_RGD1_W0)	32	RW	0000_0000
1_524C	MRC Region Descriptor Word 1 (MRC1_DOM2_RGD1_W1)	32	RW	0000_0000
1_5250	MRC Region Descriptor Word 0 (MRC1_DOM2_RGD2_W0)	32	RW	0000_0000
1_5254	MRC Region Descriptor Word 1 (MRC1_DOM2_RGD2_W1)	32	RW	0000_0000
1_5258	MRC Region Descriptor Word 0 (MRC1_DOM2_RGD3_W0)	32	RW	0000_0000
1_525C	MRC Region Descriptor Word 1 (MRC1_DOM2_RGD3_W1)	32	RW	0000_0000
1_5260	MRC Region Descriptor Word 0 (MRC1_DOM2_RGD4_W0)	32	RW	0000_0000
1_5264	MRC Region Descriptor Word 1 (MRC1_DOM2_RGD4_W1)	32	RW	0000_0000
1_5268	MRC Region Descriptor Word 0 (MRC1_DOM2_RGD5_W0)	32	RW	0000_0000
1_526C	MRC Region Descriptor Word 1 (MRC1_DOM2_RGD5_W1)	32	RW	0000_0000
1_5270	MRC Region Descriptor Word 0 (MRC1_DOM2_RGD6_W0)	32	RW	0000_0000
1_5274	MRC Region Descriptor Word 1 (MRC1_DOM2_RGD6_W1)	32	RW	0000_0000
1_5278	MRC Region Descriptor Word 0 (MRC1_DOM2_RGD7_W0)	32	RW	0000_0000
1_527C	MRC Region Descriptor Word 1 (MRC1_DOM2_RGD7_W1)	32	RW	0000_0000
1_52C0	MRC Region Descriptor NonSecure Enable (MRC1_DOM2_RGD_NSE)	32	RW	0000_0000
1_5340	MRC Region Descriptor Word 0 (MRC1_DOM3_RGD0_W0)	32	RW	0000_0000
1_5344	MRC Region Descriptor Word 1 (MRC1_DOM3_RGD0_W1)	32	RW	0000_0000
1_5348	MRC Region Descriptor Word 0 (MRC1_DOM3_RGD1_W0)	32	RW	0000_0000
1_534C	MRC Region Descriptor Word 1 (MRC1_DOM3_RGD1_W1)	32	RW	0000_0000
1_5350	MRC Region Descriptor Word 0 (MRC1_DOM3_RGD2_W0)	32	RW	0000_0000
1_5354	MRC Region Descriptor Word 1 (MRC1_DOM3_RGD2_W1)	32	RW	0000_0000
1_5358	MRC Region Descriptor Word 0 (MRC1_DOM3_RGD3_W0)	32	RW	0000_0000
1_535C	MRC Region Descriptor Word 1 (MRC1_DOM3_RGD3_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_5360	MRC Region Descriptor Word 0 (MRC1_DOM3_RGD4_W0)	32	RW	0000_0000
1_5364	MRC Region Descriptor Word 1 (MRC1_DOM3_RGD4_W1)	32	RW	0000_0000
1_5368	MRC Region Descriptor Word 0 (MRC1_DOM3_RGD5_W0)	32	RW	0000_0000
1_536C	MRC Region Descriptor Word 1 (MRC1_DOM3_RGD5_W1)	32	RW	0000_0000
1_5370	MRC Region Descriptor Word 0 (MRC1_DOM3_RGD6_W0)	32	RW	0000_0000
1_5374	MRC Region Descriptor Word 1 (MRC1_DOM3_RGD6_W1)	32	RW	0000_0000
1_5378	MRC Region Descriptor Word 0 (MRC1_DOM3_RGD7_W0)	32	RW	0000_0000
1_537C	MRC Region Descriptor Word 1 (MRC1_DOM3_RGD7_W1)	32	RW	0000_0000
1_53C0	MRC Region Descriptor NonSecure Enable (MRC1_DOM3_RGD_NSE)	32	RW	0000_0000
1_5440	MRC Region Descriptor Word 0 (MRC1_DOM4_RGD0_W0)	32	RW	0000_0000
1_5444	MRC Region Descriptor Word 1 (MRC1_DOM4_RGD0_W1)	32	RW	0000_0000
1_5448	MRC Region Descriptor Word 0 (MRC1_DOM4_RGD1_W0)	32	RW	0000_0000
1_544C	MRC Region Descriptor Word 1 (MRC1_DOM4_RGD1_W1)	32	RW	0000_0000
1_5450	MRC Region Descriptor Word 0 (MRC1_DOM4_RGD2_W0)	32	RW	0000_0000
1_5454	MRC Region Descriptor Word 1 (MRC1_DOM4_RGD2_W1)	32	RW	0000_0000
1_5458	MRC Region Descriptor Word 0 (MRC1_DOM4_RGD3_W0)	32	RW	0000_0000
1_545C	MRC Region Descriptor Word 1 (MRC1_DOM4_RGD3_W1)	32	RW	0000_0000
1_5460	MRC Region Descriptor Word 0 (MRC1_DOM4_RGD4_W0)	32	RW	0000_0000
1_5464	MRC Region Descriptor Word 1 (MRC1_DOM4_RGD4_W1)	32	RW	0000_0000
1_5468	MRC Region Descriptor Word 0 (MRC1_DOM4_RGD5_W0)	32	RW	0000_0000
1_546C	MRC Region Descriptor Word 1 (MRC1_DOM4_RGD5_W1)	32	RW	0000_0000
1_5470	MRC Region Descriptor Word 0 (MRC1_DOM4_RGD6_W0)	32	RW	0000_0000
1_5474	MRC Region Descriptor Word 1 (MRC1_DOM4_RGD6_W1)	32	RW	0000_0000
1_5478	MRC Region Descriptor Word 0 (MRC1_DOM4_RGD7_W0)	32	RW	0000_0000
1_547C	MRC Region Descriptor Word 1 (MRC1_DOM4_RGD7_W1)	32	RW	0000_0000
1_54C0	MRC Region Descriptor NonSecure Enable (MRC1_DOM4_RGD_NSE)	32	RW	0000_0000
1_5540	MRC Region Descriptor Word 0 (MRC1_DOM5_RGD0_W0)	32	RW	0000_0000
1_5544	MRC Region Descriptor Word 1 (MRC1_DOM5_RGD0_W1)	32	RW	0000_0000
1_5548	MRC Region Descriptor Word 0 (MRC1_DOM5_RGD1_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_554C	MRC Region Descriptor Word 1 (MRC1_DOM5_RGD1_W1)	32	RW	0000_0000
1_5550	MRC Region Descriptor Word 0 (MRC1_DOM5_RGD2_W0)	32	RW	0000_0000
1_5554	MRC Region Descriptor Word 1 (MRC1_DOM5_RGD2_W1)	32	RW	0000_0000
1_5558	MRC Region Descriptor Word 0 (MRC1_DOM5_RGD3_W0)	32	RW	0000_0000
1_555C	MRC Region Descriptor Word 1 (MRC1_DOM5_RGD3_W1)	32	RW	0000_0000
1_5560	MRC Region Descriptor Word 0 (MRC1_DOM5_RGD4_W0)	32	RW	0000_0000
1_5564	MRC Region Descriptor Word 1 (MRC1_DOM5_RGD4_W1)	32	RW	0000_0000
1_5568	MRC Region Descriptor Word 0 (MRC1_DOM5_RGD5_W0)	32	RW	0000_0000
1_556C	MRC Region Descriptor Word 1 (MRC1_DOM5_RGD5_W1)	32	RW	0000_0000
1_5570	MRC Region Descriptor Word 0 (MRC1_DOM5_RGD6_W0)	32	RW	0000_0000
1_5574	MRC Region Descriptor Word 1 (MRC1_DOM5_RGD6_W1)	32	RW	0000_0000
1_5578	MRC Region Descriptor Word 0 (MRC1_DOM5_RGD7_W0)	32	RW	0000_0000
1_557C	MRC Region Descriptor Word 1 (MRC1_DOM5_RGD7_W1)	32	RW	0000_0000
1_55C0	MRC Region Descriptor NonSecure Enable (MRC1_DOM5_RGD_NSE)	32	RW	0000_0000
1_5640	MRC Region Descriptor Word 0 (MRC1_DOM6_RGD0_W0)	32	RW	0000_0000
1_5644	MRC Region Descriptor Word 1 (MRC1_DOM6_RGD0_W1)	32	RW	0000_0000
1_5648	MRC Region Descriptor Word 0 (MRC1_DOM6_RGD1_W0)	32	RW	0000_0000
1_564C	MRC Region Descriptor Word 1 (MRC1_DOM6_RGD1_W1)	32	RW	0000_0000
1_5650	MRC Region Descriptor Word 0 (MRC1_DOM6_RGD2_W0)	32	RW	0000_0000
1_5654	MRC Region Descriptor Word 1 (MRC1_DOM6_RGD2_W1)	32	RW	0000_0000
1_5658	MRC Region Descriptor Word 0 (MRC1_DOM6_RGD3_W0)	32	RW	0000_0000
1_565C	MRC Region Descriptor Word 1 (MRC1_DOM6_RGD3_W1)	32	RW	0000_0000
1_5660	MRC Region Descriptor Word 0 (MRC1_DOM6_RGD4_W0)	32	RW	0000_0000
1_5664	MRC Region Descriptor Word 1 (MRC1_DOM6_RGD4_W1)	32	RW	0000_0000
1_5668	MRC Region Descriptor Word 0 (MRC1_DOM6_RGD5_W0)	32	RW	0000_0000
1_566C	MRC Region Descriptor Word 1 (MRC1_DOM6_RGD5_W1)	32	RW	0000_0000
1_5670	MRC Region Descriptor Word 0 (MRC1_DOM6_RGD6_W0)	32	RW	0000_0000
1_5674	MRC Region Descriptor Word 1 (MRC1_DOM6_RGD6_W1)	32	RW	0000_0000
1_5678	MRC Region Descriptor Word 0 (MRC1_DOM6_RGD7_W0)	32	RW	0000_0000
1_567C	MRC Region Descriptor Word 1 (MRC1_DOM6_RGD7_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_56C0	MRC Region Descriptor NonSecure Enable (MRC1_DOM6_RGD_NSE)	32	RW	0000_0000
1_5740	MRC Region Descriptor Word 0 (MRC1_DOM7_RGD0_W0)	32	RW	0000_0000
1_5744	MRC Region Descriptor Word 1 (MRC1_DOM7_RGD0_W1)	32	RW	0000_0000
1_5748	MRC Region Descriptor Word 0 (MRC1_DOM7_RGD1_W0)	32	RW	0000_0000
1_574C	MRC Region Descriptor Word 1 (MRC1_DOM7_RGD1_W1)	32	RW	0000_0000
1_5750	MRC Region Descriptor Word 0 (MRC1_DOM7_RGD2_W0)	32	RW	0000_0000
1_5754	MRC Region Descriptor Word 1 (MRC1_DOM7_RGD2_W1)	32	RW	0000_0000
1_5758	MRC Region Descriptor Word 0 (MRC1_DOM7_RGD3_W0)	32	RW	0000_0000
1_575C	MRC Region Descriptor Word 1 (MRC1_DOM7_RGD3_W1)	32	RW	0000_0000
1_5760	MRC Region Descriptor Word 0 (MRC1_DOM7_RGD4_W0)	32	RW	0000_0000
1_5764	MRC Region Descriptor Word 1 (MRC1_DOM7_RGD4_W1)	32	RW	0000_0000
1_5768	MRC Region Descriptor Word 0 (MRC1_DOM7_RGD5_W0)	32	RW	0000_0000
1_576C	MRC Region Descriptor Word 1 (MRC1_DOM7_RGD5_W1)	32	RW	0000_0000
1_5770	MRC Region Descriptor Word 0 (MRC1_DOM7_RGD6_W0)	32	RW	0000_0000
1_5774	MRC Region Descriptor Word 1 (MRC1_DOM7_RGD6_W1)	32	RW	0000_0000
1_5778	MRC Region Descriptor Word 0 (MRC1_DOM7_RGD7_W0)	32	RW	0000_0000
1_577C	MRC Region Descriptor Word 1 (MRC1_DOM7_RGD7_W1)	32	RW	0000_0000
1_57C0	MRC Region Descriptor NonSecure Enable (MRC1_DOM7_RGD_NSE)	32	RW	0000_0000
1_5840	MRC Region Descriptor Word 0 (MRC1_DOM8_RGD0_W0)	32	RW	0000_0000
1_5844	MRC Region Descriptor Word 1 (MRC1_DOM8_RGD0_W1)	32	RW	0000_0000
1_5848	MRC Region Descriptor Word 0 (MRC1_DOM8_RGD1_W0)	32	RW	0000_0000
1_584C	MRC Region Descriptor Word 1 (MRC1_DOM8_RGD1_W1)	32	RW	0000_0000
1_5850	MRC Region Descriptor Word 0 (MRC1_DOM8_RGD2_W0)	32	RW	0000_0000
1_5854	MRC Region Descriptor Word 1 (MRC1_DOM8_RGD2_W1)	32	RW	0000_0000
1_5858	MRC Region Descriptor Word 0 (MRC1_DOM8_RGD3_W0)	32	RW	0000_0000
1_585C	MRC Region Descriptor Word 1 (MRC1_DOM8_RGD3_W1)	32	RW	0000_0000
1_5860	MRC Region Descriptor Word 0 (MRC1_DOM8_RGD4_W0)	32	RW	0000_0000
1_5864	MRC Region Descriptor Word 1 (MRC1_DOM8_RGD4_W1)	32	RW	0000_0000
1_5868	MRC Region Descriptor Word 0 (MRC1_DOM8_RGD5_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_586C	MRC Region Descriptor Word 1 (MRC1_DOM8_RGD5_W1)	32	RW	0000_0000
1_5870	MRC Region Descriptor Word 0 (MRC1_DOM8_RGD6_W0)	32	RW	0000_0000
1_5874	MRC Region Descriptor Word 1 (MRC1_DOM8_RGD6_W1)	32	RW	0000_0000
1_5878	MRC Region Descriptor Word 0 (MRC1_DOM8_RGD7_W0)	32	RW	0000_0000
1_587C	MRC Region Descriptor Word 1 (MRC1_DOM8_RGD7_W1)	32	RW	0000_0000
1_58C0	MRC Region Descriptor NonSecure Enable (MRC1_DOM8_RGD_NSE)	32	RW	0000_0000
1_5940	MRC Region Descriptor Word 0 (MRC1_DOM9_RGD0_W0)	32	RW	0000_0000
1_5944	MRC Region Descriptor Word 1 (MRC1_DOM9_RGD0_W1)	32	RW	0000_0000
1_5948	MRC Region Descriptor Word 0 (MRC1_DOM9_RGD1_W0)	32	RW	0000_0000
1_594C	MRC Region Descriptor Word 1 (MRC1_DOM9_RGD1_W1)	32	RW	0000_0000
1_5950	MRC Region Descriptor Word 0 (MRC1_DOM9_RGD2_W0)	32	RW	0000_0000
1_5954	MRC Region Descriptor Word 1 (MRC1_DOM9_RGD2_W1)	32	RW	0000_0000
1_5958	MRC Region Descriptor Word 0 (MRC1_DOM9_RGD3_W0)	32	RW	0000_0000
1_595C	MRC Region Descriptor Word 1 (MRC1_DOM9_RGD3_W1)	32	RW	0000_0000
1_5960	MRC Region Descriptor Word 0 (MRC1_DOM9_RGD4_W0)	32	RW	0000_0000
1_5964	MRC Region Descriptor Word 1 (MRC1_DOM9_RGD4_W1)	32	RW	0000_0000
1_5968	MRC Region Descriptor Word 0 (MRC1_DOM9_RGD5_W0)	32	RW	0000_0000
1_596C	MRC Region Descriptor Word 1 (MRC1_DOM9_RGD5_W1)	32	RW	0000_0000
1_5970	MRC Region Descriptor Word 0 (MRC1_DOM9_RGD6_W0)	32	RW	0000_0000
1_5974	MRC Region Descriptor Word 1 (MRC1_DOM9_RGD6_W1)	32	RW	0000_0000
1_5978	MRC Region Descriptor Word 0 (MRC1_DOM9_RGD7_W0)	32	RW	0000_0000
1_597C	MRC Region Descriptor Word 1 (MRC1_DOM9_RGD7_W1)	32	RW	0000_0000
1_59C0	MRC Region Descriptor NonSecure Enable (MRC1_DOM9_RGD_NSE)	32	RW	0000_0000
1_5A40	MRC Region Descriptor Word 0 (MRC1_DOM10_RGD0_W0)	32	RW	0000_0000
1_5A44	MRC Region Descriptor Word 1 (MRC1_DOM10_RGD0_W1)	32	RW	0000_0000
1_5A48	MRC Region Descriptor Word 0 (MRC1_DOM10_RGD1_W0)	32	RW	0000_0000
1_5A4C	MRC Region Descriptor Word 1 (MRC1_DOM10_RGD1_W1)	32	RW	0000_0000
1_5A50	MRC Region Descriptor Word 0 (MRC1_DOM10_RGD2_W0)	32	RW	0000_0000
1_5A54	MRC Region Descriptor Word 1 (MRC1_DOM10_RGD2_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_5A58	MRC Region Descriptor Word 0 (MRC1_DOM10_RGD3_W0)	32	RW	0000_0000
1_5A5C	MRC Region Descriptor Word 1 (MRC1_DOM10_RGD3_W1)	32	RW	0000_0000
1_5A60	MRC Region Descriptor Word 0 (MRC1_DOM10_RGD4_W0)	32	RW	0000_0000
1_5A64	MRC Region Descriptor Word 1 (MRC1_DOM10_RGD4_W1)	32	RW	0000_0000
1_5A68	MRC Region Descriptor Word 0 (MRC1_DOM10_RGD5_W0)	32	RW	0000_0000
1_5A6C	MRC Region Descriptor Word 1 (MRC1_DOM10_RGD5_W1)	32	RW	0000_0000
1_5A70	MRC Region Descriptor Word 0 (MRC1_DOM10_RGD6_W0)	32	RW	0000_0000
1_5A74	MRC Region Descriptor Word 1 (MRC1_DOM10_RGD6_W1)	32	RW	0000_0000
1_5A78	MRC Region Descriptor Word 0 (MRC1_DOM10_RGD7_W0)	32	RW	0000_0000
1_5A7C	MRC Region Descriptor Word 1 (MRC1_DOM10_RGD7_W1)	32	RW	0000_0000
1_5AC0	MRC Region Descriptor NonSecure Enable (MRC1_DOM10_RGD_NSE)	32	RW	0000_0000
1_5B40	MRC Region Descriptor Word 0 (MRC1_DOM11_RGD0_W0)	32	RW	0000_0000
1_5B44	MRC Region Descriptor Word 1 (MRC1_DOM11_RGD0_W1)	32	RW	0000_0000
1_5B48	MRC Region Descriptor Word 0 (MRC1_DOM11_RGD1_W0)	32	RW	0000_0000
1_5B4C	MRC Region Descriptor Word 1 (MRC1_DOM11_RGD1_W1)	32	RW	0000_0000
1_5B50	MRC Region Descriptor Word 0 (MRC1_DOM11_RGD2_W0)	32	RW	0000_0000
1_5B54	MRC Region Descriptor Word 1 (MRC1_DOM11_RGD2_W1)	32	RW	0000_0000
1_5B58	MRC Region Descriptor Word 0 (MRC1_DOM11_RGD3_W0)	32	RW	0000_0000
1_5B5C	MRC Region Descriptor Word 1 (MRC1_DOM11_RGD3_W1)	32	RW	0000_0000
1_5B60	MRC Region Descriptor Word 0 (MRC1_DOM11_RGD4_W0)	32	RW	0000_0000
1_5B64	MRC Region Descriptor Word 1 (MRC1_DOM11_RGD4_W1)	32	RW	0000_0000
1_5B68	MRC Region Descriptor Word 0 (MRC1_DOM11_RGD5_W0)	32	RW	0000_0000
1_5B6C	MRC Region Descriptor Word 1 (MRC1_DOM11_RGD5_W1)	32	RW	0000_0000
1_5B70	MRC Region Descriptor Word 0 (MRC1_DOM11_RGD6_W0)	32	RW	0000_0000
1_5B74	MRC Region Descriptor Word 1 (MRC1_DOM11_RGD6_W1)	32	RW	0000_0000
1_5B78	MRC Region Descriptor Word 0 (MRC1_DOM11_RGD7_W0)	32	RW	0000_0000
1_5B7C	MRC Region Descriptor Word 1 (MRC1_DOM11_RGD7_W1)	32	RW	0000_0000
1_5BC0	MRC Region Descriptor NonSecure Enable (MRC1_DOM11_RGD_NSE)	32	RW	0000_0000
1_5C40	MRC Region Descriptor Word 0 (MRC1_DOM12_RGD0_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_5C44	MRC Region Descriptor Word 1 (MRC1_DOM12_RGD0_W1)	32	RW	0000_0000
1_5C48	MRC Region Descriptor Word 0 (MRC1_DOM12_RGD1_W0)	32	RW	0000_0000
1_5C4C	MRC Region Descriptor Word 1 (MRC1_DOM12_RGD1_W1)	32	RW	0000_0000
1_5C50	MRC Region Descriptor Word 0 (MRC1_DOM12_RGD2_W0)	32	RW	0000_0000
1_5C54	MRC Region Descriptor Word 1 (MRC1_DOM12_RGD2_W1)	32	RW	0000_0000
1_5C58	MRC Region Descriptor Word 0 (MRC1_DOM12_RGD3_W0)	32	RW	0000_0000
1_5C5C	MRC Region Descriptor Word 1 (MRC1_DOM12_RGD3_W1)	32	RW	0000_0000
1_5C60	MRC Region Descriptor Word 0 (MRC1_DOM12_RGD4_W0)	32	RW	0000_0000
1_5C64	MRC Region Descriptor Word 1 (MRC1_DOM12_RGD4_W1)	32	RW	0000_0000
1_5C68	MRC Region Descriptor Word 0 (MRC1_DOM12_RGD5_W0)	32	RW	0000_0000
1_5C6C	MRC Region Descriptor Word 1 (MRC1_DOM12_RGD5_W1)	32	RW	0000_0000
1_5C70	MRC Region Descriptor Word 0 (MRC1_DOM12_RGD6_W0)	32	RW	0000_0000
1_5C74	MRC Region Descriptor Word 1 (MRC1_DOM12_RGD6_W1)	32	RW	0000_0000
1_5C78	MRC Region Descriptor Word 0 (MRC1_DOM12_RGD7_W0)	32	RW	0000_0000
1_5C7C	MRC Region Descriptor Word 1 (MRC1_DOM12_RGD7_W1)	32	RW	0000_0000
1_5CC0	MRC Region Descriptor NonSecure Enable (MRC1_DOM12_RGD_NSE)	32	RW	0000_0000
1_5D40	MRC Region Descriptor Word 0 (MRC1_DOM13_RGD0_W0)	32	RW	0000_0000
1_5D44	MRC Region Descriptor Word 1 (MRC1_DOM13_RGD0_W1)	32	RW	0000_0000
1_5D48	MRC Region Descriptor Word 0 (MRC1_DOM13_RGD1_W0)	32	RW	0000_0000
1_5D4C	MRC Region Descriptor Word 1 (MRC1_DOM13_RGD1_W1)	32	RW	0000_0000
1_5D50	MRC Region Descriptor Word 0 (MRC1_DOM13_RGD2_W0)	32	RW	0000_0000
1_5D54	MRC Region Descriptor Word 1 (MRC1_DOM13_RGD2_W1)	32	RW	0000_0000
1_5D58	MRC Region Descriptor Word 0 (MRC1_DOM13_RGD3_W0)	32	RW	0000_0000
1_5D5C	MRC Region Descriptor Word 1 (MRC1_DOM13_RGD3_W1)	32	RW	0000_0000
1_5D60	MRC Region Descriptor Word 0 (MRC1_DOM13_RGD4_W0)	32	RW	0000_0000
1_5D64	MRC Region Descriptor Word 1 (MRC1_DOM13_RGD4_W1)	32	RW	0000_0000
1_5D68	MRC Region Descriptor Word 0 (MRC1_DOM13_RGD5_W0)	32	RW	0000_0000
1_5D6C	MRC Region Descriptor Word 1 (MRC1_DOM13_RGD5_W1)	32	RW	0000_0000
1_5D70	MRC Region Descriptor Word 0 (MRC1_DOM13_RGD6_W0)	32	RW	0000_0000
1_5D74	MRC Region Descriptor Word 1 (MRC1_DOM13_RGD6_W1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_5D78	MRC Region Descriptor Word 0 (MRC1_DOM13_RGD7_W0)	32	RW	0000_0000
1_5D7C	MRC Region Descriptor Word 1 (MRC1_DOM13_RGD7_W1)	32	RW	0000_0000
1_5DC0	MRC Region Descriptor NonSecure Enable (MRC1_DOM13_RGD_NSE)	32	RW	0000_0000
1_5E40	MRC Region Descriptor Word 0 (MRC1_DOM14_RGD0_W0)	32	RW	0000_0000
1_5E44	MRC Region Descriptor Word 1 (MRC1_DOM14_RGD0_W1)	32	RW	0000_0000
1_5E48	MRC Region Descriptor Word 0 (MRC1_DOM14_RGD1_W0)	32	RW	0000_0000
1_5E4C	MRC Region Descriptor Word 1 (MRC1_DOM14_RGD1_W1)	32	RW	0000_0000
1_5E50	MRC Region Descriptor Word 0 (MRC1_DOM14_RGD2_W0)	32	RW	0000_0000
1_5E54	MRC Region Descriptor Word 1 (MRC1_DOM14_RGD2_W1)	32	RW	0000_0000
1_5E58	MRC Region Descriptor Word 0 (MRC1_DOM14_RGD3_W0)	32	RW	0000_0000
1_5E5C	MRC Region Descriptor Word 1 (MRC1_DOM14_RGD3_W1)	32	RW	0000_0000
1_5E60	MRC Region Descriptor Word 0 (MRC1_DOM14_RGD4_W0)	32	RW	0000_0000
1_5E64	MRC Region Descriptor Word 1 (MRC1_DOM14_RGD4_W1)	32	RW	0000_0000
1_5E68	MRC Region Descriptor Word 0 (MRC1_DOM14_RGD5_W0)	32	RW	0000_0000
1_5E6C	MRC Region Descriptor Word 1 (MRC1_DOM14_RGD5_W1)	32	RW	0000_0000
1_5E70	MRC Region Descriptor Word 0 (MRC1_DOM14_RGD6_W0)	32	RW	0000_0000
1_5E74	MRC Region Descriptor Word 1 (MRC1_DOM14_RGD6_W1)	32	RW	0000_0000
1_5E78	MRC Region Descriptor Word 0 (MRC1_DOM14_RGD7_W0)	32	RW	0000_0000
1_5E7C	MRC Region Descriptor Word 1 (MRC1_DOM14_RGD7_W1)	32	RW	0000_0000
1_5EC0	MRC Region Descriptor NonSecure Enable (MRC1_DOM14_RGD_NSE)	32	RW	0000_0000
1_5F40	MRC Region Descriptor Word 0 (MRC1_DOM15_RGD0_W0)	32	RW	0000_0000
1_5F44	MRC Region Descriptor Word 1 (MRC1_DOM15_RGD0_W1)	32	RW	0000_0000
1_5F48	MRC Region Descriptor Word 0 (MRC1_DOM15_RGD1_W0)	32	RW	0000_0000
1_5F4C	MRC Region Descriptor Word 1 (MRC1_DOM15_RGD1_W1)	32	RW	0000_0000
1_5F50	MRC Region Descriptor Word 0 (MRC1_DOM15_RGD2_W0)	32	RW	0000_0000
1_5F54	MRC Region Descriptor Word 1 (MRC1_DOM15_RGD2_W1)	32	RW	0000_0000
1_5F58	MRC Region Descriptor Word 0 (MRC1_DOM15_RGD3_W0)	32	RW	0000_0000
1_5F5C	MRC Region Descriptor Word 1 (MRC1_DOM15_RGD3_W1)	32	RW	0000_0000
1_5F60	MRC Region Descriptor Word 0 (MRC1_DOM15_RGD4_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1_5F64	MRC Region Descriptor Word 1 (MRC1_DOM15_RGD4_W1)	32	RW	0000_0000
1_5F68	MRC Region Descriptor Word 0 (MRC1_DOM15_RGD5_W0)	32	RW	0000_0000
1_5F6C	MRC Region Descriptor Word 1 (MRC1_DOM15_RGD5_W1)	32	RW	0000_0000
1_5F70	MRC Region Descriptor Word 0 (MRC1_DOM15_RGD6_W0)	32	RW	0000_0000
1_5F74	MRC Region Descriptor Word 1 (MRC1_DOM15_RGD6_W1)	32	RW	0000_0000
1_5F78	MRC Region Descriptor Word 0 (MRC1_DOM15_RGD7_W0)	32	RW	0000_0000
1_5F7C	MRC Region Descriptor Word 1 (MRC1_DOM15_RGD7_W1)	32	RW	0000_0000
1_5FC0	MRC Region Descriptor NonSecure Enable (MRC1_DOM15_RGD_NSE)	32	RW	0000_0000

22.7.6.2 TRDC Register (TRDC_CR)

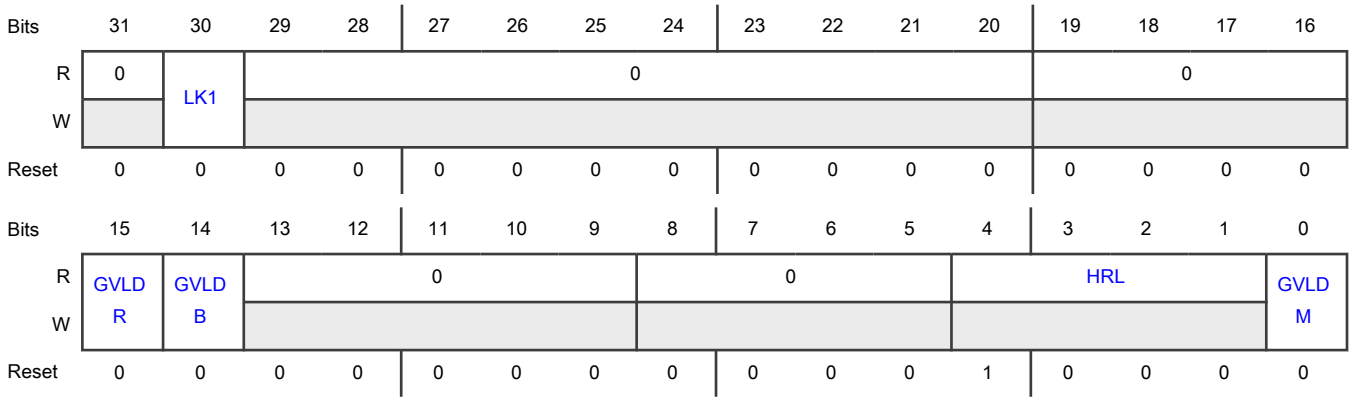
This register provides status about the TRDC and global enable bits for the entire module's operation. A fully operational TRDC requires all global bits GVLD {R,B,M} to be asserted. Undefined behavior results if they are not all asserted. If there are no region checkers (MRCs) present in the configuration, GVLDR need not be asserted for a fully operational TRDC.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_CR	0h

Diagram



Fields

Field	Description
31 —	Reserved
30 LK1	Lock Status This field is the lock status of the TRDC_CR. This field is set when all GVLD bits are set; GVLDR=GVLDB=GVLDM=1b1. Once set, this bit remains asserted until the next reset. 0 - The CR can be written by any secure privileged write. 1 - The CR is locked (read-only) until the next reset.
29-20 —	Reserved
19-16 —	Reserved
15 GVLDR	Global Valid for Memory Region Checkers TRDC global MRC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC MRCs are disabled. 1 - TRDC MRCs are enabled.
14 GVLDB	Global Valid for Memory Block Checkers TRDC global MBC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC MBCs are disabled. 1 - TRDC MBCs are enabled.
13-9 —	Reserved
8-5 —	Reserved
4-1 HRL	Hardware Revision Level This read-only field specifies the TRDC’s hardware and definition revision level. It can be read by software to determine the functional definition of the module.
0 GVLDM	Global Valid for Domain Assignment Controllers TRDC global DAC enable/disable. Once set, this bit remains set until the next reset. 0 - TRDC DACs are disabled. 1 - TRDC DACs are enabled.

22.7.6.3 TRDC Hardware Configuration Register 0 (TRDC_HWCFG0)

This read-only register contains information on the TRDC's hardware configuration. Specifically, it defines the number of implemented domains and bus initiators along with the number of instances of memory block checkers (MBCs) and memory region checkers (MRCs). The register value at reset is device-specific. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG0	F0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MID				NMRC				0				NMBC			
W																
Reset	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NMSTR								0				NDID			
W																
Reset	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0

Fields

Field	Description
31-29 MID	Module ID This field defines major version ID of this module.
28-24 NMRC	Number of MRCs This field defines the number of MRCs on the device [1-16].
23-20 —	Reserved
19-16 NMBC	Number of MBCs This field defines the number of MBCs on the device [1-8].
15-8 NMSTR	Number of bus initiators This read-only field defines the number of bus initiators.
7-5 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
4-0 NDID	Number of domains This read-only field defines the number of domains on the device [1-16].

22.7.6.4 TRDC Hardware Configuration Register 1 (TRDC_HWCFG1)

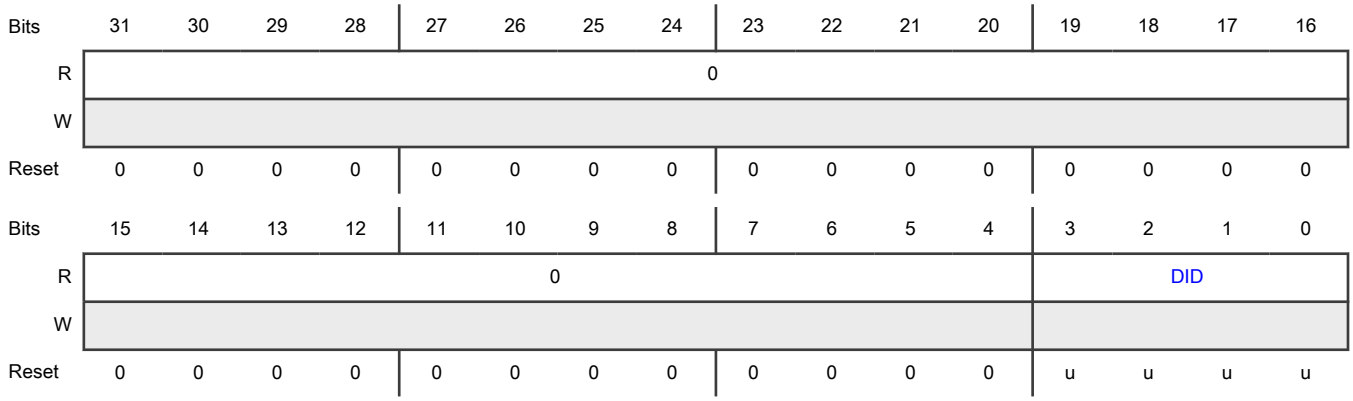
This register contains information on the TRDC's hardware configuration. It provides a mechanism for software to determine its domain number by simply reading the register. See Domain error capture management for more details on typical usage. Attempted writes are error terminated.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_HWCFG1	F4h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 DID	Domain identifier number This field provides the domain number [0-15] of the requesting bus initiator.

22.7.6.5 Domain Assignment Configuration Register (DACFG2 - DACFG4)

This register defines the number of implemented domain assignment registers for bus initiator m, where m+1 can specify from 1 to 64 bus initiators. These registers are organized as a byte-sized data array and can be read using 8-, 16- or 32-bit accesses. An all-zero value (NCM = 0, NMDAR = 0) indicates a non-existent bus initiator. Attempted writes are error terminated.

Register read will not return transfer error when DACFG for a initiator doesn't exist but it is in the same 32-bit group as DACFG for an existing initiator. For example, when there are only 2 DACFG0-1 registers for 2 initiator, and initiators for DACFG2-3 don't exist, then access to DACFG2-3 won't return transfer error.

Typically, processor bus initiators have one or more domain assignment registers, while non-processor initiators have a single domain assignment register.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
DACFG2	102h
DACFG3	103h
DACFG4	104h

Diagram

Bits	7	6	5	4	3	2	1	0
R	NCM	0			NMDAR			
W								
Reset	1	0	0	0	0	0	0	1

Fields

Field	Description
7 NCM	Non-CPU Initiator This read-only field signals that bus initiator m is a non-CPU initiator. It specifies that the format of the associated MDA_Wr_m register defines a non-processor domain assignment. This field is zero for a non-existent bus initiator. 0 - Bus initiator is a processor. 1 - Bus initiator is a non-processor.
6-4 —	Reserved
3-0 NMDAR	Number of initiator domain assignment registers for bus initiator m This read-only field specifies the number of registers associated with the initiator domain assignment register for a given bus initiator. The value is limited to the range [0-8], where zero indicates a non-existent bus initiator and non-zero values indicate the number of implemented registers associated with this MDAm.

22.7.6.6 TRDC IDAU Control Register (TRDC_IDAU_CR)

This register defines the configuration for the Implementation-Defined Attribution Unit which is tightly-coupled to the TZ-M extensions in the processor core. Many of the fields in this register explicitly control processor TZ-M functions.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_IDAU_CR	1C0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PCURRNS	0	0	0	0	0	0							0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			LKSAU	LKNSMPU	LKSMPU	LKNSVTOR	LKSVTAI...	0	SAUDIS	MPUNSDIS	MPUSDIS	CFGSECE...	0		VLD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Fields

Field	Description
31 PCURRNS	Processor current security Current security state of the processor. 0 - Processor is in Secure state 1 - Processor is in Nonsecure state
30 —	Reserved
29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25-19 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
18-16 —	Reserved
15-13 —	Reserved
12 LKSAU	Lock SAU Asserting this bit prevents changes to the processor's Secure SAU memory regions already programmed. All processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the SAU_CTRL, SAU_RNR, SAU_RBAR and SAU_RLAR registers from software or from a debug agent connected to the processor
11 LKNSMPU	Lock Nonsecure MPU Asserting this bit prevents changes to the processor's Nonsecure MPU memory regions already programmed. All processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the MPU_CTRL_NS, MPU_RNR_NS, MPU_RBAR_NS, MPU_RLAR_NS, MPU_RBAR_A_NSn and MPU_RLAR_A_NSn from software or from a debug agent connected to the processor
10 LKSMPU	Lock Secure MPU Asserting this signal prevents changes to the processor's programmed Secure MPU memory regions and all processor writes to the registers are ignored. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock these registers 1 - Disable writes to the MPU_CTRL, MPU_RNR, MPU_RBAR, MPU_RLAR, MPU_RBAR_An and MPU_RLAR_An from software or from a debug agent connected to the processor in Secure state
9 LKNSVTOR	Lock Nonsecure Vector Table Offset Register Asserting this signal prevents changes to the processor's Nonsecure vector table base address. This bit is sticky, once set, it can only be cleared with a reset. 0 - Unlock this register 1 - Disable writes to the VTOR_NS register
8 LKSVTAIRCR	Lock Secure VTOR, Application interrupt and Reset Control Registers This bit is sticky, once set, it can only be cleared with a reset. Asserting this signal prevents processor changes to: • The Secure vector table base address.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Handling of Secure interrupt priority. • BusFault, HardFault, and NMI security target settings in the processor. 0 - Unlock these registers 1 - Disable writes to the VTOR_S, AIRCR[PRIS], and AIRCR[BFHFNMINS] registers
7 —	Reserved
6 SAUDIS	Security Attribution Unit Disable 0 - SAU is enabled 1 - SAU is disabled
5 MPUNSDIS	NonSecure Memory Protection Unit Disabled 0 - Nonsecure MPU is enabled 1 - Nonsecure MPU is disabled
4 MPUSDIS	Secure Memory Protection Unit Disabled 0 - Secure MPU is enabled 1 - Secure MPU is disabled
3 CFGSECEXT	Configure Security Extension 0 - Armv8M Security Extension is disabled 1 - Armv8-M Security Extension is enabled
2-1 —	Reserved
0 VLD	Valid When VLD =0, all address attribute from IDAU is nonSecure. When VLD=1, values in other fields of this register are valid.

22.7.6.7 TRDC FLW Control (TRDC_FLW_CTL)

This register provides control of the FLW = Flash Logical Window operation.

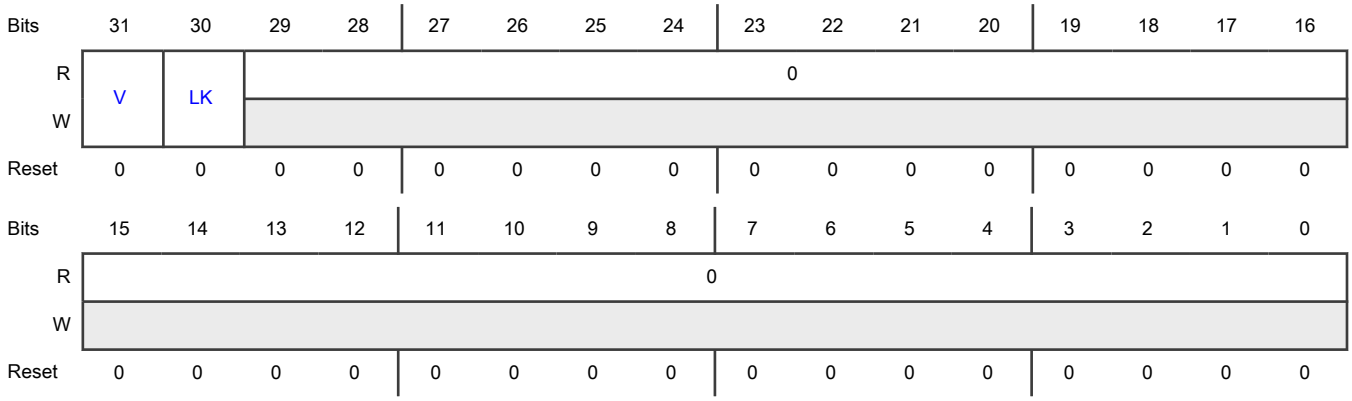
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_FLW_CTL	1E0h

Diagram



Fields

Field	Description
31 V	Valid bit 0 - FLW function is disabled. 1 - FLW function is enabled.
30 LK	Lock bit 0 - FLW registers may be modified. 1 - FLW registers are locked until the next reset.
29-0 —	Reserved

22.7.6.8 TRDC FLW Physical Base (TRDC_FLW_PBASE)

This read-only register gives the physical base address of the Flash Logical Window.

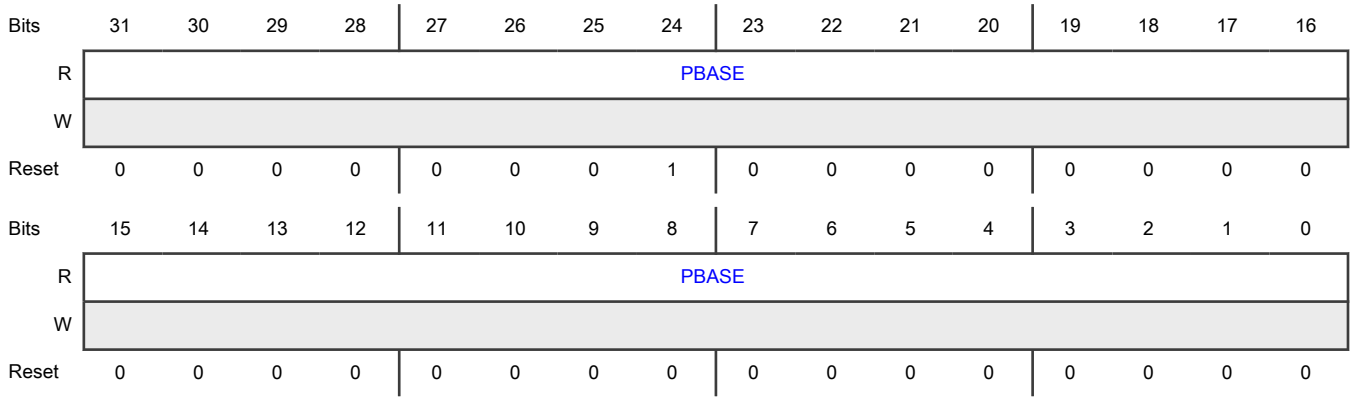
This register exists for this device but has not effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
TRDC_FLW_PBASE	1E4h

Diagram



Fields

Field	Description
31-0 PBASE	Physical base address Physical address of the base of the FLW (mod 32KBytes).

22.7.6.9 TRDC FLW Array Base (TRDC_FLW_ABASE)

This register gives the flash array base address of the Flash Logical Window.

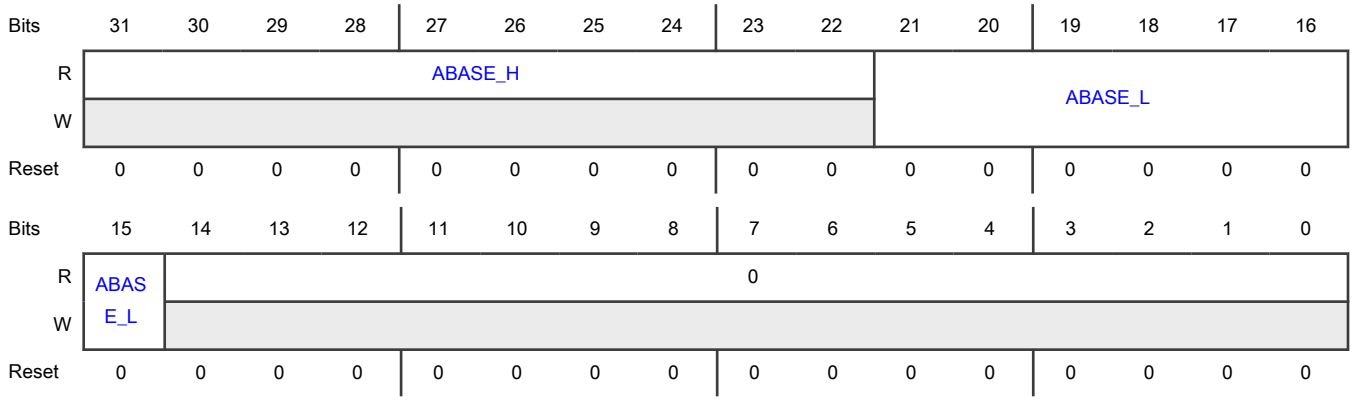
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write. This register is not writable if TRDC_FLW_CTL[LK]=1.

Offset

Register	Offset
TRDC_FLW_ABASE	1E8h

Diagram



Fields

Field	Description
31-22 ABASE_H	Array base address high Flash array address upper bits of the base of the FLW (mod 32KBytes).
21-15 ABASE_L	Array base address low Flash array address lower bits of the base of the FLW (mod 32KBytes).
14-0 —	Reserved

22.7.6.10 TRDC FLW Block Count (TRDC_FLW_BCNT)

This register gives the size of the Flash Logic Window in 32KByte blocks.

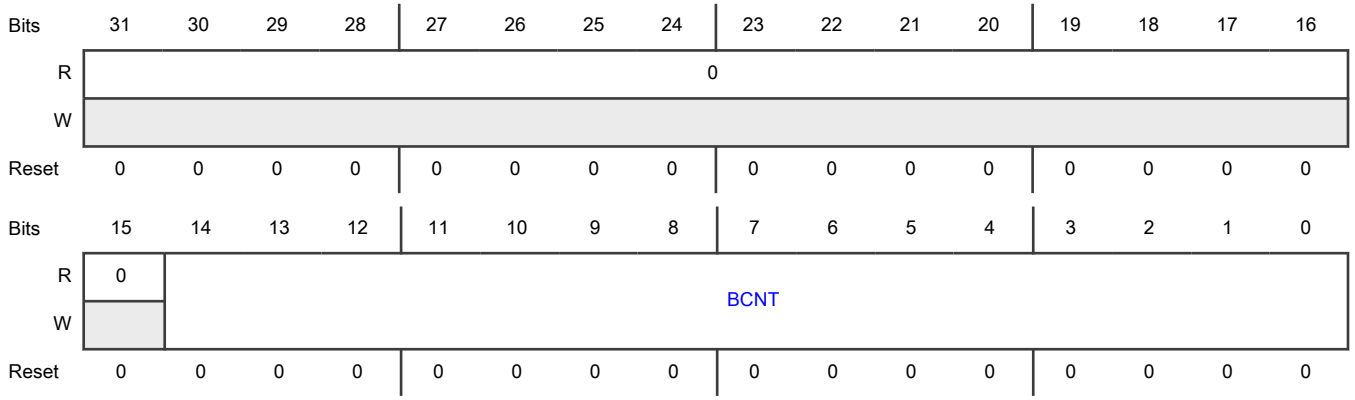
This register exists for this device but has no effect on device operation.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write. This register is not writable if TRDC_FLW_CTL[LK]=1.

Offset

Register	Offset
TRDC_FLW_BCNT	1ECh

Diagram



Fields

Field	Description
31-15 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
14-0 BCNT	Block Count Size of FLW in number of 32KByte blocks.

22.7.6.11 TRDC Fault Domain ID (TRDC_FDID)

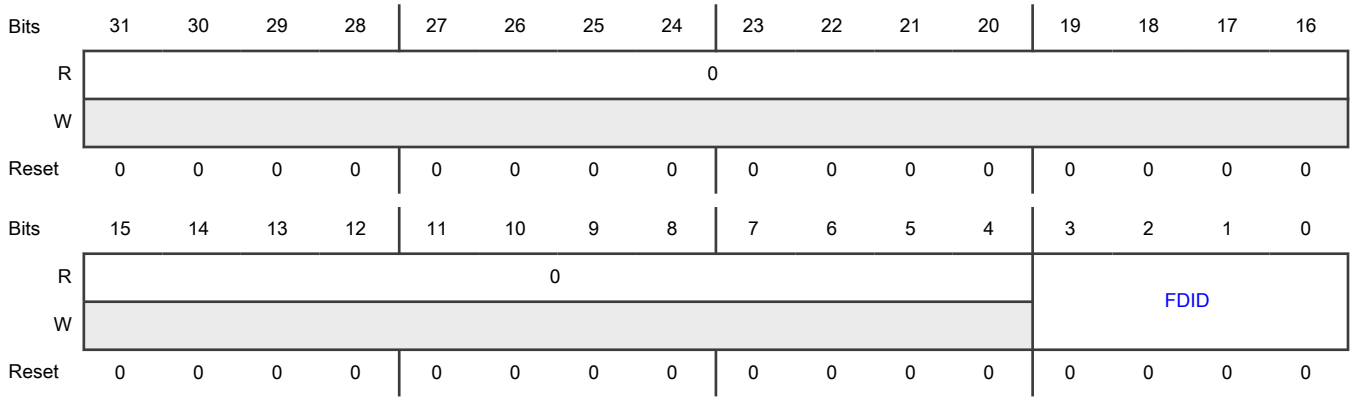
In the event of an access error, this register is used to specify the domainID of the faulting reference before indexing into the Domain Error registers.

If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
TRDC_FDID	1FCh

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 FDID	Domain ID of Faulted Access This field indicates the domainID of the fault used to index the array of domain error information. The user queries the DERRLOCx registers to find the DomainID of the faulting access and must write this register with the domain ID before reading the Domain Error registers.

22.7.6.12 TRDC Domain Error Location Register (TRDC_DERRLOC0 - TRDC_DERRLOC15)

This array of read-only registers provide the instance number of the submodule where (an) access violation(s) occurred. These registers are organized as a word array, indexed by the faulting domain number, d. The two fields of this register provide a bitmap

of instances associated with all submodules containing captured error information for that domain. These instance numbers are then used as indices into the DERR_W0_*i*, DERR_W1_*i*, and DERR_W3_*i*register arrays. See Domain error capture management for more details.

When an access violation is detected by either a Memory Region Checker (MRC) or a Memory Block Checker (MBC), address and attribute information of the offending access is captured. Using the faulting domainID number as the index, d, this array of read-only registers provide additional information signaling the instance number of the submodule where the access violation(s) occurred. Since the resulting exception handler needs the submodule instance to retrieve the captured address and attribute information from DERR_W0_*i* and DERR_W1_*i*, these registers provide the instance number details.

Attempted writes are error terminated.

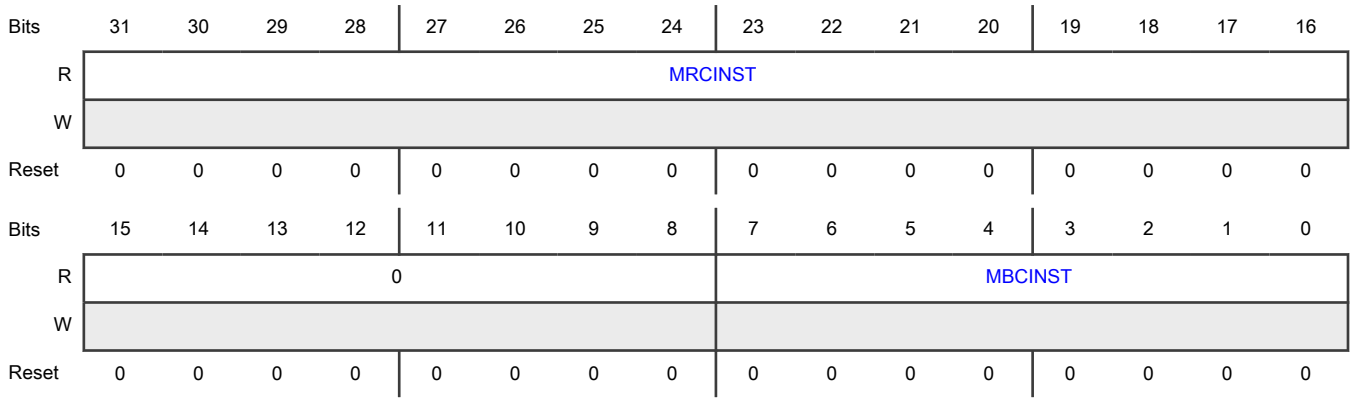
Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

For d = 0 to 15:

Register	Offset
TRDC_DERRLOCd	200h + (d × 4h)

Diagram



Fields

Field	Description
31-16 MRCINST	MRC instance This field is a bitmap indicating the presence of a detected access violation for domain d in the implemented instances of the MRC. The least-significant bit of this field (bit 16) corresponds to MRC instance 0. The most-significant bit of this field (bit 31) corresponds to MRC instance 15 (MRC instance = i-16 where i is the register bit number), and so on. Multiple bits can be set at any time indicating access violations for this domain have been detected across multiple instances of the MRCs. For each bit in this field: • 0 - The memory region checker has not detected an access violation or is not physically present. • 1 - The memory region checker has detected one or more access violations for this domain.
15-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7-0 MBCINST	MBC instance This field is a bitmap indicating the presence of a detected access violation for domain d in the implemented instances of the MBC. The least-significant bit of this field (bit 0) corresponds to MBC instance 0. The most-significant bit of this field (bit 7) corresponds to MBC instance 7 Multiple bits can be set at any time indicating access violations for this domain have been detected across multiple instances of the MBCs. For each bit in this field: • 0 - The memory block checker has not detected an access violation or is not physically present. • 1 - The memory block checker has detected one or more access violations for this domain.

22.7.6.13 MBC Domain Error Word0 Register (MBC0_DERR_W0 - MBC1_DERR_W0)

This read-only register array provides the address of an access violation detected by memory block checker (MBC). These registers are organized as a word array, which is indexed by the MBC instance number. That is, the index, i, of this array is the instance number of MBC with the access violation. The submodule instance numbers are provided by the DERRLOC registers. The memory mapped error capture detail registers are organized as 24 sequential 16 byte entries.

When an access violation is detected and the offending information captured, subsequent updates to this register are disabled until the required data pattern is written to the DERR_W3_i register. At that time, this register is cleared and re-enabled to capture the next access violation.

Attempted writes are error terminated as are attempted reads of an MBC instance that is not physically present.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
MBC0_DERR_W0	400h
MBC1_DERR_W0	410h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 EADDR	Error address This is the unaliased virtual address of the access that generated the access violation.

22.7.6.14 MBC Domain Error Word1 Register (MBC0_DERR_W1 - MBC1_DERR_W1)

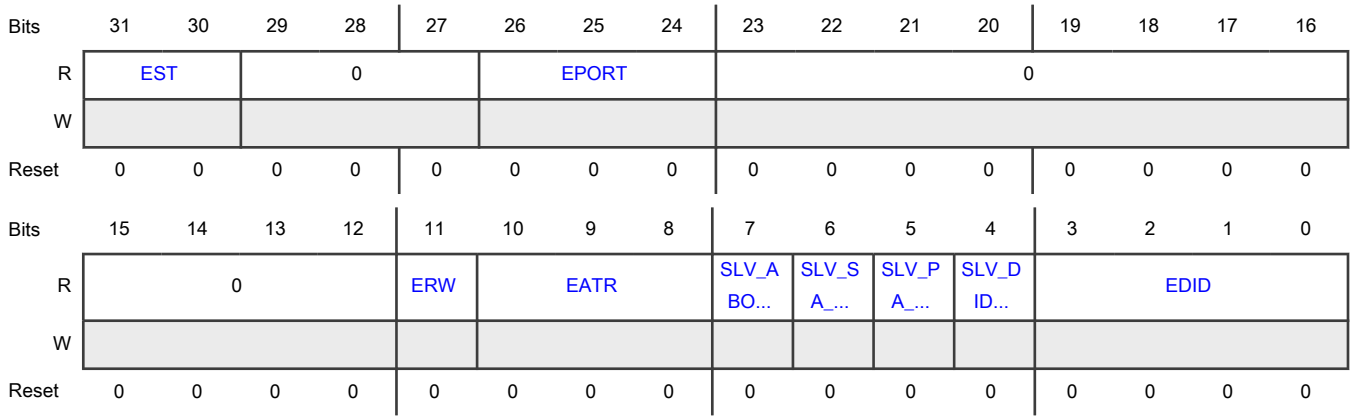
This read-only register array provides the attributes of an access violation detected by Memory Block Checker (MBC). These registers are organized as a word array, which is indexed by the violating submodule instance number. Refer to register Domain Error Location(DERRLOCd), Domain Error Word0 Register(DERR_W0_i), and Domain error capture management for more information.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
MBC0_DERR_W1	404h
MBC1_DERR_W1	414h

Diagram



Fields

Field	Description
31-30 EST	Error state This field signals the state of access violations for this domain in this instance of the block checker or region checker. Once an access violation has been detected and the faulting address and attribute information stored, subsequent errors are simply recorded as an overrun condition without any data captured. After retrieving the faulting address and attributes, the error capture mechanism must be rearmed by performing a write to DERR_W3_i.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	00 - No access violation has been detected. 01 - No access violation has been detected. 10 - A single access violation has been detected. 11 - Multiple access violations for this domain have been detected by this submodule instance. Only the address and attribute information for the first error have been captured in DERR_W0_i and DERR_W1_i.
29-27 —	Reserved
26-24 EPORT	Error port This field identifies the encoded port number of the MBC that detected the access violation. The MBC port number connection is device-specific. See the chip configuration details for more information. This is the MBC target that has the violation. 000 - mbcxslv0 001 - mbcxslv1 010 - mbcxslv2 011 - mbcxslv3
23-12 —	Reserved
11 ERW	Error read/write This field signals whether the captured access violation occurred on a read or write reference. 0 - Read access 1 - Write access
10-8 EATR	Error attributes This field captures attributes of the access violation. 000 - Secure user mode, instruction fetch access. 001 - Secure user mode, data access. 010 - Secure privileged mode, instruction fetch access. 011 - Secure privileged mode, data access. 100 - Nonsecure user mode, instruction fetch access. 101 - Nonsecure user mode, data access. 110 - Nonsecure privileged mode, instruction fetch access. 111 - Nonsecure privileged mode, data access.
7	Bus protect error

Table continues on the next page...

Table continued from the previous page...

Field	Description
SLV_ABORT	This field captures the bus protect error from target which cause access abort.
6 SLV_SA_ERR	Secure attribute check error This field captures the secure attribute check error from target.
5 SLV_PA_ERR	Privilege attribute check error This field captures the privilege attribute check error from target.
4 SLV_DID_ERR	DID check error This field captures the DID check error from target.
3-0 EDID	Error domain identifier This field captures the domain identifier of the access violation.

22.7.6.15 MBC Domain Error Word3 Register (MBC0_DERR_W3 - MBC1_DERR_W3)

This register is used to rearm the error capture logic and clear the DERR_W0_i and DERR_W1_i registers. After the domain access violation error details have been read, typically in an exception service routine, a 32-bit word write to this register is required to rearm the error capture logic.

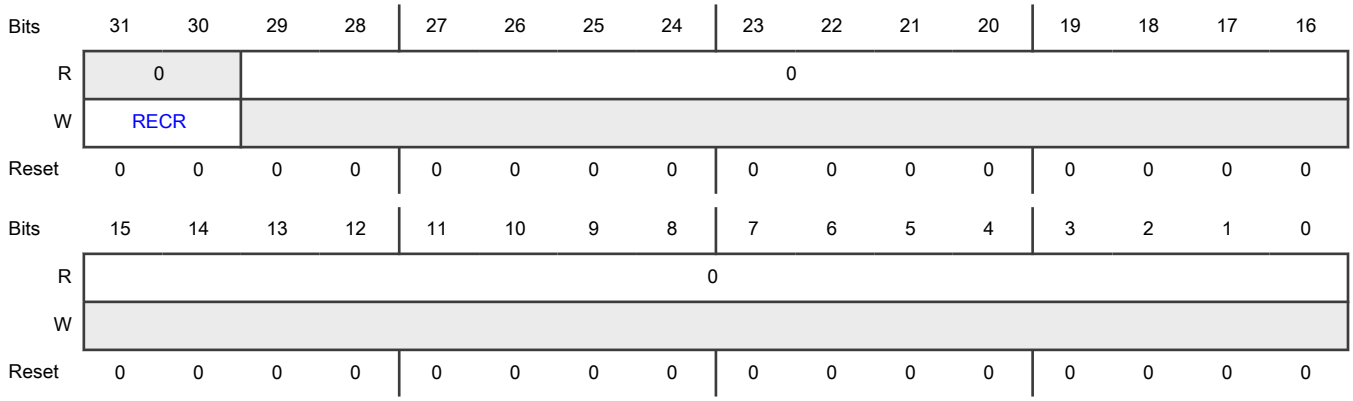
A read of this location returns zeroes. Attempted reads of a Memory Region Checker (MRC) or Memory Block Checker (MBC) instance that is not physically present are error terminated.

See Domain error capture management for more details.

Offset

Register	Offset
MBC0_DERR_W3	40Ch
MBC1_DERR_W3	41Ch

Diagram



Fields

Field	Description
31-30 RECR	Rearm Error Capture Registers This 2-bit, write-only field controls the rearming of the domain error capture registers. Once an access violation has been detected with the faulting address and attribute information stored, subsequent errors are simply recorded as an overrun condition without any data captured. Writing 01b to this field rearms the error capture mechanism and clears the DERR_W0_i and DERR_W1_i registers. A write of any value other than 01b has no effect.
29-0 —	Reserved

22.7.6.16 MRC Domain Error Word0 Register (MRC0_DERR_W0 - MRC1_DERR_W0)

This read-only register array provides the address of an access violation detected by memory region checker (MRC). These registers are organized as a word array, which is indexed by the MRC instance number. That is, the index, i, of this array is the instance number of MRC with the access violation. The submodule instance numbers are provided by the DERRLOC registers. The memory mapped error capture detail registers are organized as 24 sequential 16 byte entries.

When an access violation is detected and the offending information captured, subsequent updates to this register are disabled until the required data pattern is written to the DERR_W3_i register. At that time, this register is cleared and re-enabled to capture the next access violation.

Attempted writes are error terminated as are attempted reads of an MRC instance that is not physically present.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
MRC0_DERR_W0	480h
MRC1_DERR_W0	490h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 EADDR	Error address This is the unaliased virtual address of the access that generated the access violation.

22.7.6.17 MRC Domain Error Word1 Register (MRC0_DERR_W1 - MRC1_DERR_W1)

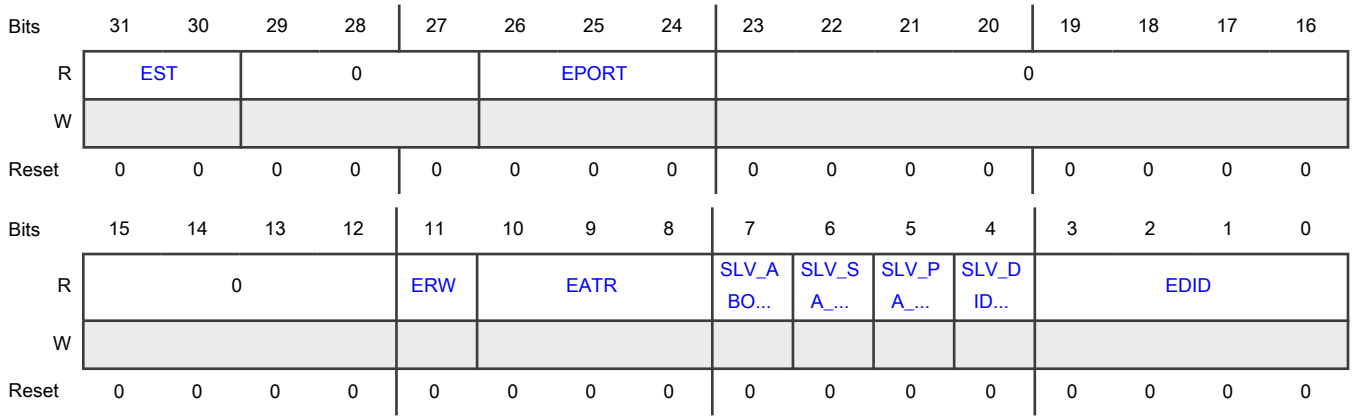
This read-only register array provides the attributes of an access violation detected by memory region checker (MRC). These registers are organized as a word array, which is indexed by the violating submodule instance number. Refer to register Domain Error Location(DERRLOCd), Domain Error Word0 Register(DERR_W0_i), and Domain error capture management for more information.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read

Offset

Register	Offset
MRC0_DERR_W1	484h
MRC1_DERR_W1	494h

Diagram



Fields

Field	Description
31-30 EST	Error state This field signals the state of access violations for this domain in this instance of the block checker or region checker. Once an access violation has been detected and the faulting address and attribute information stored, subsequent errors are simply recorded as an overrun condition without any data captured. After retrieving the faulting address and attributes, the error capture mechanism must be rearmed by performing a write to DERR_W3_i.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	00 - No access violation has been detected. 01 - No access violation has been detected. 10 - A single access violation has been detected. 11 - Multiple access violations for this domain have been detected by this submodule instance. Only the address and attribute information for the first error have been captured in DERR_W0_i and DERR_W1_i.
29-27 —	Reserved
26-24 EPORT	Error port This field identifies the encoded port number of the MRC that detected the access violation. The MRC port number connection is device-specific. See the chip configuration details for more information.
23-12 —	Reserved
11 ERW	Error read/write This field signals whether the captured access violation occurred on a read or write reference. 0 - Read access 1 - Write access
10-8 EATR	Error attributes This field captures certain attributes of the access violation. 000 - Secure user mode, instruction fetch access. 001 - Secure user mode, data access. 010 - Secure privileged mode, instruction fetch access. 011 - Secure privileged mode, data access. 100 - Nonsecure user mode, instruction fetch access. 101 - Nonsecure user mode, data access. 110 - Nonsecure privileged mode, instruction fetch access. 111 - Nonsecure privileged mode, data access.
7 SLV_ABORT	Bus protect error This field captures the bus protect error from target which cause access abort.
6 SLV_SA_ERR	Secure attribute check error This field captures the secure attribute check error from target.
5	Privilege attribute check error

Table continues on the next page...

Table continued from the previous page...

Field	Description
SLV_PA_ERR	This field captures the privilege attribute check error from target.
4 SLV_DID_ERR	DID check error This field captures the DID check error from target.
3-0 EDID	Error domain identifier This field captures the domain identifier of the access violation.

22.7.6.18 MRC Domain Error Word3 Register (MRC0_DERR_W3 - MRC1_DERR_W3)

This register is used to rearm the error capture logic and clear the DERR_W0_i and DERR_W1_i registers. After the domain access violation error details have been read, typically in an exception service routine, a 32-bit word write to this register is required to rearm the error capture logic.

A read of this location returns zeroes. Attempted reads of a memory region checker (MRC) or Memory Block Checker (MBC) instance that is not physically present are error terminated.

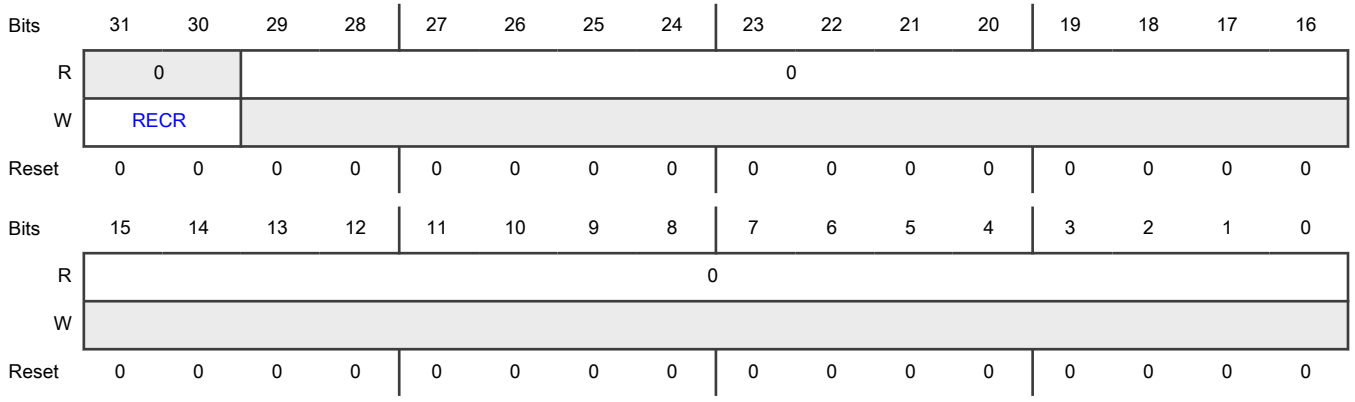
See Domain error capture management for more details.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/Write

Offset

Register	Offset
MRC0_DERR_W3	48Ch
MRC1_DERR_W3	49Ch

Diagram



Fields

Field	Description
31-30 RECR	Rearm Error Capture Registers This 2-bit, write-only field controls the rearming of the domain error capture registers. Once an access violation has been detected with the faulting address and attribute information stored, subsequent errors are simply recorded as an overrun condition without any data captured. Writing 01b to this field rearms the error capture mechanism and clears the DERR_W0_i and DERR_W1_i registers. A write of any value other than 01b has no effect.
29-0 —	Reserved

22.7.6.19 DAC Initiator Domain Assignment Register (MDA_W0_2_DFMT1 - MDA_W0_4_DFMT1)

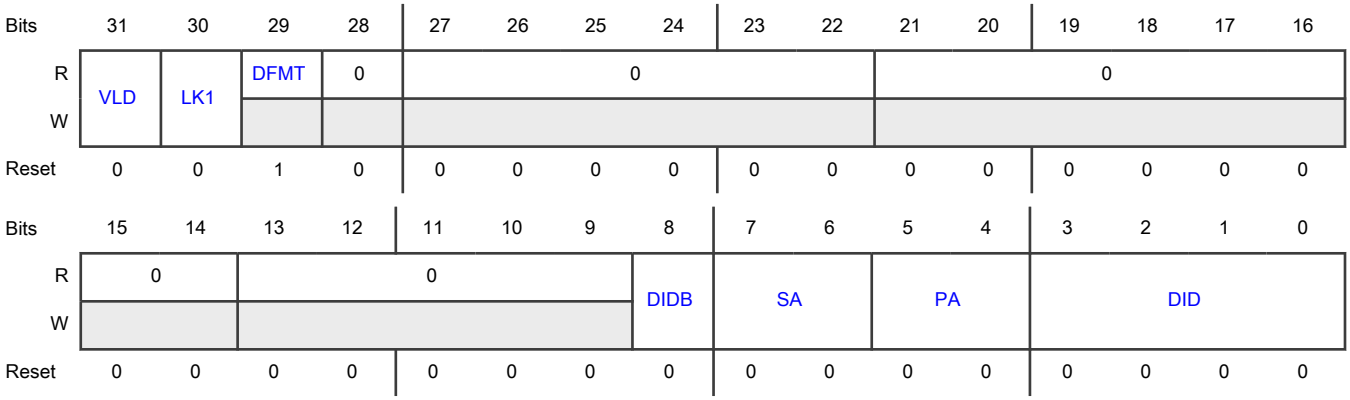
This register is identical to Initiator Domain Assignment (MDA_Wr_m_DFMT0) register except that the domain format field, DFMT, is 1 and the PID field is not used. This format supports two different specifications of the DID for non-core bus initiators.

Access: If TZ-M is enabled, then SecurePrivileged, else NonsecurePriv Read/write.

Offset

Register	Offset
MDA_W0_2_DFMT1	840h
MDA_W0_3_DFMT1	860h
MDA_W0_4_DFMT1	880h

Diagram



Fields

Field	Description
31	Valid

Table continues on the next page...

Table continued from the previous page...

Field	Description
VLD	This field indicates the domain assignment is valid. It is further qualified by TRDC_CR[GVLDM] = 1. If TRDC_CR[GVLDM] is cleared, the DID output is defined by the SoC-specific *default DID* value. See the TRDC chip configuration section for more details. If both MDA_Wr_m[VLD] and TRDC_CR[GVLDM] are asserted, the DID output is defined by the remaining contents of this register. 0 - The Wr domain assignment is invalid. 1 - The Wr domain assignment is valid.
30 LK1	1-bit Lock This field provides a locking mechanism that can be used to limit the ability to write the register. Once set, this bit remains asserted until the next reset. 0 - Register can be written by any secure privileged write. 1 - Register is locked (read-only) until the next reset.
29 DFMT	Domain format Identifies this register's domain assignment format. **NOTE** This bitfield access is ROO 0 - Processor-core domain assignment 1 - Non-processor domain assignment
28 —	Reserved
27-22 —	Reserved
21-16 —	Reserved
15-14 —	Reserved
13-9 —	Reserved
8 DIDB	DID Bypass If asserted, this bit enables the bypassing of an input DID value as the domain identifier for this non-processor bus initiator. This capability allows non-processor bus initiators, for example, a DMA to masquerade as a processor. Once set, this field is “sticky” and remains set until the next reset.

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>DAC</th><th>DID Input</th></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>DMA Domain ID</td></tr>
<tr><td>2</td><td>USB Domain ID</td></tr>
</table>
0 - Use MDAn[3:0] as the domain identifier.
1 - Use the DID input as the domain identifier.
</td></tr>
<tr><td>7-6
SA</td><td>Secure attribute
This field defines the secure/nonsecure attribute for non-processor cores.
NOTE
The bus initiator’s input secure/nonsecure attribute is used if SA = 1X, or this VLD = 0. If TZM_ENB = 0, the initiator attribute is forced to nonsecure. A nonsecure write cannot program this field to 2'b00, which is a security level higher than the mode of the process that is writing it.
Reset value of SA:
• if TZM_ENB=0, SA reset value = 2'b01
• if TZM_ENB=1, SA reset value = 2'b00
00 - Force the bus attribute for this initiator to secure.
01 - Force the bus attribute for this initiator to nonsecure.
10 - Use the bus initiator’s secure/nonsecure attribute directly.
11 - Use the bus initiator’s secure/nonsecure attribute directly.</td></tr>
<tr><td>5-4
PA</td><td>Privileged attribute
This field defines the privileged/user attribute for non-processor cores.
NOTE
The bus initiator’s input privileged/user attribute is used if PA = 1X, or this VLD = 0.
00 - Force the bus attribute for this initiator to user.
01 - Force the bus attribute for this initiator to privileged.
10 - Use the bus initiator’s privileged/user attribute directly.
11 - Use the bus initiator’s privileged/user attribute directly.</td></tr>
<tr><td>3-0
DID</td><td>Domain identifier
This 4-bit field is the domain ID attribute that is sent on the accesses from the bus initiator connected to the DAC when DIDB=0.
Write a value greater than the number of DIDs is illegal.</td></tr>
</table>

22.7.6.20 MBC Global Configuration Register (MBC0_MEM0_GLBCFG - MBC1_MEM3_GLBCFG)

These MBC global configuration read-only registers contain information on the MBC's hardware configuration. Specifically, it defines the number of memory blocks and the size of each block in each MBC mem (r).

NOTE

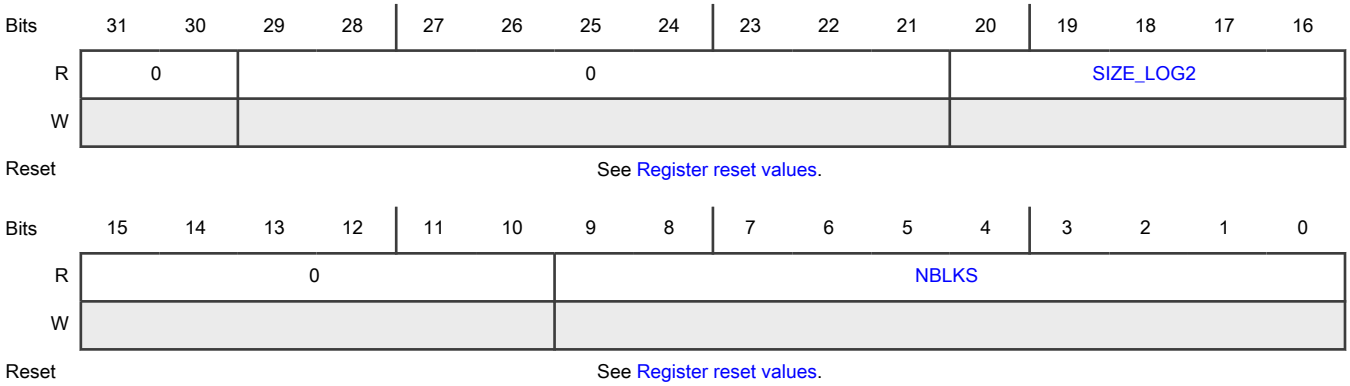
An all zero value (SIZE_LOG2 = 0, NBLKS = 0) indicates a non-existent memory. For the number of memories implemented in the MBCs, refer to chip-specific information

Offset

For m = 0 to 1; r = 0 to 3:

Register	Offset
MBCm_MEMr_GLBCFG	1_0000h + (m × 2000h) + (r × 4h)

Diagram



Register reset values

Register	Reset value
MBC0_MEM0_GLBCFG	000C_0080h
MBC0_MEM1_GLBCFG–MBC0_MEM2_GLBCFG	000C_0001h
MBC0_MEM3_GLBCFG	000C_0011h
MBC1_MEM0_GLBCFG	000C_0080h
MBC1_MEM1_GLBCFG–MBC1_MEM3_GLBCFG	000C_0001h

Fields

Field	Description
31-30 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
29-21 —	Reserved
20-16 SIZE_LOG2	Log2 size per block For example SIZE_LOG2=0x0C is 2^12=4 KB blocks.
15-10 —	Reserved
9-0 NBLKS	Number of blocks in this memory

22.7.6.21 MBC NonSecure Enable Block Index (MBC0_NSE_BLK_INDEX - MBC1_NSE_BLK_INDEX)

This R/W register defines the selected memories that are affected by the writes to the NSE_BLK_SET and NSE_BLK_CLR registers.

Offset

Register	Offset
MBC0_NSE_BLK_INDEX	1_0010h
MBC1_NSE_BLK_INDEX	1_2010h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL9	DID_SEL8	DID_SEL7	DID_SEL6	DID_SEL5	DID_SEL4	DID_SEL3	DID_SEL2	DID_SEL1	DID_SEL0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				MEM_SEL				0		WNDX				0	AI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 DID_SELn	DID Select Destination domain bitmap select. 0 - No effect. 1 - Selects NSE bits for this domain.
15-12 —	Reserved
11-8 MEM_SEL	Memory Select Destination memory bitmap select. • Bit [11] - MBC MEM 3 • Bit [10] - MBC MEM 2 • Bit [9] - MBC MEM 1 • Bit [8] - MBC MEM 0
7-6 —	Reserved
5-2 WNDX	Word index into the block NSE bitmap. It selects the BLK_NSE_Wn register, where WNDX determines the value of n.
1 —	Reserved
0 AI	Auto Increment 0 - No effect. 1 - Add 1 to the WNDX field after the register write.

22.7.6.22 MBC NonSecure Enable Block Set (MBC0_NSE_BLK_SET - MBC1_NSE_BLK_SET)

A write to this register sets the appropriate NSE Bits for the selected domains and memories defined at the word location NSE_BLK_INDEX[WDNX]. If NSE_BLK_INDEX[AI] = 1, then the NSE_BLK_INDEX[WDNX] field is incremented by 1 (modulo-16) after the write is completed. This register reads as zero.

Offset

Register	Offset
MBC0_NSE_BLK_SET	1_0014h
MBC1_NSE_BLK_SET	1_2014h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	W1SET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	W1SET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 W1SET	Write-1 Set If set to 1, sets appropriate NSE bit for the selected domains and memories defined at the word location NSE_BLK_INDEX[WDNX]. Write with value 0 has no effect.

22.7.6.23 MBC NonSecure Enable Block Clear (MBC0_NSE_BLK_CLR - MBC1_NSE_BLK_CLR)

A write to this location clears the appropriate NSE bits as defined by the W1CLR[n]=1 for the selected domains and memories Defined at the word location NSE_BLK_INDEX[WDNX]. If NSE_BLK_INDEX[AI] = 1, then the NSE_BLK_INDEX[WDNX] field is incremented by 1 (modulo-16) after the write is completed. This register reads as zero.

Offset

Register	Offset
MBC0_NSE_BLK_CLR	1_0018h
MBC1_NSE_BLK_CLR	1_2018h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	W1CLR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	W1CLR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 W1CLR	Write-1 Clear If set to 1, Clear the appropriate NSE bit for the selected domains and memories defined at the word location NSE_BLK_INDEX[WDNX]. Write with value 0 has no effect.

22.7.6.24 MBC NonSecure Enable Block Clear All (MBC0_NSE_BLK_CLR_ALL - MBC1_NSE_BLK_CLR_ALL)

A write to this location clears all the block NSE bits for the selected domains and memories defined in the write data. This register reads as zero.

Offset

Register	Offset
MBC0_NSE_BLK_CLR_ALL	1_001Ch
MBC1_NSE_BLK_CLR_ALL	1_201Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL...	DID_SEL9	DID_SEL8	DID_SEL7	DID_SEL6	DID_SEL5	DID_SEL4	DID_SEL3	DID_SEL2	DID_SEL1	DID_SEL0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0				0							
W					MEMSEL											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 DID_SELn	DID Select Destination domain bitmap select. 0 - No effect. 1 - Clear all NSE bits for this domain.
15-12 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
11-8 MEMSEL	Memory Select Destination memory bitmap select. • Bit [11] - clear all MEM 3 NSE bits • Bit [10] - clear all MEM 2 NSE bits • Bit [9] - clear all MEM 1 NSE bits • Bit [8] - clear all MEM 0 NSE bits
7-0 —	Reserved

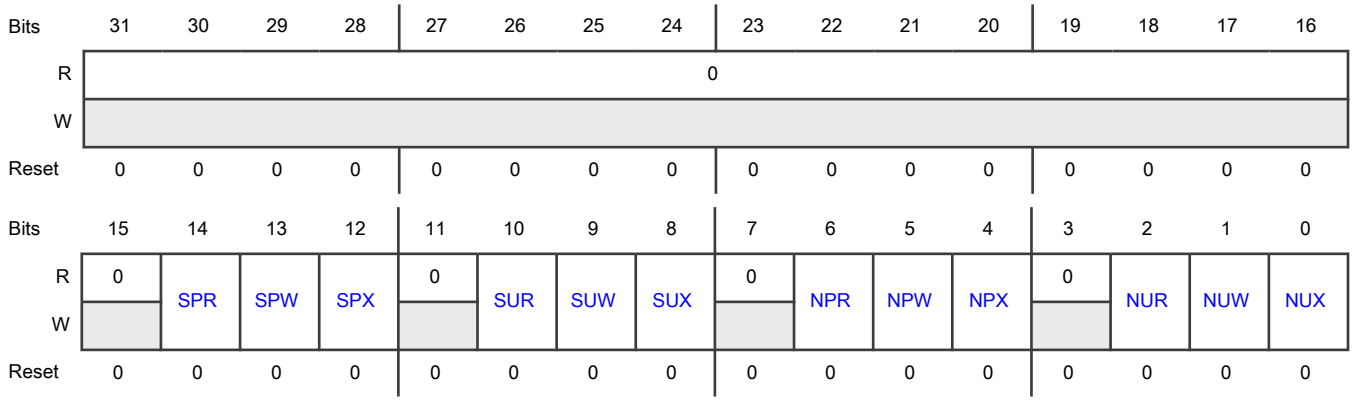
22.7.6.25 MBC Global Access Control (MBC0_MEMN_GLBAC0)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User.

Offset

Register	Offset
MBC0_MEMN_GLBAC0	1_0020h

Diagram



Fields

Field	Description
31-15 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

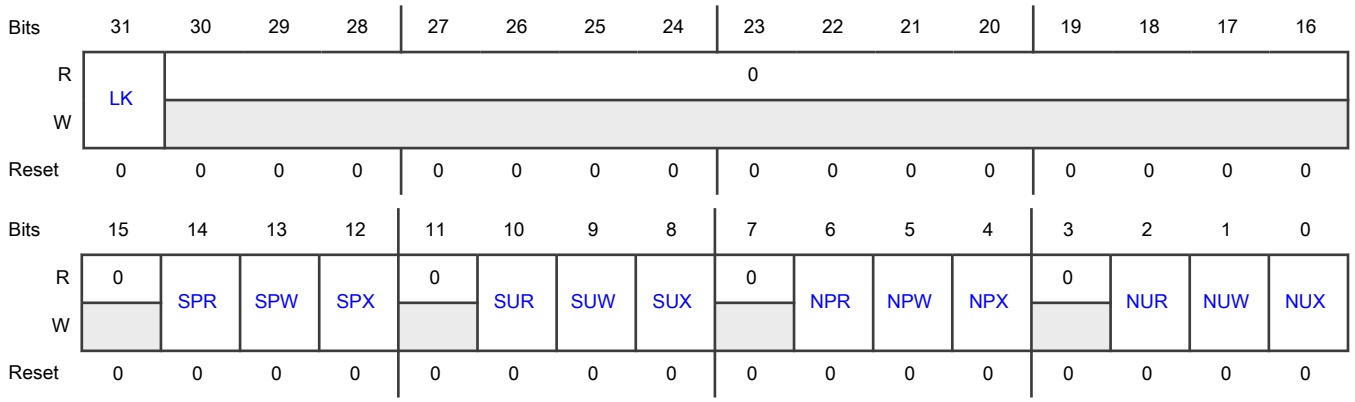
22.7.6.26 MBC Global Access Control (MBC0_MEMN_GLBAC1 - MBC0_MEMN_GLBAC7)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

Offset

Register	Offset
MBC0_MEMN_GLBAC1	1_0024h
MBC0_MEMN_GLBAC2	1_0028h
MBC0_MEMN_GLBAC3	1_002Ch
MBC0_MEMN_GLBAC4	1_0030h
MBC0_MEMN_GLBAC5	1_0034h
MBC0_MEMN_GLBAC6	1_0038h
MBC0_MEMN_GLBAC7	1_003Ch

Diagram



Fields

Field	Description
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0 - This register is not locked and can be altered. 1 - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13	SecurePriv Write

Table continues on the next page...

Table continued from the previous page...

Field	Description
SPW	SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

22.7.6.27 MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W0 - MBC1_DOM15_MEM3_BLK_CFG_W0)

MBC[m]_DOM[d]_MEM[s]_BLK_CFG_W[w], where,

- m - mbc index
- d - domain index
- s - memory target index
- w - word index

These registers are read/write for both the alternate view of the NSE bit and the 3-bit MBACSEL field. This is an array of 4 bit fields defining the block configuration for the given submemory. Each 4-bit field includes an alternative view of the associated NSE bit plus a 3-bit access control select that selects the global access control value that applies to the referenced memory block.

For a given memory block, B, if the value programmed in the MBACSELn field selects a locked MBC_MEMN_GLBACr register, the MBACSEL field is locked until the next reset. Depending on BLK_CFG_W, the NSEn/MBASELn fields correspond to different blocks. The following table describes the relationship between W (word index) and B (block number).

MEM0 supports up to 512 blocks ; W0 - W63

MEM1-3 supports up to 64 block ; W0 - W7

NOTE

This register is shown with fields such that it covers entire 32 bit register. However, actual fields may be less if number of blocks are such that fields don't align to 32 bits. These absent fields may be shown but user can refer to table below to deduce actual fields and treat absent fields as reserved.

Table 116. Block Config Word to Block Number Relationship

Block Config Word	NSE7/ MBACSEL7	NSE6/ MBACSEL6	NSE5/ MBACSEL5	NSE4/ MBACSEL4	NSE3/ MBACSEL3	NSE2/ MBACSEL2	NSE1/ MBACSEL1	NSE0/ MBACSEL0
W0	block 7	block 6	block 5	block 4	block 3	block 2	block 1	block 0
W1	block 15	block 14	block 13	block 12	block 11	block 10	block 9	block 8
W2	block 23	block 22	block 21	block 20	block 19	block 18	block 17	block 16
W3	block 31	block 30	block 29	block 28	block 27	block 26	block 25	block 24
W4	block 39	block 38	block 37	block 36	block 35	block 34	block 33	block 32
W5	block 47	block 46	block 45	block 44	block 43	block 42	block 41	block 40
W6	block 55	block 54	block 53	block 52	block 51	block 50	block 49	block 48
W7	block 63	block 62	block 61	block 60	block 59	block 58	block 57	block 56
For MEM0, block 64 - block 511								
W8	block 71	block 70	block 69	block 68	block 67	block 66	block 65	block 64
. . .								
W63	block 511	block 510	block 509	block 508	block 507	block 506	block 505	block 504

Offset

Register	Offset
MBC0_DOM0_MEM0_BLK_CFG_W0	1_0040h
MBC0_DOM0_MEM0_BLK_CFG_W1	1_0044h
MBC0_DOM0_MEM0_BLK_CFG_W2	1_0048h
MBC0_DOM0_MEM0_BLK_CFG_W3	1_004Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM0_MEM0_BLK_CFG_W4	1_0050h
MBC0_DOM0_MEM0_BLK_CFG_W5	1_0054h
MBC0_DOM0_MEM0_BLK_CFG_W6	1_0058h
MBC0_DOM0_MEM0_BLK_CFG_W7	1_005Ch
MBC0_DOM0_MEM0_BLK_CFG_W8	1_0060h
MBC0_DOM0_MEM0_BLK_CFG_W9	1_0064h
MBC0_DOM0_MEM0_BLK_CFG_W10	1_0068h
MBC0_DOM0_MEM0_BLK_CFG_W11	1_006Ch
MBC0_DOM0_MEM0_BLK_CFG_W12	1_0070h
MBC0_DOM0_MEM0_BLK_CFG_W13	1_0074h
MBC0_DOM0_MEM0_BLK_CFG_W14	1_0078h
MBC0_DOM0_MEM0_BLK_CFG_W15	1_007Ch
MBC0_DOM0_MEM1_BLK_CFG_W0	1_0180h
MBC0_DOM0_MEM2_BLK_CFG_W0	1_01A8h
MBC0_DOM0_MEM3_BLK_CFG_W0	1_01D0h
MBC0_DOM0_MEM3_BLK_CFG_W1	1_01D4h
MBC0_DOM0_MEM3_BLK_CFG_W2	1_01D8h
MBC0_DOM1_MEM0_BLK_CFG_W0	1_0240h
MBC0_DOM1_MEM0_BLK_CFG_W1	1_0244h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM1_MEM0_BLK_CFG_W2	1_0248h
MBC0_DOM1_MEM0_BLK_CFG_W3	1_024Ch
MBC0_DOM1_MEM0_BLK_CFG_W4	1_0250h
MBC0_DOM1_MEM0_BLK_CFG_W5	1_0254h
MBC0_DOM1_MEM0_BLK_CFG_W6	1_0258h
MBC0_DOM1_MEM0_BLK_CFG_W7	1_025Ch
MBC0_DOM1_MEM0_BLK_CFG_W8	1_0260h
MBC0_DOM1_MEM0_BLK_CFG_W9	1_0264h
MBC0_DOM1_MEM0_BLK_CFG_W10	1_0268h
MBC0_DOM1_MEM0_BLK_CFG_W11	1_026Ch
MBC0_DOM1_MEM0_BLK_CFG_W12	1_0270h
MBC0_DOM1_MEM0_BLK_CFG_W13	1_0274h
MBC0_DOM1_MEM0_BLK_CFG_W14	1_0278h
MBC0_DOM1_MEM0_BLK_CFG_W15	1_027Ch
MBC0_DOM1_MEM1_BLK_CFG_W0	1_0380h
MBC0_DOM1_MEM2_BLK_CFG_W0	1_03A8h
MBC0_DOM1_MEM3_BLK_CFG_W0	1_03D0h
MBC0_DOM1_MEM3_BLK_CFG_W1	1_03D4h
MBC0_DOM1_MEM3_BLK_CFG_W2	1_03D8h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM2_MEM0_BLK_CFG_W0	1_0440h
MBC0_DOM2_MEM0_BLK_CFG_W1	1_0444h
MBC0_DOM2_MEM0_BLK_CFG_W2	1_0448h
MBC0_DOM2_MEM0_BLK_CFG_W3	1_044Ch
MBC0_DOM2_MEM0_BLK_CFG_W4	1_0450h
MBC0_DOM2_MEM0_BLK_CFG_W5	1_0454h
MBC0_DOM2_MEM0_BLK_CFG_W6	1_0458h
MBC0_DOM2_MEM0_BLK_CFG_W7	1_045Ch
MBC0_DOM2_MEM0_BLK_CFG_W8	1_0460h
MBC0_DOM2_MEM0_BLK_CFG_W9	1_0464h
MBC0_DOM2_MEM0_BLK_CFG_W10	1_0468h
MBC0_DOM2_MEM0_BLK_CFG_W11	1_046Ch
MBC0_DOM2_MEM0_BLK_CFG_W12	1_0470h
MBC0_DOM2_MEM0_BLK_CFG_W13	1_0474h
MBC0_DOM2_MEM0_BLK_CFG_W14	1_0478h
MBC0_DOM2_MEM0_BLK_CFG_W15	1_047Ch
MBC0_DOM2_MEM1_BLK_CFG_W0	1_0580h
MBC0_DOM2_MEM2_BLK_CFG_W0	1_05A8h
MBC0_DOM2_MEM3_BLK_CFG_W0	1_05D0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM2_MEM3_BLK_CFG_W1	1_05D4h
MBC0_DOM2_MEM3_BLK_CFG_W2	1_05D8h
MBC0_DOM3_MEM0_BLK_CFG_W0	1_0640h
MBC0_DOM3_MEM0_BLK_CFG_W1	1_0644h
MBC0_DOM3_MEM0_BLK_CFG_W2	1_0648h
MBC0_DOM3_MEM0_BLK_CFG_W3	1_064Ch
MBC0_DOM3_MEM0_BLK_CFG_W4	1_0650h
MBC0_DOM3_MEM0_BLK_CFG_W5	1_0654h
MBC0_DOM3_MEM0_BLK_CFG_W6	1_0658h
MBC0_DOM3_MEM0_BLK_CFG_W7	1_065Ch
MBC0_DOM3_MEM0_BLK_CFG_W8	1_0660h
MBC0_DOM3_MEM0_BLK_CFG_W9	1_0664h
MBC0_DOM3_MEM0_BLK_CFG_W10	1_0668h
MBC0_DOM3_MEM0_BLK_CFG_W11	1_066Ch
MBC0_DOM3_MEM0_BLK_CFG_W12	1_0670h
MBC0_DOM3_MEM0_BLK_CFG_W13	1_0674h
MBC0_DOM3_MEM0_BLK_CFG_W14	1_0678h
MBC0_DOM3_MEM0_BLK_CFG_W15	1_067Ch
MBC0_DOM3_MEM1_BLK_CFG_W0	1_0780h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM3_MEM2_BLK_CFG_W0	1_07A8h
MBC0_DOM3_MEM3_BLK_CFG_W0	1_07D0h
MBC0_DOM3_MEM3_BLK_CFG_W1	1_07D4h
MBC0_DOM3_MEM3_BLK_CFG_W2	1_07D8h
MBC0_DOM4_MEM0_BLK_CFG_W0	1_0840h
MBC0_DOM4_MEM0_BLK_CFG_W1	1_0844h
MBC0_DOM4_MEM0_BLK_CFG_W2	1_0848h
MBC0_DOM4_MEM0_BLK_CFG_W3	1_084Ch
MBC0_DOM4_MEM0_BLK_CFG_W4	1_0850h
MBC0_DOM4_MEM0_BLK_CFG_W5	1_0854h
MBC0_DOM4_MEM0_BLK_CFG_W6	1_0858h
MBC0_DOM4_MEM0_BLK_CFG_W7	1_085Ch
MBC0_DOM4_MEM0_BLK_CFG_W8	1_0860h
MBC0_DOM4_MEM0_BLK_CFG_W9	1_0864h
MBC0_DOM4_MEM0_BLK_CFG_W10	1_0868h
MBC0_DOM4_MEM0_BLK_CFG_W11	1_086Ch
MBC0_DOM4_MEM0_BLK_CFG_W12	1_0870h
MBC0_DOM4_MEM0_BLK_CFG_W13	1_0874h
MBC0_DOM4_MEM0_BLK_CFG_W14	1_0878h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM4_MEM0_BLK_CFG_W15	1_087Ch
MBC0_DOM4_MEM1_BLK_CFG_W0	1_0980h
MBC0_DOM4_MEM2_BLK_CFG_W0	1_09A8h
MBC0_DOM4_MEM3_BLK_CFG_W0	1_09D0h
MBC0_DOM4_MEM3_BLK_CFG_W1	1_09D4h
MBC0_DOM4_MEM3_BLK_CFG_W2	1_09D8h
MBC0_DOM5_MEM0_BLK_CFG_W0	1_0A40h
MBC0_DOM5_MEM0_BLK_CFG_W1	1_0A44h
MBC0_DOM5_MEM0_BLK_CFG_W2	1_0A48h
MBC0_DOM5_MEM0_BLK_CFG_W3	1_0A4Ch
MBC0_DOM5_MEM0_BLK_CFG_W4	1_0A50h
MBC0_DOM5_MEM0_BLK_CFG_W5	1_0A54h
MBC0_DOM5_MEM0_BLK_CFG_W6	1_0A58h
MBC0_DOM5_MEM0_BLK_CFG_W7	1_0A5Ch
MBC0_DOM5_MEM0_BLK_CFG_W8	1_0A60h
MBC0_DOM5_MEM0_BLK_CFG_W9	1_0A64h
MBC0_DOM5_MEM0_BLK_CFG_W10	1_0A68h
MBC0_DOM5_MEM0_BLK_CFG_W11	1_0A6Ch
MBC0_DOM5_MEM0_BLK_CFG_W12	1_0A70h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM5_MEM0_BLK_CFG_W13	1_0A74h
MBC0_DOM5_MEM0_BLK_CFG_W14	1_0A78h
MBC0_DOM5_MEM0_BLK_CFG_W15	1_0A7Ch
MBC0_DOM5_MEM1_BLK_CFG_W0	1_0B80h
MBC0_DOM5_MEM2_BLK_CFG_W0	1_0BA8h
MBC0_DOM5_MEM3_BLK_CFG_W0	1_0BD0h
MBC0_DOM5_MEM3_BLK_CFG_W1	1_0BD4h
MBC0_DOM5_MEM3_BLK_CFG_W2	1_0BD8h
MBC0_DOM6_MEM0_BLK_CFG_W0	1_0C40h
MBC0_DOM6_MEM0_BLK_CFG_W1	1_0C44h
MBC0_DOM6_MEM0_BLK_CFG_W2	1_0C48h
MBC0_DOM6_MEM0_BLK_CFG_W3	1_0C4Ch
MBC0_DOM6_MEM0_BLK_CFG_W4	1_0C50h
MBC0_DOM6_MEM0_BLK_CFG_W5	1_0C54h
MBC0_DOM6_MEM0_BLK_CFG_W6	1_0C58h
MBC0_DOM6_MEM0_BLK_CFG_W7	1_0C5Ch
MBC0_DOM6_MEM0_BLK_CFG_W8	1_0C60h
MBC0_DOM6_MEM0_BLK_CFG_W9	1_0C64h
MBC0_DOM6_MEM0_BLK_CFG_W10	1_0C68h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM6_MEM0_BLK_CFG_W11	1_0C6Ch
MBC0_DOM6_MEM0_BLK_CFG_W12	1_0C70h
MBC0_DOM6_MEM0_BLK_CFG_W13	1_0C74h
MBC0_DOM6_MEM0_BLK_CFG_W14	1_0C78h
MBC0_DOM6_MEM0_BLK_CFG_W15	1_0C7Ch
MBC0_DOM6_MEM1_BLK_CFG_W0	1_0D80h
MBC0_DOM6_MEM2_BLK_CFG_W0	1_0DA8h
MBC0_DOM6_MEM3_BLK_CFG_W0	1_0DD0h
MBC0_DOM6_MEM3_BLK_CFG_W1	1_0DD4h
MBC0_DOM6_MEM3_BLK_CFG_W2	1_0DD8h
MBC0_DOM7_MEM0_BLK_CFG_W0	1_0E40h
MBC0_DOM7_MEM0_BLK_CFG_W1	1_0E44h
MBC0_DOM7_MEM0_BLK_CFG_W2	1_0E48h
MBC0_DOM7_MEM0_BLK_CFG_W3	1_0E4Ch
MBC0_DOM7_MEM0_BLK_CFG_W4	1_0E50h
MBC0_DOM7_MEM0_BLK_CFG_W5	1_0E54h
MBC0_DOM7_MEM0_BLK_CFG_W6	1_0E58h
MBC0_DOM7_MEM0_BLK_CFG_W7	1_0E5Ch
MBC0_DOM7_MEM0_BLK_CFG_W8	1_0E60h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM7_MEM0_BLK_CFG_W9	1_0E64h
MBC0_DOM7_MEM0_BLK_CFG_W10	1_0E68h
MBC0_DOM7_MEM0_BLK_CFG_W11	1_0E6Ch
MBC0_DOM7_MEM0_BLK_CFG_W12	1_0E70h
MBC0_DOM7_MEM0_BLK_CFG_W13	1_0E74h
MBC0_DOM7_MEM0_BLK_CFG_W14	1_0E78h
MBC0_DOM7_MEM0_BLK_CFG_W15	1_0E7Ch
MBC0_DOM7_MEM1_BLK_CFG_W0	1_0F80h
MBC0_DOM7_MEM2_BLK_CFG_W0	1_0FA8h
MBC0_DOM7_MEM3_BLK_CFG_W0	1_0FD0h
MBC0_DOM7_MEM3_BLK_CFG_W1	1_0FD4h
MBC0_DOM7_MEM3_BLK_CFG_W2	1_0FD8h
MBC0_DOM8_MEM0_BLK_CFG_W0	1_1040h
MBC0_DOM8_MEM0_BLK_CFG_W1	1_1044h
MBC0_DOM8_MEM0_BLK_CFG_W2	1_1048h
MBC0_DOM8_MEM0_BLK_CFG_W3	1_104Ch
MBC0_DOM8_MEM0_BLK_CFG_W4	1_1050h
MBC0_DOM8_MEM0_BLK_CFG_W5	1_1054h
MBC0_DOM8_MEM0_BLK_CFG_W6	1_1058h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM8_MEM0_BLK_CFG_W7	1_105Ch
MBC0_DOM8_MEM0_BLK_CFG_W8	1_1060h
MBC0_DOM8_MEM0_BLK_CFG_W9	1_1064h
MBC0_DOM8_MEM0_BLK_CFG_W10	1_1068h
MBC0_DOM8_MEM0_BLK_CFG_W11	1_106Ch
MBC0_DOM8_MEM0_BLK_CFG_W12	1_1070h
MBC0_DOM8_MEM0_BLK_CFG_W13	1_1074h
MBC0_DOM8_MEM0_BLK_CFG_W14	1_1078h
MBC0_DOM8_MEM0_BLK_CFG_W15	1_107Ch
MBC0_DOM8_MEM1_BLK_CFG_W0	1_1180h
MBC0_DOM8_MEM2_BLK_CFG_W0	1_11A8h
MBC0_DOM8_MEM3_BLK_CFG_W0	1_11D0h
MBC0_DOM8_MEM3_BLK_CFG_W1	1_11D4h
MBC0_DOM8_MEM3_BLK_CFG_W2	1_11D8h
MBC0_DOM9_MEM0_BLK_CFG_W0	1_1240h
MBC0_DOM9_MEM0_BLK_CFG_W1	1_1244h
MBC0_DOM9_MEM0_BLK_CFG_W2	1_1248h
MBC0_DOM9_MEM0_BLK_CFG_W3	1_124Ch
MBC0_DOM9_MEM0_BLK_CFG_W4	1_1250h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM9_MEM0_BLK_CFG_W5	1_1254h
MBC0_DOM9_MEM0_BLK_CFG_W6	1_1258h
MBC0_DOM9_MEM0_BLK_CFG_W7	1_125Ch
MBC0_DOM9_MEM0_BLK_CFG_W8	1_1260h
MBC0_DOM9_MEM0_BLK_CFG_W9	1_1264h
MBC0_DOM9_MEM0_BLK_CFG_W10	1_1268h
MBC0_DOM9_MEM0_BLK_CFG_W11	1_126Ch
MBC0_DOM9_MEM0_BLK_CFG_W12	1_1270h
MBC0_DOM9_MEM0_BLK_CFG_W13	1_1274h
MBC0_DOM9_MEM0_BLK_CFG_W14	1_1278h
MBC0_DOM9_MEM0_BLK_CFG_W15	1_127Ch
MBC0_DOM9_MEM1_BLK_CFG_W0	1_1380h
MBC0_DOM9_MEM2_BLK_CFG_W0	1_13A8h
MBC0_DOM9_MEM3_BLK_CFG_W0	1_13D0h
MBC0_DOM9_MEM3_BLK_CFG_W1	1_13D4h
MBC0_DOM9_MEM3_BLK_CFG_W2	1_13D8h
MBC0_DOM10_MEM0_BLK_CFG_W0	1_1440h
MBC0_DOM10_MEM0_BLK_CFG_W1	1_1444h
MBC0_DOM10_MEM0_BLK_CFG_W2	1_1448h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM10_MEM0_BLK_CFG_W3	1_144Ch
MBC0_DOM10_MEM0_BLK_CFG_W4	1_1450h
MBC0_DOM10_MEM0_BLK_CFG_W5	1_1454h
MBC0_DOM10_MEM0_BLK_CFG_W6	1_1458h
MBC0_DOM10_MEM0_BLK_CFG_W7	1_145Ch
MBC0_DOM10_MEM0_BLK_CFG_W8	1_1460h
MBC0_DOM10_MEM0_BLK_CFG_W9	1_1464h
MBC0_DOM10_MEM0_BLK_CFG_W10	1_1468h
MBC0_DOM10_MEM0_BLK_CFG_W11	1_146Ch
MBC0_DOM10_MEM0_BLK_CFG_W12	1_1470h
MBC0_DOM10_MEM0_BLK_CFG_W13	1_1474h
MBC0_DOM10_MEM0_BLK_CFG_W14	1_1478h
MBC0_DOM10_MEM0_BLK_CFG_W15	1_147Ch
MBC0_DOM10_MEM1_BLK_CFG_W0	1_1580h
MBC0_DOM10_MEM2_BLK_CFG_W0	1_15A8h
MBC0_DOM10_MEM3_BLK_CFG_W0	1_15D0h
MBC0_DOM10_MEM3_BLK_CFG_W1	1_15D4h
MBC0_DOM10_MEM3_BLK_CFG_W2	1_15D8h
MBC0_DOM11_MEM0_BLK_CFG_W0	1_1640h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM11_MEM0_BLK_CFG_W1	1_1644h
MBC0_DOM11_MEM0_BLK_CFG_W2	1_1648h
MBC0_DOM11_MEM0_BLK_CFG_W3	1_164Ch
MBC0_DOM11_MEM0_BLK_CFG_W4	1_1650h
MBC0_DOM11_MEM0_BLK_CFG_W5	1_1654h
MBC0_DOM11_MEM0_BLK_CFG_W6	1_1658h
MBC0_DOM11_MEM0_BLK_CFG_W7	1_165Ch
MBC0_DOM11_MEM0_BLK_CFG_W8	1_1660h
MBC0_DOM11_MEM0_BLK_CFG_W9	1_1664h
MBC0_DOM11_MEM0_BLK_CFG_W10	1_1668h
MBC0_DOM11_MEM0_BLK_CFG_W11	1_166Ch
MBC0_DOM11_MEM0_BLK_CFG_W12	1_1670h
MBC0_DOM11_MEM0_BLK_CFG_W13	1_1674h
MBC0_DOM11_MEM0_BLK_CFG_W14	1_1678h
MBC0_DOM11_MEM0_BLK_CFG_W15	1_167Ch
MBC0_DOM11_MEM1_BLK_CFG_W0	1_1780h
MBC0_DOM11_MEM2_BLK_CFG_W0	1_17A8h
MBC0_DOM11_MEM3_BLK_CFG_W0	1_17D0h
MBC0_DOM11_MEM3_BLK_CFG_W1	1_17D4h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM11_MEM3_BLK_CFG_W2	1_17D8h
MBC0_DOM12_MEM0_BLK_CFG_W0	1_1840h
MBC0_DOM12_MEM0_BLK_CFG_W1	1_1844h
MBC0_DOM12_MEM0_BLK_CFG_W2	1_1848h
MBC0_DOM12_MEM0_BLK_CFG_W3	1_184Ch
MBC0_DOM12_MEM0_BLK_CFG_W4	1_1850h
MBC0_DOM12_MEM0_BLK_CFG_W5	1_1854h
MBC0_DOM12_MEM0_BLK_CFG_W6	1_1858h
MBC0_DOM12_MEM0_BLK_CFG_W7	1_185Ch
MBC0_DOM12_MEM0_BLK_CFG_W8	1_1860h
MBC0_DOM12_MEM0_BLK_CFG_W9	1_1864h
MBC0_DOM12_MEM0_BLK_CFG_W10	1_1868h
MBC0_DOM12_MEM0_BLK_CFG_W11	1_186Ch
MBC0_DOM12_MEM0_BLK_CFG_W12	1_1870h
MBC0_DOM12_MEM0_BLK_CFG_W13	1_1874h
MBC0_DOM12_MEM0_BLK_CFG_W14	1_1878h
MBC0_DOM12_MEM0_BLK_CFG_W15	1_187Ch
MBC0_DOM12_MEM1_BLK_CFG_W0	1_1980h
MBC0_DOM12_MEM2_BLK_CFG_W0	1_19A8h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM12_MEM3_BLK_CFG_W0	1_19D0h
MBC0_DOM12_MEM3_BLK_CFG_W1	1_19D4h
MBC0_DOM12_MEM3_BLK_CFG_W2	1_19D8h
MBC0_DOM13_MEM0_BLK_CFG_W0	1_1A40h
MBC0_DOM13_MEM0_BLK_CFG_W1	1_1A44h
MBC0_DOM13_MEM0_BLK_CFG_W2	1_1A48h
MBC0_DOM13_MEM0_BLK_CFG_W3	1_1A4Ch
MBC0_DOM13_MEM0_BLK_CFG_W4	1_1A50h
MBC0_DOM13_MEM0_BLK_CFG_W5	1_1A54h
MBC0_DOM13_MEM0_BLK_CFG_W6	1_1A58h
MBC0_DOM13_MEM0_BLK_CFG_W7	1_1A5Ch
MBC0_DOM13_MEM0_BLK_CFG_W8	1_1A60h
MBC0_DOM13_MEM0_BLK_CFG_W9	1_1A64h
MBC0_DOM13_MEM0_BLK_CFG_W10	1_1A68h
MBC0_DOM13_MEM0_BLK_CFG_W11	1_1A6Ch
MBC0_DOM13_MEM0_BLK_CFG_W12	1_1A70h
MBC0_DOM13_MEM0_BLK_CFG_W13	1_1A74h
MBC0_DOM13_MEM0_BLK_CFG_W14	1_1A78h
MBC0_DOM13_MEM0_BLK_CFG_W15	1_1A7Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM13_MEM1_BLK_CFG_W0	1_1B80h
MBC0_DOM13_MEM2_BLK_CFG_W0	1_1BA8h
MBC0_DOM13_MEM3_BLK_CFG_W0	1_1BD0h
MBC0_DOM13_MEM3_BLK_CFG_W1	1_1BD4h
MBC0_DOM13_MEM3_BLK_CFG_W2	1_1BD8h
MBC0_DOM14_MEM0_BLK_CFG_W0	1_1C40h
MBC0_DOM14_MEM0_BLK_CFG_W1	1_1C44h
MBC0_DOM14_MEM0_BLK_CFG_W2	1_1C48h
MBC0_DOM14_MEM0_BLK_CFG_W3	1_1C4Ch
MBC0_DOM14_MEM0_BLK_CFG_W4	1_1C50h
MBC0_DOM14_MEM0_BLK_CFG_W5	1_1C54h
MBC0_DOM14_MEM0_BLK_CFG_W6	1_1C58h
MBC0_DOM14_MEM0_BLK_CFG_W7	1_1C5Ch
MBC0_DOM14_MEM0_BLK_CFG_W8	1_1C60h
MBC0_DOM14_MEM0_BLK_CFG_W9	1_1C64h
MBC0_DOM14_MEM0_BLK_CFG_W10	1_1C68h
MBC0_DOM14_MEM0_BLK_CFG_W11	1_1C6Ch
MBC0_DOM14_MEM0_BLK_CFG_W12	1_1C70h
MBC0_DOM14_MEM0_BLK_CFG_W13	1_1C74h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM14_MEM0_BLK_CFG_W14	1_1C78h
MBC0_DOM14_MEM0_BLK_CFG_W15	1_1C7Ch
MBC0_DOM14_MEM1_BLK_CFG_W0	1_1D80h
MBC0_DOM14_MEM2_BLK_CFG_W0	1_1DA8h
MBC0_DOM14_MEM3_BLK_CFG_W0	1_1DD0h
MBC0_DOM14_MEM3_BLK_CFG_W1	1_1DD4h
MBC0_DOM14_MEM3_BLK_CFG_W2	1_1DD8h
MBC0_DOM15_MEM0_BLK_CFG_W0	1_1E40h
MBC0_DOM15_MEM0_BLK_CFG_W1	1_1E44h
MBC0_DOM15_MEM0_BLK_CFG_W2	1_1E48h
MBC0_DOM15_MEM0_BLK_CFG_W3	1_1E4Ch
MBC0_DOM15_MEM0_BLK_CFG_W4	1_1E50h
MBC0_DOM15_MEM0_BLK_CFG_W5	1_1E54h
MBC0_DOM15_MEM0_BLK_CFG_W6	1_1E58h
MBC0_DOM15_MEM0_BLK_CFG_W7	1_1E5Ch
MBC0_DOM15_MEM0_BLK_CFG_W8	1_1E60h
MBC0_DOM15_MEM0_BLK_CFG_W9	1_1E64h
MBC0_DOM15_MEM0_BLK_CFG_W10	1_1E68h
MBC0_DOM15_MEM0_BLK_CFG_W11	1_1E6Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM15_MEM0_BLK_CFG_W12	1_1E70h
MBC0_DOM15_MEM0_BLK_CFG_W13	1_1E74h
MBC0_DOM15_MEM0_BLK_CFG_W14	1_1E78h
MBC0_DOM15_MEM0_BLK_CFG_W15	1_1E7Ch
MBC0_DOM15_MEM1_BLK_CFG_W0	1_1F80h
MBC0_DOM15_MEM2_BLK_CFG_W0	1_1FA8h
MBC0_DOM15_MEM3_BLK_CFG_W0	1_1FD0h
MBC0_DOM15_MEM3_BLK_CFG_W1	1_1FD4h
MBC0_DOM15_MEM3_BLK_CFG_W2	1_1FD8h
MBC1_DOM0_MEM0_BLK_CFG_W0	1_2040h
MBC1_DOM0_MEM0_BLK_CFG_W1	1_2044h
MBC1_DOM0_MEM0_BLK_CFG_W2	1_2048h
MBC1_DOM0_MEM0_BLK_CFG_W3	1_204Ch
MBC1_DOM0_MEM0_BLK_CFG_W4	1_2050h
MBC1_DOM0_MEM0_BLK_CFG_W5	1_2054h
MBC1_DOM0_MEM0_BLK_CFG_W6	1_2058h
MBC1_DOM0_MEM0_BLK_CFG_W7	1_205Ch
MBC1_DOM0_MEM0_BLK_CFG_W8	1_2060h
MBC1_DOM0_MEM0_BLK_CFG_W9	1_2064h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM0_MEM0_BLK_CFG_W10	1_2068h
MBC1_DOM0_MEM0_BLK_CFG_W11	1_206Ch
MBC1_DOM0_MEM0_BLK_CFG_W12	1_2070h
MBC1_DOM0_MEM0_BLK_CFG_W13	1_2074h
MBC1_DOM0_MEM0_BLK_CFG_W14	1_2078h
MBC1_DOM0_MEM0_BLK_CFG_W15	1_207Ch
MBC1_DOM0_MEM1_BLK_CFG_W0	1_2180h
MBC1_DOM0_MEM2_BLK_CFG_W0	1_21A8h
MBC1_DOM0_MEM3_BLK_CFG_W0	1_21D0h
MBC1_DOM1_MEM0_BLK_CFG_W0	1_2240h
MBC1_DOM1_MEM0_BLK_CFG_W1	1_2244h
MBC1_DOM1_MEM0_BLK_CFG_W2	1_2248h
MBC1_DOM1_MEM0_BLK_CFG_W3	1_224Ch
MBC1_DOM1_MEM0_BLK_CFG_W4	1_2250h
MBC1_DOM1_MEM0_BLK_CFG_W5	1_2254h
MBC1_DOM1_MEM0_BLK_CFG_W6	1_2258h
MBC1_DOM1_MEM0_BLK_CFG_W7	1_225Ch
MBC1_DOM1_MEM0_BLK_CFG_W8	1_2260h
MBC1_DOM1_MEM0_BLK_CFG_W9	1_2264h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM1_MEM0_BLK_CFG_W10	1_2268h
MBC1_DOM1_MEM0_BLK_CFG_W11	1_226Ch
MBC1_DOM1_MEM0_BLK_CFG_W12	1_2270h
MBC1_DOM1_MEM0_BLK_CFG_W13	1_2274h
MBC1_DOM1_MEM0_BLK_CFG_W14	1_2278h
MBC1_DOM1_MEM0_BLK_CFG_W15	1_227Ch
MBC1_DOM1_MEM1_BLK_CFG_W0	1_2380h
MBC1_DOM1_MEM2_BLK_CFG_W0	1_23A8h
MBC1_DOM1_MEM3_BLK_CFG_W0	1_23D0h
MBC1_DOM2_MEM0_BLK_CFG_W0	1_2440h
MBC1_DOM2_MEM0_BLK_CFG_W1	1_2444h
MBC1_DOM2_MEM0_BLK_CFG_W2	1_2448h
MBC1_DOM2_MEM0_BLK_CFG_W3	1_244Ch
MBC1_DOM2_MEM0_BLK_CFG_W4	1_2450h
MBC1_DOM2_MEM0_BLK_CFG_W5	1_2454h
MBC1_DOM2_MEM0_BLK_CFG_W6	1_2458h
MBC1_DOM2_MEM0_BLK_CFG_W7	1_245Ch
MBC1_DOM2_MEM0_BLK_CFG_W8	1_2460h
MBC1_DOM2_MEM0_BLK_CFG_W9	1_2464h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM2_MEM0_BLK_CFG_W10	1_2468h
MBC1_DOM2_MEM0_BLK_CFG_W11	1_246Ch
MBC1_DOM2_MEM0_BLK_CFG_W12	1_2470h
MBC1_DOM2_MEM0_BLK_CFG_W13	1_2474h
MBC1_DOM2_MEM0_BLK_CFG_W14	1_2478h
MBC1_DOM2_MEM0_BLK_CFG_W15	1_247Ch
MBC1_DOM2_MEM1_BLK_CFG_W0	1_2580h
MBC1_DOM2_MEM2_BLK_CFG_W0	1_25A8h
MBC1_DOM2_MEM3_BLK_CFG_W0	1_25D0h
MBC1_DOM3_MEM0_BLK_CFG_W0	1_2640h
MBC1_DOM3_MEM0_BLK_CFG_W1	1_2644h
MBC1_DOM3_MEM0_BLK_CFG_W2	1_2648h
MBC1_DOM3_MEM0_BLK_CFG_W3	1_264Ch
MBC1_DOM3_MEM0_BLK_CFG_W4	1_2650h
MBC1_DOM3_MEM0_BLK_CFG_W5	1_2654h
MBC1_DOM3_MEM0_BLK_CFG_W6	1_2658h
MBC1_DOM3_MEM0_BLK_CFG_W7	1_265Ch
MBC1_DOM3_MEM0_BLK_CFG_W8	1_2660h
MBC1_DOM3_MEM0_BLK_CFG_W9	1_2664h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM3_MEM0_BLK_CFG_W10	1_2668h
MBC1_DOM3_MEM0_BLK_CFG_W11	1_266Ch
MBC1_DOM3_MEM0_BLK_CFG_W12	1_2670h
MBC1_DOM3_MEM0_BLK_CFG_W13	1_2674h
MBC1_DOM3_MEM0_BLK_CFG_W14	1_2678h
MBC1_DOM3_MEM0_BLK_CFG_W15	1_267Ch
MBC1_DOM3_MEM1_BLK_CFG_W0	1_2780h
MBC1_DOM3_MEM2_BLK_CFG_W0	1_27A8h
MBC1_DOM3_MEM3_BLK_CFG_W0	1_27D0h
MBC1_DOM4_MEM0_BLK_CFG_W0	1_2840h
MBC1_DOM4_MEM0_BLK_CFG_W1	1_2844h
MBC1_DOM4_MEM0_BLK_CFG_W2	1_2848h
MBC1_DOM4_MEM0_BLK_CFG_W3	1_284Ch
MBC1_DOM4_MEM0_BLK_CFG_W4	1_2850h
MBC1_DOM4_MEM0_BLK_CFG_W5	1_2854h
MBC1_DOM4_MEM0_BLK_CFG_W6	1_2858h
MBC1_DOM4_MEM0_BLK_CFG_W7	1_285Ch
MBC1_DOM4_MEM0_BLK_CFG_W8	1_2860h
MBC1_DOM4_MEM0_BLK_CFG_W9	1_2864h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM4_MEM0_BLK_CFG_W10	1_2868h
MBC1_DOM4_MEM0_BLK_CFG_W11	1_286Ch
MBC1_DOM4_MEM0_BLK_CFG_W12	1_2870h
MBC1_DOM4_MEM0_BLK_CFG_W13	1_2874h
MBC1_DOM4_MEM0_BLK_CFG_W14	1_2878h
MBC1_DOM4_MEM0_BLK_CFG_W15	1_287Ch
MBC1_DOM4_MEM1_BLK_CFG_W0	1_2980h
MBC1_DOM4_MEM2_BLK_CFG_W0	1_29A8h
MBC1_DOM4_MEM3_BLK_CFG_W0	1_29D0h
MBC1_DOM5_MEM0_BLK_CFG_W0	1_2A40h
MBC1_DOM5_MEM0_BLK_CFG_W1	1_2A44h
MBC1_DOM5_MEM0_BLK_CFG_W2	1_2A48h
MBC1_DOM5_MEM0_BLK_CFG_W3	1_2A4Ch
MBC1_DOM5_MEM0_BLK_CFG_W4	1_2A50h
MBC1_DOM5_MEM0_BLK_CFG_W5	1_2A54h
MBC1_DOM5_MEM0_BLK_CFG_W6	1_2A58h
MBC1_DOM5_MEM0_BLK_CFG_W7	1_2A5Ch
MBC1_DOM5_MEM0_BLK_CFG_W8	1_2A60h
MBC1_DOM5_MEM0_BLK_CFG_W9	1_2A64h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM5_MEM0_BLK_CFG_W10	1_2A68h
MBC1_DOM5_MEM0_BLK_CFG_W11	1_2A6Ch
MBC1_DOM5_MEM0_BLK_CFG_W12	1_2A70h
MBC1_DOM5_MEM0_BLK_CFG_W13	1_2A74h
MBC1_DOM5_MEM0_BLK_CFG_W14	1_2A78h
MBC1_DOM5_MEM0_BLK_CFG_W15	1_2A7Ch
MBC1_DOM5_MEM1_BLK_CFG_W0	1_2B80h
MBC1_DOM5_MEM2_BLK_CFG_W0	1_2BA8h
MBC1_DOM5_MEM3_BLK_CFG_W0	1_2BD0h
MBC1_DOM6_MEM0_BLK_CFG_W0	1_2C40h
MBC1_DOM6_MEM0_BLK_CFG_W1	1_2C44h
MBC1_DOM6_MEM0_BLK_CFG_W2	1_2C48h
MBC1_DOM6_MEM0_BLK_CFG_W3	1_2C4Ch
MBC1_DOM6_MEM0_BLK_CFG_W4	1_2C50h
MBC1_DOM6_MEM0_BLK_CFG_W5	1_2C54h
MBC1_DOM6_MEM0_BLK_CFG_W6	1_2C58h
MBC1_DOM6_MEM0_BLK_CFG_W7	1_2C5Ch
MBC1_DOM6_MEM0_BLK_CFG_W8	1_2C60h
MBC1_DOM6_MEM0_BLK_CFG_W9	1_2C64h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM6_MEM0_BLK_CFG_W10	1_2C68h
MBC1_DOM6_MEM0_BLK_CFG_W11	1_2C6Ch
MBC1_DOM6_MEM0_BLK_CFG_W12	1_2C70h
MBC1_DOM6_MEM0_BLK_CFG_W13	1_2C74h
MBC1_DOM6_MEM0_BLK_CFG_W14	1_2C78h
MBC1_DOM6_MEM0_BLK_CFG_W15	1_2C7Ch
MBC1_DOM6_MEM1_BLK_CFG_W0	1_2D80h
MBC1_DOM6_MEM2_BLK_CFG_W0	1_2DA8h
MBC1_DOM6_MEM3_BLK_CFG_W0	1_2DD0h
MBC1_DOM7_MEM0_BLK_CFG_W0	1_2E40h
MBC1_DOM7_MEM0_BLK_CFG_W1	1_2E44h
MBC1_DOM7_MEM0_BLK_CFG_W2	1_2E48h
MBC1_DOM7_MEM0_BLK_CFG_W3	1_2E4Ch
MBC1_DOM7_MEM0_BLK_CFG_W4	1_2E50h
MBC1_DOM7_MEM0_BLK_CFG_W5	1_2E54h
MBC1_DOM7_MEM0_BLK_CFG_W6	1_2E58h
MBC1_DOM7_MEM0_BLK_CFG_W7	1_2E5Ch
MBC1_DOM7_MEM0_BLK_CFG_W8	1_2E60h
MBC1_DOM7_MEM0_BLK_CFG_W9	1_2E64h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM7_MEM0_BLK_CFG_W10	1_2E68h
MBC1_DOM7_MEM0_BLK_CFG_W11	1_2E6Ch
MBC1_DOM7_MEM0_BLK_CFG_W12	1_2E70h
MBC1_DOM7_MEM0_BLK_CFG_W13	1_2E74h
MBC1_DOM7_MEM0_BLK_CFG_W14	1_2E78h
MBC1_DOM7_MEM0_BLK_CFG_W15	1_2E7Ch
MBC1_DOM7_MEM1_BLK_CFG_W0	1_2F80h
MBC1_DOM7_MEM2_BLK_CFG_W0	1_2FA8h
MBC1_DOM7_MEM3_BLK_CFG_W0	1_2FD0h
MBC1_DOM8_MEM0_BLK_CFG_W0	1_3040h
MBC1_DOM8_MEM0_BLK_CFG_W1	1_3044h
MBC1_DOM8_MEM0_BLK_CFG_W2	1_3048h
MBC1_DOM8_MEM0_BLK_CFG_W3	1_304Ch
MBC1_DOM8_MEM0_BLK_CFG_W4	1_3050h
MBC1_DOM8_MEM0_BLK_CFG_W5	1_3054h
MBC1_DOM8_MEM0_BLK_CFG_W6	1_3058h
MBC1_DOM8_MEM0_BLK_CFG_W7	1_305Ch
MBC1_DOM8_MEM0_BLK_CFG_W8	1_3060h
MBC1_DOM8_MEM0_BLK_CFG_W9	1_3064h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM8_MEM0_BLK_CFG_W10	1_3068h
MBC1_DOM8_MEM0_BLK_CFG_W11	1_306Ch
MBC1_DOM8_MEM0_BLK_CFG_W12	1_3070h
MBC1_DOM8_MEM0_BLK_CFG_W13	1_3074h
MBC1_DOM8_MEM0_BLK_CFG_W14	1_3078h
MBC1_DOM8_MEM0_BLK_CFG_W15	1_307Ch
MBC1_DOM8_MEM1_BLK_CFG_W0	1_3180h
MBC1_DOM8_MEM2_BLK_CFG_W0	1_31A8h
MBC1_DOM8_MEM3_BLK_CFG_W0	1_31D0h
MBC1_DOM9_MEM0_BLK_CFG_W0	1_3240h
MBC1_DOM9_MEM0_BLK_CFG_W1	1_3244h
MBC1_DOM9_MEM0_BLK_CFG_W2	1_3248h
MBC1_DOM9_MEM0_BLK_CFG_W3	1_324Ch
MBC1_DOM9_MEM0_BLK_CFG_W4	1_3250h
MBC1_DOM9_MEM0_BLK_CFG_W5	1_3254h
MBC1_DOM9_MEM0_BLK_CFG_W6	1_3258h
MBC1_DOM9_MEM0_BLK_CFG_W7	1_325Ch
MBC1_DOM9_MEM0_BLK_CFG_W8	1_3260h
MBC1_DOM9_MEM0_BLK_CFG_W9	1_3264h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM9_MEM0_BLK_CFG_W10	1_3268h
MBC1_DOM9_MEM0_BLK_CFG_W11	1_326Ch
MBC1_DOM9_MEM0_BLK_CFG_W12	1_3270h
MBC1_DOM9_MEM0_BLK_CFG_W13	1_3274h
MBC1_DOM9_MEM0_BLK_CFG_W14	1_3278h
MBC1_DOM9_MEM0_BLK_CFG_W15	1_327Ch
MBC1_DOM9_MEM1_BLK_CFG_W0	1_3380h
MBC1_DOM9_MEM2_BLK_CFG_W0	1_33A8h
MBC1_DOM9_MEM3_BLK_CFG_W0	1_33D0h
MBC1_DOM10_MEM0_BLK_CFG_W0	1_3440h
MBC1_DOM10_MEM0_BLK_CFG_W1	1_3444h
MBC1_DOM10_MEM0_BLK_CFG_W2	1_3448h
MBC1_DOM10_MEM0_BLK_CFG_W3	1_344Ch
MBC1_DOM10_MEM0_BLK_CFG_W4	1_3450h
MBC1_DOM10_MEM0_BLK_CFG_W5	1_3454h
MBC1_DOM10_MEM0_BLK_CFG_W6	1_3458h
MBC1_DOM10_MEM0_BLK_CFG_W7	1_345Ch
MBC1_DOM10_MEM0_BLK_CFG_W8	1_3460h
MBC1_DOM10_MEM0_BLK_CFG_W9	1_3464h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM10_MEM0_BLK_CFG_W10	1_3468h
MBC1_DOM10_MEM0_BLK_CFG_W11	1_346Ch
MBC1_DOM10_MEM0_BLK_CFG_W12	1_3470h
MBC1_DOM10_MEM0_BLK_CFG_W13	1_3474h
MBC1_DOM10_MEM0_BLK_CFG_W14	1_3478h
MBC1_DOM10_MEM0_BLK_CFG_W15	1_347Ch
MBC1_DOM10_MEM1_BLK_CFG_W0	1_3580h
MBC1_DOM10_MEM2_BLK_CFG_W0	1_35A8h
MBC1_DOM10_MEM3_BLK_CFG_W0	1_35D0h
MBC1_DOM11_MEM0_BLK_CFG_W0	1_3640h
MBC1_DOM11_MEM0_BLK_CFG_W1	1_3644h
MBC1_DOM11_MEM0_BLK_CFG_W2	1_3648h
MBC1_DOM11_MEM0_BLK_CFG_W3	1_364Ch
MBC1_DOM11_MEM0_BLK_CFG_W4	1_3650h
MBC1_DOM11_MEM0_BLK_CFG_W5	1_3654h
MBC1_DOM11_MEM0_BLK_CFG_W6	1_3658h
MBC1_DOM11_MEM0_BLK_CFG_W7	1_365Ch
MBC1_DOM11_MEM0_BLK_CFG_W8	1_3660h
MBC1_DOM11_MEM0_BLK_CFG_W9	1_3664h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM11_MEM0_BLK_CFG_W10	1_3668h
MBC1_DOM11_MEM0_BLK_CFG_W11	1_366Ch
MBC1_DOM11_MEM0_BLK_CFG_W12	1_3670h
MBC1_DOM11_MEM0_BLK_CFG_W13	1_3674h
MBC1_DOM11_MEM0_BLK_CFG_W14	1_3678h
MBC1_DOM11_MEM0_BLK_CFG_W15	1_367Ch
MBC1_DOM11_MEM1_BLK_CFG_W0	1_3780h
MBC1_DOM11_MEM2_BLK_CFG_W0	1_37A8h
MBC1_DOM11_MEM3_BLK_CFG_W0	1_37D0h
MBC1_DOM12_MEM0_BLK_CFG_W0	1_3840h
MBC1_DOM12_MEM0_BLK_CFG_W1	1_3844h
MBC1_DOM12_MEM0_BLK_CFG_W2	1_3848h
MBC1_DOM12_MEM0_BLK_CFG_W3	1_384Ch
MBC1_DOM12_MEM0_BLK_CFG_W4	1_3850h
MBC1_DOM12_MEM0_BLK_CFG_W5	1_3854h
MBC1_DOM12_MEM0_BLK_CFG_W6	1_3858h
MBC1_DOM12_MEM0_BLK_CFG_W7	1_385Ch
MBC1_DOM12_MEM0_BLK_CFG_W8	1_3860h
MBC1_DOM12_MEM0_BLK_CFG_W9	1_3864h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM12_MEM0_BLK_CFG_W10	1_3868h
MBC1_DOM12_MEM0_BLK_CFG_W11	1_386Ch
MBC1_DOM12_MEM0_BLK_CFG_W12	1_3870h
MBC1_DOM12_MEM0_BLK_CFG_W13	1_3874h
MBC1_DOM12_MEM0_BLK_CFG_W14	1_3878h
MBC1_DOM12_MEM0_BLK_CFG_W15	1_387Ch
MBC1_DOM12_MEM1_BLK_CFG_W0	1_3980h
MBC1_DOM12_MEM2_BLK_CFG_W0	1_39A8h
MBC1_DOM12_MEM3_BLK_CFG_W0	1_39D0h
MBC1_DOM13_MEM0_BLK_CFG_W0	1_3A40h
MBC1_DOM13_MEM0_BLK_CFG_W1	1_3A44h
MBC1_DOM13_MEM0_BLK_CFG_W2	1_3A48h
MBC1_DOM13_MEM0_BLK_CFG_W3	1_3A4Ch
MBC1_DOM13_MEM0_BLK_CFG_W4	1_3A50h
MBC1_DOM13_MEM0_BLK_CFG_W5	1_3A54h
MBC1_DOM13_MEM0_BLK_CFG_W6	1_3A58h
MBC1_DOM13_MEM0_BLK_CFG_W7	1_3A5Ch
MBC1_DOM13_MEM0_BLK_CFG_W8	1_3A60h
MBC1_DOM13_MEM0_BLK_CFG_W9	1_3A64h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM13_MEM0_BLK_CFG_W10	1_3A68h
MBC1_DOM13_MEM0_BLK_CFG_W11	1_3A6Ch
MBC1_DOM13_MEM0_BLK_CFG_W12	1_3A70h
MBC1_DOM13_MEM0_BLK_CFG_W13	1_3A74h
MBC1_DOM13_MEM0_BLK_CFG_W14	1_3A78h
MBC1_DOM13_MEM0_BLK_CFG_W15	1_3A7Ch
MBC1_DOM13_MEM1_BLK_CFG_W0	1_3B80h
MBC1_DOM13_MEM2_BLK_CFG_W0	1_3BA8h
MBC1_DOM13_MEM3_BLK_CFG_W0	1_3BD0h
MBC1_DOM14_MEM0_BLK_CFG_W0	1_3C40h
MBC1_DOM14_MEM0_BLK_CFG_W1	1_3C44h
MBC1_DOM14_MEM0_BLK_CFG_W2	1_3C48h
MBC1_DOM14_MEM0_BLK_CFG_W3	1_3C4Ch
MBC1_DOM14_MEM0_BLK_CFG_W4	1_3C50h
MBC1_DOM14_MEM0_BLK_CFG_W5	1_3C54h
MBC1_DOM14_MEM0_BLK_CFG_W6	1_3C58h
MBC1_DOM14_MEM0_BLK_CFG_W7	1_3C5Ch
MBC1_DOM14_MEM0_BLK_CFG_W8	1_3C60h
MBC1_DOM14_MEM0_BLK_CFG_W9	1_3C64h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM14_MEM0_BLK_CFG_W10	1_3C68h
MBC1_DOM14_MEM0_BLK_CFG_W11	1_3C6Ch
MBC1_DOM14_MEM0_BLK_CFG_W12	1_3C70h
MBC1_DOM14_MEM0_BLK_CFG_W13	1_3C74h
MBC1_DOM14_MEM0_BLK_CFG_W14	1_3C78h
MBC1_DOM14_MEM0_BLK_CFG_W15	1_3C7Ch
MBC1_DOM14_MEM1_BLK_CFG_W0	1_3D80h
MBC1_DOM14_MEM2_BLK_CFG_W0	1_3DA8h
MBC1_DOM14_MEM3_BLK_CFG_W0	1_3DD0h
MBC1_DOM15_MEM0_BLK_CFG_W0	1_3E40h
MBC1_DOM15_MEM0_BLK_CFG_W1	1_3E44h
MBC1_DOM15_MEM0_BLK_CFG_W2	1_3E48h
MBC1_DOM15_MEM0_BLK_CFG_W3	1_3E4Ch
MBC1_DOM15_MEM0_BLK_CFG_W4	1_3E50h
MBC1_DOM15_MEM0_BLK_CFG_W5	1_3E54h
MBC1_DOM15_MEM0_BLK_CFG_W6	1_3E58h
MBC1_DOM15_MEM0_BLK_CFG_W7	1_3E5Ch
MBC1_DOM15_MEM0_BLK_CFG_W8	1_3E60h
MBC1_DOM15_MEM0_BLK_CFG_W9	1_3E64h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM15_MEM0_BLK_CFG_W10	1_3E68h
MBC1_DOM15_MEM0_BLK_CFG_W11	1_3E6Ch
MBC1_DOM15_MEM0_BLK_CFG_W12	1_3E70h
MBC1_DOM15_MEM0_BLK_CFG_W13	1_3E74h
MBC1_DOM15_MEM0_BLK_CFG_W14	1_3E78h
MBC1_DOM15_MEM0_BLK_CFG_W15	1_3E7Ch
MBC1_DOM15_MEM1_BLK_CFG_W0	1_3F80h
MBC1_DOM15_MEM2_BLK_CFG_W0	1_3FA8h
MBC1_DOM15_MEM3_BLK_CFG_W0	1_3FD0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NSE7	MBACSEL7			NSE6	MBACSEL6			NSE5	MBACSEL5			NSE4	MBACSEL4		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NSE3	MBACSEL3			NSE2	MBACSEL2			NSE1	MBACSEL1			NSE0	MBACSEL0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 NSE7	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
30-28 MBACSEL7	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL7 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
27 NSE6	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
26-24 MBACSEL6	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL6 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
23	NonSecure Enable for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
NSE5	0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
22-20 MBACSEL5	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL5 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
19 NSE4	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
18-16 MBACSEL4	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL4 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
	111 - select MBC_MEMN_GLBAC7 access control policy for block B
15 NSE3	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
14-12 MBACSEL3	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL3 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
11 NSE2	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
10-8 MBACSEL2	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL2 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
7 NSE1	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
6-4 MBACSEL1	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL1 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B 011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B
3 NSE0	NonSecure Enable for block B 0 - Secure read and write to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
2-0 MBACSEL0	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL0 = r and MBC_MEMN_GLBACr[LK] = 1. 000 - select MBC_MEMN_GLBAC0 access control policy for block B 001 - select MBC_MEMN_GLBAC1 access control policy for block B 010 - select MBC_MEMN_GLBAC2 access control policy for block B

Table continues on the next page...

Table continued from the previous page...

Field	Description
	011 - select MBC_MEMN_GLBAC3 access control policy for block B 100 - select MBC_MEMN_GLBAC4 access control policy for block B 101 - select MBC_MEMN_GLBAC5 access control policy for block B 110 - select MBC_MEMN_GLBAC6 access control policy for block B 111 - select MBC_MEMN_GLBAC7 access control policy for block B

22.7.6.28 MBC Memory Block NonSecure Enable Word (MBC0_DOM0_MEM0_BLK_NSE_W0 - MBC1_DOM15_MEM3_BLK_NSE_W0)

MBC[m]_DOM[d]_MEM[s]_BLK_NSE_W[w], where,

- m - mbc index
- d - domain index
- s - memory target index
- w - word index

This is the bitmap of the individual NSE bits for the memory block.

NOTE

This register is shown with fields such that it covers entire 32 bit register. However, actual fields may be less if number of blocks are such that fields don't align to 32 bits. These absent fields may be shown but user can refer to table below to deduce actual fields and treat absent fields as reserved.

The following table describes the relationship between W (word index) and B (block number).

Table 117. Block Config Word to Block Number Relationship

Block NSE Word	BIT31	BIT30-BIT1	BIT0
W0	block 31	block 30 - block 1	block 0
W1	block 63	block 62 - block 33	block 32
For MEM0, block 64 - block 511			
W2	block 95	block 94 - block 65	block 64
.			
.			
.			
W15	block 511	block 510 - block 481	block 480

The NSE bitmap can be accessed directly by register reads and writes to these locations, or "indirectly" through writes to NSE_SET, NSE_CLR, NSE_CLR_ALL, and BLK_CFG[NSEn] registers.

Offset

Register	Offset
MBC0_DOM0_MEM0_BLK_NSE_W0	1_0140h
MBC0_DOM0_MEM0_BLK_NSE_W1	1_0144h
MBC0_DOM0_MEM0_BLK_NSE_W2	1_0148h
MBC0_DOM0_MEM0_BLK_NSE_W3	1_014Ch
MBC0_DOM0_MEM1_BLK_NSE_W0	1_01A0h
MBC0_DOM0_MEM2_BLK_NSE_W0	1_01C8h
MBC0_DOM0_MEM3_BLK_NSE_W0	1_01F0h
MBC0_DOM1_MEM0_BLK_NSE_W0	1_0340h
MBC0_DOM1_MEM0_BLK_NSE_W1	1_0344h
MBC0_DOM1_MEM0_BLK_NSE_W2	1_0348h
MBC0_DOM1_MEM0_BLK_NSE_W3	1_034Ch
MBC0_DOM1_MEM1_BLK_NSE_W0	1_03A0h
MBC0_DOM1_MEM2_BLK_NSE_W0	1_03C8h
MBC0_DOM1_MEM3_BLK_NSE_W0	1_03F0h
MBC0_DOM2_MEM0_BLK_NSE_W0	1_0540h
MBC0_DOM2_MEM0_BLK_NSE_W1	1_0544h
MBC0_DOM2_MEM0_BLK_NSE_W2	1_0548h
MBC0_DOM2_MEM0_BLK_NSE_W3	1_054Ch
MBC0_DOM2_MEM1_BLK_NSE_W0	1_05A0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM2_MEM2_BLK_NSE_W0	1_05C8h
MBC0_DOM2_MEM3_BLK_NSE_W0	1_05F0h
MBC0_DOM3_MEM0_BLK_NSE_W0	1_0740h
MBC0_DOM3_MEM0_BLK_NSE_W1	1_0744h
MBC0_DOM3_MEM0_BLK_NSE_W2	1_0748h
MBC0_DOM3_MEM0_BLK_NSE_W3	1_074Ch
MBC0_DOM3_MEM1_BLK_NSE_W0	1_07A0h
MBC0_DOM3_MEM2_BLK_NSE_W0	1_07C8h
MBC0_DOM3_MEM3_BLK_NSE_W0	1_07F0h
MBC0_DOM4_MEM0_BLK_NSE_W0	1_0940h
MBC0_DOM4_MEM0_BLK_NSE_W1	1_0944h
MBC0_DOM4_MEM0_BLK_NSE_W2	1_0948h
MBC0_DOM4_MEM0_BLK_NSE_W3	1_094Ch
MBC0_DOM4_MEM1_BLK_NSE_W0	1_09A0h
MBC0_DOM4_MEM2_BLK_NSE_W0	1_09C8h
MBC0_DOM4_MEM3_BLK_NSE_W0	1_09F0h
MBC0_DOM5_MEM0_BLK_NSE_W0	1_0B40h
MBC0_DOM5_MEM0_BLK_NSE_W1	1_0B44h
MBC0_DOM5_MEM0_BLK_NSE_W2	1_0B48h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM5_MEM0_BLK_NSE_W3	1_0B4Ch
MBC0_DOM5_MEM1_BLK_NSE_W0	1_0BA0h
MBC0_DOM5_MEM2_BLK_NSE_W0	1_0BC8h
MBC0_DOM5_MEM3_BLK_NSE_W0	1_0BF0h
MBC0_DOM6_MEM0_BLK_NSE_W0	1_0D40h
MBC0_DOM6_MEM0_BLK_NSE_W1	1_0D44h
MBC0_DOM6_MEM0_BLK_NSE_W2	1_0D48h
MBC0_DOM6_MEM0_BLK_NSE_W3	1_0D4Ch
MBC0_DOM6_MEM1_BLK_NSE_W0	1_0DA0h
MBC0_DOM6_MEM2_BLK_NSE_W0	1_0DC8h
MBC0_DOM6_MEM3_BLK_NSE_W0	1_0DF0h
MBC0_DOM7_MEM0_BLK_NSE_W0	1_0F40h
MBC0_DOM7_MEM0_BLK_NSE_W1	1_0F44h
MBC0_DOM7_MEM0_BLK_NSE_W2	1_0F48h
MBC0_DOM7_MEM0_BLK_NSE_W3	1_0F4Ch
MBC0_DOM7_MEM1_BLK_NSE_W0	1_0FA0h
MBC0_DOM7_MEM2_BLK_NSE_W0	1_0FC8h
MBC0_DOM7_MEM3_BLK_NSE_W0	1_0FF0h
MBC0_DOM8_MEM0_BLK_NSE_W0	1_1140h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM8_MEM0_BLK_NSE_W1	1_1144h
MBC0_DOM8_MEM0_BLK_NSE_W2	1_1148h
MBC0_DOM8_MEM0_BLK_NSE_W3	1_114Ch
MBC0_DOM8_MEM1_BLK_NSE_W0	1_11A0h
MBC0_DOM8_MEM2_BLK_NSE_W0	1_11C8h
MBC0_DOM8_MEM3_BLK_NSE_W0	1_11F0h
MBC0_DOM9_MEM0_BLK_NSE_W0	1_1340h
MBC0_DOM9_MEM0_BLK_NSE_W1	1_1344h
MBC0_DOM9_MEM0_BLK_NSE_W2	1_1348h
MBC0_DOM9_MEM0_BLK_NSE_W3	1_134Ch
MBC0_DOM9_MEM1_BLK_NSE_W0	1_13A0h
MBC0_DOM9_MEM2_BLK_NSE_W0	1_13C8h
MBC0_DOM9_MEM3_BLK_NSE_W0	1_13F0h
MBC0_DOM10_MEM0_BLK_NSE_W0	1_1540h
MBC0_DOM10_MEM0_BLK_NSE_W1	1_1544h
MBC0_DOM10_MEM0_BLK_NSE_W2	1_1548h
MBC0_DOM10_MEM0_BLK_NSE_W3	1_154Ch
MBC0_DOM10_MEM1_BLK_NSE_W0	1_15A0h
MBC0_DOM10_MEM2_BLK_NSE_W0	1_15C8h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM10_MEM3_BLK_NSE_W0	1_15F0h
MBC0_DOM11_MEM0_BLK_NSE_W0	1_1740h
MBC0_DOM11_MEM0_BLK_NSE_W1	1_1744h
MBC0_DOM11_MEM0_BLK_NSE_W2	1_1748h
MBC0_DOM11_MEM0_BLK_NSE_W3	1_174Ch
MBC0_DOM11_MEM1_BLK_NSE_W0	1_17A0h
MBC0_DOM11_MEM2_BLK_NSE_W0	1_17C8h
MBC0_DOM11_MEM3_BLK_NSE_W0	1_17F0h
MBC0_DOM12_MEM0_BLK_NSE_W0	1_1940h
MBC0_DOM12_MEM0_BLK_NSE_W1	1_1944h
MBC0_DOM12_MEM0_BLK_NSE_W2	1_1948h
MBC0_DOM12_MEM0_BLK_NSE_W3	1_194Ch
MBC0_DOM12_MEM1_BLK_NSE_W0	1_19A0h
MBC0_DOM12_MEM2_BLK_NSE_W0	1_19C8h
MBC0_DOM12_MEM3_BLK_NSE_W0	1_19F0h
MBC0_DOM13_MEM0_BLK_NSE_W0	1_1B40h
MBC0_DOM13_MEM0_BLK_NSE_W1	1_1B44h
MBC0_DOM13_MEM0_BLK_NSE_W2	1_1B48h
MBC0_DOM13_MEM0_BLK_NSE_W3	1_1B4Ch

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC0_DOM13_MEM1_BLK_NSE_W0	1_1BA0h
MBC0_DOM13_MEM2_BLK_NSE_W0	1_1BC8h
MBC0_DOM13_MEM3_BLK_NSE_W0	1_1BF0h
MBC0_DOM14_MEM0_BLK_NSE_W0	1_1D40h
MBC0_DOM14_MEM0_BLK_NSE_W1	1_1D44h
MBC0_DOM14_MEM0_BLK_NSE_W2	1_1D48h
MBC0_DOM14_MEM0_BLK_NSE_W3	1_1D4Ch
MBC0_DOM14_MEM1_BLK_NSE_W0	1_1DA0h
MBC0_DOM14_MEM2_BLK_NSE_W0	1_1DC8h
MBC0_DOM14_MEM3_BLK_NSE_W0	1_1DF0h
MBC0_DOM15_MEM0_BLK_NSE_W0	1_1F40h
MBC0_DOM15_MEM0_BLK_NSE_W1	1_1F44h
MBC0_DOM15_MEM0_BLK_NSE_W2	1_1F48h
MBC0_DOM15_MEM0_BLK_NSE_W3	1_1F4Ch
MBC0_DOM15_MEM1_BLK_NSE_W0	1_1FA0h
MBC0_DOM15_MEM2_BLK_NSE_W0	1_1FC8h
MBC0_DOM15_MEM3_BLK_NSE_W0	1_1FF0h
MBC1_DOM0_MEM0_BLK_NSE_W0	1_2140h
MBC1_DOM0_MEM0_BLK_NSE_W1	1_2144h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM0_MEM0_BLK_NSE_W2	1_2148h
MBC1_DOM0_MEM0_BLK_NSE_W3	1_214Ch
MBC1_DOM0_MEM1_BLK_NSE_W0	1_21A0h
MBC1_DOM0_MEM2_BLK_NSE_W0	1_21C8h
MBC1_DOM0_MEM3_BLK_NSE_W0	1_21F0h
MBC1_DOM1_MEM0_BLK_NSE_W0	1_2340h
MBC1_DOM1_MEM0_BLK_NSE_W1	1_2344h
MBC1_DOM1_MEM0_BLK_NSE_W2	1_2348h
MBC1_DOM1_MEM0_BLK_NSE_W3	1_234Ch
MBC1_DOM1_MEM1_BLK_NSE_W0	1_23A0h
MBC1_DOM1_MEM2_BLK_NSE_W0	1_23C8h
MBC1_DOM1_MEM3_BLK_NSE_W0	1_23F0h
MBC1_DOM2_MEM0_BLK_NSE_W0	1_2540h
MBC1_DOM2_MEM0_BLK_NSE_W1	1_2544h
MBC1_DOM2_MEM0_BLK_NSE_W2	1_2548h
MBC1_DOM2_MEM0_BLK_NSE_W3	1_254Ch
MBC1_DOM2_MEM1_BLK_NSE_W0	1_25A0h
MBC1_DOM2_MEM2_BLK_NSE_W0	1_25C8h
MBC1_DOM2_MEM3_BLK_NSE_W0	1_25F0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM3_MEM0_BLK_NSE_W0	1_2740h
MBC1_DOM3_MEM0_BLK_NSE_W1	1_2744h
MBC1_DOM3_MEM0_BLK_NSE_W2	1_2748h
MBC1_DOM3_MEM0_BLK_NSE_W3	1_274Ch
MBC1_DOM3_MEM1_BLK_NSE_W0	1_27A0h
MBC1_DOM3_MEM2_BLK_NSE_W0	1_27C8h
MBC1_DOM3_MEM3_BLK_NSE_W0	1_27F0h
MBC1_DOM4_MEM0_BLK_NSE_W0	1_2940h
MBC1_DOM4_MEM0_BLK_NSE_W1	1_2944h
MBC1_DOM4_MEM0_BLK_NSE_W2	1_2948h
MBC1_DOM4_MEM0_BLK_NSE_W3	1_294Ch
MBC1_DOM4_MEM1_BLK_NSE_W0	1_29A0h
MBC1_DOM4_MEM2_BLK_NSE_W0	1_29C8h
MBC1_DOM4_MEM3_BLK_NSE_W0	1_29F0h
MBC1_DOM5_MEM0_BLK_NSE_W0	1_2B40h
MBC1_DOM5_MEM0_BLK_NSE_W1	1_2B44h
MBC1_DOM5_MEM0_BLK_NSE_W2	1_2B48h
MBC1_DOM5_MEM0_BLK_NSE_W3	1_2B4Ch
MBC1_DOM5_MEM1_BLK_NSE_W0	1_2BA0h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM5_MEM2_BLK_NSE_W0	1_2BC8h
MBC1_DOM5_MEM3_BLK_NSE_W0	1_2BF0h
MBC1_DOM6_MEM0_BLK_NSE_W0	1_2D40h
MBC1_DOM6_MEM0_BLK_NSE_W1	1_2D44h
MBC1_DOM6_MEM0_BLK_NSE_W2	1_2D48h
MBC1_DOM6_MEM0_BLK_NSE_W3	1_2D4Ch
MBC1_DOM6_MEM1_BLK_NSE_W0	1_2DA0h
MBC1_DOM6_MEM2_BLK_NSE_W0	1_2DC8h
MBC1_DOM6_MEM3_BLK_NSE_W0	1_2DF0h
MBC1_DOM7_MEM0_BLK_NSE_W0	1_2F40h
MBC1_DOM7_MEM0_BLK_NSE_W1	1_2F44h
MBC1_DOM7_MEM0_BLK_NSE_W2	1_2F48h
MBC1_DOM7_MEM0_BLK_NSE_W3	1_2F4Ch
MBC1_DOM7_MEM1_BLK_NSE_W0	1_2FA0h
MBC1_DOM7_MEM2_BLK_NSE_W0	1_2FC8h
MBC1_DOM7_MEM3_BLK_NSE_W0	1_2FF0h
MBC1_DOM8_MEM0_BLK_NSE_W0	1_3140h
MBC1_DOM8_MEM0_BLK_NSE_W1	1_3144h
MBC1_DOM8_MEM0_BLK_NSE_W2	1_3148h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM8_MEM0_BLK_NSE_W3	1_314Ch
MBC1_DOM8_MEM1_BLK_NSE_W0	1_31A0h
MBC1_DOM8_MEM2_BLK_NSE_W0	1_31C8h
MBC1_DOM8_MEM3_BLK_NSE_W0	1_31F0h
MBC1_DOM9_MEM0_BLK_NSE_W0	1_3340h
MBC1_DOM9_MEM0_BLK_NSE_W1	1_3344h
MBC1_DOM9_MEM0_BLK_NSE_W2	1_3348h
MBC1_DOM9_MEM0_BLK_NSE_W3	1_334Ch
MBC1_DOM9_MEM1_BLK_NSE_W0	1_33A0h
MBC1_DOM9_MEM2_BLK_NSE_W0	1_33C8h
MBC1_DOM9_MEM3_BLK_NSE_W0	1_33F0h
MBC1_DOM10_MEM0_BLK_NSE_W0	1_3540h
MBC1_DOM10_MEM0_BLK_NSE_W1	1_3544h
MBC1_DOM10_MEM0_BLK_NSE_W2	1_3548h
MBC1_DOM10_MEM0_BLK_NSE_W3	1_354Ch
MBC1_DOM10_MEM1_BLK_NSE_W0	1_35A0h
MBC1_DOM10_MEM2_BLK_NSE_W0	1_35C8h
MBC1_DOM10_MEM3_BLK_NSE_W0	1_35F0h
MBC1_DOM11_MEM0_BLK_NSE_W0	1_3740h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM11_MEM0_BLK_NSE_W1	1_3744h
MBC1_DOM11_MEM0_BLK_NSE_W2	1_3748h
MBC1_DOM11_MEM0_BLK_NSE_W3	1_374Ch
MBC1_DOM11_MEM1_BLK_NSE_W0	1_37A0h
MBC1_DOM11_MEM2_BLK_NSE_W0	1_37C8h
MBC1_DOM11_MEM3_BLK_NSE_W0	1_37F0h
MBC1_DOM12_MEM0_BLK_NSE_W0	1_3940h
MBC1_DOM12_MEM0_BLK_NSE_W1	1_3944h
MBC1_DOM12_MEM0_BLK_NSE_W2	1_3948h
MBC1_DOM12_MEM0_BLK_NSE_W3	1_394Ch
MBC1_DOM12_MEM1_BLK_NSE_W0	1_39A0h
MBC1_DOM12_MEM2_BLK_NSE_W0	1_39C8h
MBC1_DOM12_MEM3_BLK_NSE_W0	1_39F0h
MBC1_DOM13_MEM0_BLK_NSE_W0	1_3B40h
MBC1_DOM13_MEM0_BLK_NSE_W1	1_3B44h
MBC1_DOM13_MEM0_BLK_NSE_W2	1_3B48h
MBC1_DOM13_MEM0_BLK_NSE_W3	1_3B4Ch
MBC1_DOM13_MEM1_BLK_NSE_W0	1_3BA0h
MBC1_DOM13_MEM2_BLK_NSE_W0	1_3BC8h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
MBC1_DOM13_MEM3_BLK_NSE_W0	1_3BF0h
MBC1_DOM14_MEM0_BLK_NSE_W0	1_3D40h
MBC1_DOM14_MEM0_BLK_NSE_W1	1_3D44h
MBC1_DOM14_MEM0_BLK_NSE_W2	1_3D48h
MBC1_DOM14_MEM0_BLK_NSE_W3	1_3D4Ch
MBC1_DOM14_MEM1_BLK_NSE_W0	1_3DA0h
MBC1_DOM14_MEM2_BLK_NSE_W0	1_3DC8h
MBC1_DOM14_MEM3_BLK_NSE_W0	1_3DF0h
MBC1_DOM15_MEM0_BLK_NSE_W0	1_3F40h
MBC1_DOM15_MEM0_BLK_NSE_W1	1_3F44h
MBC1_DOM15_MEM0_BLK_NSE_W2	1_3F48h
MBC1_DOM15_MEM0_BLK_NSE_W3	1_3F4Ch
MBC1_DOM15_MEM1_BLK_NSE_W0	1_3FA0h
MBC1_DOM15_MEM2_BLK_NSE_W0	1_3FC8h
MBC1_DOM15_MEM3_BLK_NSE_W0	1_3FF0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BIT31	BIT30	BIT29	BIT28	BIT27	BIT26	BIT25	BIT24	BIT23	BIT22	BIT21	BIT20	BIT19	BIT18	BIT17	BIT16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BITb	Bit b NonSecure Enable [b = 0 - 31] 0 - Secure read and write to block B are based on corresponding MBACSEL field in register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), secure execution and nonsecure accesses to block B are not allowed. 1 - Secure accesses to block B are not allowed, nonsecure accesses to block B are based on corresponding MBACSEL field in register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).

22.7.6.29 MBC Global Access Control (MBC1_MEMN_GLBAC0)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User.

Offset

Register	Offset
MBC1_MEMN_GLBAC0	1_2020h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SPR	SPW	SPX	0	SUR	SUW	SUX	0	NPR	NPW	NPX	0	NUR	NUW	NUX
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6	NonsecurePriv Read

Table continues on the next page...

Table continued from the previous page...

Field	Description
NPR	NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

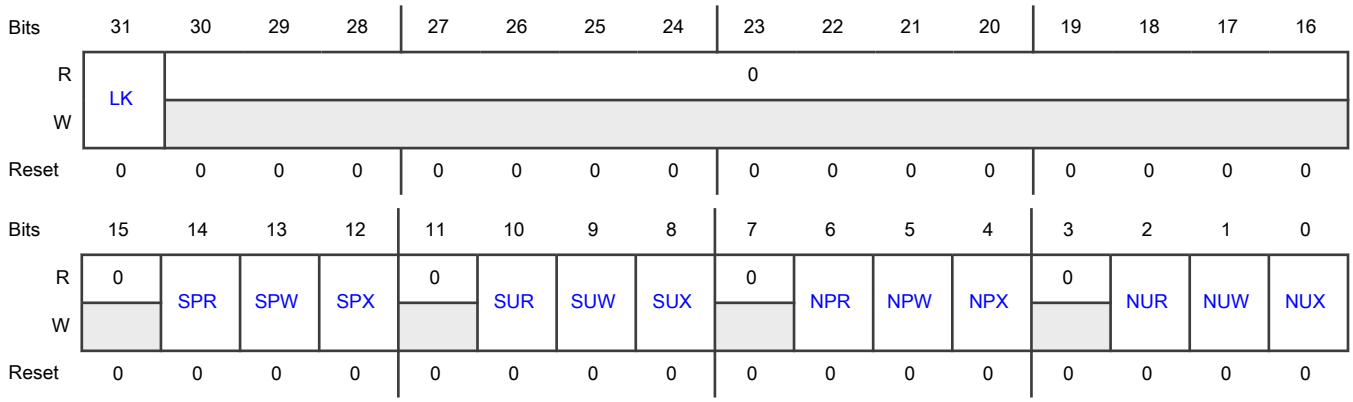
22.7.6.30 MBC Global Access Control (MBC1_MEMN_GLBAC1 - MBC1_MEMN_GLBAC7)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

Offset

Register	Offset
MBC1_MEMN_GLBAC1	1_2024h
MBC1_MEMN_GLBAC2	1_2028h
MBC1_MEMN_GLBAC3	1_202Ch
MBC1_MEMN_GLBAC4	1_2030h
MBC1_MEMN_GLBAC5	1_2034h
MBC1_MEMN_GLBAC6	1_2038h
MBC1_MEMN_GLBAC7	1_203Ch

Diagram



Fields

Field	Description
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0 - This register is not locked and can be altered. 1 - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13	SecurePriv Write

Table continues on the next page...

Table continued from the previous page...

Field	Description
SPW	SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

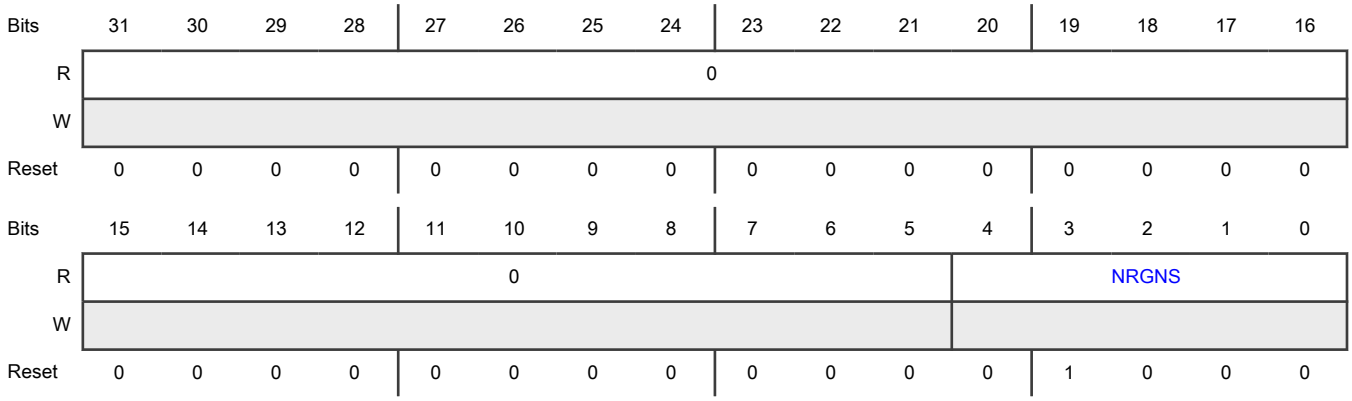
22.7.6.31 MRC Global Configuration Register (MRC0_GLBCFG - MRC1_GLBCFG)

This read-only register describes the number of region descriptors for this memory.

Offset

Register	Offset
MRC0_GLBCFG	1_4000h
MRC1_GLBCFG	1_5000h

Diagram



Fields

Field	Description
31-5 —	Reserved
4-0 NRGNS	Number of regions [1-16]

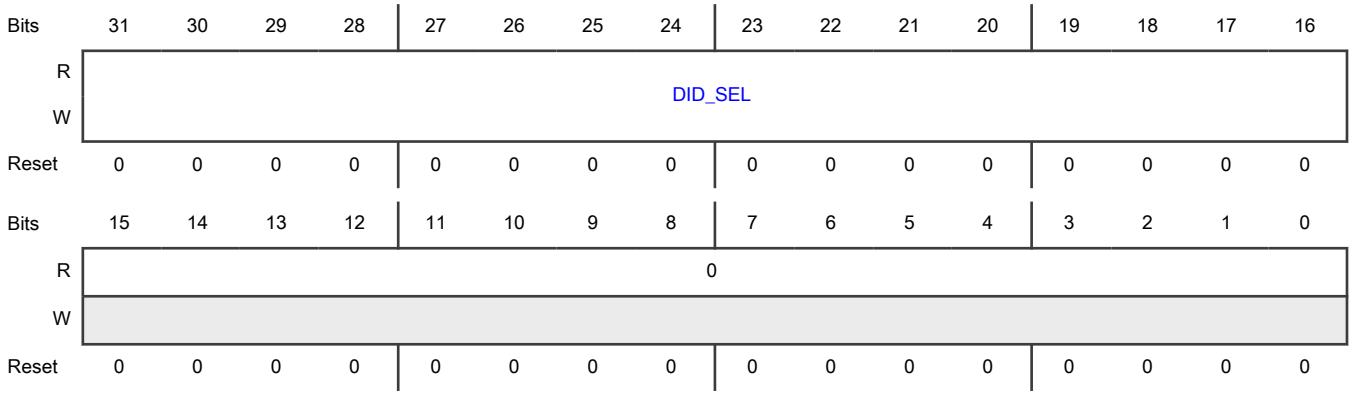
22.7.6.32 MRC NonSecure Enable Region Indirect (MRC0_NSE_RGN_INDIRECT - MRC1_NSE_RGN_INDIRECT)

This R/W register defines the selected domains that are affected by writes to the NSE_RGN_SET and NSE_RGN_CLR registers.

Offset

Register	Offset
MRC0_NSE_RGN_INDIRECT	1_4010h
MRC1_NSE_RGN_INDIRECT	1_5010h

Diagram



Fields

Field	Description
31-16 DID_SEL	DID Select Destination domain bitmap select. • Bit [31] = 1, update domain 15 • Bit [30] = 1, update domain 14 • Bit [29] = 1, update domain 13 • Bit [28] = 1, update domain 12 • Bit [27] = 1, update domain 11 • Bit [26] = 1, update domain 10 • Bit [25] = 1, update domain 9 • Bit [24] = 1, update domain 8 • Bit [23] = 1, update domain 7 • Bit [22] = 1, update domain 6 • Bit [21] = 1, update domain 5 • Bit [20] = 1, update domain 4 • Bit [19] = 1, update domain 3 • Bit [18] = 1, update domain 2 • Bit [17] = 1, update domain 1 • Bit [16] = 1, update domain 0
15-0 —	Reserved

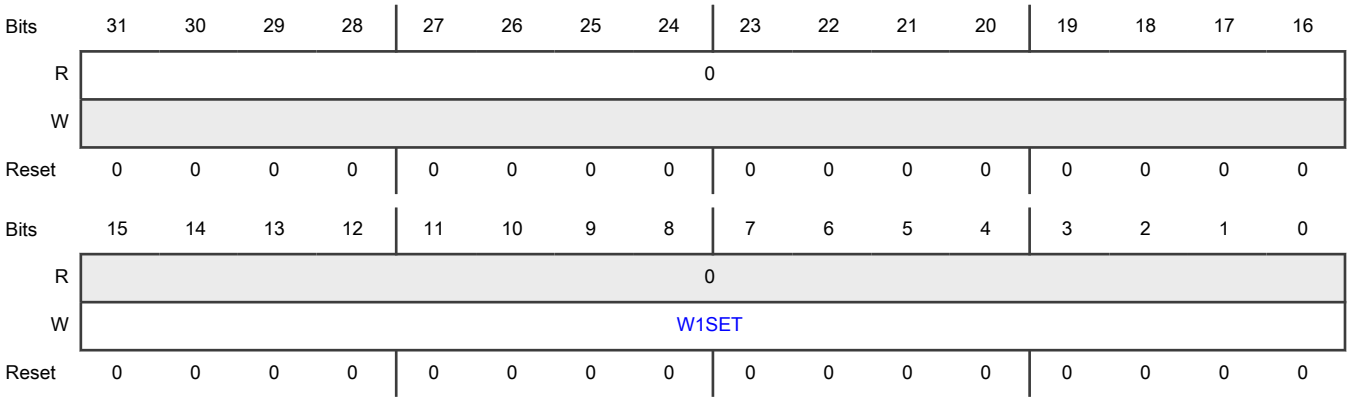
22.7.6.33 MRC NonSecure Enable Region Set (MRC0_NSE_RGN_SET - MRC1_NSE_RGN_SET)

A write to this location sets the appropriate NSE bits in region descriptors for the selected domain in MRC NonSecure Enable Region Indirect register. This register reads as 0.

Offset

Register	Offset
MRC0_NSE_RGN_SET	1_4014h
MRC1_NSE_RGN_SET	1_5014h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 W1SET	Write-1 Set • Bit [15] = Set the NSE bits for RGD15 • Bit [14] = Set the NSE bits for RGD14 • Bit [13] = Set the NSE bits for RGD13 • Bit [12] = Set the NSE bits for RGD12 • Bit [11] = Set the NSE bits for RGD11 • Bit [10] = Set the NSE bits for RGD10 • Bit [9] = Set the NSE bits for RGD9 • Bit [8] = Set the NSE bits for RGD8 • Bit [7] = Set the NSE bits for RGD7 • Bit [6] = Set the NSE bits for RGD6 • Bit [5] = Set the NSE bits for RGD5 • Bit [4] = Set the NSE bits for RGD4 • Bit [3] = Set the NSE bits for RGD3 • Bit [2] = Set the NSE bits for RGD2

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Bit [1] = Set the NSE bits for RGD1 • Bit [0] = Set the NSE bits for RGD0

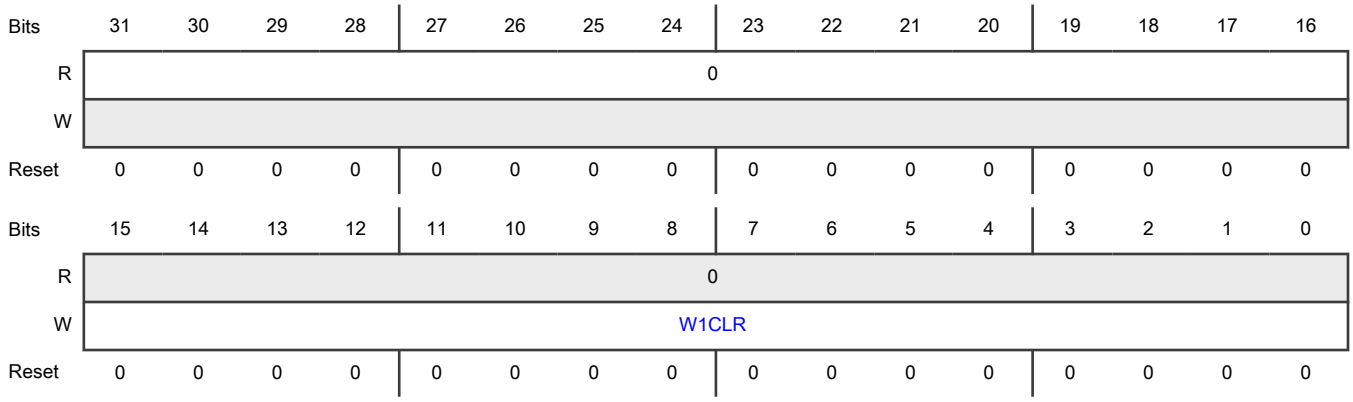
22.7.6.34 MRC NonSecure Enable Region Clear (MRC0_NSE_RGN_CLR - MRC1_NSE_RGN_CLR)

A write to this location clears the appropriate NSE bits in region descriptors for the selected domain in MRC NonSecure Enable Region Indirect register. This register reads as 0.

Offset

Register	Offset
MRC0_NSE_RGN_CLR	1_4018h
MRC1_NSE_RGN_CLR	1_5018h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 W1CLR	Write-1 Clear • Bit [15] = Clear the NSE bits for RGD15 • Bit [14] = Clear the NSE bits for RGD14 • Bit [13] = Clear the NSE bits for RGD13 • Bit [12] = Clear the NSE bits for RGD12

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Bit [11] = Clear the NSE bits for RGD11 • Bit [10] = Clear the NSE bits for RGD10 • Bit [9] = Clear the NSE bits for RGD9 • Bit [8] = Clear the NSE bits for RGD8 • Bit [7] = Clear the NSE bits for RGD7 • Bit [6] = Clear the NSE bits for RGD6 • Bit [5] = Clear the NSE bits for RGD5 • Bit [4] = Clear the NSE bits for RGD4 • Bit [3] = Clear the NSE bits for RGD3 • Bit [2] = Clear the NSE bits for RGD2 • Bit [1] = Clear the NSE bits for RGD1 • Bit [0] = Clear the NSE bits for RGD0

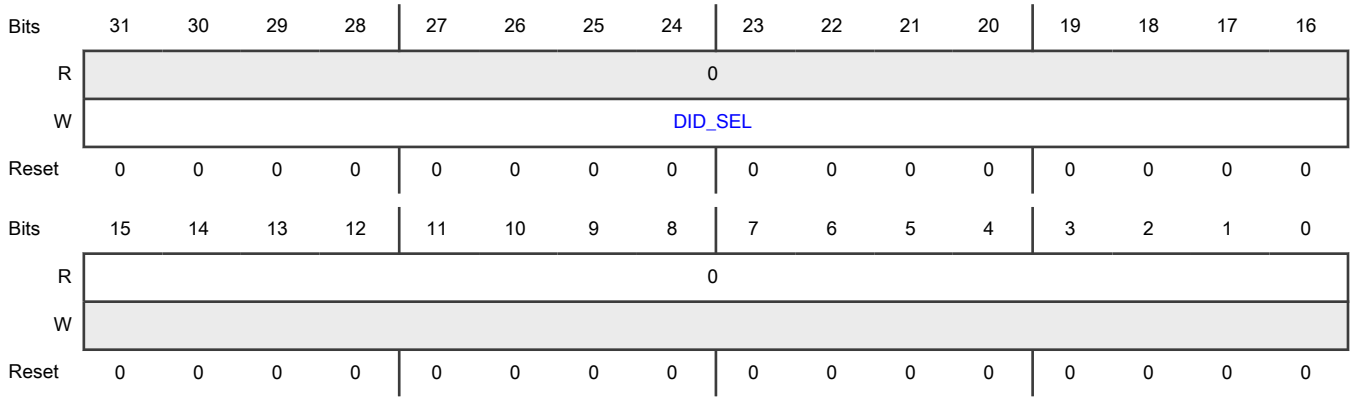
22.7.6.35 MRC NonSecure Enable Region Clear All (MRC0_NSE_RGN_CLR_ALL - MRC1_NSE_RGN_CLR_ALL)

A write to this register clears all the region NSE bits for the selected domains defined in the write data.

Offset

Register	Offset
MRC0_NSE_RGN_CLR_ALL	1_401Ch
MRC1_NSE_RGN_CLR_ALL	1_501Ch

Diagram



Fields

Field	Description
31-16 DID_SEL	DID Select Destination domain bitmap select. • Bit [31] = 1 clear NSE bits for domain 15 • Bit [30] = 1 clear NSE bits for domain 14 • Bit [29] = 1 clear NSE bits for domain 13 • Bit [28] = 1 clear NSE bits for domain 12 • Bit [27] = 1 clear NSE bits for domain 11 • Bit [26] = 1 clear NSE bits for domain 10 • Bit [25] = 1 clear NSE bits for domain 9 • Bit [24] = 1 clear NSE bits for domain 8 • Bit [23] = 1 clear NSE bits for domain 7 • Bit [22] = 1 clear NSE bits for domain 6 • Bit [21] = 1 clear NSE bits for domain 5 • Bit [20] = 1 clear NSE bits for domain 4 • Bit [19] = 1 clear NSE bits for domain 3 • Bit [18] = 1 clear NSE bits for domain 2 • Bit [17] = 1 clear NSE bits for domain 1 • Bit [16] = 1 clear NSE bits for domain 0
15-0 —	Reserved

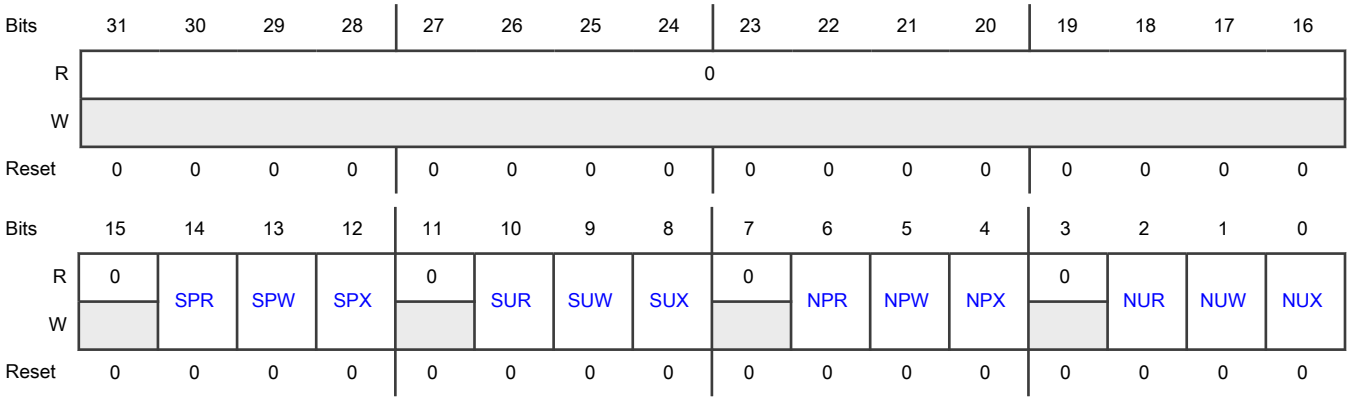
22.7.6.36 MRC Global Access Control (MRC0_GLBAC0)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, NonsecureUser. The lock bit provides the ability to lock these control access flags, to lock the MRC_BLK_CFG[MRACSEL] field, and to lock the MRC_RGD[STRT_ADDR, END_ADDR, VLD] fields when a locked MRC_GLBAC register is selected.

Offset

Register	Offset
MRC0_GLBAC0	1_4020h

Diagram



Fields

Field	Description
31-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9	SecureUser Write

Table continues on the next page...

Table continued from the previous page...

Field	Description
SUW	SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

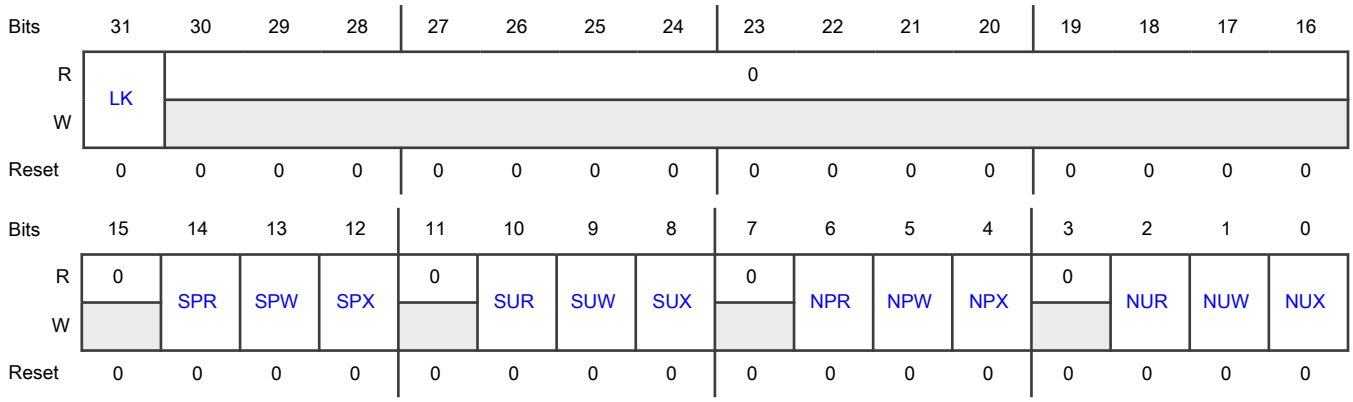
22.7.6.37 MRC Global Access Control (MRC0_GLBAC1 - MRC0_GLBAC7)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, NonsecureUser. The lock bit provides the ability to lock these control access flags, to lock the MRC_BLK_CFG[MRACSEL] field, and to lock the MRC_RGD[STRT_ADDR, END_ADDR, VLD] fields when a locked MRC_GLBAC register is selected.

Offset

Register	Offset
MRC0_GLBAC1	1_4024h
MRC0_GLBAC2	1_4028h
MRC0_GLBAC3	1_402Ch
MRC0_GLBAC4	1_4030h
MRC0_GLBAC5	1_4034h
MRC0_GLBAC6	1_4038h
MRC0_GLBAC7	1_403Ch

Diagram



Fields

Field	Description
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0 - This register is not locked and can be altered. 1 - This register is locked (read-only) and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

22.7.6.38 MRC Region Descriptor Word 0 (MRC0_DOM0_RGD0_W0 - MRC1_DOM15_RGD7_W0)

MRC[m]_DOM[d]_RGD[r]_W[w], where,

- m - mrc index
- d - domain index
- r - region descriptor index
- w - word index

The MRC[m]_DOM[d]_RGD[r]_W[w] registers provide a 3-dimensional data structure for defining access control policies per domain for each supported memory region.

Memory map/register definition:

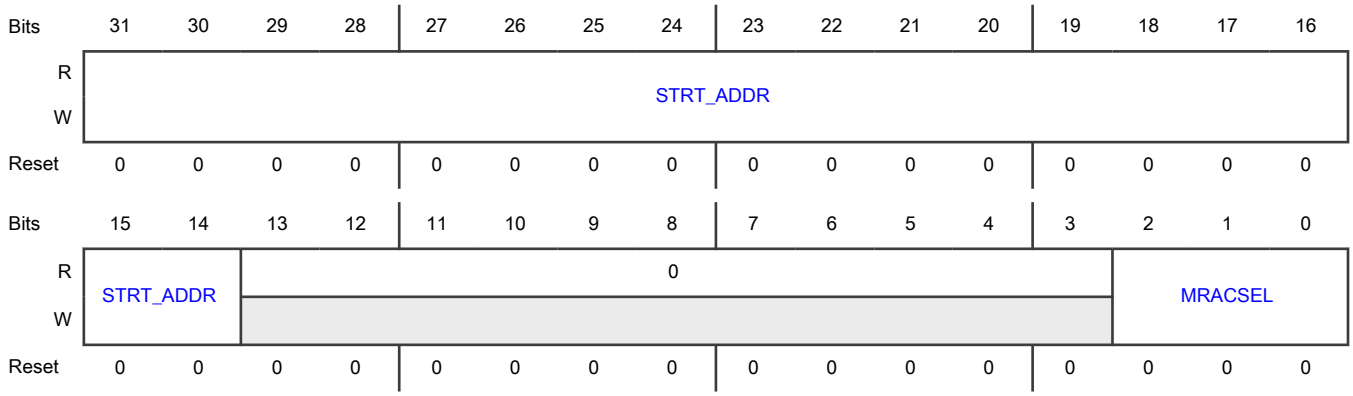
There two word-size registers (W[w]) defined in each memory region. There are up to 16 memory region checkers (MRCs), m, each supporting up to 16 memory region descriptors (RGD), r. The number of MRCs is defined by HWCFG0[NMRC].

Offset

For m = 0 to 1; d = 0 to 15; r = 0 to 7:

Register	Offset
MRCm_DOMd_RGDr_W0	1_4040h + (m × 1000h) + (d × 100h) + (r × 8h)

Diagram



Fields

Field	Description
31-14 STRT_ADDR	Start Address 0-mod-16K physical start address [31:14] of region r. **NOTE** This field is locked (read-only) if MRACSEL = r and MRC_GLBACr[LK] = 1.
13-3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
2-0 MRACSEL	Memory Region Access Control Select This field selects the global access control associated with region r. **NOTE** This field is locked (read-only) if MRACSEL = r and MRC_GLBACr[LK] = 1. 000 - Select MRC_GLBAC0 access control policy 001 - Select MRC_GLBAC1 access control policy 010 - Select MRC_GLBAC2 access control policy 011 - Select MRC_GLBAC3 access control policy 100 - Select MRC_GLBAC4 access control policy 101 - Select MRC_GLBAC5 access control policy 110 - Select MRC_GLBAC6 access control policy 111 - Select MRC_GLBAC7 access control policy

22.7.6.39 MRC Region Descriptor Word 1 (MRC0_DOM0_RGD0_W1 - MRC1_DOM15_RGD7_W1)

MRC[m]_DOM[d]_RGD[r]_W[w], where,

- m - mrc index
- d - domain index
- r - region descriptor index
- w - word index

The MRC[m]_[d]_rgd[r]_w[w] registers provide a 3-dimensional data structure for defining access control policies per domain for each supported memory region.

Memory map/register definition:

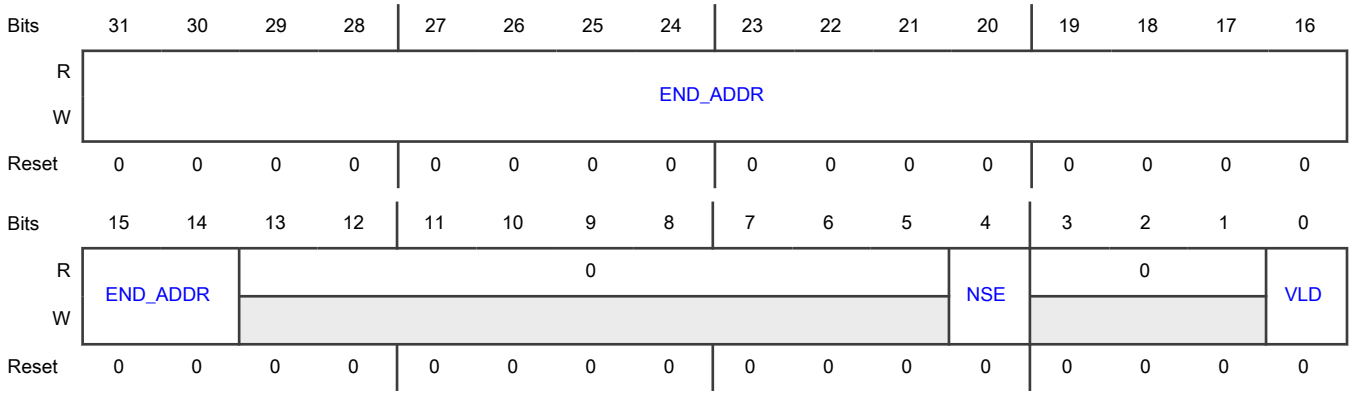
There two word-size registers (W[w]) defined in each memory region. There are up to 16 memory region checkers (MRCs), m, each supporting up to 16 region descriptors (RGD), r. The number of MRCs is defined by HWCFG0[NMRC].

Offset

For m = 0 to 1; d = 0 to 15; r = 0 to 7:

Register	Offset
MRCm_DOMd_RGDr_W1	1_4044h + (m × 1000h) + (d × 100h) + (r × 8h)

Diagram



Fields

Field	Description
31-14 END_ADDR	End Address (16K-1)-mod-16K physical end address [31:14] of region r. **NOTE** This field is locked (read-only) if MRACSEL = r and MRC_GLBACr[LK] = 1.
13-5 —	Reserved
4 NSE	NonSecure Enable Alternate view of NSE flag for Region r. 0 - Secure accesses to region r are based on corresponding MRACSEL field in this register (MRCm_DOMd_RGDr_Ww[MRACSEL]), nonsecure accesses to region r are not allowed. 1 - Secure accesses to region r are not allowed, nonsecure accesses to region r are based on corresponding MRACSEL field in this register (MRCm_DOMd_RGDr_Ww[MRACSEL]).
3-1 —	Reserved
0 VLD	Valid If set, this region is valid and accesses are checked for this region. **NOTE** This field is locked (read-only) if MRACSEL = r and MRC_GLBACr[LK] = 1.

22.7.6.40 MRC Region Descriptor NonSecure Enable (MRC0_DOM0_RGD_NSE - MRC1_DOM15_RGD_NSE)

This register is the bitmap of the individual NSE bits for the region descriptors. Bit n corresponds to region n.

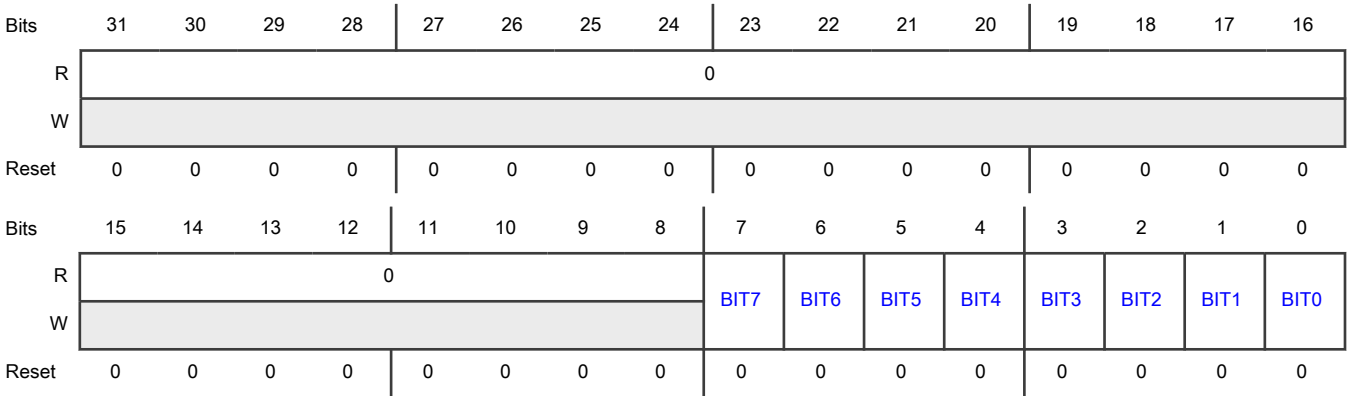
These bits can also be written or "indirectly" through writes to NSE_SET, NSE_CLR, NSE_CLR_ALL, and RGD_W0[NSE] registers.

Offset

For m = 0 to 1; d = 0 to 15:

Register	Offset
MRCm_DOMd_RGD_NSE	1_40C0h + (m × 1000h) + (d × 100h)

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 BITn	Bit n NonSecure Enable [n = 0 - 15] 0 - Secure accesses to region r are based on corresponding MRACSEL field in this register (MRCm_DOMd_RGDr_Ww[MRACSEL]), nonsecure accesses to region r are not allowed. 1 - Secure accesses to region r are not allowed, nonsecure accesses to region r are based on corresponding MRACSEL field in register (MRCm_DOMd_RGDr_Ww[MRACSEL]).

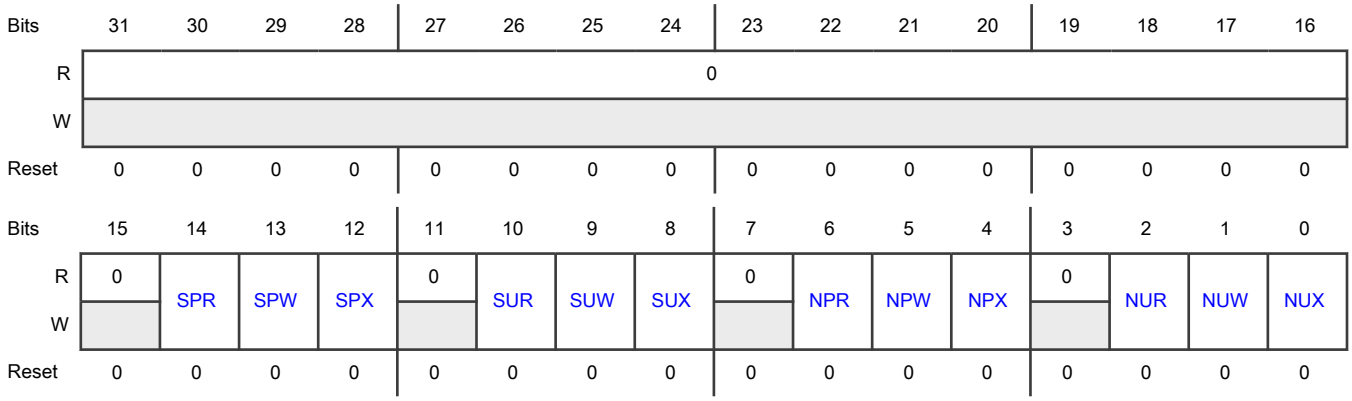
22.7.6.41 MRC Global Access Control (MRC1_GLBAC0)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, NonsecureUser. The lock bit provides the ability to lock these control access flags, to lock the MRC_BLK_CFG[MRACSEL] field, and to lock the MRC_RGD[STRT_ADDR, END_ADDR, VLD] fields when a locked MRC_GLBAC register is selected.

Offset

Register	Offset
MRC1_GLBAC0	1_5020h

Diagram



Fields

Field	Description
31-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9	SecureUser Write

Table continues on the next page...

Table continued from the previous page...

Field	Description
SUW	SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

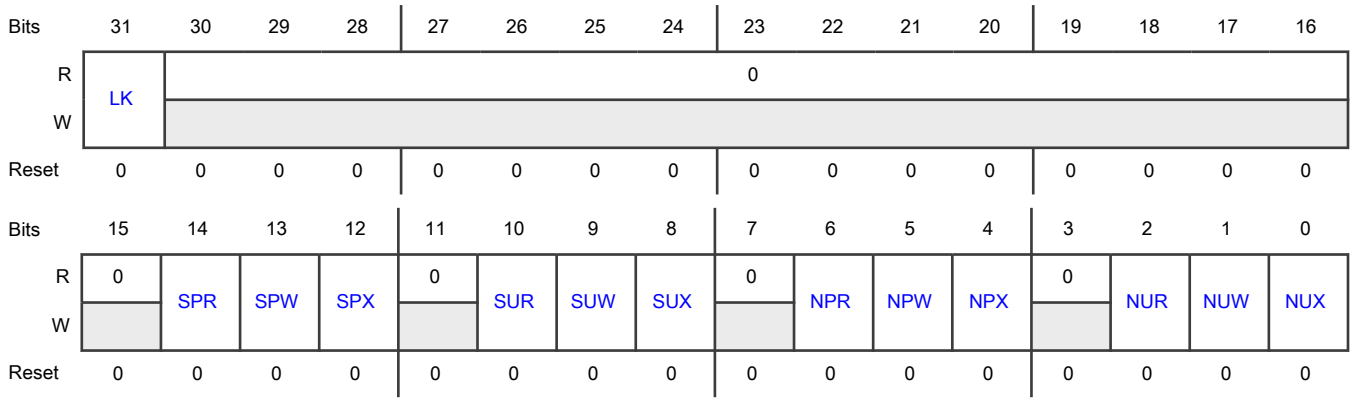
22.7.6.42 MRC Global Access Control (MRC1_GLBAC1 - MRC1_GLBAC7)

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, NonsecureUser. The lock bit provides the ability to lock these control access flags, to lock the MRC_BLK_CFG[MRACSEL] field, and to lock the MRC_RGD[STRT_ADDR, END_ADDR, VLD] fields when a locked MRC_GLBAC register is selected.

Offset

Register	Offset
MRC1_GLBAC1	1_5024h
MRC1_GLBAC2	1_5028h
MRC1_GLBAC3	1_502Ch
MRC1_GLBAC4	1_5030h
MRC1_GLBAC5	1_5034h
MRC1_GLBAC6	1_5038h
MRC1_GLBAC7	1_503Ch

Diagram



Fields

Field	Description
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0 - This register is not locked and can be altered. 1 - This register is locked (read-only) and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0 - Read access is not allowed in Secure Privilege mode. 1 - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0 - Write access is not allowed in Secure Privilege mode. 1 - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0 - Execute access is not allowed in Secure Privilege mode. 1 - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0 - Read access is not allowed in Secure User mode. 1 - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0 - Write access is not allowed in Secure User mode. 1 - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Execute access is not allowed in Secure User mode. 1 - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0 - Read access is not allowed in Nonsecure Privilege mode. 1 - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0 - Write access is not allowed in Nonsecure Privilege mode. 1 - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0 - Execute access is not allowed in Nonsecure Privilege mode. 1 - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0 - Read access is not allowed in Nonsecure User mode. 1 - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0 - Write access is not allowed in Nonsecure User mode. 1 - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0 - Execute access is not allowed in Nonsecure User mode. 1 - Execute access is allowed in Nonsecure User mode.

Chapter 23
System Counter (SYS_CTR)

23.1 Chip-specific SYS_CTR information

Table 118. Reference links to related information

Topic	Related module	Reference
Full description	Arm Cortex CM33 Platform (CM33)	CM33 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

NOTE

Software can read A55 core's generic timer counter value via A55 CP15 to get the SYS_CTR value.

23.1.1 Module instances

This chip has one instance of the SYS_CTR module.

23.2 Overview

SYS_CTR is a programmable system counter that provides you with a shared time base to multiple processors. It is intended for applications in which the counter is always powered on and supports multiple unrelated clocks.

23.2.1 Block diagram

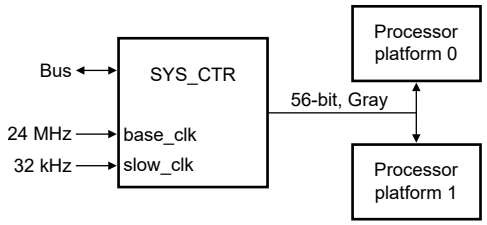


Figure 44. Block diagram

23.2.2 Features

- Two counter clock frequencies:
  - Base clock for normal operation
  - Alternate slow clock for low-power operation
- 56-bit counter width
- Gray-coded counter output for distribution to the processor timers
- Two compare frames, with each frame:

— Containing a 64-bit compare value

— Supporting a programmable interrupt generation

23.3 Functional description

SYS_CTR inputs two counter clock sources and outputs a gray-coded counter value and interrupt signals (one per compare frame) to the platform's interrupt controller.

23.3.1 Modes of operation

Table 119. Modes of operation

Mode	Function
Normal	The counter increments on the rising edge of the selected clock when enabled in Normal mode.
Debug	You can configure the SYS_CTR debug operation via Counter Control (CNTCR): • If CNTCR[HDBG] = 0, SYS_CTR continues to operate. • If CNTCR[HDBG] = 1, SYS_CTR halts when its debug input signal is asserted.

23.3.2 Operation

After reset, SYS_CTR is disabled with the count value reset to 0 and base frequency selected. After the counter is enabled, it increments the appropriate value on each rising edge of the selected clock. Because SYS_CTR handles a 56-bit count value across multiple clock domains, synchronization is necessary. SYS_CTR provides synchronization mechanisms between the various clock domains.

After the system switches the counter's clock source, there is a short pause while the clock multiplexer is handling the clock transition. To maintain an accurate count value, the clock control logic employs two offset counters: one for base-to-slow transition and the other one for slow-to-base transition. These offset counters only operate during the clock transition time to compensate for the idled source clock. Both counters run off the base clock. The transition offset values are added to the system count value at an appropriate time after the counter's clock is restored.

NOTE

Both the base clock and alternate clock must be running when changing frequencies.

23.3.3 Clocks

Table 120. Clocks

Clock	Description
base_clk	Base clock This clock is used during normal operation.
slow_clk	Slow clock This clock is used during low-power mode. It is internally divided by 64 before use.

23.3.4 Interrupts

Each SYS_CTR compare frame (SYS_CTR_COMPARE) is capable of generating one maskable interrupt.

23.4 External signals

This module has no external signals.

23.5 Initialization

To initialize SYS_CTR:

1. Write an initial value to Counter Count Value Low (CNTCV0) and Counter Count Value High (CNTCV1), if a different value is desired.
2. Write 1 to CNTCR[EN] to enable the counter.

23.6 Application information

To use a compare frame:

1. Write the desired count value into Compare Count Value Low (CMPCVL0 - CMPCVL1) and Compare Count Value High (CMPCVH0 - CMPCVH1).
2. Enable interrupts using CMPCR0[IMASK], if required.
3. Write 1 to CMPCR0[EN] to enable the compare function.
4. Read CMPCR0[ISTAT] to check whether the count value is reached.

23.7 Programming model

SYS_CTR contains three programming model sections. Each section has an independent module enable to allow individual secure access control. See the following table for details.

Table 121. Programming model

Programming model	Description
Control	Controls the counter frequency, enable, debug, and count value. This model also contains the frequency-modes table that supplies control and status for the frequency of SYS_CTR.
Read	Allows you to read the count value.
Compare	Controls the compare value frequency, enable, interrupt mask, and interrupt status for each compare frame.

23.8 Memory map and register definition

23.8.1 SYS_CTR_CONTROL register descriptions

NOTE

Write or read access to reserved locations does not generate a transfer error.

NOTE

All 32-bit registers allow only 32-bit wide write operations, and are aligned to 32 bits.

23.8.1.1 SYS_CTR_CONTROL memory map

SYS_CTR_CONTROL base address: 4429_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Counter Control (CNTCR)	32	RW	0000_0000
4	Counter Status (CNTSR)	32	R	0000_0100
8	Counter Count Value Low (CNTCV0)	32	RW	0000_0000
C	Counter Count Value High (CNTCV1)	32	RW	0000_0000
20	Frequency-Modes Table 0 (CNTFID0)	32	R	016E_3600
24	Frequency-Modes Table 1 (CNTFID1)	32	R	0000_0200
28	Frequency-Modes Table 2 (CNTFID2)	32	R	0000_0000
C0	Counter Control 2 (CNTCR2)	32	RW	0000_0000
FD0	Counter ID (CNTID0)	32	R	0000_0000

23.8.1.2 Counter Control (CNTCR)

Defines the basic operating configuration of SYS_CTR, which operates using a fixed base frequency. However, the counter can increment at a lower alternate frequency than the base frequency, using a correspondingly larger increment. For example, the counter can increment by 46,875 and run at a frequency of 1/46875 of the base frequency, which is the normal operating frequency. The alternate frequency is a significantly lower operating frequency used to reduce power consumption. The frequency-modes table identifies the two available frequencies. These two frequencies are the base frequency (table entry 0) and the lower, alternate frequency (table entry 1). Writing 1 to CNTCR[FCR1] selects the alternate frequency, and writing 1 to CNTCR[FCR0] selects the base frequency. Writing 1 or writing 0 to both FCR0 and FCR1 has no effect and the frequency does not change.

Offset

Register	Offset
CNTCR	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						FCR1	FCR0	Reserved						HDBG	EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-10 —	Reserved
9 FCR1	Frequency Change Request, ID 1 Selects frequency-modes table entry 1. 0 - No change 1 - Alternate frequency
8 FCR0	Frequency Change Request, ID 0 Selects frequency-modes table entry 0. 0 - No change 1 - Base frequency
7-2 —	Reserved
1 HDBG	Enable Debug Halt Specifies whether the assertion of the debug input is ignored. 0 - Ignored 1 - Causes SYS_CTR to halt
0 EN	Enable Counting Enables counting. 0 - Disable 1 - Enable

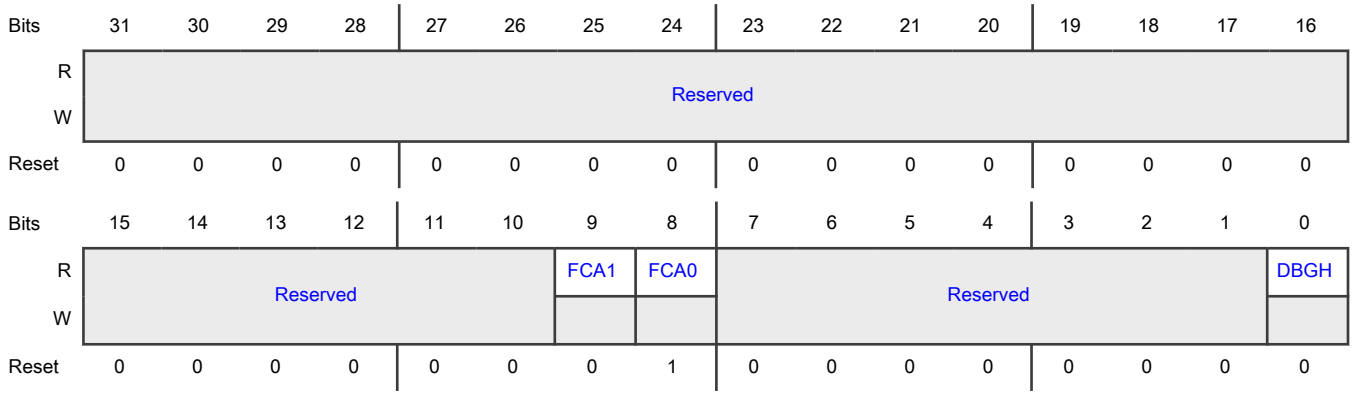
23.8.1.3 Counter Status (CNTSR)

Provides information concerning the clock frequency and debug state of SYS_CTR.

Offset

Register	Offset
CNTSR	4h

Diagram



Fields

Field	Description
31-10 —	Reserved
9 FCA1	Frequency Change Acknowledge, ID 1 Indicates whether the alternate frequency, entry 1, is selected. 0 - Not selected 1 - Selected
8 FCA0	Frequency Change Acknowledge, ID 0 Indicates whether the base frequency, entry 0, is selected. 0 - Not selected 1 - Selected
7-1 —	Reserved
0 DBGH	Debug Halt Indicates whether debug halted the counter. 0 - Did not halt 1 - Halted

23.8.1.4 Counter Count Value Low (CNTCV0)

Specifies the current count value bits (31–0).

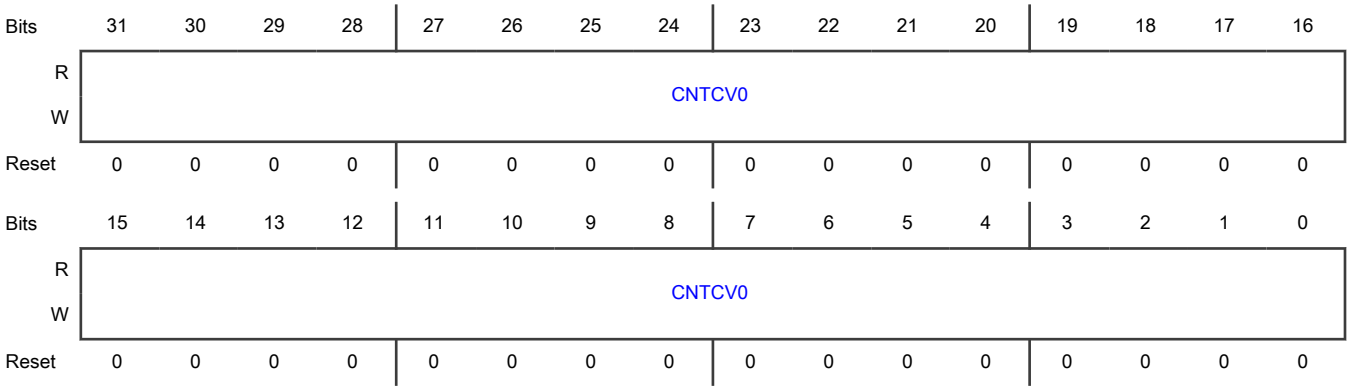
NOTE

You must write to the CNTCV registers when operating on the base frequency only. Writes to these registers when running on the alternate frequency may have unpredictable results.

Offset

Register	Offset
CNTCV0	8h

Diagram



Fields

Field	Description
31-0 CNTCV0	Counter Count Value Bits [31:0] Specifies the lower 32 bits of the 56-bit system count value.

23.8.1.5 Counter Count Value High (CNTCV1)

Specifies the current count value bits (63–32).

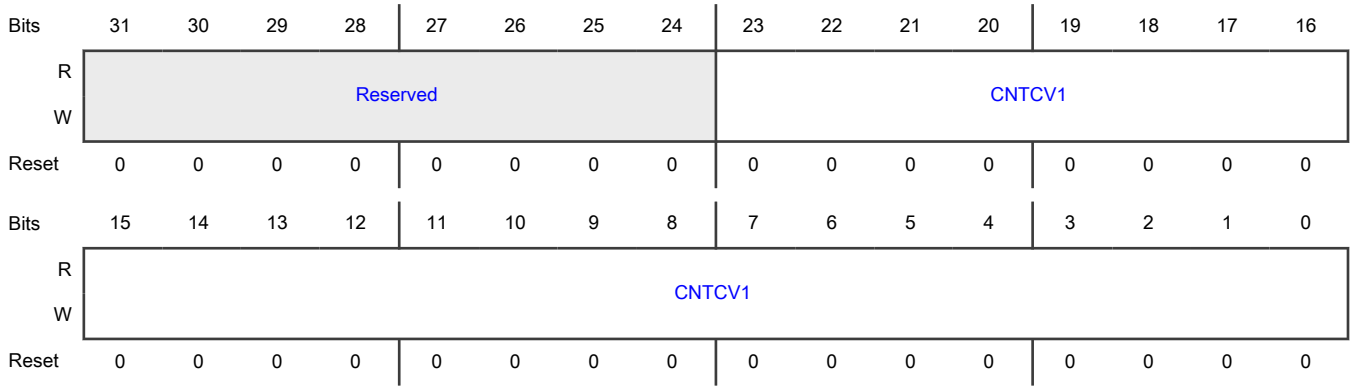
NOTE

You must write to the CNTCV registers when operating on the base frequency only. Writes to these registers when running on the alternate frequency may have unpredictable results.

Offset

Register	Offset
CNTCV1	Ch

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 CNTCV1	Counter Count Value Bits [55:32] Specifies the upper 24 bits of the 56-bit system count value.

23.8.1.6 Frequency-Modes Table 0 (CNTFID0)

Indicates the frequency-modes table starting at offset 20h.

Table entries contain 32 bits, and each entry specifies the SYS_CTR update frequency in Hz. The first entry in the table specifies the base frequency of SYS_CTR. To ensure that the overall counter accuracy is maintained, any subsequent entries in the table are exact divisors of the base frequency.

When SYS_CTR is operating at a lower frequency than the base frequency, the increment applied at each counter update is base_frequency ÷ selected_frequency; a 32-bit word of 0 value marks the end of the table. That is, the word of memory immediately after the last entry in the table is 0.

Offset

Register	Offset
CNTFID0	20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CNTFID0															
W																
Reset	0	0	0	0	0	0	0	1	0	1	1	0	1	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNTFID0															
W																
Reset	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CNTFID0	Counter Frequency ID 0 Indicates the base frequency, 24 MHz.

23.8.1.7 Frequency-Modes Table 1 (CNTFID1)

Indicates the frequency-modes table starting at offset 20h.

Table entries contain 32 bits, and each entry specifies the SYS_CTR update frequency in Hz. The first entry in the table specifies the base frequency of SYS_CTR. To ensure that the overall counter accuracy is maintained, any subsequent entries in the table are exact divisors of the base frequency.

When SYS_CTR is operating at a lower frequency than the base frequency, the increment applied at each counter update is base frequency ÷ selected frequency; a 32-bit word of 0 value marks the end of the table. That is, the word of memory immediately after the last entry in the table is 0.

Offset

Register	Offset
CNTFID1	24h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CNTFID1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNTFID1															
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CNTFID1	Counter Frequency ID 1 Indicates the alternate frequency, 512 Hz.

23.8.1.8 Frequency-Modes Table 2 (CNTFID2)

Indicates the frequency-modes table starting at offset 20h.

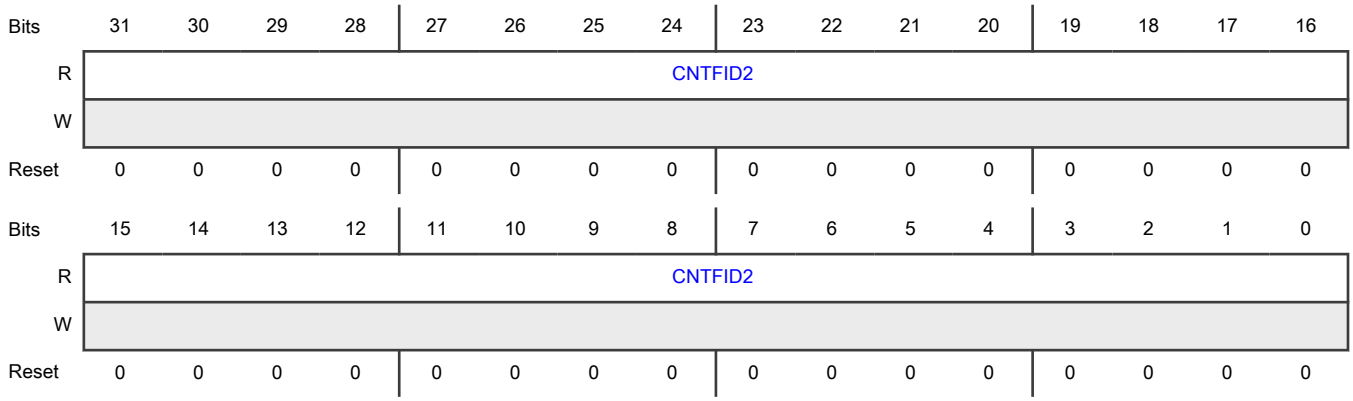
Table entries are 32 bits, and each entry specifies the SYS_CTR update frequency in Hz. The first entry in the table specifies the base frequency of SYS_CTR. To ensure that the overall counter accuracy is maintained, any subsequent entries in the table are exact divisors of the base frequency.

When SYS_CTR is operating at a lower frequency than the base frequency, the increment applied at each counter update is base frequency ÷ selected frequency; a 32-bit word of 0 value marks the end of the table. That is, the word of memory immediately after the last entry in the table is 0.

Offset

Register	Offset
CNTFID2	28h

Diagram



Fields

Field	Description
31-0 CNTFID2	Counter Frequency ID 2 Indicates the end of the frequency-modes table.

23.8.1.9 Counter Control 2 (CNTCR2)

Enables automatic configuration of SYS_CTR via hardware.

SYS_CTR operates using a fixed base frequency. However, the counter can increment at a lower alternate frequency than the base frequency, using a correspondingly larger increment. For example, the counter can increment by 46,875 and run

at a frequency of 1/46875 of the base frequency, which is the normal operating frequency. The alternate frequency is a significantly lower operating frequency used to reduce power consumption. The frequency-modes table identifies the two available frequencies. These two frequencies are the base frequency (table entry 0), and the lower and alternate frequency (table entry 1). Writing 1 to CNTCR2[HWFC_EN] enables hardware to automatically switch to the lower, alternate frequency clock after the chip enters low-power mode. When exiting this mode, hardware automatically switches back to the base frequency. If CNTCR2[HWFC_EN] = 0, frequency changes are fully under software control via Counter Control (CNTCR).

Offset

Register	Offset
CNTCR2	C0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															HWFC_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-1 —	Reserved
0 HWFC_EN	Hardware Frequency Change Enable Enables automatic configuration of SYS_CTR via hardware. If this field is 1, it performs the same frequency change request that you can do using CNTCR[FCR0] and CNTCR[FCR1]. 0 - No effect 1 - Same as performed via software

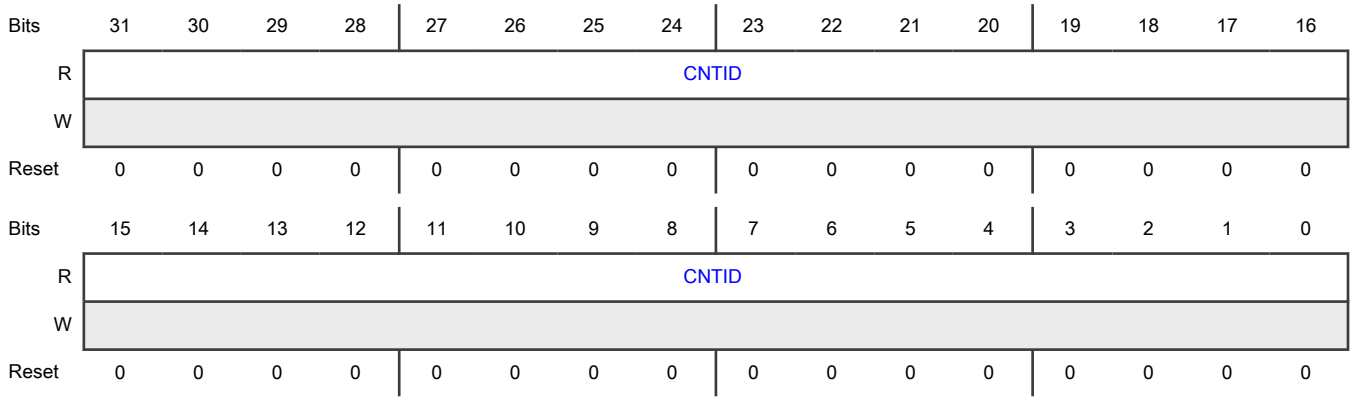
23.8.1.10 Counter ID (CNTID0)

Indicates the architectural version 0 of SYS_CTR.

Offset

Register	Offset
CNTID0	FD0h

Diagram



Fields

Field	Description
31-0 CNTID	Counter Identification Specifies counter ID 0, the architectural version of this system counter.

23.8.2 SYS_CTR_READ register descriptions

These read frame registers read the same values as the control frame registers for the count value and counter ID. They are processed via a separate mechanism from the control frame to allow Nonsecure and User mode accesses.

NOTE

Write or read access to reserved locations does not generate a transfer error.

NOTE

All 32-bit registers allow only 32-bit wide write operations, and are aligned to 32 bits.

23.8.2.1 SYS_CTR_READ memory map

SYS_CTR_READ base address: 442B_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8	Counter Count Value Low (CNTCV0)	32	R	0000_0000
C	Counter Count Value High (CNTCV1)	32	R	0000_0000
FD0	Counter ID (CNTID0)	32	R	0000_0000

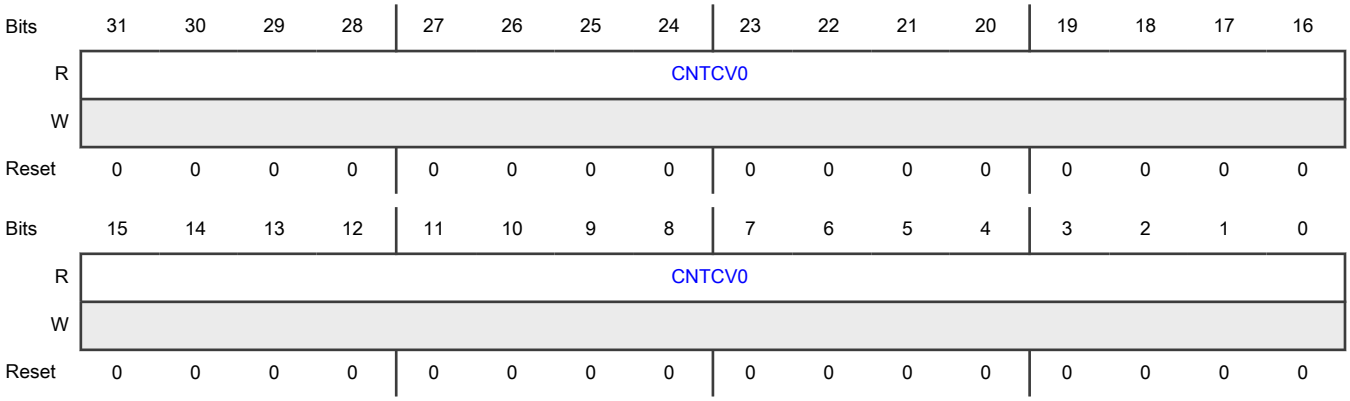
23.8.2.2 Counter Count Value Low (CNTCV0)

Indicates the current count value bits, 31–0.

Offset

Register	Offset
CNTCV0	8h

Diagram



Fields

Field	Description
31-0 CNTCV0	Counter Count Value Bits [31:0] Indicates the lower 32 bits of the 56-bit system count value.

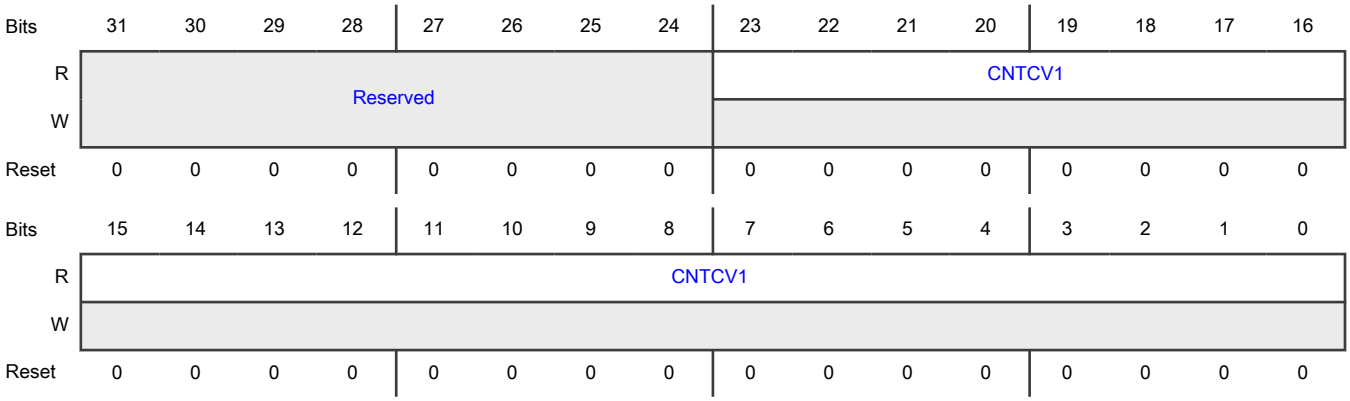
23.8.2.3 Counter Count Value High (CNTCV1)

Indicates the current count value bits 63–32.

Offset

Register	Offset
CNTCV1	Ch

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 CNTCV1	Counter Count Value Bits [55:32] Indicates the upper 24 bits of the 56-bit system count value. Bits 63–56 are always zero.

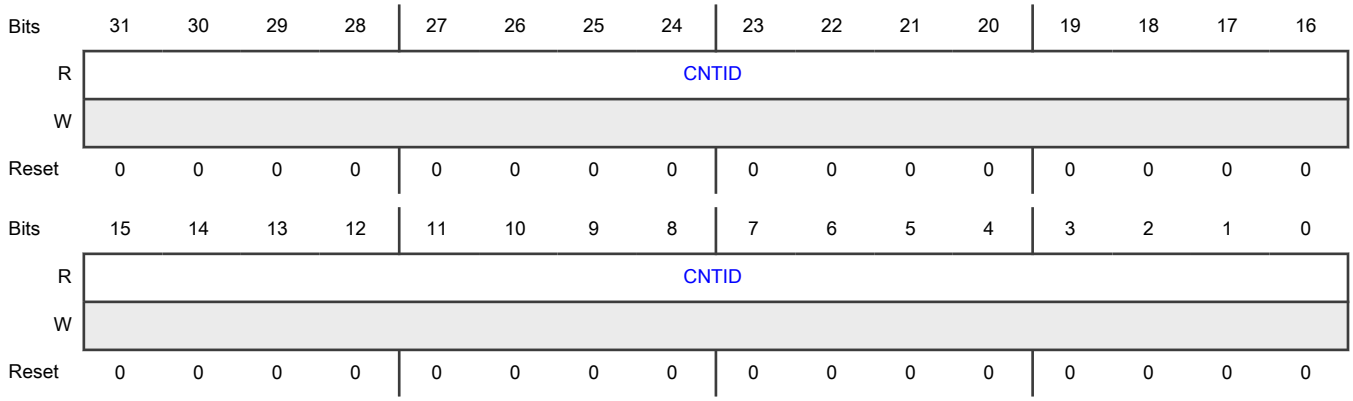
23.8.2.4 Counter ID (CNTID0)

Indicates the architectural version 0 of SYS_CTR.

Offset

Register	Offset
CNTID0	FD0h

Diagram



Fields

Field	Description
31-0 CNTID	Counter Identification Indicates counter ID 0, the architectural version of this system counter.

23.8.3 SYS_CTR_COMPARE register descriptions

Each compare frame:

- Consists of a 256-byte region.
- Has its own compare value and control register.
- Is capable of generating a maskable interrupt.

NOTE

Write or read access to reserved locations does not generate a transfer error.

NOTE

All 32-bit registers allow only 32-bit wide write operations, and are aligned to 32 bits.

23.8.3.1 SYS_CTR_COMPARE memory map

SYS_CTR_COMPARE base address: 442A_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
20	Compare Count Value Low (CMPCVL0)	32	RW	0000_0000
24	Compare Count Value High (CMPCVH0)	32	RW	0000_0000
2C	Compare Control (CMPCR0)	32	RW	0000_0000
120	Compare Count Value Low (CMPCVL1)	32	RW	0000_0000
124	Compare Count Value High (CMPCVH1)	32	RW	0000_0000
12C	Compare Control (CMPCR1)	32	RW	0000_0000
FD0	Counter ID (CNTID0)	32	R	0000_0000

23.8.3.2 Compare Count Value Low (CMPCVL0 - CMPCVL1)

Indicates the current count value bits 31–0.

NOTE

You must write to this set of registers when operating on the base frequency only. Writes to these registers when running on the alternate frequency may have unpredictable results.

Offset

Register	Offset
CMPCVL0	20h
CMPCVL1	120h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CMPCV0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMPCV0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CMPCV0	Compare Count Value Bits [31:0] Specifies the lower 32 bits of the 56-bit compare value.

23.8.3.3 Compare Count Value High (CMPCVH0 - CMPCVH1)

Indicates the current count value bits 63–32.

NOTE

You must write to this set of registers when operating on the base frequency only. Writes to these registers when running on the alternate frequency may have unpredictable results.

Offset

Register	Offset
CMPCVH0	24h
CMPCVH1	124h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								CMPCV1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMPCV1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23-0 CMPCV1	Compare Count Value Bits [55:32] Specifies the upper 24 bits of the 56-bit compare value. Bits[63:56] are always zero.

23.8.3.4 Compare Control (CMPCR0 - CMPCR1)

Provides control and status of the compare function. When enabled, CMPCR0[ISTAT] indicates whether the counter value is greater than or equal to the value in CMPCV*n*. The ISTAT equation is:

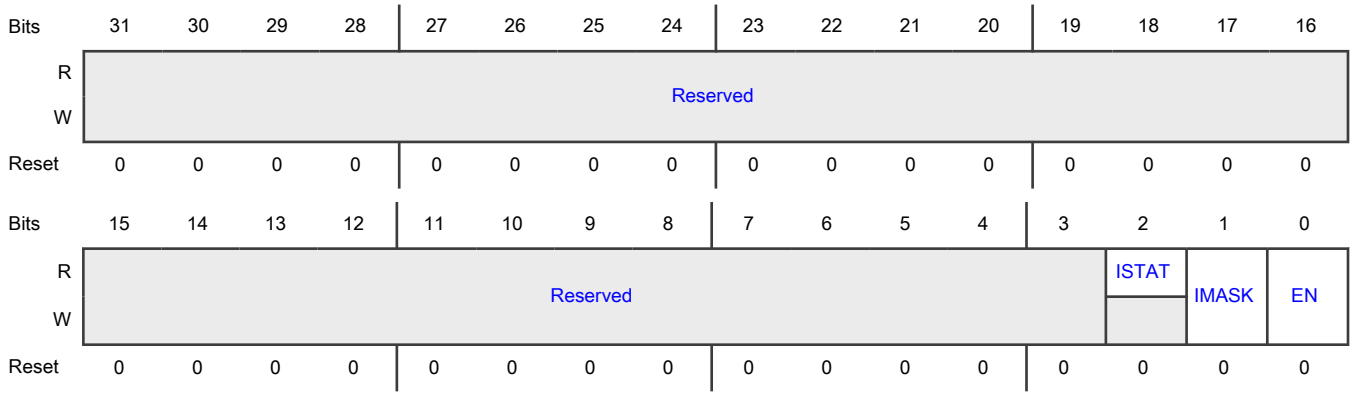
Value of CMPCR0[ISTAT] = CNTCV*n* ≥ CMPCV*n*

CMPCR0[ISTAT] takes no account of the value of CMPCR0[IMASK]. If CMPCR0[ISTAT] = 1 and CMPCR0[IMASK] = 0, the interrupt request is asserted. Writing 0 to CMPCR0[EN] makes CMPCR0[ISTAT] = 0 and negates the interrupt output signal.

Offset

Register	Offset
CMPCR0	2Ch
CMPCR1	12Ch

Diagram



Fields

Field	Description
31-3 —	Reserved
2	Compare Interrupt Status

Table continues on the next page...

Table continued from the previous page...

Field	Description
ISTAT	Indicates whether the counter value is less than, greater than, or equal to the compare value. 0 - Either less than the compare value or compare is disabled 1 - Greater than or equal to the compare value and compare is enabled
1 IMASK	Interrupt Request Mask Specifies whether the interrupt output signal is masked. 0 - Not masked 1 - Masked
0 EN	Compare Enable Enables the compare function. 0 - Disable 1 - Enable

23.8.3.5 Counter ID (CNTID0)

Indicates the architectural version 0 of SYS_CTR.

Offset

Register	Offset
CNTID0	FD0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CNTID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNTID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CNTID	Counter Identification Indicates counter ID 0, the architectural version of this system counter.

Chapter 24
System Debug

24.1 Overview

The chip debug architecture is based on Arm CoreSight technology, with support for the Cortex-A55, Cortex-M33, and EdgeLock Secure Enclave cores.

24.1.1 Block Diagram

The block diagram for the system level debug architecture is provided below.

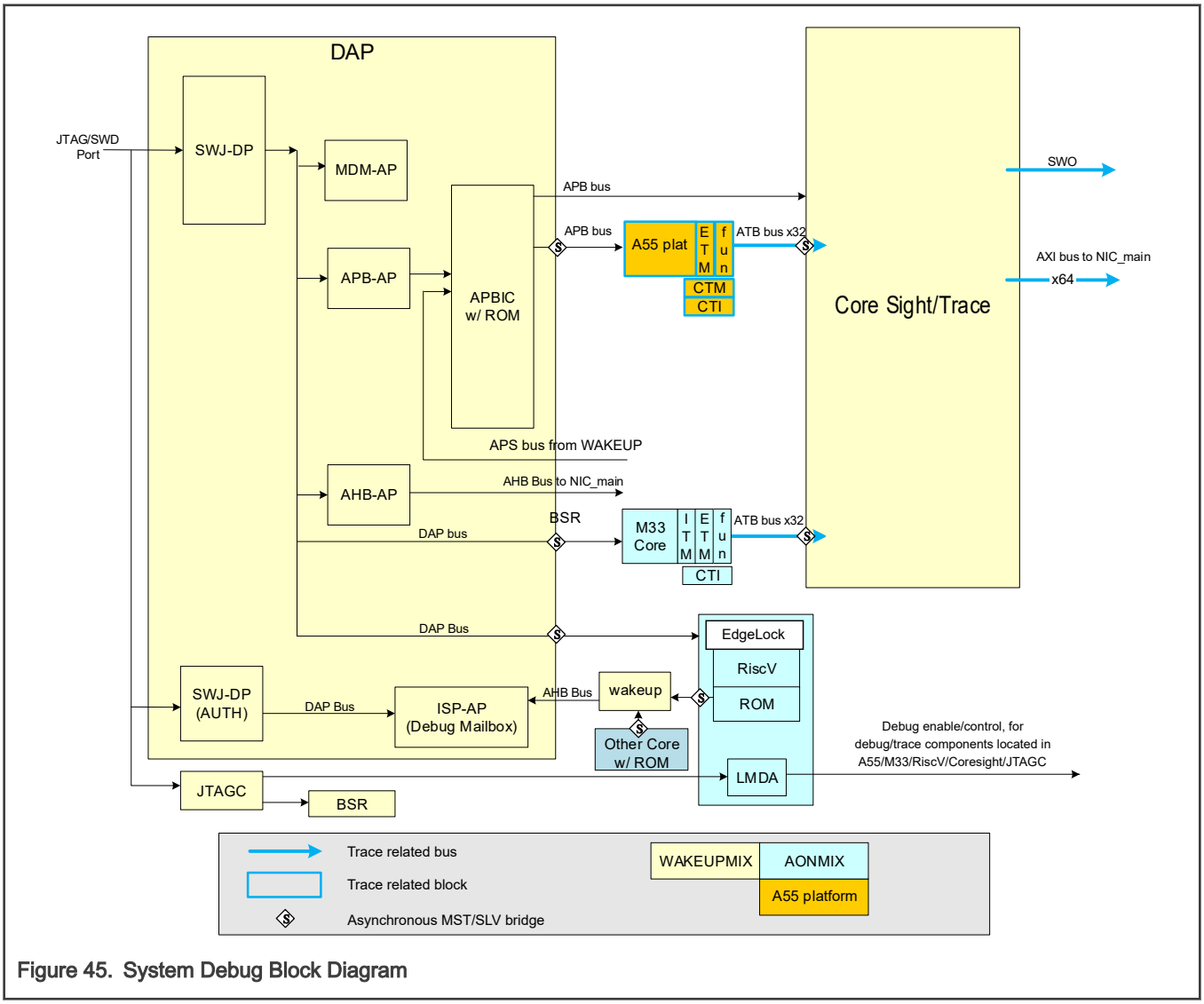


Figure 45. System Debug Block Diagram

The debug/trace components are listed below:

- Debug Access Port (DAP) - Has Serial Wire port (SWJ-DP) which interprets the data coming in and routes to appropriate Access Port (AP).
- Serial Wire port (SWJ-DP) - Consists of a wrapper around the JTAG-DP and SW-DP. It selects JTAG or SWD as the connection mechanism and enables either JTAG-DP or SW-DP as the interface to the various on chip debug/trace components. It is enabled only after debug is authenticated.

- SWJ-DP (AUTH) is another SWJ-DP instance dedicated for ISP-AP. It is enabled only when debug authentication is required, means it will be disabled if debug is authenticated.
- JTAGC
  - JTAGC and DAP/SWJ-DP share the same JTAG interface from PAD.
  - Achieve boundary scan testing, with an external BSR connected.
  - Supports Challenge-response authentication, with LMDA in Edgelock secure enclave.
  - Supports IEEE 1149.6/AC JTAG.
- Debug Access port for Cortex-A55 platform.
- Debug Access port for Coresight.
- Cortex-M33 DAP-AP - Debug Access port for M33 core.
- RISC-V DAP-AP - Debug Access port for RISC-V core in Edgelock secure enclave.
- ISP-AP - Debug Access port for Debug Mailbox, to allow the debugger to communicate with onboard ROM code.
- MDM-AP registers for debugger to control multi-core halt/resume cores.
- Coresight components.
- Cross-Trigger Interface (CTI) and Cross-Trigger Matrix (CTM).

24.1.2 Features

The key features of the debug system include:

- Supports 1149.1/ARM SWD interface. Supports both 5-pins (JTAG) and 2-pins (SWD) interfaces, to make efficient use of package pins, serial wire shares the JTAG pins use an auto-detect mechanism.

NOTE

1149.7 Compact JTAG is NOT supported.

- Supports both non-intrusive and halt-mode trace/debug options
- Supports trace for various cores, including a trace Memory Controller (TMC) to enable capturing trace
  - 4KB ETF in SOC trace block
  - ETR (4G memory range, 64-bit wide) to allow routing trace data to system memory
  - CoreSight Serial Wire Output
- Supports cross trigger between the Cortex-A55 and Cortex-M33
- MDM-AP registers for debugger to control multi-core halt/resume cores
- Supports narrow timestamp for trace components
- Supports ISP-AP (Debug Mailbox), allowing the debugger to communicate with onboard ROM code, providing debug authentication based on messages
- Supports challenge-response authentication

NOTE

Arm ELA-500 and Coresight SoC-600 are not supported.

NOTE

When CM33 core is under software reset (core only and not its debug components), FCLK is cut to the DAP AHB AP of CM33 core. At this time debugger can not configure debug components of CM33. The work around is, after the core comes out of reset it will be gated by TROUT of CM33 CPU WAIT, debugger can configure the debug components while the wait is asserted and once it is done. TROUT can be configured to ungate the core and debug from 1st instruction could be possible.

NOTE

External debugger can access memory inside CM33 through AHBAP when cm33_spiden and cm33_dbgen are enabled by security core.

24.2 Functional description

24.2.1 Debug Access Port (DAP) TAP

DAP is a standard Arm component, comprising of several components. These components are used to access the DAP from an external debugger and Access Ports to access on-chip debug system resources. The DAP supports 1149.1/Arm SW-DP interface, which means that the JTAG interface can be operate in standard 5-pin JTAG-DP interface or in 2-pin SW-DP interface. The following figure shows the connectivity between the DAP and the pads.

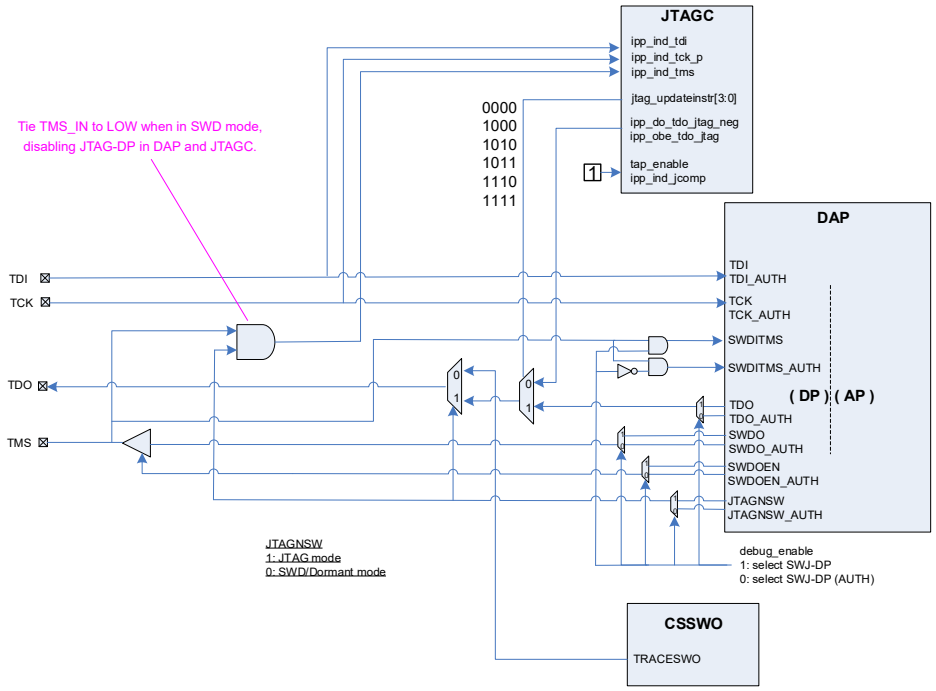


Figure 46. JTAG Pad Connectivity

For PAD descriptions, see External Signals.

24.2.2 JTAGC and JTAG mux

The JTAGC and SWJ-DP/SWJ-DP (AUTH) share the same 5-pin JTAG interface from PAD. SWJ-DP and SWJ-DP (AUTH) will not be enabled at the same time, controlled by the authentication status from LMDA.If debug is authenticated, select SWJ-DP, otherwise select SWJ-DP (AUTH).

SW-DP pins (TCK,TMS) are connected directly from and to SoC pins.

- When SW-DP is enabled, both JTAGC and JTAG-DP will be disabled.

- When SW-DP is disabled, the output to TDO from JTAGC and JTAG-DP will be selected by a mux and that mux is controlled by decode of the IR inside the JTAGC (JTAGC output port jtag_updateinstr[4:0]), and then connected to SoC pins for debug and/or test purposes.

Selection is based on the following table of decodes.

Table 122. Instruction Register

Instruction	Value	Module
EXTEST (Arm)	4'b0000	Arm JTAG-DP
IDCODE (JTAGC)	4'b0001	JTAGC
SAMPLE/PRELOAD	4'b0010	JTAGC
SAMPLE	4'b0011	JTAGC
EXTEST (JTAGC)	4'b0100	JTAGC
TEST_LEAKAGE	4'b0101	JTAGC
ENABLE_TEST_CTRL	4'b0110	JTAGC
ENABLE_CENSOR_CTRL	4'b0111	JTAGC
ABORT (Arm)	4'b1000	Arm JTAG-DP
HIGHZ	4'b1001	JTAGC
DPACC (Arm)	4'b1010	Arm JTAG-DP
APACC (Arm)	4'b1011	Arm JTAG-DP
CLAMP	4'b1100	JTAGC
ENABLE_SOC_DATA1	4'b1101	JTAGC
IDCODE (Arm)	4'b1110	Arm JTAG-DP
BYPASS (Arm)	4'b1111	Arm JTAG-DP

Partial of TARGETID[31:0] input for SWJ-DP should also be connected to JTAGC, for coherence, as below:

Table 123. JTAG TARGETID

JTAGC	SWJ-DP	description
jtagc_pin_plug[9:0]	TARGETID[21:12]	Part Identification Number
jtagc_prn_plug[3:0]	TARGETID[31:28]	Part Revision Number
jtagc_jcomp_plug[0:J-1]	1'b1 (J=1)	JTAG JCOMP value

24.2.3 AP Select Decoding

The recommendation for AP selects decoding is provided in the following table.

NOTE

SWJ-DP (AUTH) has only one DAP slot (the first DAP slot) targeting ISP-AP, so it has no DAPBUSIC decoding logic.

The DAPBUSIC in the table below is only for SWJ-DP rather than SWJ-DP (AUTH).

Table 124. Address Decoding for DAPBUSIC

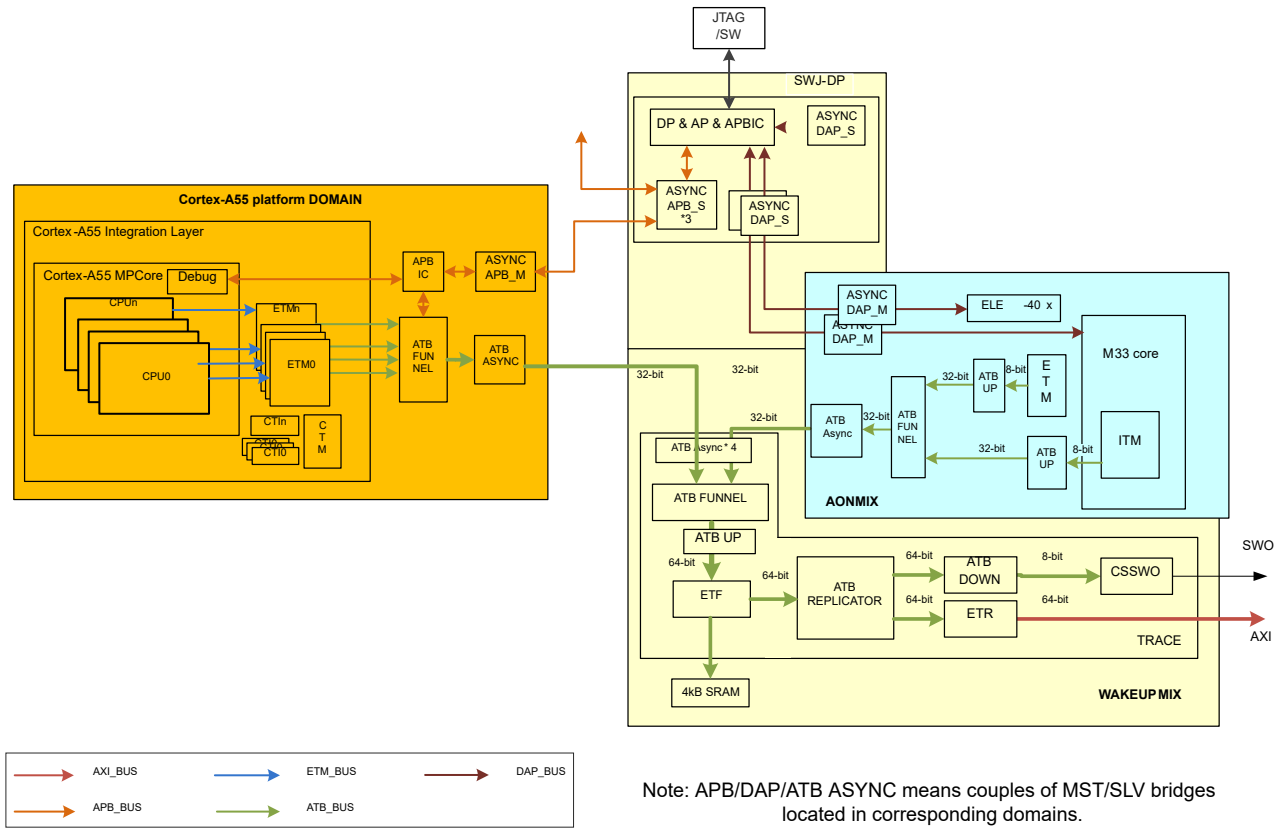
Destination	DAPBUSIC_AP_SEL[31:24]
AHB-AP to System Bus	8'h00 (M0)
APB-AP0 to CoreSight System	8'h01 (M1)
APB-AP0 to A-core Platform	Refer to Address decoding for APBIC
DAP-AP to M33	8'h03 (M3)
DAP-AP to Edgelock secure enclave RISC-V core	8'h04 (M4)
Debug Mailbox ISP-AP to Edgelock secure enclave	Reserved
MDM-AP	8'h06 (M6)
Reserved (Default AP response)	8'h07 -8'hFF

Table 125. Address Decoding for APBIC

Destination	APBIC_TA_REG[24:16], 64KB
DAPROM - Debug Access Port ROM	9'b000000000
Reserved	9'b000000001 - 9'b001111111
A-core Platform	9'b010000000 - 9'b011111111
Reserved	9'b100000000
Reserved	9'b100000001
CoreSight Serial Wire Output	9'b100000010
CoreSight Embedded Trace FIFO	9'b100000011
CoreSight Embedded Trace Router	9'b100000100
CoreSight Cross Trigger Interface	9'b100000101
CoreSight Cross Trigger Interface	9'b100000110
Reserved	9'b100000111 - 9'b111111111

24.2.4 Trace

The debug and trace implementation for the Cortex-A55 and Cortex-M33 is shown in the following figure.



Note: APB/DAP/ATB ASYNC means couples of MST/SLV bridges located in corresponding domains.

Figure 47. Trace Diagram

- Trace sources include ETM in Cortex-A55 and ITM/ETM in Cortex-M33, and ETM for DSP.
- One Embedded Trace FIFO (ETF, can operate as ETB, programmable), 4KB size as 128x256 (width x depth) with single port of SRAM.
- ATB funnel can capture trace data from Cortex-A55, Cortex-M33 ETM and Cortex-M33 ITM.
- ETR - Routing trace to system enabling to reuse SoC resources. (4G memory range, 64-bit wide).

24.2.5 Embedded Cross Trigger (ECT)

The ECT for coreSight consists of a few CTIs and CTM. The diagram below shows the structure of the ECT. The ECT enables subsystems to interact, that is cross trigger with each other. It is used for multiple core run control and trace cross trigger.

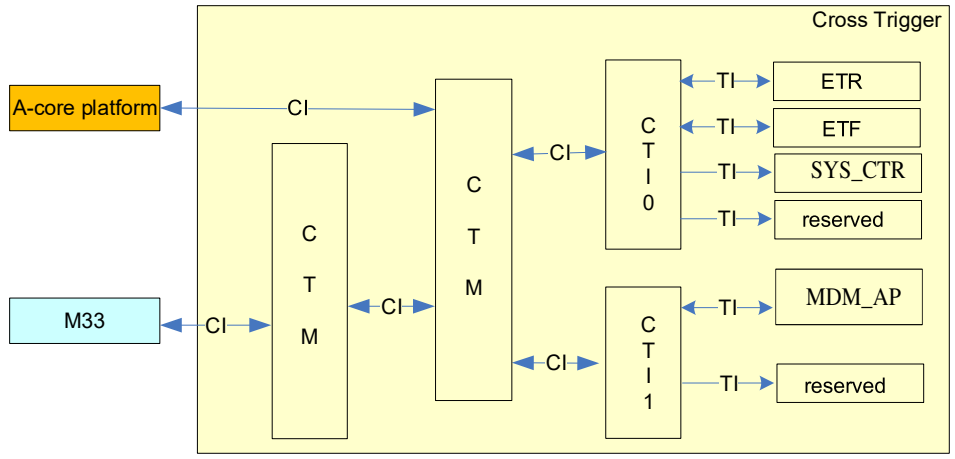


Figure 48. Embedded Cross Trigger Diagram

24.2.6 Timestamp Generation

There is no Arm Coresight Timestamp Generator in the system, and instead the System Counter is the source of the time used by all of the following components:

- System Counter internal counters and comparators.
- Cortex-A55 timestamp value (for trace).
- Cortex-M33 timestamp value (for trace).
- TSTMR1 and TSTMR2 read-only timestamp snapshot registers.
- Cortex-A55 Arm Generic Timer input.

The Cortex-A55 Arm Generic Timer is provided the full width uncompressed timestamp directly from the System Counter, without going through any Coresight Narrow timestamp encoder or decoder blocks. The Coresight Narrow timestamp IP are not used for the Cortex-A Generic Timer inputs in order to provide the maximum accuracy for the Cortex-A functional usage, as recommended by Arm. See the diagram below for details.

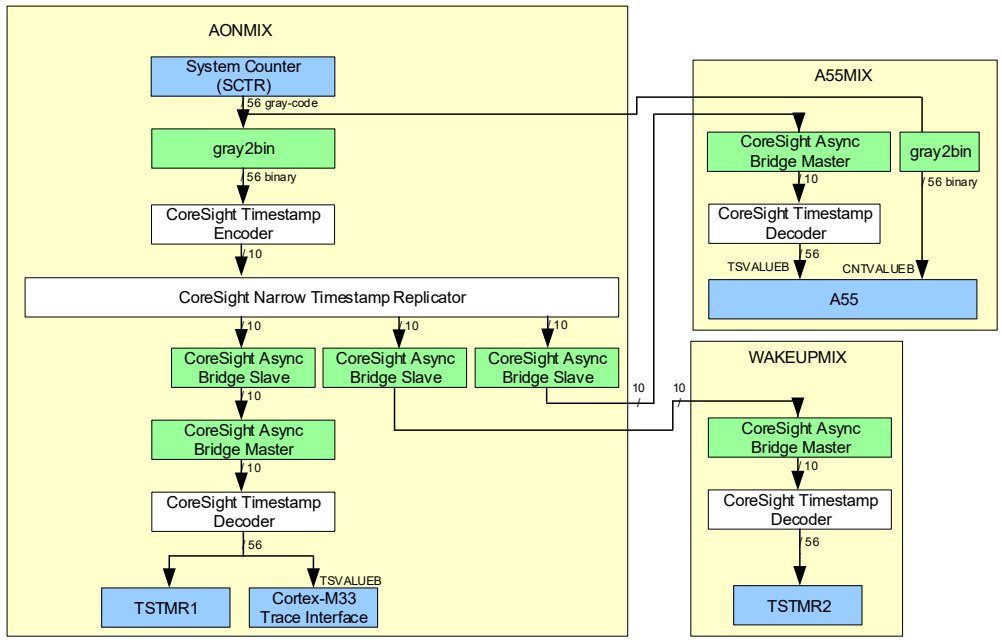


Figure 49. Timestamp Distribution

24.2.7 Debug Access Control from ROM Code

Once debug is authenticated, an external debugger tool may break code execution at any moment. This can potentially be during the boot process, when the ROM code is executing. If this occurs, the normal ROM execution is interrupted. The chip environment is uncertain (for example, the CLOCK initialization status, the interrupt status, the MPU settings).

When the debug tool loads the Flashloader, it will reset the core and halt the core first and then load the Flashloader. In such case, the Cortex-M33 system will be reset, and the ROM execution would be interrupted, and the delay time is not guaranteed before loading the Flashloader. Tasks such as the analog trim and clock initialization may be interrupted.

To address this, the chip allows the ROM code to receive a request whenever a debug request is received, gate the debug request until the ROM code enters a safe state to allow the debugger to continue. This is for reasons purely related to debug ease-of-use, not for reasons related to security.

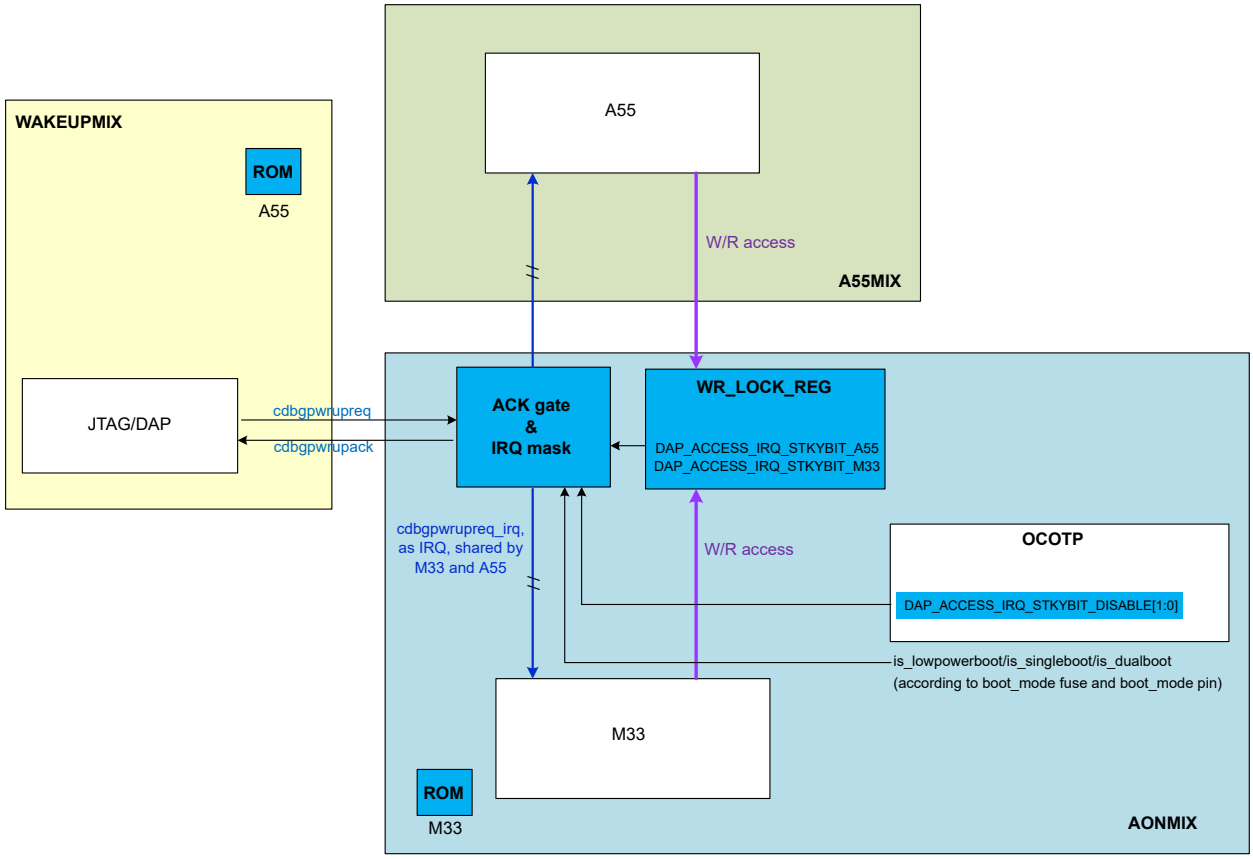


Figure 50. Debug Access Control from ROM Code

Details for the fuse controlling the debug access control from ROM code is provided in the following table.

Table 126. Fuse definition controlling debug access control from ROM code

DAP_ACCESS_IRQ_ STKYBIT_DISABLE[1]	DAP_ACCESS_IRQ_ STKYBIT_DISABLE[0]	Debug Control Logic
0	X	Disable DAP Access qualified by IRQ and sticky-bit logic. Direct DAP Access.
1	0	Enable DAP Access qualified by IRQ and sticky-bit logic.
1	1	Disable DAP Access qualified by IRQ and sticky-bit logic. Direct DAP Access.

DAP_ACCESS_IRQ_STKYBIT_DISABLE supports BootROM control of debug enable, and both fuse bits are in a section of BootROM that may be controlled by either NXP or OEM. By default, this feature is disabled when no fuses are blown.

This feature for the Cortex-M33 and Cortex-A55 will be enabled or disabled at the same time. Therefore, there is no case where one is enabled but another is disabled.

- For low power boot mode (Cortex-M boot), will select Cortex-M33's sticky bit.
- For single boot mode (Cortex-A boot), will select A55's sticky bit.

Pseudo code for the solution is pasted below:

```
      assign dap_access_irq_fuse_disabled =  (DAP_ACCESS_IRQ_STKYBIT_DISABLE[1] == 0) ||
((DAP_ACCESS_IRQ_STKYBIT_DISABLE[1] == 1) && (DAP_ACCESS_IRQ_STKYBIT_DISABLE[0] == 1));

      assign dap_accessible = dap_access_irq_fuse_disabled?  1'b1 :
      (is_lowpowerboot) ?  (DAP_ACCESS_IRQ_STKYBIT_M33) :
      (is_singleboot) ?   (DAP_ACCESS_IRQ_STKYBIT_A55) :
      (is_dualboot)?   (DAP_ACCESS_IRQ_STKYBIT_M33 && DAP_ACCESS_IRQ_STKYBIT_A55) : 1'b1;

      assign cdbgpwrupack = cdbgpwrupreq && fuse_latched && dap_accessible;
      assign cdbgpwrupreq_irq = cdbgpwrupreq && (~cdbgpwrupack);
```

NOTE

- DAP_ACCESS_IRQ_STKYBIT_M33 is a W/R register controlled by the Cortex-M33 ROM. Once set to "1", it will be "1" unless there is a reset.
- DAP_ACCESS_IRQ_STKYBIT_A55 is a W/R register controlled by the Cortex-A55 ROM. Once set to "1", it will be "1" unless there is a reset.
- cdbgpwrupreq_irq will be the final IRQ, shared by the Cortex-M33 and Cortex-A55.

24.3 External Signals

The JTAG signals are outlined in the following table.

Table 127. External signals

Name	I/O	JTAG	SWD	Pullup (PU) / Pulldown (PD)
TMS/SWDIO	I/IO	Mode selection	Data In/Out	PU
TCLK/SWCLK	I	Clock	Clock	PD
TDI	I	Data input	-	PU
TDO	O	Data output	-	NC

Chapter 25
JTAG Controller (JTAGC)

25.1 Chip-specific JTAGC information

Table 128. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing

The JTAG_IDCODE is the 32-bit identifier for the silicon for the purpose of debug and boundary scan access.

Table 129. JTAG IDCODE definition

Field	Assignment
31-28	0000
27-22	100010
21-17	01001
16-12	01000
11-1	000_00001110
0	1

25.1.1 Module instances

This device has one instance of the JTAGC module.

25.2 Overview

JTAGC allows you to test chip functionality and connectivity while remaining transparent to system logic when not in test mode. Testing is performed via a boundary scan technique, as defined in the IEEE 1149.1-2001 standard. All data input to and output from JTAGC is communicated in serial format.

25.2.1 Block diagram

The following is a simplified block diagram of the JTAG Controller (JTAGC) block. See the chip-specific configuration information within this chapter as well as Register description for more information about the JTAGC registers.

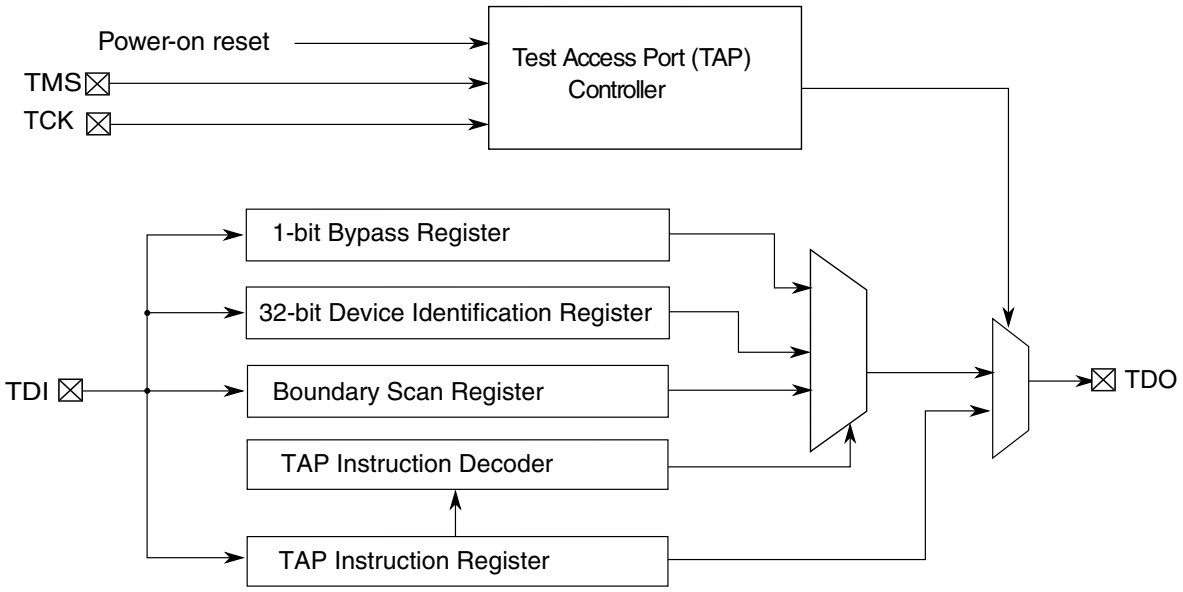


Figure 51. JTAG (IEEE 1149.1) block diagram

25.2.2 Features

The JTAGC block is compliant with the IEEE 1149.1-2001 standard, and supports the following features:

- IEEE 1149.1-2001 TAP interface
  - Four pins (TDI, TMS, TCK, and TDO)
- Instruction register that supports several IEEE 1149.1–2001 defined instructions as well as several public and private device-specific instructions (see JTAGC block instructions for a list of supported instructions).
- Bypass register, boundary scan register, and device identification register
- TAP controller state machine that controls the operation of the data registers, instruction register, and associated circuitry

25.3 Functional description

25.3.1 Modes of operation

The JTAGC block uses a power-on reset indication as its primary reset signals. Several IEEE 1149.1–2001 defined test modes are supported, as well as a bypass mode.

25.3.1.1 IEEE 1149.1–2001 defined test modes

The JTAGC block supports several IEEE 1149.1–2001 defined test modes. A test mode is selected by loading the appropriate instruction into the instruction register when the JTAGC is enabled. Supported test instructions include EXTEST, HIGHZ, CLAMP, PRELOAD.

Each instruction defines the set of data register(s) that may operate and interact with the on-chip system logic when the instruction is current. Only one test data register path is enabled to shift data between TDI and TDO for each instruction.

The boundary scan register is enabled for serial access between TDI and TDO when the EXTEST, PRELOAD instructions are active. The single-bit bypass register shift stage is enabled for serial access between TDI and TDO when the following instructions are active:

- BYPASS
- HIGHZ

- CLAMP

The functionality of each test mode is explained in more detail in JTAGC block instructions.

25.3.1.2 Bypass mode

When no test operation is required, the BYPASS instruction can be loaded to place the JTAGC block into bypass mode. When in bypass mode, the single-bit bypass shift register is used to provide a minimum-length serial path to shift data between TDI and TDO.

25.3.2 IEEE 1149.1-2001 (JTAG) TAP

The JTAGC block uses the IEEE 1149.1-2001 TAP for accessing registers. This port can be shared with other TAP controllers on the MCU. Ownership of the port is determined by the value of the currently loaded instruction.

Data is shifted between TDI and TDO through the selected register (instruction, test data, or the bypass register) starting with the least significant bit, as illustrated in the following figure.

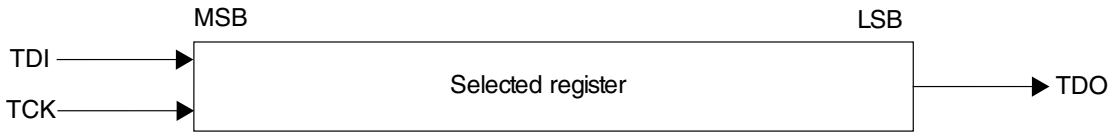
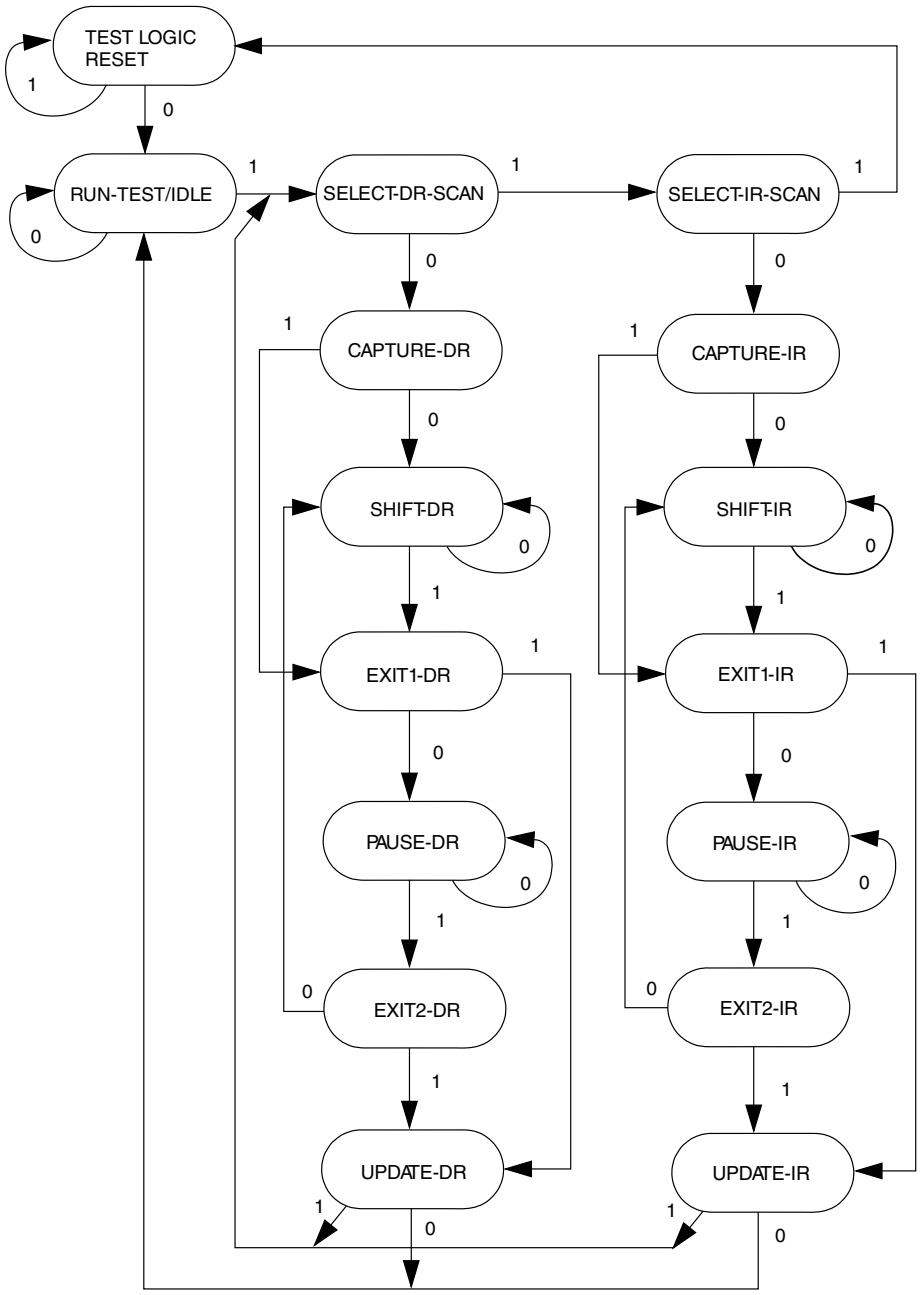


Figure 52. Shifting data through a register

25.3.3 TAP controller state machine

The TAP controller is a synchronous state machine that interprets the sequence of logical values on the TMS pin.

The following figure shows the machine's states. The value shown next to each state is the value of the TMS signal sampled on the rising edge of the TCK signal. As the figure shows, holding TMS at logic 1 when clocking TCK through a sufficient number of rising edges also causes the state machine to enter the Test-Logic-Reset state.



The value shown adjacent to each state transition in this figure represents the value of TMS at the time of a rising edge of TCK.

Figure 53. IEEE 1149.1-2001 TAP controller finite state machine

25.3.3.1 Enabling the TAP controller

The JTAGC TAP controller is enabled by setting the JTAGC enable to a logic 1 value.

25.3.3.2 Selecting an IEEE 1149.1-2001 register

Access to the JTAGC data registers is achieved by loading the instruction register with any of the JTAGC block instructions when the JTAGC is enabled. Instructions are shifted in via the Select-IR-Scan path and loaded in the Update-IR state. At this point, all data register access is performed via the Select-DR-Scan path.

The Select-DR-Scan path is used to read or write the register data by shifting in the data (LSB first) during the Shift-DR state. When reading a register, the register value is loaded into the IEEE 1149.1-2001 shifter during the Capture-DR state. When writing a register, the value is loaded from the IEEE 1149.1-2001 shifter to the register during the Update-DR state. When reading a register, there is no requirement to shift out the entire register contents. Shifting may be terminated after the required number of bits have been acquired.

25.3.4 JTAGC block instructions

The JTAGC block implements the IEEE 1149.1-2001 defined instructions. See the IEEE 1149.1-2001 standard for more details. All undefined opcodes are reserved.

NOTE

See the chip specific information JTAGC block instructions for applicable to your device.

25.3.4.1 IDCODE instruction

IDCODE selects the 32-bit device identification register as the shift path between TDI and TDO. This instruction allows interrogation of the MCU to determine its version number and other part identification data. IDCODE is the instruction placed into the instruction register when the JTAGC block is reset.

25.3.4.2 PRELOAD instruction

The PRELOAD instruction has two functions:

- The PRELOAD portion of the instruction initializes the boundary scan register cells before selecting the EXTEST or CLAMP instructions to perform boundary scan tests. This is achieved by shifting in initialization data to the boundary scan register during the Shift-DR state. The initialization data is transferred to the parallel outputs of the boundary scan register cells on the falling edge of TCK in the Update-DR state. The data is applied to the external output pins by the EXTEST or CLAMP instruction. System operation is not affected.

25.3.4.3 EXTEST external test instruction

EXTEST selects the boundary scan register as the shift path between TDI and TDO. It allows testing of off-chip circuitry and board-level interconnections by driving preloaded data contained in the boundary scan register onto the system output pins. Typically, the data is loaded into the boundary scan register using the PRELOAD instruction before the selection of EXTEST. EXTEST asserts the internal system reset for the MCU to force a predictable internal state when performing external boundary scan operations.

25.3.4.4 HIGHZ instruction

HIGHZ selects the bypass register as the shift path between TDI and TDO. When HIGHZ is active all output drivers are placed in an inactive drive state (for example, high impedance). HIGHZ also asserts the internal system reset for the MCU to force a predictable internal state.

25.3.4.5 CLAMP instruction

CLAMP allows the state of signals driven from MCU pins to be determined from the boundary scan register when the bypass register is selected as the serial path between TDI and TDO. CLAMP enhances test efficiency by reducing the overall shift path to a single bit (the bypass register) when conducting an EXTEST type of instruction through the boundary scan register. CLAMP also asserts the internal system reset for the MCU to force a predictable internal state.

25.3.4.6 BYPASS instruction

BYPASS selects the bypass register, creating a single-bit shift register path between TDI and TDO. BYPASS enhances test efficiency by reducing the overall shift path when no test operation of the MCU is required. This allows more rapid movement of test data to and from other components on a board that are required to perform test functions. When the BYPASS instruction is active the MCU operates normally.

25.3.5 Boundary scan

The boundary scan technique allows signals at component boundaries to be controlled and observed through the shift-register stage associated with each pad. Each stage is part of a larger boundary scan register cell, and cells for each pad are interconnected serially to form a shift-register chain around the border of the design. The boundary scan register consists of this shift-register chain, and is connected between TDI and TDO when the EXTEST, PRELOAD instructions are loaded. The shift-register chain contains a serial input and serial output, as well as clock and control signals.

25.3.6 Clocking

There is one clock for this module. For more information See External signals.

25.3.7 Interrupts

This module has no interrupts.

25.3.8 Reset

The JTAGC block is placed in reset when:

- Power-on reset is asserted
- TMS input is held high for enough consecutive rising edges of TCK to sequence the TAP controller state machine into the Test-Logic-Reset state

Holding TMS high for five consecutive rising edges of TCK guarantees entry into the Test-Logic-Reset state regardless of the current TAP controller state. Asserting power-on reset results in asynchronous entry into the reset state.

When in reset, the following actions occur:

- The TAP controller is forced into the Test-Logic-Reset state, thereby disabling the test logic and allowing normal operation of the MCU.
- The instruction register is loaded with the IDCODE instruction.

25.4 External signals

JTAGC includes a set of signals that connect to off-chip development tools and allow access to test support functions. The JTAGC signals are outlined in the following table and described in the following sections.

Table 130. JTAG signal properties

Name	I/O	Function	Reset state
TCK	Input	Test clock	Weak pulldown
TDI	Input	Test data in	Weak pullup
TDO	Output	Test data out	High Z[1]
TMS	Input	Test mode select	Weak pullup

1. TDO output buffer is disabled when the JTAGC is not in the Shift-IR or Shift-DR states. A weak pull may be implemented at the TDO pad for use when JTAGC is inactive.

25.4.1 Test clock input (TCK)

Test Clock Input (TCK) is an input pin used to synchronize the test logic and control register access through the TAP.

25.4.2 Test data input (TDI)

Test Data Input (TDI) is an input pin that receives serial test instructions and data. TDI is sampled on the rising edge of TCK.

25.4.3 Test data output (TDO)

Test Data Output (TDO) is an output pin that transmits serial output for test instructions and data. TDO is tristateable and is actively driven only in the Shift-IR and Shift-DR states of the TAP controller state machine, which is described in TAP controller state machine.

25.4.4 Test mode select (TMS)

Test Mode Select (TMS) is an input pin used to sequence the IEEE 1149.1-2001 test control state machine. TMS is sampled on the rising edge of TCK.

25.5 Initialization

To initialize the JTAGC block and enable access to registers, the following sequence is required:

1. Place the JTAGC in reset through TAP controller state machine transitions controlled by TMS.
2. Load the appropriate instruction for the test or action to be performed.

25.6 Application information

The test logic is a static logic design, and TCK can be stopped in either a high or low state without loss of data. However, the system clock is not synchronized to TCK internally. Any mixed operation using both the test logic and the system functional logic requires external synchronization.

25.7 Register description

This section provides a detailed description of the JTAGC block registers accessible through the TAP interface, including data registers and the instruction register. Individual bit-level descriptions and reset states of each register are included. These registers are not memory-mapped and can only be accessed through the TAP.

25.7.1 Instruction register

The JTAGC block uses a 4-bit instruction register as shown in the following figure.

The instruction register allows instructions to be loaded into the block to select the test to be performed, or the test data register to be accessed, or both. Instructions are shifted in through TDI when the TAP controller is in the Shift-IR state, and latched on the falling edge of TCK in the Update-IR state. The latched instruction value can only be changed in the Update-IR and Test-Logic-Reset TAP controller states.

Synchronous entry into the Test-Logic-Reset state results in the IDCODE instruction being loaded on the falling edge of TCK. Asynchronous entry into the Test-Logic-Reset state results in asynchronous loading of the IDCODE instruction. During the Capture-IR TAP controller state, the instruction shift register is loaded with the value 0b1, making this value the register's read value when the TAP controller is sequenced into the Shift-IR state.

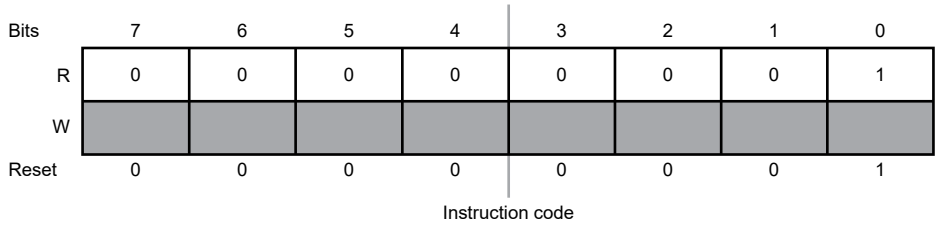


Figure 54. Instruction register

25.7.2 Bypass register

The bypass register is a single-bit shift register path selected for serial data transfer between TDI and TDO when the following instructions are active:

- BYPASS
- HIGHZ
- CLAMP

After entry into the Capture-DR state, the single-bit shift register is set to a logic 0. Therefore, the first bit shifted out after selecting the bypass register is always a logic 0.

25.7.3 Device identification register

The device identification (JTAG ID) register, shown in the following figure, allows the revision number, part number, manufacturer, and design center responsible for the design of the part to be determined through the TAP.

The device identification register is selected for serial data transfer between TDI and TDO when the IDCODE instruction is active. Entry into the Capture-DR state when the device identification register is selected loads the IDCODE into the shift register to be shifted out on TDO in the Shift-DR state. No action occurs in the Update-DR state.

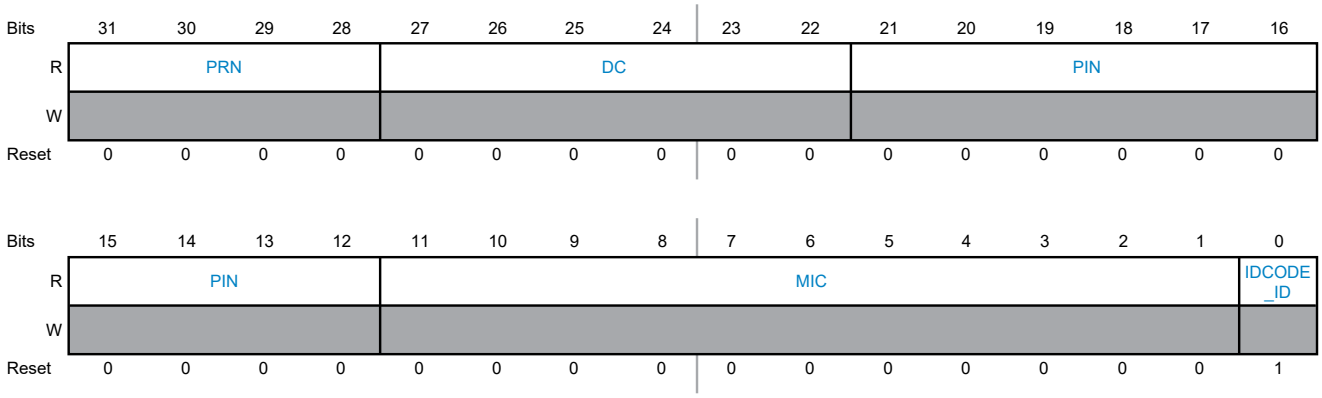


Figure 55. Device identification register

The following table describes the device identification register functions. The device identification register values are described in the chip-specific JTAGC information.

Table 131. Device identification register field descriptions

Field	Function
PRN	Part revision number Contains the revision number of the part.
DC	Design center Indicates the design center.
PIN	Part identification number Contains the part number of the device.
MIC	Manufacturer identity code

Table continues on the next page...

Table 131. Device identification register field descriptions (continued)

Field	Function
	Contains the reduced Joint Electron Device Engineering Council (JEDEC) ID .
IDCODE ID	IDCODE register ID Identifies this register as the device identification register and not the bypass register. Always set to 1.

25.7.4 Boundary scan register

The boundary scan register is connected between TDI and TDO when the EXTEST, PRELOAD instructions are active. It is used to:

- Capture input pin data
- Force fixed values on output pins
- Select a logic value and direction for bidirectional pins

Each bit of the boundary scan register represents a separate boundary scan register cell, as described in the IEEE 1149.1-2001 standard and discussed in Boundary scan. The size of the boundary scan register and bit ordering is device-dependent and can be found in the device BSDL file.

25.8 Glossary

TAP	Test access port
TCK	Test clock
TDI	Test data in
TDO	Test data out
TMS	Test mode select
JCOMP	JTAG compliance

Chapter 26
External Signals and Pin Multiplexing

26.1 Overview

The chip contains a limited number of pins, most of which have multiple signal options. These signal-to-pin and pin-to-signal options are selected by the input-output multiplexer called IOMUX. The IOMUX is also used to configure other pin characteristics, such as voltage level, drive strength, and hysteresis.

The muxing options table lists the external signals grouped by the module instance, the muxing options for each signal, and the registers used to route the signal to the chosen pad.

26.1.1 Muxing Options

Please refer the Pinlist spreadsheet attached with this RM for more details.

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

NOTE

LPUART DCB_B is the same as LPUART DCD_B.

NOTE

Chapter 27
IOMUX Controller (IOMUXC)

27.1 Overview

The IOMUX Controller (IOMUXC), together with the IOMUX, enables the IC to share one pad to several functional blocks. This sharing is done by multiplexing the pad's input and output signals.

Every module requires a specific pad setting (such as pull up or keeper), and for each pad, there are up to 8 muxing options (called ALT modes). The pad settings parameters are controlled by the IOMUXC.

The IOMUX consists only of combinatorial logic combined from several basic IOMUX cells. Each basic IOMUX cell handles only one pad signal's muxing.

27.1.1 Block diagram

The figure below illustrates the IOMUX/IOMUXC connectivity in the system.

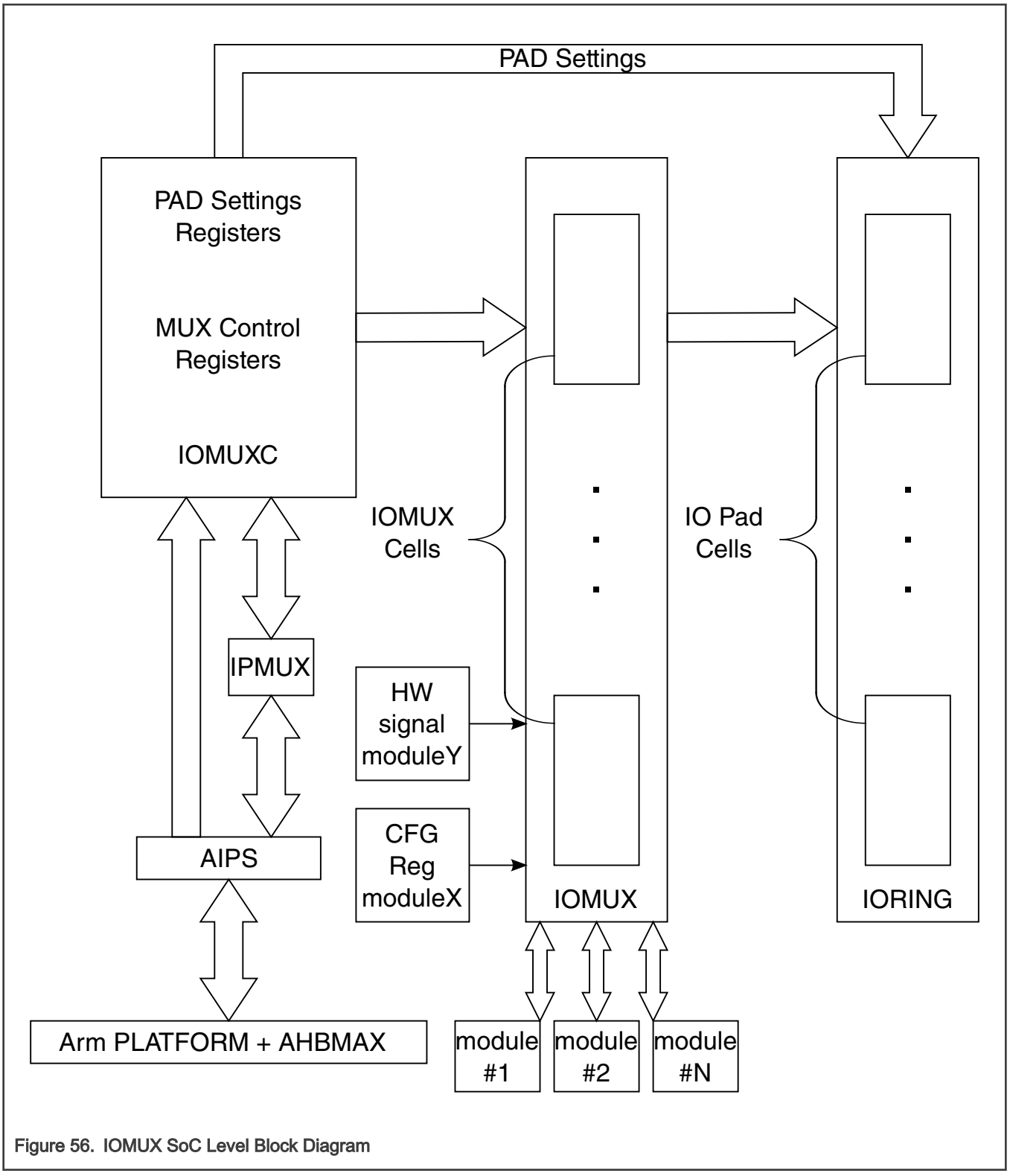


Figure 56. IOMUX SoC Level Block Diagram

27.1.2 Features

The IOMUXC features include:

- 32-bit software mux control registers (SW_MUX_CTL_PAD_<PAD NAME>) to configure 1 of 12 alternate (ALT) MUX_MODE fields of each pad and to enable the forcing of an input path of the pad(s) (SION bit).
- 32-bit software pad control registers (SW_PAD_CTL_PAD_<PAD_NAME>) to configure specific pad settings of each pad.
- 32-bit input select control registers to control the input path to a module when more than one pad drives this module input.

Each SW MUX/PAD CTL IOMUXC register handles only one pad.

Only the minimum number of registers required by software are implemented by hardware. For example, if only ALT0 and ALT1 modes are used on Pad x then only one bit register will be generated as the MUX_MODE control field in the software mux control register of Pad x.

The software mux control registers may allow the forcing of pads to become input (input path enabled) regardless of the functional direction driven. This may be useful for loopback and GPIO data capture.

27.2 Functional description

This section provides a complete functional description of the block.

The IOMUX consists of a number of basic iomux cell units. If only one functional mode is required for a specific pad, it would have been no need for IOMUX and the signals can be connected directly from the module to the I/O. The IOMUX cell is required whenever two or more functional modes are required for a specific pad or when one functional mode and the one test mode are required.

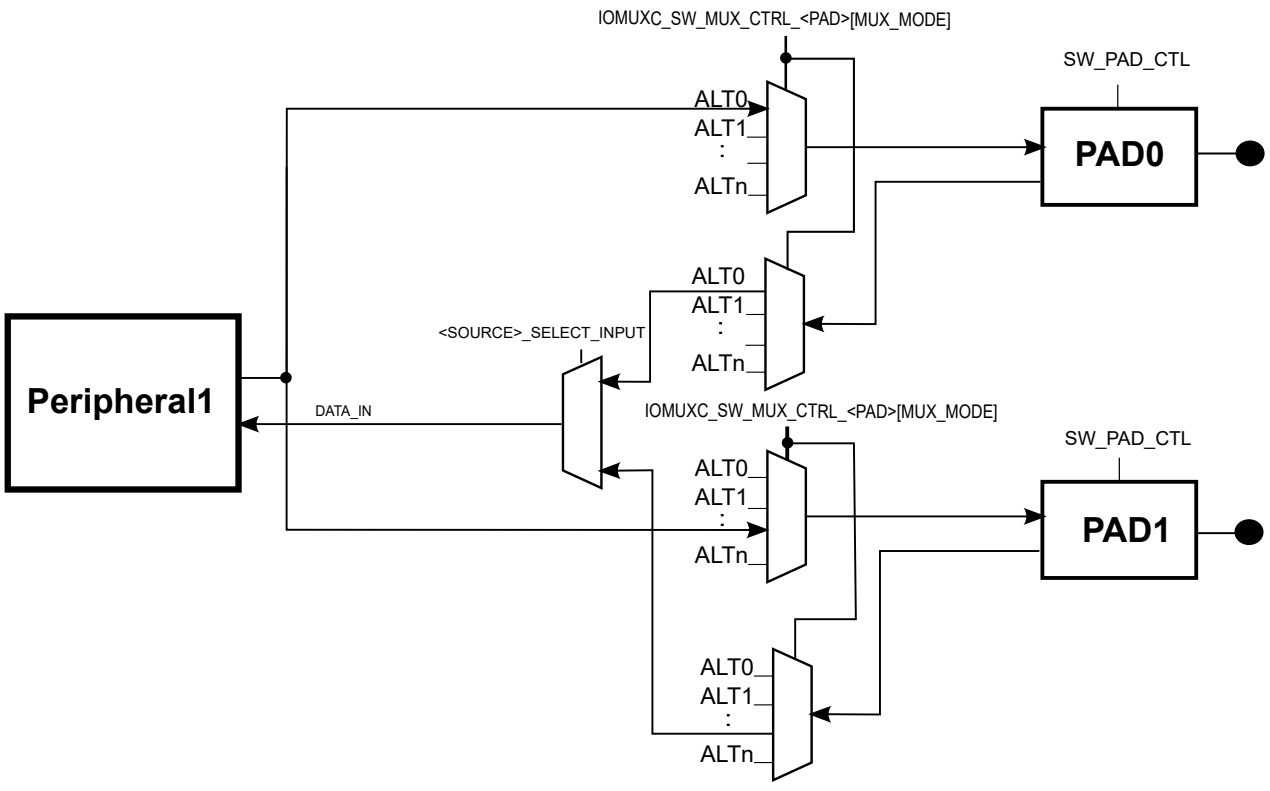


Figure 57. IOMUX Cell Block Diagram

27.2.1 SW Loopback through SION bit

A limited option exists to override the default pad functionality and force the input path to be active regardless of the value driven by the corresponding module. This can be done by setting the SION (Software Input On) bit in the IOMUXC_SW_MUX_CTL register (when available) to "1".

Uses include:

- LoopBack - Module x drives the pad and also receives pad value as an input.

27.2.2 Daisy chain - multi pads driving same module input pin

In some cases, more than one pad may drive a single module input pin. Such cases require the addition of one more level of IOMUXing; all of these input signals are muxed, and a dedicated software controlled register controls the mux in order to select the required input path.

A module port involved in "daisy chain" requires two software configuration commands, one for selecting the mode for this pad (programable via the SW_MUX_CTL_<PAD> registers) and one for defining it as the input path (via the daisy chain registers).

This means that a module port involved in "daisy chain" requires two software configuration commands, one for selecting the mode for this pad (programable via the SW_MUX_CTL_<PAD> registers) and one for defining it as the input path (via the daisy chain registers).

The daisy chain is illustrated in the figure below.

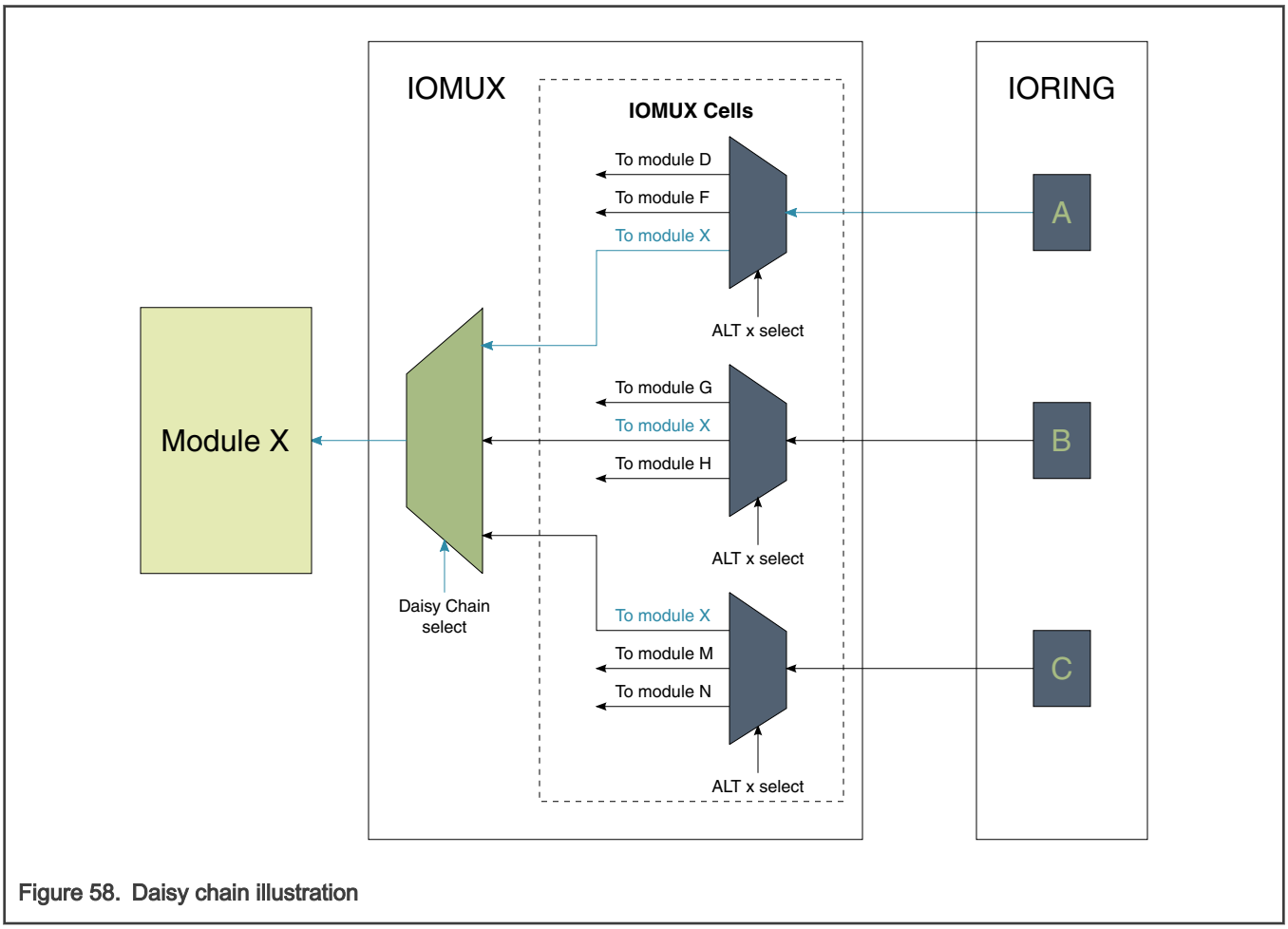


Figure 58. Daisy chain illustration

27.2.3 Clocks

The table found here describes the clock sources for IOMUXC. Please see Clock Controller Module (CCM) chapter for clock setting, configuration and gating information.

Table 132. IOMUXC Clocks

Clock Name	Description
ipg_clk_s	Peripheral access clock

27.3 Memory Map and register definition

This section includes the IOMUXC module memory map and detailed descriptions of all registers.

NOTE

Write/Read to the Reserved locations will not generate a transfer error.

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC in the registers below.

27.3.1 iomuxc register descriptions

27.3.1.1 iomuxc memory map

IOMUXC1 base address: 443C_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	SW_MUX_CTL_PAD_DAP_TDI SW MUX Control Register (SW_MUX_CTL_PAD_DAP_TDI)	32	RW	0000_0000
4	SW_MUX_CTL_PAD_DAP_TMS_SWDIO SW MUX Control Register (SW_MUX_CTL_PAD_DAP_TMS_SWDIO)	32	RW	0000_0000
8	SW_MUX_CTL_PAD_DAP_TCLK_SWCLK SW MUX Control Register (SW_MUX_CTL_PAD_DAP_TCLK_SWCLK)	32	RW	0000_0000
C	SW_MUX_CTL_PAD_DAP_TDO_TRACESWO SW MUX Control Register (SW_MUX_CTL_PAD_DAP_TDO_TRACESWO)	32	RW	0000_0000
10	SW_MUX_CTL_PAD_GPIO_IO00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO00)	32	RW	0000_0000
14	SW_MUX_CTL_PAD_GPIO_IO01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO01)	32	RW	0000_0000
18	SW_MUX_CTL_PAD_GPIO_IO02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO02)	32	RW	0000_0000
1C	SW_MUX_CTL_PAD_GPIO_IO03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO03)	32	RW	0000_0000
20	SW_MUX_CTL_PAD_GPIO_IO04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO04)	32	RW	0000_0000
24	SW_MUX_CTL_PAD_GPIO_IO05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO05)	32	RW	0000_0000
28	SW_MUX_CTL_PAD_GPIO_IO06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO06)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2C	SW_MUX_CTL_PAD_GPIO_IO07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO07)	32	RW	0000_0000
30	SW_MUX_CTL_PAD_GPIO_IO08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO08)	32	RW	0000_0000
34	SW_MUX_CTL_PAD_GPIO_IO09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO09)	32	RW	0000_0000
38	SW_MUX_CTL_PAD_GPIO_IO10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO10)	32	RW	0000_0000
3C	SW_MUX_CTL_PAD_GPIO_IO11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO11)	32	RW	0000_0000
40	SW_MUX_CTL_PAD_GPIO_IO12 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO12)	32	RW	0000_0000
44	SW_MUX_CTL_PAD_GPIO_IO13 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO13)	32	RW	0000_0000
48	SW_MUX_CTL_PAD_GPIO_IO14 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO14)	32	RW	0000_0000
4C	SW_MUX_CTL_PAD_GPIO_IO15 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO15)	32	RW	0000_0000
50	SW_MUX_CTL_PAD_GPIO_IO16 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO16)	32	RW	0000_0000
54	SW_MUX_CTL_PAD_GPIO_IO17 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO17)	32	RW	0000_0000
58	SW_MUX_CTL_PAD_GPIO_IO18 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO18)	32	RW	0000_0000
5C	SW_MUX_CTL_PAD_GPIO_IO19 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO19)	32	RW	0000_0000
60	SW_MUX_CTL_PAD_GPIO_IO20 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO20)	32	RW	0000_0000
64	SW_MUX_CTL_PAD_GPIO_IO21 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO21)	32	RW	0000_0000
68	SW_MUX_CTL_PAD_GPIO_IO22 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO22)	32	RW	0000_0000
6C	SW_MUX_CTL_PAD_GPIO_IO23 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO23)	32	RW	0000_0000
70	SW_MUX_CTL_PAD_GPIO_IO24 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO24)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
74	SW_MUX_CTL_PAD_GPIO_IO25 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO25)	32	RW	0000_0000
78	SW_MUX_CTL_PAD_GPIO_IO26 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO26)	32	RW	0000_0000
7C	SW_MUX_CTL_PAD_GPIO_IO27 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO27)	32	RW	0000_0000
80	SW_MUX_CTL_PAD_GPIO_IO28 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO28)	32	RW	0000_0000
84	SW_MUX_CTL_PAD_GPIO_IO29 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO29)	32	RW	0000_0000
88	SW_MUX_CTL_PAD_CCM_CLKO1 SW MUX Control Register (SW_MUX_CTL_PAD_CCM_CLKO1)	32	RW	0000_0000
8C	SW_MUX_CTL_PAD_CCM_CLKO2 SW MUX Control Register (SW_MUX_CTL_PAD_CCM_CLKO2)	32	RW	0000_0000
90	SW_MUX_CTL_PAD_CCM_CLKO3 SW MUX Control Register (SW_MUX_CTL_PAD_CCM_CLKO3)	32	RW	0000_0005
94	SW_MUX_CTL_PAD_CCM_CLKO4 SW MUX Control Register (SW_MUX_CTL_PAD_CCM_CLKO4)	32	RW	0000_0005
98	SW_MUX_CTL_PAD_ENET1_MDC SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_MDC)	32	RW	0000_0005
9C	SW_MUX_CTL_PAD_ENET1_MDIO SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_MDIO)	32	RW	0000_0005
A0	SW_MUX_CTL_PAD_ENET1_TD3 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_TD3)	32	RW	0000_0005
A4	SW_MUX_CTL_PAD_ENET1_TD2 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_TD2)	32	RW	0000_0005
A8	SW_MUX_CTL_PAD_ENET1_TD1 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_TD1)	32	RW	0000_0005
AC	SW_MUX_CTL_PAD_ENET1_TD0 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_TD0)	32	RW	0000_0005
B0	SW_MUX_CTL_PAD_ENET1_TX_CTL SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_TX_CTL)	32	RW	0000_0005
B4	SW_MUX_CTL_PAD_ENET1_TXC SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_TXC)	32	RW	0000_0005
B8	SW_MUX_CTL_PAD_ENET1_RX_CTL SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_RX_CTL)	32	RW	0000_0005

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
BC	SW_MUX_CTL_PAD_ENET1_RXC SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_RXC)	32	RW	0000_0005
C0	SW_MUX_CTL_PAD_ENET1_RD0 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_RD0)	32	RW	0000_0005
C4	SW_MUX_CTL_PAD_ENET1_RD1 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_RD1)	32	RW	0000_0005
C8	SW_MUX_CTL_PAD_ENET1_RD2 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_RD2)	32	RW	0000_0005
CC	SW_MUX_CTL_PAD_ENET1_RD3 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_RD3)	32	RW	0000_0005
D0	SW_MUX_CTL_PAD_ENET2_MDC SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_MDC)	32	RW	0000_0005
D4	SW_MUX_CTL_PAD_ENET2_MDIO SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_MDIO)	32	RW	0000_0005
D8	SW_MUX_CTL_PAD_ENET2_TD3 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_TD3)	32	RW	0000_0005
DC	SW_MUX_CTL_PAD_ENET2_TD2 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_TD2)	32	RW	0000_0005
E0	SW_MUX_CTL_PAD_ENET2_TD1 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_TD1)	32	RW	0000_0005
E4	SW_MUX_CTL_PAD_ENET2_TD0 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_TD0)	32	RW	0000_0005
E8	SW_MUX_CTL_PAD_ENET2_TX_CTL SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_TX_CTL)	32	RW	0000_0005
EC	SW_MUX_CTL_PAD_ENET2_TXC SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_TXC)	32	RW	0000_0005
F0	SW_MUX_CTL_PAD_ENET2_RX_CTL SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_RX_CTL)	32	RW	0000_0005
F4	SW_MUX_CTL_PAD_ENET2_RXC SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_RXC)	32	RW	0000_0005
F8	SW_MUX_CTL_PAD_ENET2_RD0 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_RD0)	32	RW	0000_0005
FC	SW_MUX_CTL_PAD_ENET2_RD1 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_RD1)	32	RW	0000_0005
100	SW_MUX_CTL_PAD_ENET2_RD2 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_RD2)	32	RW	0000_0005

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
104	SW_MUX_CTL_PAD_ENET2_RD3 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_RD3)	32	RW	0000_0005
108	SW_MUX_CTL_PAD_SD1_CLK SW MUX Control Register (SW_MUX_CTL_PAD_SD1_CLK)	32	RW	0000_0005
10C	SW_MUX_CTL_PAD_SD1_CMD SW MUX Control Register (SW_MUX_CTL_PAD_SD1_CMD)	32	RW	0000_0005
110	SW_MUX_CTL_PAD_SD1_DATA0 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA0)	32	RW	0000_0005
114	SW_MUX_CTL_PAD_SD1_DATA1 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA1)	32	RW	0000_0005
118	SW_MUX_CTL_PAD_SD1_DATA2 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA2)	32	RW	0000_0005
11C	SW_MUX_CTL_PAD_SD1_DATA3 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA3)	32	RW	0000_0005
120	SW_MUX_CTL_PAD_SD1_DATA4 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA4)	32	RW	0000_0005
124	SW_MUX_CTL_PAD_SD1_DATA5 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA5)	32	RW	0000_0005
128	SW_MUX_CTL_PAD_SD1_DATA6 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA6)	32	RW	0000_0005
12C	SW_MUX_CTL_PAD_SD1_DATA7 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA7)	32	RW	0000_0005
130	SW_MUX_CTL_PAD_SD1_STROBE SW MUX Control Register (SW_MUX_CTL_PAD_SD1_STROBE)	32	RW	0000_0005
134	SW_MUX_CTL_PAD_SD2_VSELECT SW MUX Control Register (SW_MUX_CTL_PAD_SD2_VSELECT)	32	RW	0000_0005
138	SW_MUX_CTL_PAD_SD3_CLK SW MUX Control Register (SW_MUX_CTL_PAD_SD3_CLK)	32	RW	0000_0005
13C	SW_MUX_CTL_PAD_SD3_CMD SW MUX Control Register (SW_MUX_CTL_PAD_SD3_CMD)	32	RW	0000_0005
140	SW_MUX_CTL_PAD_SD3_DATA0 SW MUX Control Register (SW_MUX_CTL_PAD_SD3_DATA0)	32	RW	0000_0005
144	SW_MUX_CTL_PAD_SD3_DATA1 SW MUX Control Register (SW_MUX_CTL_PAD_SD3_DATA1)	32	RW	0000_0005
148	SW_MUX_CTL_PAD_SD3_DATA2 SW MUX Control Register (SW_MUX_CTL_PAD_SD3_DATA2)	32	RW	0000_0005

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
14C	SW_MUX_CTL_PAD_SD3_DATA3 SW MUX Control Register (SW_MUX_CTL_PAD_SD3_DATA3)	32	RW	0000_0005
150	SW_MUX_CTL_PAD_SD2_CD_B SW MUX Control Register (SW_MUX_CTL_PAD_SD2_CD_B)	32	RW	0000_0005
154	SW_MUX_CTL_PAD_SD2_CLK SW MUX Control Register (SW_MUX_CTL_PAD_SD2_CLK)	32	RW	0000_0005
158	SW_MUX_CTL_PAD_SD2_CMD SW MUX Control Register (SW_MUX_CTL_PAD_SD2_CMD)	32	RW	0000_0005
15C	SW_MUX_CTL_PAD_SD2_DATA0 SW MUX Control Register (SW_MUX_CTL_PAD_SD2_DATA0)	32	RW	0000_0005
160	SW_MUX_CTL_PAD_SD2_DATA1 SW MUX Control Register (SW_MUX_CTL_PAD_SD2_DATA1)	32	RW	0000_0005
164	SW_MUX_CTL_PAD_SD2_DATA2 SW MUX Control Register (SW_MUX_CTL_PAD_SD2_DATA2)	32	RW	0000_0005
168	SW_MUX_CTL_PAD_SD2_DATA3 SW MUX Control Register (SW_MUX_CTL_PAD_SD2_DATA3)	32	RW	0000_0005
16C	SW_MUX_CTL_PAD_SD2_RESET_B SW MUX Control Register (SW_MUX_CTL_PAD_SD2_RESET_B)	32	RW	0000_0005
170	SW_MUX_CTL_PAD_I2C1_SCL SW MUX Control Register (SW_MUX_CTL_PAD_I2C1_SCL)	32	RW	0000_0005
174	SW_MUX_CTL_PAD_I2C1_SDA SW MUX Control Register (SW_MUX_CTL_PAD_I2C1_SDA)	32	RW	0000_0005
178	SW_MUX_CTL_PAD_I2C2_SCL SW MUX Control Register (SW_MUX_CTL_PAD_I2C2_SCL)	32	RW	0000_0005
17C	SW_MUX_CTL_PAD_I2C2_SDA SW MUX Control Register (SW_MUX_CTL_PAD_I2C2_SDA)	32	RW	0000_0005
180	SW_MUX_CTL_PAD_UART1_RXD SW MUX Control Register (SW_MUX_CTL_PAD_UART1_RXD)	32	RW	0000_0005
184	SW_MUX_CTL_PAD_UART1_TXD SW MUX Control Register (SW_MUX_CTL_PAD_UART1_TXD)	32	RW	0000_0005
188	SW_MUX_CTL_PAD_UART2_RXD SW MUX Control Register (SW_MUX_CTL_PAD_UART2_RXD)	32	RW	0000_0005
18C	SW_MUX_CTL_PAD_UART2_TXD SW MUX Control Register (SW_MUX_CTL_PAD_UART2_TXD)	32	RW	0000_0005
190	SW_MUX_CTL_PAD_PDM_CLK SW MUX Control Register (SW_MUX_CTL_PAD_PDM_CLK)	32	RW	0000_0005

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
194	SW_MUX_CTL_PAD_PDM_BIT_STREAM0 SW MUX Control Register (SW_MUX_CTL_PAD_PDM_BIT_STREAM0)	32	RW	0000_0005
198	SW_MUX_CTL_PAD_PDM_BIT_STREAM1 SW MUX Control Register (SW_MUX_CTL_PAD_PDM_BIT_STREAM1)	32	RW	0000_0005
19C	SW_MUX_CTL_PAD_SAI1_TXFS SW MUX Control Register (SW_MUX_CTL_PAD_SAI1_TXFS)	32	RW	0000_0005
1A0	SW_MUX_CTL_PAD_SAI1_TXC SW MUX Control Register (SW_MUX_CTL_PAD_SAI1_TXC)	32	RW	0000_0005
1A4	SW_MUX_CTL_PAD_SAI1_TXD0 SW MUX Control Register (SW_MUX_CTL_PAD_SAI1_TXD0)	32	RW	0000_0005
1A8	SW_MUX_CTL_PAD_SAI1_RXD0 SW MUX Control Register (SW_MUX_CTL_PAD_SAI1_RXD0)	32	RW	0000_0005
1AC	SW_MUX_CTL_PAD_WDOG_ANY SW MUX Control Register (SW_MUX_CTL_PAD_WDOG_ANY)	32	RW	0000_0000
1B0	SW_PAD_CTL_PAD_DAP_TDI SW PAD Control Register (SW_PAD_CTL_PAD_DAP_TDI)	32	RW	0000_031E
1B4	SW_PAD_CTL_PAD_DAP_TMS_SWDIO SW PAD Control Register (SW_PAD_CTL_PAD_DAP_TMS_SWDIO)	32	RW	0000_031E
1B8	SW_PAD_CTL_PAD_DAP_TCLK_SWCLK SW PAD Control Register (SW_PAD_CTL_PAD_DAP_TCLK_SWCLK)	32	RW	0000_051E
1BC	SW_PAD_CTL_PAD_DAP_TDO_TRACESWO SW PAD Control Register (SW_PAD_CTL_PAD_DAP_TDO_TRACESWO)	32	RW	0000_011E
1C0	SW_PAD_CTL_PAD_GPIO_IO00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO00)	32	RW	0000_051E
1C4	SW_PAD_CTL_PAD_GPIO_IO01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO01)	32	RW	0000_051E
1C8	SW_PAD_CTL_PAD_GPIO_IO02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO02)	32	RW	0000_051E
1CC	SW_PAD_CTL_PAD_GPIO_IO03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO03)	32	RW	0000_051E
1D0	SW_PAD_CTL_PAD_GPIO_IO04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO04)	32	RW	0000_051E
1D4	SW_PAD_CTL_PAD_GPIO_IO05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO05)	32	RW	0000_051E
1D8	SW_PAD_CTL_PAD_GPIO_IO06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO06)	32	RW	0000_051E

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1DC	SW_PAD_CTL_PAD_GPIO_IO07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO07)	32	RW	0000_051E
1E0	SW_PAD_CTL_PAD_GPIO_IO08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO08)	32	RW	0000_051E
1E4	SW_PAD_CTL_PAD_GPIO_IO09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO09)	32	RW	0000_051E
1E8	SW_PAD_CTL_PAD_GPIO_IO10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO10)	32	RW	0000_051E
1EC	SW_PAD_CTL_PAD_GPIO_IO11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO11)	32	RW	0000_051E
1F0	SW_PAD_CTL_PAD_GPIO_IO12 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO12)	32	RW	0000_051E
1F4	SW_PAD_CTL_PAD_GPIO_IO13 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO13)	32	RW	0000_051E
1F8	SW_PAD_CTL_PAD_GPIO_IO14 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO14)	32	RW	0000_051E
1FC	SW_PAD_CTL_PAD_GPIO_IO15 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO15)	32	RW	0000_051E
200	SW_PAD_CTL_PAD_GPIO_IO16 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO16)	32	RW	0000_051E
204	SW_PAD_CTL_PAD_GPIO_IO17 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO17)	32	RW	0000_051E
208	SW_PAD_CTL_PAD_GPIO_IO18 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO18)	32	RW	0000_051E
20C	SW_PAD_CTL_PAD_GPIO_IO19 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO19)	32	RW	0000_051E
210	SW_PAD_CTL_PAD_GPIO_IO20 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO20)	32	RW	0000_051E
214	SW_PAD_CTL_PAD_GPIO_IO21 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO21)	32	RW	0000_051E
218	SW_PAD_CTL_PAD_GPIO_IO22 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO22)	32	RW	0000_051E
21C	SW_PAD_CTL_PAD_GPIO_IO23 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO23)	32	RW	0000_051E
220	SW_PAD_CTL_PAD_GPIO_IO24 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO24)	32	RW	0000_051E

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
224	SW_PAD_CTL_PAD_GPIO_IO25 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO25)	32	RW	0000_051E
228	SW_PAD_CTL_PAD_GPIO_IO26 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO26)	32	RW	0000_051E
22C	SW_PAD_CTL_PAD_GPIO_IO27 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO27)	32	RW	0000_051E
230	SW_PAD_CTL_PAD_GPIO_IO28 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO28)	32	RW	0000_051E
234	SW_PAD_CTL_PAD_GPIO_IO29 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO29)	32	RW	0000_051E
238	SW_PAD_CTL_PAD_CCM_CLKO1 SW PAD Control Register (SW_PAD_CTL_PAD_CCM_CLKO1)	32	RW	0000_051E
23C	SW_PAD_CTL_PAD_CCM_CLKO2 SW PAD Control Register (SW_PAD_CTL_PAD_CCM_CLKO2)	32	RW	0000_051E
240	SW_PAD_CTL_PAD_CCM_CLKO3 SW PAD Control Register (SW_PAD_CTL_PAD_CCM_CLKO3)	32	RW	0000_051E
244	SW_PAD_CTL_PAD_CCM_CLKO4 SW PAD Control Register (SW_PAD_CTL_PAD_CCM_CLKO4)	32	RW	0000_051E
248	SW_PAD_CTL_PAD_ENET1_MDC SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_MDC)	32	RW	0000_051E
24C	SW_PAD_CTL_PAD_ENET1_MDIO SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_MDIO)	32	RW	0000_051E
250	SW_PAD_CTL_PAD_ENET1_TD3 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_TD3)	32	RW	0000_051E
254	SW_PAD_CTL_PAD_ENET1_TD2 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_TD2)	32	RW	0000_051E
258	SW_PAD_CTL_PAD_ENET1_TD1 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_TD1)	32	RW	0000_051E
25C	SW_PAD_CTL_PAD_ENET1_TD0 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_TD0)	32	RW	0000_051E
260	SW_PAD_CTL_PAD_ENET1_TX_CTL SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_TX_CTL)	32	RW	0000_051E
264	SW_PAD_CTL_PAD_ENET1_TXC SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_TXC)	32	RW	0000_051E
268	SW_PAD_CTL_PAD_ENET1_RX_CTL SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_RX_CTL)	32	RW	0000_051E

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
26C	SW_PAD_CTL_PAD_ENET1_RXC SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_RXC)	32	RW	0000_051E
270	SW_PAD_CTL_PAD_ENET1_RD0 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_RD0)	32	RW	0000_051E
274	SW_PAD_CTL_PAD_ENET1_RD1 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_RD1)	32	RW	0000_051E
278	SW_PAD_CTL_PAD_ENET1_RD2 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_RD2)	32	RW	0000_051E
27C	SW_PAD_CTL_PAD_ENET1_RD3 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_RD3)	32	RW	0000_051E
280	SW_PAD_CTL_PAD_ENET2_MDC SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_MDC)	32	RW	0000_051E
284	SW_PAD_CTL_PAD_ENET2_MDIO SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_MDIO)	32	RW	0000_051E
288	SW_PAD_CTL_PAD_ENET2_TD3 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_TD3)	32	RW	0000_051E
28C	SW_PAD_CTL_PAD_ENET2_TD2 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_TD2)	32	RW	0000_051E
290	SW_PAD_CTL_PAD_ENET2_TD1 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_TD1)	32	RW	0000_051E
294	SW_PAD_CTL_PAD_ENET2_TD0 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_TD0)	32	RW	0000_051E
298	SW_PAD_CTL_PAD_ENET2_TX_CTL SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_TX_CTL)	32	RW	0000_051E
29C	SW_PAD_CTL_PAD_ENET2_TXC SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_TXC)	32	RW	0000_051E
2A0	SW_PAD_CTL_PAD_ENET2_RX_CTL SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_RX_CTL)	32	RW	0000_051E
2A4	SW_PAD_CTL_PAD_ENET2_RXC SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_RXC)	32	RW	0000_051E
2A8	SW_PAD_CTL_PAD_ENET2_RD0 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_RD0)	32	RW	0000_051E
2AC	SW_PAD_CTL_PAD_ENET2_RD1 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_RD1)	32	RW	0000_051E
2B0	SW_PAD_CTL_PAD_ENET2_RD2 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_RD2)	32	RW	0000_051E

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2B4	SW_PAD_CTL_PAD_ENET2_RD3 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_RD3)	32	RW	0000_051E
2B8	SW_PAD_CTL_PAD_SD1_CLK SW PAD Control Register (SW_PAD_CTL_PAD_SD1_CLK)	32	RW	0000_051E
2BC	SW_PAD_CTL_PAD_SD1_CMD SW PAD Control Register (SW_PAD_CTL_PAD_SD1_CMD)	32	RW	0000_051E
2C0	SW_PAD_CTL_PAD_SD1_DATA0 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA0)	32	RW	0000_051E
2C4	SW_PAD_CTL_PAD_SD1_DATA1 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA1)	32	RW	0000_051E
2C8	SW_PAD_CTL_PAD_SD1_DATA2 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA2)	32	RW	0000_051E
2CC	SW_PAD_CTL_PAD_SD1_DATA3 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA3)	32	RW	0000_051E
2D0	SW_PAD_CTL_PAD_SD1_DATA4 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA4)	32	RW	0000_051E
2D4	SW_PAD_CTL_PAD_SD1_DATA5 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA5)	32	RW	0000_051E
2D8	SW_PAD_CTL_PAD_SD1_DATA6 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA6)	32	RW	0000_051E
2DC	SW_PAD_CTL_PAD_SD1_DATA7 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA7)	32	RW	0000_051E
2E0	SW_PAD_CTL_PAD_SD1_STROBE SW PAD Control Register (SW_PAD_CTL_PAD_SD1_STROBE)	32	RW	0000_051E
2E4	SW_PAD_CTL_PAD_SD2_VSELECT SW PAD Control Register (SW_PAD_CTL_PAD_SD2_VSELECT)	32	RW	0000_051E
2E8	SW_PAD_CTL_PAD_SD3_CLK SW PAD Control Register (SW_PAD_CTL_PAD_SD3_CLK)	32	RW	0000_051E
2EC	SW_PAD_CTL_PAD_SD3_CMD SW PAD Control Register (SW_PAD_CTL_PAD_SD3_CMD)	32	RW	0000_051E
2F0	SW_PAD_CTL_PAD_SD3_DATA0 SW PAD Control Register (SW_PAD_CTL_PAD_SD3_DATA0)	32	RW	0000_051E
2F4	SW_PAD_CTL_PAD_SD3_DATA1 SW PAD Control Register (SW_PAD_CTL_PAD_SD3_DATA1)	32	RW	0000_051E
2F8	SW_PAD_CTL_PAD_SD3_DATA2 SW PAD Control Register (SW_PAD_CTL_PAD_SD3_DATA2)	32	RW	0000_051E

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2FC	SW_PAD_CTL_PAD_SD3_DATA3 SW PAD Control Register (SW_PAD_CTL_PAD_SD3_DATA3)	32	RW	0000_051E
300	SW_PAD_CTL_PAD_SD2_CD_B SW PAD Control Register (SW_PAD_CTL_PAD_SD2_CD_B)	32	RW	0000_051E
304	SW_PAD_CTL_PAD_SD2_CLK SW PAD Control Register (SW_PAD_CTL_PAD_SD2_CLK)	32	RW	0000_051E
308	SW_PAD_CTL_PAD_SD2_CMD SW PAD Control Register (SW_PAD_CTL_PAD_SD2_CMD)	32	RW	0000_051E
30C	SW_PAD_CTL_PAD_SD2_DATA0 SW PAD Control Register (SW_PAD_CTL_PAD_SD2_DATA0)	32	RW	0000_051E
310	SW_PAD_CTL_PAD_SD2_DATA1 SW PAD Control Register (SW_PAD_CTL_PAD_SD2_DATA1)	32	RW	0000_051E
314	SW_PAD_CTL_PAD_SD2_DATA2 SW PAD Control Register (SW_PAD_CTL_PAD_SD2_DATA2)	32	RW	0000_051E
318	SW_PAD_CTL_PAD_SD2_DATA3 SW PAD Control Register (SW_PAD_CTL_PAD_SD2_DATA3)	32	RW	0000_051E
31C	SW_PAD_CTL_PAD_SD2_RESET_B SW PAD Control Register (SW_PAD_CTL_PAD_SD2_RESET_B)	32	RW	0000_051E
320	SW_PAD_CTL_PAD_I2C1_SCL SW PAD Control Register (SW_PAD_CTL_PAD_I2C1_SCL)	32	RW	0000_051E
324	SW_PAD_CTL_PAD_I2C1_SDA SW PAD Control Register (SW_PAD_CTL_PAD_I2C1_SDA)	32	RW	0000_051E
328	SW_PAD_CTL_PAD_I2C2_SCL SW PAD Control Register (SW_PAD_CTL_PAD_I2C2_SCL)	32	RW	0000_051E
32C	SW_PAD_CTL_PAD_I2C2_SDA SW PAD Control Register (SW_PAD_CTL_PAD_I2C2_SDA)	32	RW	0000_051E
330	SW_PAD_CTL_PAD_UART1_RXD SW PAD Control Register (SW_PAD_CTL_PAD_UART1_RXD)	32	RW	0000_051E
334	SW_PAD_CTL_PAD_UART1_TXD SW PAD Control Register (SW_PAD_CTL_PAD_UART1_TXD)	32	RW	0000_051E
338	SW_PAD_CTL_PAD_UART2_RXD SW PAD Control Register (SW_PAD_CTL_PAD_UART2_RXD)	32	RW	0000_051E
33C	SW_PAD_CTL_PAD_UART2_TXD SW PAD Control Register (SW_PAD_CTL_PAD_UART2_TXD)	32	RW	0000_051E
340	SW_PAD_CTL_PAD_PDM_CLK SW PAD Control Register (SW_PAD_CTL_PAD_PDM_CLK)	32	RW	0000_051E

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
344	SW_PAD_CTL_PAD_PDM_BIT_STREAM0 SW PAD Control Register (SW_PAD_CTL_PAD_PDM_BIT_STREAM0)	32	RW	0000_051E
348	SW_PAD_CTL_PAD_PDM_BIT_STREAM1 SW PAD Control Register (SW_PAD_CTL_PAD_PDM_BIT_STREAM1)	32	RW	0000_051E
34C	SW_PAD_CTL_PAD_SAI1_TXFS SW PAD Control Register (SW_PAD_CTL_PAD_SAI1_TXFS)	32	RW	0000_051E
350	SW_PAD_CTL_PAD_SAI1_TXC SW PAD Control Register (SW_PAD_CTL_PAD_SAI1_TXC)	32	RW	0000_051E
354	SW_PAD_CTL_PAD_SAI1_TXD0 SW PAD Control Register (SW_PAD_CTL_PAD_SAI1_TXD0)	32	RW	0000_051E
358	SW_PAD_CTL_PAD_SAI1_RXD0 SW PAD Control Register (SW_PAD_CTL_PAD_SAI1_RXD0)	32	RW	0000_051E
35C	SW_PAD_CTL_PAD_WDOG_ANY SW PAD Control Register (SW_PAD_CTL_PAD_WDOG_ANY)	32	RW	0000_031E
360	CAN1_IPP_IND_CANRX_SELECT_INPUT DAISY Register (CAN1_IPP_IND_CANRX_SELECT_INPUT)	32	RW	0000_0000
364	CAN2_IPP_IND_CANRX_SELECT_INPUT DAISY Register (CAN2_IPP_IND_CANRX_SELECT_INPUT)	32	RW	0000_0000
36C	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_0 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_0)	32	RW	0000_0000
370	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_1 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_1)	32	RW	0000_0000
374	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_2 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_2)	32	RW	0000_0000
378	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_3 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_3)	32	RW	0000_0000
37C	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_4 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_4)	32	RW	0000_0000
380	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_5 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_5)	32	RW	0000_0000
384	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_6 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_6)	32	RW	0000_0000
388	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_7 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_7)	32	RW	0000_0000
38C	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_8 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_8)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
390	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_9 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_9)	32	RW	0000_0000
394	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_10 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_10)	32	RW	0000_0000
398	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_11 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_11)	32	RW	0000_0000
39C	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_13 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_13)	32	RW	0000_0000
3A0	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_14 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_14)	32	RW	0000_0000
3A4	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_15 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_15)	32	RW	0000_0000
3A8	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_16 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_16)	32	RW	0000_0000
3AC	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_17 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_17)	32	RW	0000_0000
3B0	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_18 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_18)	32	RW	0000_0000
3B4	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_20 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_20)	32	RW	0000_0000
3B8	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_22 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_22)	32	RW	0000_0000
3BC	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_23 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_23)	32	RW	0000_0000
3C0	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_24 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_24)	32	RW	0000_0000
3C4	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_25 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_25)	32	RW	0000_0000
3C8	FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_27 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_27)	32	RW	0000_0000
3CC	I3C2_PIN_SCL_IN_SELECT_INPUT DAISY Register (I3C2_PIN_SCL_IN_SELECT_INPUT)	32	RW	0000_0000
3D0	I3C2_PIN_SDA_IN_SELECT_INPUT DAISY Register (I3C2_PIN_SDA_IN_SELECT_INPUT)	32	RW	0000_0000
3D4	JTAG_MUX_TCK_SELECT_INPUT DAISY Register (JTAG_MUX_TCK_SELECT_INPUT)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
3D8	JTAG_MUX_TDI_SELECT_INPUT DAISY Register (JTAG_MUX_TDI_SELECT_INPUT)	32	RW	0000_0000
3DC	JTAG_MUX_TMS_SELECT_INPUT DAISY Register (JTAG_MUX_TMS_SELECT_INPUT)	32	RW	0000_0000
3E0	LPI2C3_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C3_IPP_IND_LPI2C_SCL_SELECT_INPUT)	32	RW	0000_0000
3E4	LPI2C3_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C3_IPP_IND_LPI2C_SDA_SELECT_INPUT)	32	RW	0000_0000
3E8	LPI2C5_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C5_IPP_IND_LPI2C_SCL_SELECT_INPUT)	32	RW	0000_0000
3EC	LPI2C5_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C5_IPP_IND_LPI2C_SDA_SELECT_INPUT)	32	RW	0000_0000
3F0	LPI2C6_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C6_IPP_IND_LPI2C_SCL_SELECT_INPUT)	32	RW	0000_0000
3F4	LPI2C6_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C6_IPP_IND_LPI2C_SDA_SELECT_INPUT)	32	RW	0000_0000
3F8	LPI2C7_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C7_IPP_IND_LPI2C_SCL_SELECT_INPUT)	32	RW	0000_0000
3FC	LPI2C7_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C7_IPP_IND_LPI2C_SDA_SELECT_INPUT)	32	RW	0000_0000
400	LPI2C8_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C8_IPP_IND_LPI2C_SCL_SELECT_INPUT)	32	RW	0000_0000
404	LPI2C8_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C8_IPP_IND_LPI2C_SDA_SELECT_INPUT)	32	RW	0000_0000
408	LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_0 DAISY Register (LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_0)	32	RW	0000_0000
40C	LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_1 DAISY Register (LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_1)	32	RW	0000_0000
410	LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_2 DAISY Register (LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_2)	32	RW	0000_0000
414	LPUART3_IPP_IND_LPUART_CTS_N_SELECT_INPUT DAISY Register (LPUART3_IPP_IND_LPUART_CTS_N_SELECT_INPUT)	32	RW	0000_0000
418	LPUART3_IPP_IND_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART3_IPP_IND_LPUART_RXD_SELECT_INPUT)	32	RW	0000_0000
41C	LPUART3_IPP_IND_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART3_IPP_IND_LPUART_TXD_SELECT_INPUT)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
420	LPUART4_IPP_IND_LPUART_CTS_N_SELECT_INPUT DAISY Register (LPUART4_IPP_IND_LPUART_CTS_N_SELECT_INPUT)	32	RW	0000_0000
424	LPUART4_IPP_IND_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART4_IPP_IND_LPUART_RXD_SELECT_INPUT)	32	RW	0000_0000
428	LPUART4_IPP_IND_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART4_IPP_IND_LPUART_TXD_SELECT_INPUT)	32	RW	0000_0000
42C	LPUART5_IPP_IND_LPUART_CTS_N_SELECT_INPUT DAISY Register (LPUART5_IPP_IND_LPUART_CTS_N_SELECT_INPUT)	32	RW	0000_0000
430	LPUART5_IPP_IND_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART5_IPP_IND_LPUART_RXD_SELECT_INPUT)	32	RW	0000_0000
434	LPUART5_IPP_IND_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART5_IPP_IND_LPUART_TXD_SELECT_INPUT)	32	RW	0000_0000
438	PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_0 DAISY Register (PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_0)	32	RW	0000_0000
43C	PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_1 DAISY Register (PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_1)	32	RW	0000_0000
440	PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_2 DAISY Register (PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_2)	32	RW	0000_0000
444	PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_3 DAISY Register (PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_3)	32	RW	0000_0000
448	SAI1_IPP_IND_SAI_MCLK_SELECT_INPUT DAISY Register (SAI1_IPP_IND_SAI_MCLK_SELECT_INPUT)	32	RW	0000_0000
44C	SAI3_IPP_IND_SAI_RXBCLK_SELECT_INPUT DAISY Register (SAI3_IPP_IND_SAI_RXBCLK_SELECT_INPUT)	32	RW	0000_0000
450	SAI3_IPP_IND_SAI_RXSYNC_SELECT_INPUT DAISY Register (SAI3_IPP_IND_SAI_RXSYNC_SELECT_INPUT)	32	RW	0000_0000
454	SPDIF_SPDIF_I_SELECT_INPUT DAISY Register (SPDIF_SPDIF_I_SELECT_INPUT)	32	RW	0000_0000
458	USDHC3_IPP_CARD_CLK_IN_SELECT_INPUT DAISY Register (USDHC3_IPP_CARD_CLK_IN_SELECT_INPUT)	32	RW	0000_0000
45C	USDHC3_IPP_CMD_IN_SELECT_INPUT DAISY Register (USDHC3_IPP_CMD_IN_SELECT_INPUT)	32	RW	0000_0000
460	USDHC3_IPP_DAT0_IN_SELECT_INPUT DAISY Register (USDHC3_IPP_DAT0_IN_SELECT_INPUT)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
464	USDHC3_IPP_DAT1_IN_SELECT_INPUT DAISY Register (USDHC3_IPP_DAT1_IN_SELECT_INPUT)	32	RW	0000_0000
468	USDHC3_IPP_DAT2_IN_SELECT_INPUT DAISY Register (USDHC3_IPP_DAT2_IN_SELECT_INPUT)	32	RW	0000_0000
46C	USDHC3_IPP_DAT3_IN_SELECT_INPUT DAISY Register (USDHC3_IPP_DAT3_IN_SELECT_INPUT)	32	RW	0000_0000

27.3.1.2 SW_MUX_CTL_PAD_DAP_TDI SW MUX Control Register (SW_MUX_CTL_PAD_DAP_TDI)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_DAP_TDI	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Force input path of pad DAP_TDI
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DAP_TDI. 000 - Select mux mode: ALT0 mux port: JTAG_MUX_TDI of instance: jtag_mux 001 - Select mux mode: ALT1 mux port: MQS2_LEFT of instance: mqs2 011 - Select mux mode: ALT3 mux port: CAN2_TX of instance: can2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO30 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO3_IO28 of instance: gpio3 110 - Select mux mode: ALT6 mux port: LPUART5_RX of instance: lpuart5

27.3.1.3 SW_MUX_CTL_PAD_DAP_TMS_SWDIO SW MUX Control Register (SW_MUX_CTL_PAD_DAP_TMS_SWDIO)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_DAP_TMS_SWDIO	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad DAP_TMS_SWDIO
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DAP_TMS_SWDIO. 000 - Select mux mode: ALT0 mux port: JTAG_MUX_TMS of instance: jtag_mux 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO31 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO3_IO29 of instance: gpio3 110 - Select mux mode: ALT6 mux port: LPUART5_RTS_B of instance: lpuart5

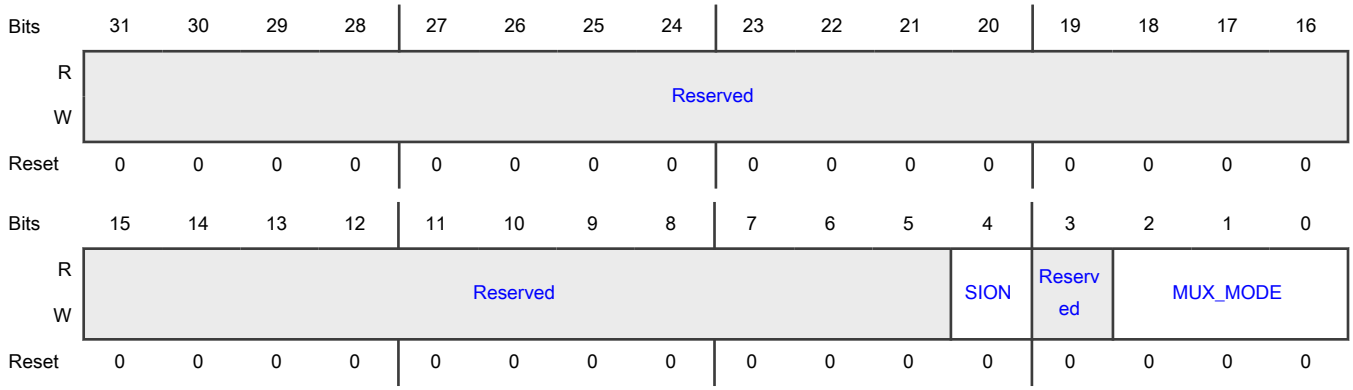
27.3.1.4 SW_MUX_CTL_PAD_DAP_TCLK_SWCLK SW MUX Control Register (SW_MUX_CTL_PAD_DAP_TCLK_SWCLK)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_DAP_TCLK_SWCLK	8h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad DAP_TCLK_SWCLK
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DAP_TCLK_SWCLK. 000 - Select mux mode: ALT0 mux port: JTAG_MUX_TCK of instance: jtag_mux 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO30 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO30 of instance: gpio3 110 - Select mux mode: ALT6 mux port: LPUART5_CTS_B of instance: lpuart5

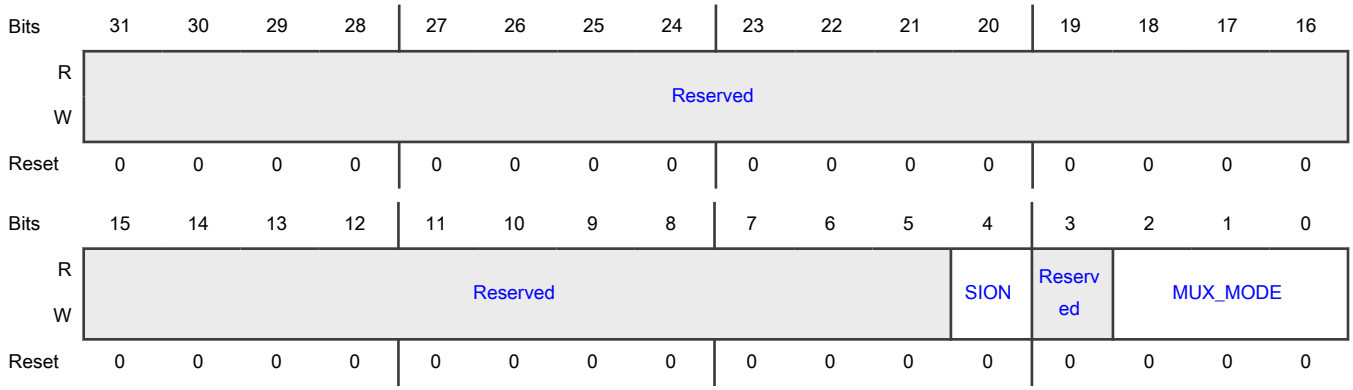
27.3.1.5 SW_MUX_CTL_PAD_DAP_TDO_TRACESWO SW MUX Control Register (SW_MUX_CTL_PAD_DAP_TDO_TRACESWO)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_DAP_TDO_TRACESWO	Ch

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad DAP_TDO_TRACESWO
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DAP_TDO_TRACESWO. 000 - Select mux mode: ALT0 mux port: JTAG_MUX_TDO of instance: jtag_mux 001 - Select mux mode: ALT1 mux port: MQS2_RIGHT of instance: mqs2 011 - Select mux mode: ALT3 mux port: CAN2_RX of instance: can2 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO31 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO31 of instance: gpio3 110 - Select mux mode: ALT6 mux port: LPUART5_TX of instance: lpuart5

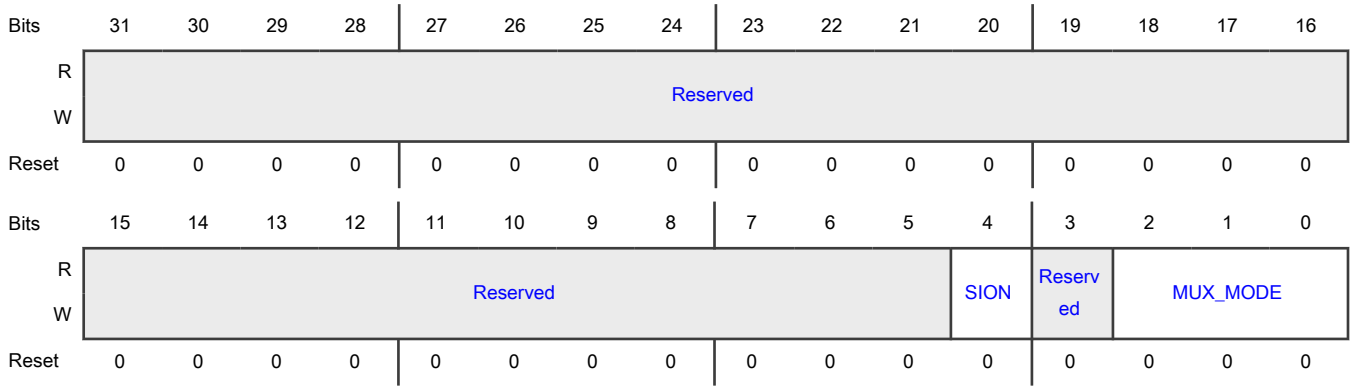
27.3.1.6 SW_MUX_CTL_PAD_GPIO_IO00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO00)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO00	10h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO00
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO00. 000 - Select mux mode: ALT0 mux port: GPIO2_IO00 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPI2C3_SDA of instance: lpi2c3 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_CLK of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_CLK of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI6_PCS0 of instance: lpspi6 101 - Select mux mode: ALT5 mux port: LPUART5_TX of instance: lpuart5 110 - Select mux mode: ALT6 mux port: LPI2C5_SDA of instance: lpi2c5 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO00 of instance: flexio1

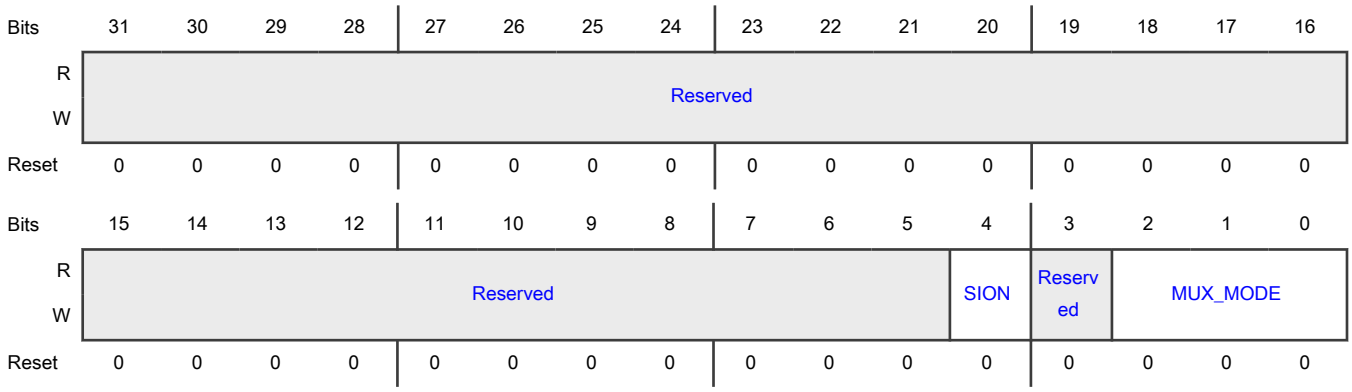
27.3.1.7 SW_MUX_CTL_PAD_GPIO_IO01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO01)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO01	14h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO01
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO01. 000 - Select mux mode: ALT0 mux port: GPIO2_IO01 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPI2C3_SCL of instance: lpi2c3 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_DATA00 of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DE of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI6_SIN of instance: lpspi6 101 - Select mux mode: ALT5 mux port: LPUART5_RX of instance: lpuart5 110 - Select mux mode: ALT6 mux port: LPI2C5_SCL of instance: lpi2c5 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO01 of instance: flexio1

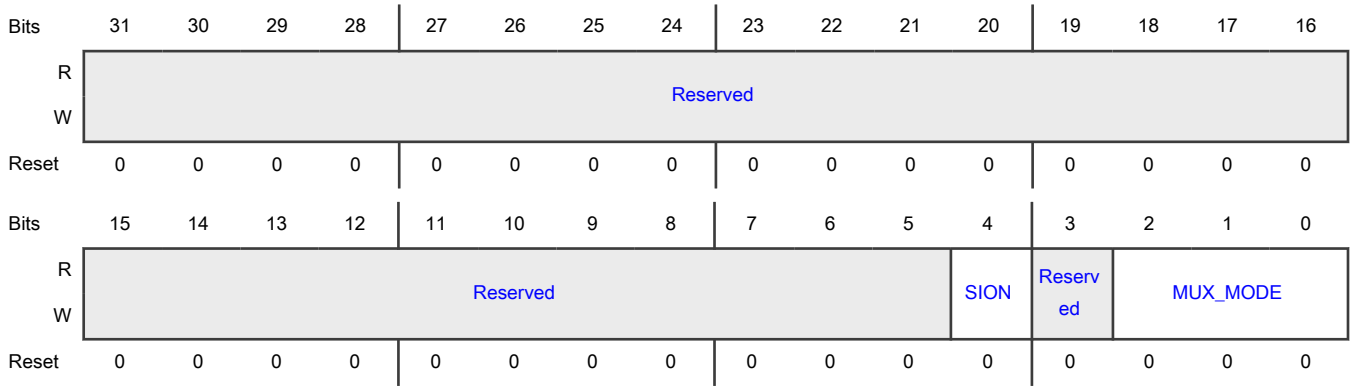
27.3.1.8 SW_MUX_CTL_PAD_GPIO_IO02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO02)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO02	18h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO02
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO02. 000 - Select mux mode: ALT0 mux port: GPIO2_IO02 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPI2C4_SDA of instance: lpi2c4 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_VSYNC of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_VSYNC of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI6_SOUT of instance: lpspi6

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - Select mux mode: ALT5 mux port: LPUART5_CTS_B of instance: lpuart5 110 - Select mux mode: ALT6 mux port: LPI2C6_SDA of instance: lpi2c6 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO02 of instance: flexio1

27.3.1.9 SW_MUX_CTL_PAD_GPIO_IO03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO03)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO03	1Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO03
3 —	- Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO03. 000 - Select mux mode: ALT0 mux port: GPIO2_IO03 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPI2C4_SCL of instance: lpi2c4 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_HSYNC of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_HSYNC of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI6_SCK of instance: lpspi6 101 - Select mux mode: ALT5 mux port: LPUART5_RTS_B of instance: lpuart5 110 - Select mux mode: ALT6 mux port: LPI2C6_SCL of instance: lpi2c6 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO03 of instance: flexio1

27.3.1.10 SW_MUX_CTL_PAD_GPIO_IO04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO04)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO04	20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5	-

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO04
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO04. 000 - Select mux mode: ALT0 mux port: GPIO2_IO04 of instance: gpio2 001 - Select mux mode: ALT1 mux port: TPM3_CH0 of instance: tpm3 010 - Select mux mode: ALT2 mux port: PDM_CLK of instance: pdm 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA00 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI7_PCS0 of instance: lpspi7 101 - Select mux mode: ALT5 mux port: LPUART6_TX of instance: lpuart6 110 - Select mux mode: ALT6 mux port: LPI2C6_SDA of instance: lpi2c6 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO04 of instance: flexio1

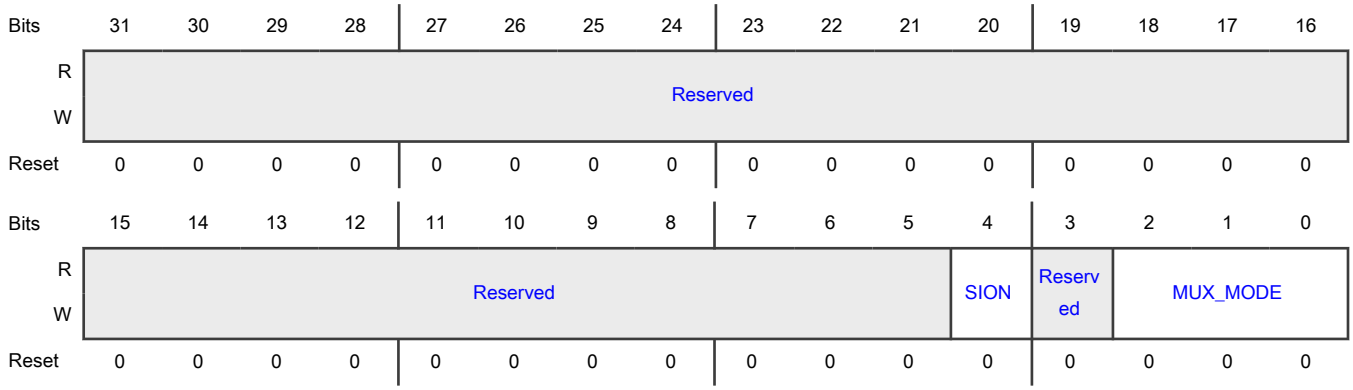
27.3.1.11 SW_MUX_CTL_PAD_GPIO_IO05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO05)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO05	24h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO05
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO05. 000 - Select mux mode: ALT0 mux port: GPIO2_IO05 of instance: gpio2 001 - Select mux mode: ALT1 mux port: TPM4_CH0 of instance: tpm4 010 - Select mux mode: ALT2 mux port: PDM_BIT_STREAM00 of instance: pdm 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA01 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI7_SIN of instance: lpspi7 101 - Select mux mode: ALT5 mux port: LPUART6_RX of instance: lpuart6 110 - Select mux mode: ALT6 mux port: LPI2C6_SCL of instance: lpi2c6 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO05 of instance: flexio1

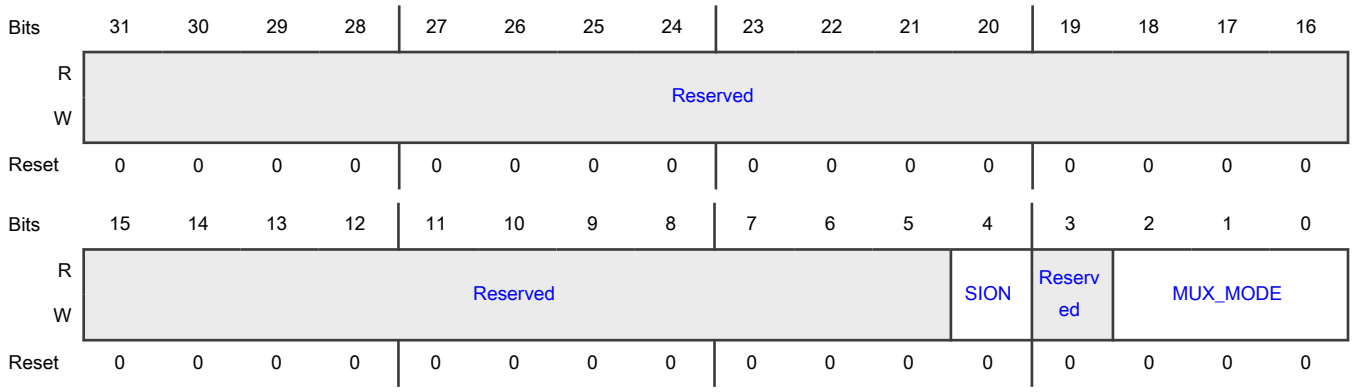
27.3.1.12 SW_MUX_CTL_PAD_GPIO_IO06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO06)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO06	28h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO06
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO06. 000 - Select mux mode: ALT0 mux port: GPIO2_IO06 of instance: gpio2 001 - Select mux mode: ALT1 mux port: TPM5_CH0 of instance: tpm5 010 - Select mux mode: ALT2 mux port: PDM_BIT_STREAM01 of instance: pdm 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA02 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI7_SOUT of instance: lpspi7 101 - Select mux mode: ALT5 mux port: LPUART6_CTS_B of instance: lpuart6 110 - Select mux mode: ALT6 mux port: LPI2C7_SDA of instance: lpi2c7 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO06 of instance: flexio1

27.3.1.13 SW_MUX_CTL_PAD_GPIO_IO07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO07)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO07	2Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO07
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO07. 000 - Select mux mode: ALT0 mux port: GPIO2_IO07 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPSPI3_PCS1 of instance: lpspi3 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_DATA01 of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA03 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI7_SCK of instance: lpspi7

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - Select mux mode: ALT5 mux port: LPUART6_RTS_B of instance: lpuart6 110 - Select mux mode: ALT6 mux port: LPI2C7_SCL of instance: lpi2c7 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO07 of instance: flexio1

27.3.1.14 SW_MUX_CTL_PAD_GPIO_IO08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO08)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO08	30h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO08
3 —	- Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO08. 000 - Select mux mode: ALT0 mux port: GPIO2_IO08 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPSPI3_PCS0 of instance: lpspi3 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_DATA02 of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA04 of instance: mediamix 100 - Select mux mode: ALT4 mux port: TPM6_CH0 of instance: tpm6 101 - Select mux mode: ALT5 mux port: LPUART7_TX of instance: lpuart7 110 - Select mux mode: ALT6 mux port: LPI2C7_SDA of instance: lpi2c7 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO08 of instance: flexio1

27.3.1.15 SW_MUX_CTL_PAD_GPIO_IO09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO09)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO09	34h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5	-

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO09
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO09. 000 - Select mux mode: ALT0 mux port: GPIO2_IO09 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPSPI3_SIN of instance: lpspi3 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_DATA03 of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA05 of instance: mediamix 100 - Select mux mode: ALT4 mux port: TPM3_EXTCLK of instance: tpm3 101 - Select mux mode: ALT5 mux port: LPUART7_RX of instance: lpuart7 110 - Select mux mode: ALT6 mux port: LPI2C7_SCL of instance: lpi2c7 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO09 of instance: flexio1

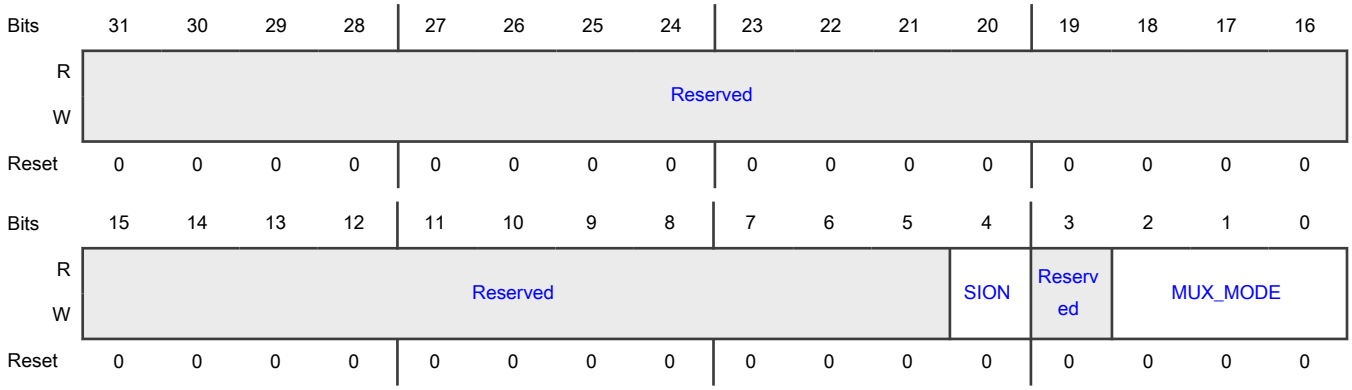
27.3.1.16 SW_MUX_CTL_PAD_GPIO_IO10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO10)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO10	38h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO10
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO10. 000 - Select mux mode: ALT0 mux port: GPIO2_IO10 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPSPI3_SOUT of instance: lpspi3 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_DATA04 of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA06 of instance: mediamix 100 - Select mux mode: ALT4 mux port: TPM4_EXTCLK of instance: tpm4 101 - Select mux mode: ALT5 mux port: LPUART7_CTS_B of instance: lpuart7 110 - Select mux mode: ALT6 mux port: LPI2C8_SDA of instance: lpi2c8 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO10 of instance: flexio1

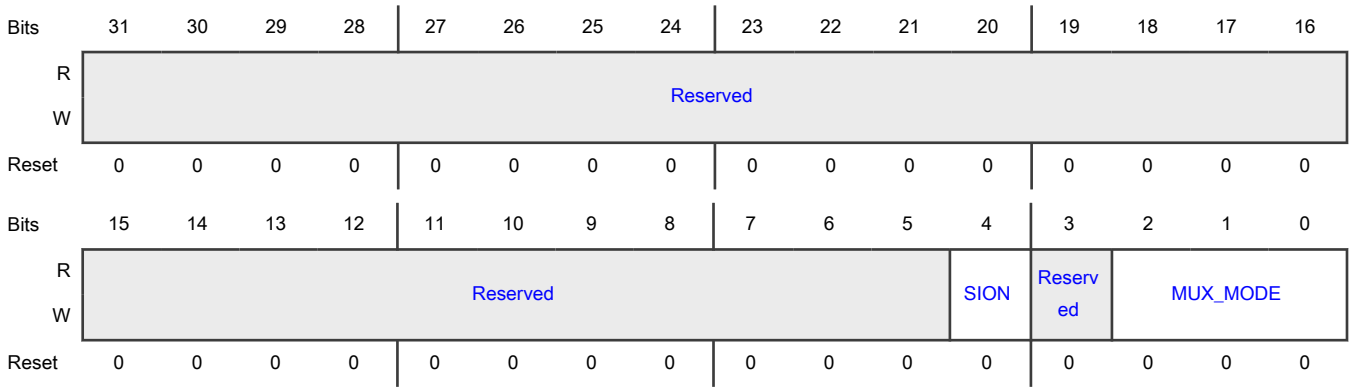
27.3.1.17 SW_MUX_CTL_PAD_GPIO_IO11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO11)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO11	3Ch

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO11
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO11. 000 - Select mux mode: ALT0 mux port: GPIO2_IO11 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPSPI3_SCK of instance: lpspi3 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_DATA05 of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA07 of instance: mediamix 100 - Select mux mode: ALT4 mux port: TPM5_EXTCLK of instance: tpm5 101 - Select mux mode: ALT5 mux port: LPUART7_RTS_B of instance: lpuart7 110 - Select mux mode: ALT6 mux port: LPI2C8_SCL of instance: lpi2c8 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO11 of instance: flexio1

27.3.1.18 SW_MUX_CTL_PAD_GPIO_IO12 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO12)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO12	40h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO12
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO12. 000 - Select mux mode: ALT0 mux port: GPIO2_IO12 of instance: gpio2 001 - Select mux mode: ALT1 mux port: TPM3_CH2 of instance: tpm3 010 - Select mux mode: ALT2 mux port: PDM_BIT_STREAM02 of instance: pdm 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA08 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI8_PCS0 of instance: lpspi8

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - Select mux mode: ALT5 mux port: LPUART8_TX of instance: lpuart8 110 - Select mux mode: ALT6 mux port: LPI2C8_SDA of instance: lpi2c8 111 - Select mux mode: ALT7 mux port: SAI3_RX_SYNC of instance: sai3

27.3.1.19 SW_MUX_CTL_PAD_GPIO_IO13 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO13)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO13	44h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO13
3 —	- Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO13. 000 - Select mux mode: ALT0 mux port: GPIO2_IO13 of instance: gpio2 001 - Select mux mode: ALT1 mux port: TPM4_CH2 of instance: tpm4 010 - Select mux mode: ALT2 mux port: PDM_BIT_STREAM03 of instance: pdm 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA09 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI8_SIN of instance: lpspi8 101 - Select mux mode: ALT5 mux port: LPUART8_RX of instance: lpuart8 110 - Select mux mode: ALT6 mux port: LPI2C8_SCL of instance: lpi2c8 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO13 of instance: flexio1

27.3.1.20 SW_MUX_CTL_PAD_GPIO_IO14 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO14)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO14	48h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5	-

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO14
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO14. 000 - Select mux mode: ALT0 mux port: GPIO2_IO14 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPUART3_TX of instance: lpuart3 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_DATA06 of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA10 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI8_SOUT of instance: lpspi8 101 - Select mux mode: ALT5 mux port: LPUART8_CTS_B of instance: lpuart8 110 - Select mux mode: ALT6 mux port: LPUART4_TX of instance: lpuart4 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO14 of instance: flexio1

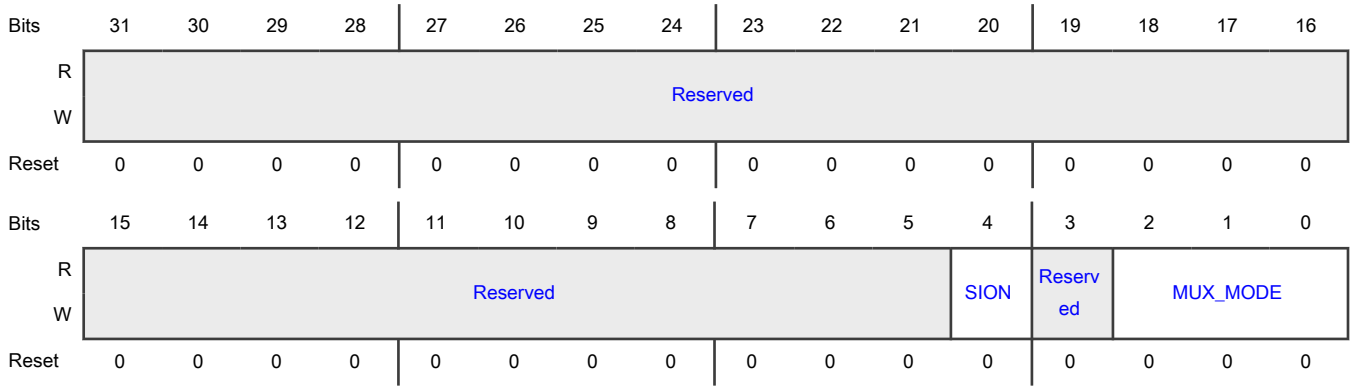
27.3.1.21 SW_MUX_CTL_PAD_GPIO_IO15 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO15)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO15	4Ch

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO15
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO15. 000 - Select mux mode: ALT0 mux port: GPIO2_IO15 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPUART3_RX of instance: lpuart3 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_DATA07 of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA11 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI8_SCK of instance: lpspi8 101 - Select mux mode: ALT5 mux port: LPUART8_RTS_B of instance: lpuart8 110 - Select mux mode: ALT6 mux port: LPUART4_RX of instance: lpuart4 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO15 of instance: flexio1

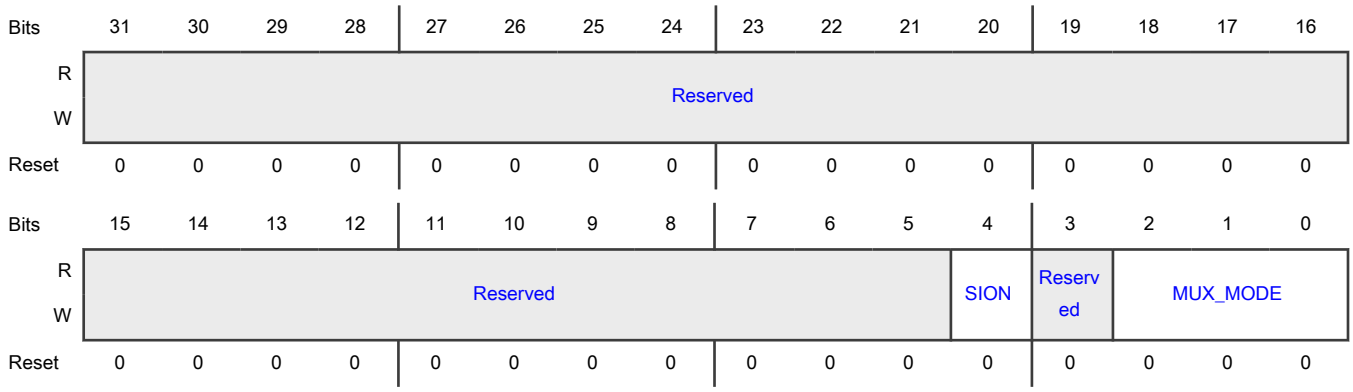
27.3.1.22 SW_MUX_CTL_PAD_GPIO_IO16 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO16)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO16	50h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO16
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO16. 000 - Select mux mode: ALT0 mux port: GPIO2_IO16 of instance: gpio2 001 - Select mux mode: ALT1 mux port: SAI3_TX_BCLK of instance: sai3 010 - Select mux mode: ALT2 mux port: PDM_BIT_STREAM02 of instance: pdm 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA12 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPUART3_CTS_B of instance: lpuart3 101 - Select mux mode: ALT5 mux port: LPSPI4_PCS2 of instance: lpspi4 110 - Select mux mode: ALT6 mux port: LPUART4_CTS_B of instance: lpuart4 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO16 of instance: flexio1

27.3.1.23 SW_MUX_CTL_PAD_GPIO_IO17 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO17)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO17	54h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO17
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO17. 000 - Select mux mode: ALT0 mux port: GPIO2_IO17 of instance: gpio2 001 - Select mux mode: ALT1 mux port: SAI3_MCLK of instance: sai3 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_DATA08 of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA13 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPUART3_RTS_B of instance: lpuart3

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - Select mux mode: ALT5 mux port: LPSPI4_PCS1 of instance: lpspi4 110 - Select mux mode: ALT6 mux port: LPUART4_RTS_B of instance: lpuart4 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO17 of instance: flexio1

27.3.1.24 SW_MUX_CTL_PAD_GPIO_IO18 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO18)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO18	58h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO18
3 —	- Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO18. 000 - Select mux mode: ALT0 mux port: GPIO2_IO18 of instance: gpio2 001 - Select mux mode: ALT1 mux port: SAI3_RX_BCLK of instance: sai3 010 - Select mux mode: ALT2 mux port: MEDIAMIX_CAM_DATA09 of instance: mediamix 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA14 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI5_PCS0 of instance: lpspi5 101 - Select mux mode: ALT5 mux port: LPSPI4_PCS0 of instance: lpspi4 110 - Select mux mode: ALT6 mux port: TPM5_CH2 of instance: tpm5 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO18 of instance: flexio1

27.3.1.25 SW_MUX_CTL_PAD_GPIO_IO19 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO19)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO19	5Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5	-

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO19
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO19. 000 - Select mux mode: ALT0 mux port: GPIO2_IO19 of instance: gpio2 001 - Select mux mode: ALT1 mux port: SAI3_RX_SYNC of instance: sai3 010 - Select mux mode: ALT2 mux port: PDM_BIT_STREAM03 of instance: pdm 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA15 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI5_SIN of instance: lpspi5 101 - Select mux mode: ALT5 mux port: LPSPI4_SIN of instance: lpspi4 110 - Select mux mode: ALT6 mux port: TPM6_CH2 of instance: tpm6 111 - Select mux mode: ALT7 mux port: SAI3_TX_DATA00 of instance: sai3

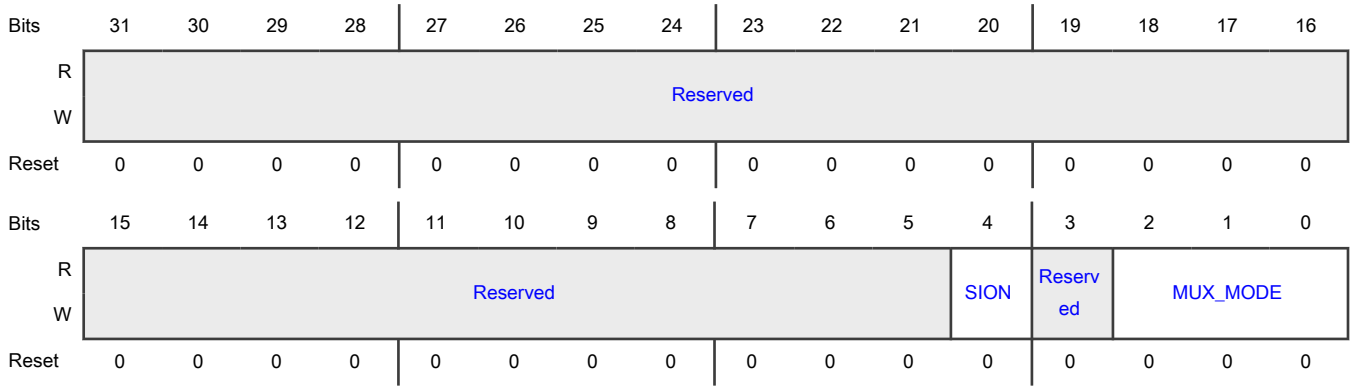
27.3.1.26 SW_MUX_CTL_PAD_GPIO_IO20 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO20)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO20	60h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO20
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO20. 000 - Select mux mode: ALT0 mux port: GPIO2_IO20 of instance: gpio2 001 - Select mux mode: ALT1 mux port: SAI3_RX_DATA00 of instance: sai3 010 - Select mux mode: ALT2 mux port: PDM_BIT_STREAM00 of instance: pdm 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA16 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI5_SOUT of instance: lpspi5 101 - Select mux mode: ALT5 mux port: LPSPI4_SOUT of instance: lpspi4 110 - Select mux mode: ALT6 mux port: TPM3_CH1 of instance: tpm3 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO20 of instance: flexio1

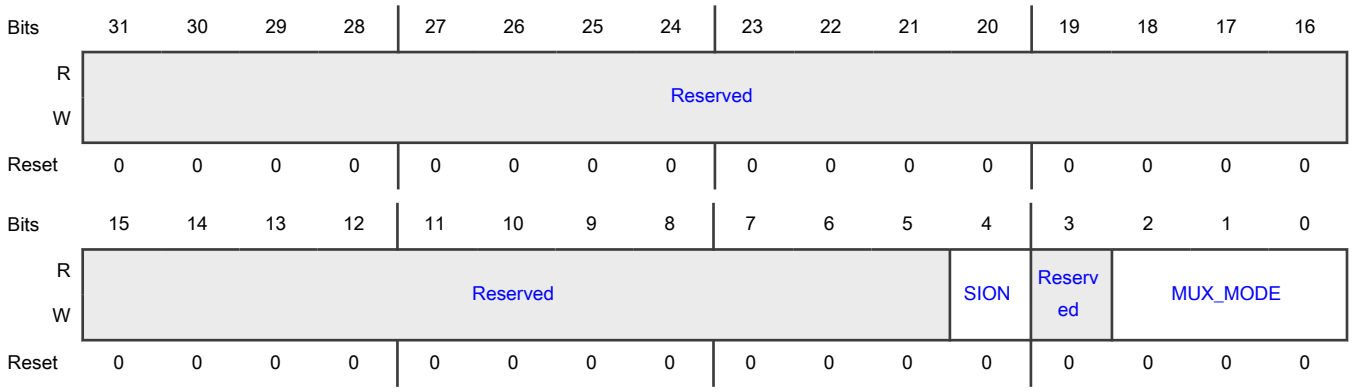
27.3.1.27 SW_MUX_CTL_PAD_GPIO_IO21 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO21)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO21	64h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO21
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO21. 000 - Select mux mode: ALT0 mux port: GPIO2_IO21 of instance: gpio2 001 - Select mux mode: ALT1 mux port: SAI3_TX_DATA00 of instance: sai3 010 - Select mux mode: ALT2 mux port: PDM_CLK of instance: pdm 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA17 of instance: mediamix 100 - Select mux mode: ALT4 mux port: LPSPI5_SCK of instance: lpspi5 101 - Select mux mode: ALT5 mux port: LPSPI4_SCK of instance: lpspi4 110 - Select mux mode: ALT6 mux port: TPM4_CH1 of instance: tpm4 111 - Select mux mode: ALT7 mux port: SAI3_RX_BCLK of instance: sai3

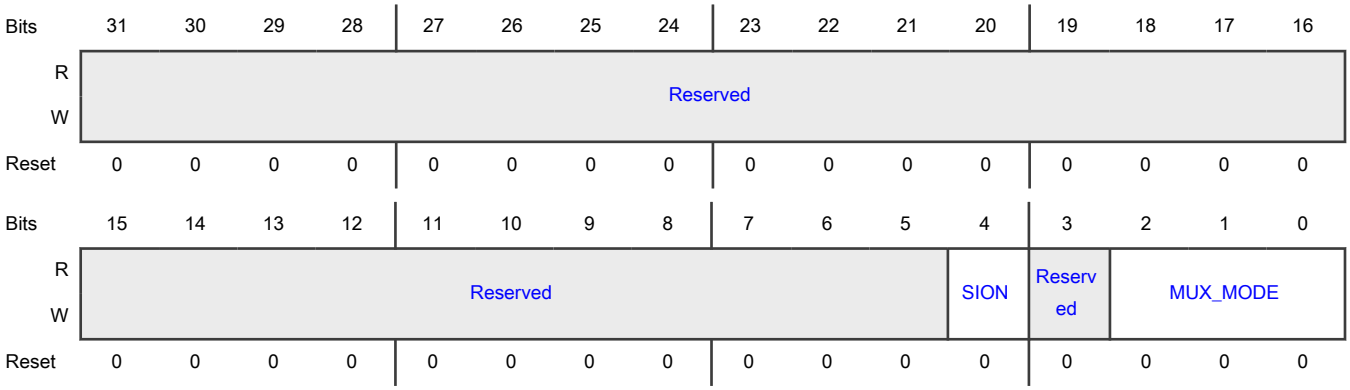
27.3.1.28 SW_MUX_CTL_PAD_GPIO_IO22 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO22)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO22	68h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO22
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO22. 000 - Select mux mode: ALT0 mux port: GPIO2_IO22 of instance: gpio2 001 - Select mux mode: ALT1 mux port: USDHC3_CLK of instance: usdhc3 010 - Select mux mode: ALT2 mux port: SPDIF_IN of instance: spdif 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA18 of instance: mediamix 100 - Select mux mode: ALT4 mux port: TPM5_CH1 of instance: tpm5

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - Select mux mode: ALT5 mux port: TPM6_EXTCLK of instance: tpm6 110 - Select mux mode: ALT6 mux port: LPI2C5_SDA of instance: lpi2c5 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO22 of instance: flexio1

27.3.1.29 SW_MUX_CTL_PAD_GPIO_IO23 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO23)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO23	6Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO23
3 —	- Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_IO23. 000 - Select mux mode: ALT0 mux port: GPIO2_IO23 of instance: gpio2 001 - Select mux mode: ALT1 mux port: USDHC3_CMD of instance: usdhc3 010 - Select mux mode: ALT2 mux port: SPDIF_OUT of instance: spdif 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA19 of instance: mediamix 100 - Select mux mode: ALT4 mux port: TPM6_CH1 of instance: tpm6 110 - Select mux mode: ALT6 mux port: LPI2C5_SCL of instance: lpi2c5 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO23 of instance: flexio1

27.3.1.30 SW_MUX_CTL_PAD_GPIO_IO24 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO24)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO24	70h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO24
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_IO24. 000 - Select mux mode: ALT0 mux port: GPIO2_IO24 of instance: gpio2 001 - Select mux mode: ALT1 mux port: USDHC3_DATA0 of instance: usdhc3 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA20 of instance: mediamix 100 - Select mux mode: ALT4 mux port: TPM3_CH3 of instance: tpm3 101 - Select mux mode: ALT5 mux port: JTAG_MUX_TDO of instance: jtag_mux 110 - Select mux mode: ALT6 mux port: LPSPI6_PCS1 of instance: lpspi6 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO24 of instance: flexio1

27.3.1.31 SW_MUX_CTL_PAD_GPIO_IO25 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO25)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO25	74h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO25
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO25. 000 - Select mux mode: ALT0 mux port: GPIO2_IO25 of instance: gpio2 001 - Select mux mode: ALT1 mux port: USDHC3_DATA1 of instance: usdhc3 010 - Select mux mode: ALT2 mux port: CAN2_TX of instance: can2 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA21 of instance: mediamix 100 - Select mux mode: ALT4 mux port: TPM4_CH3 of instance: tpm4 101 - Select mux mode: ALT5 mux port: JTAG_MUX_TCK of instance: jtag_mux 110 - Select mux mode: ALT6 mux port: LPSPI7_PCS1 of instance: lpspi7 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO25 of instance: flexio1

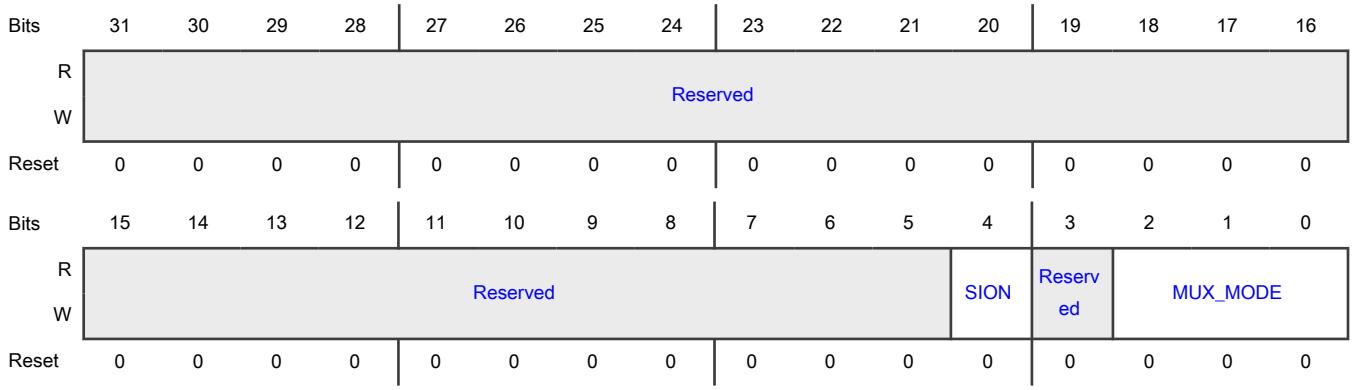
27.3.1.32 SW_MUX_CTL_PAD_GPIO_IO26 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO26)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO26	78h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO26
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO26. 000 - Select mux mode: ALT0 mux port: GPIO2_IO26 of instance: gpio2 001 - Select mux mode: ALT1 mux port: USDHC3_DATA2 of instance: usdhc3 010 - Select mux mode: ALT2 mux port: PDM_BIT_STREAM01 of instance: pdm 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA22 of instance: mediamix 100 - Select mux mode: ALT4 mux port: TPM5_CH3 of instance: tpm5 101 - Select mux mode: ALT5 mux port: JTAG_MUX_TDI of instance: jtag_mux 110 - Select mux mode: ALT6 mux port: LPSPI8_PCS1 of instance: lpspi8 111 - Select mux mode: ALT7 mux port: SAI3_TX_SYNC of instance: sai3

27.3.1.33 SW_MUX_CTL_PAD_GPIO_IO27 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO27)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO27	7Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO27
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_IO27. 000 - Select mux mode: ALT0 mux port: GPIO2_IO27 of instance: gpio2 001 - Select mux mode: ALT1 mux port: USDHC3_DATA3 of instance: usdhc3 010 - Select mux mode: ALT2 mux port: CAN2_RX of instance: can2 011 - Select mux mode: ALT3 mux port: MEDIAMIX_DISP_DATA23 of instance: mediamix 100 - Select mux mode: ALT4 mux port: TPM6_CH3 of instance: tpm6 101 - Select mux mode: ALT5 mux port: JTAG_MUX_TMS of instance: jtag_mux 110 - Select mux mode: ALT6 mux port: LPSPI5_PCS1 of instance: lpspi5 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO27 of instance: flexio1

27.3.1.34 SW_MUX_CTL_PAD_GPIO_IO28 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO28)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO28	80h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO28
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: GPIO_IO28. 000 - Select mux mode: ALT0 mux port: GPIO2_IO28 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPI2C3_SDA of instance: lpi2c3 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO28 of instance: flexio1

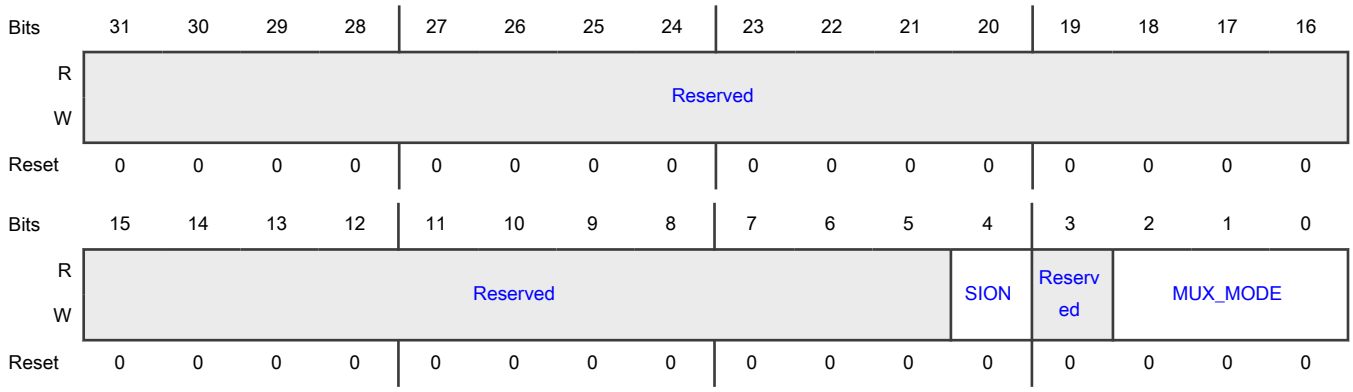
27.3.1.35 SW_MUX_CTL_PAD_GPIO_IO29 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_IO29)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_IO29	84h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad GPIO_IO29
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: GPIO_IO29. 000 - Select mux mode: ALT0 mux port: GPIO2_IO29 of instance: gpio2 001 - Select mux mode: ALT1 mux port: LPI2C3_SCL of instance: lpi2c3 111 - Select mux mode: ALT7 mux port: FLEXIO1_FLEXIO29 of instance: flexio1

27.3.1.36 SW_MUX_CTL_PAD_CCM_CLKO1 SW MUX Control Register (SW_MUX_CTL_PAD_CCM_CLKO1)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_CCM_CLKO1	88h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad CCM_CLKO1
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: CCM_CLKO1. 000 - Select mux mode: ALT0 mux port: CCMSRCGPCMIX_CLKO1 of instance: ccmsrcgpcmix 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO26 of instance: flexio1 101 - Select mux mode: GPIO3_IO26 of instance: gpio3

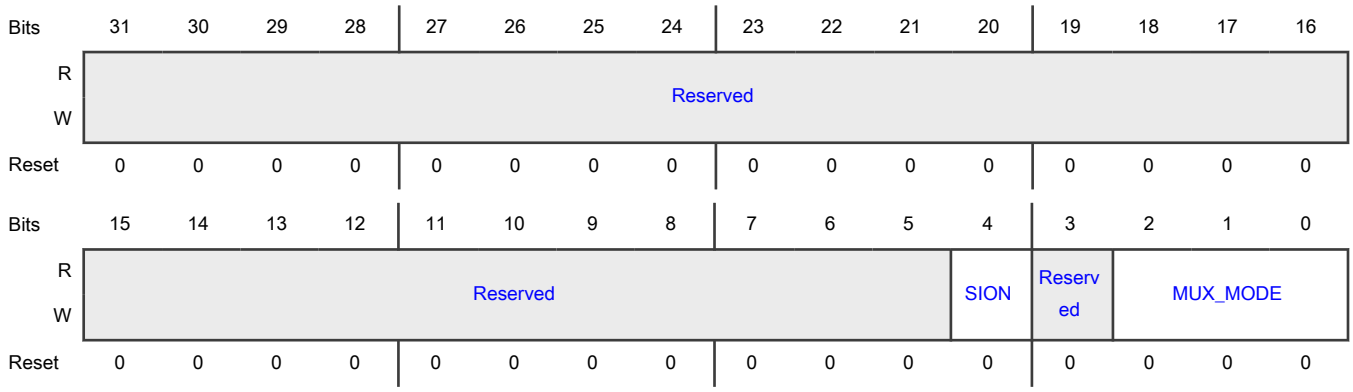
27.3.1.37 SW_MUX_CTL_PAD_CCM_CLKO2 SW MUX Control Register (SW_MUX_CTL_PAD_CCM_CLKO2)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_CCM_CLKO2	8Ch

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad CCM_CLKO2
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: CCM_CLKO2. 000 - Select mux mode: ALT0 mux port: CCMSRCGPCMIX_CLKO2 of instance: ccmsrcgpcmix 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO27 of instance: flexio1 101 - Select mux mode: GPIO3_IO27 of instance: gpio3

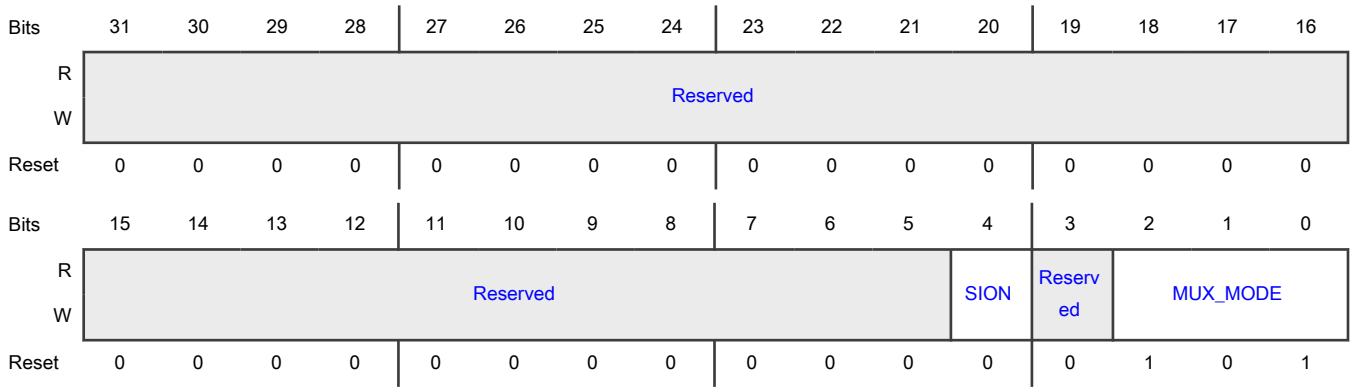
27.3.1.38 SW_MUX_CTL_PAD_CCM_CLKO3 SW MUX Control Register (SW_MUX_CTL_PAD_CCM_CLKO3)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_CCM_CLKO3	90h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad CCM_CLKO3
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: CCM_CLKO3. 000 - Select mux mode: ALT0 mux port: CCMSRCGPCMIX_CLKO3 of instance: ccmsrcgpcmix 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO28 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO28 of instance: gpio4

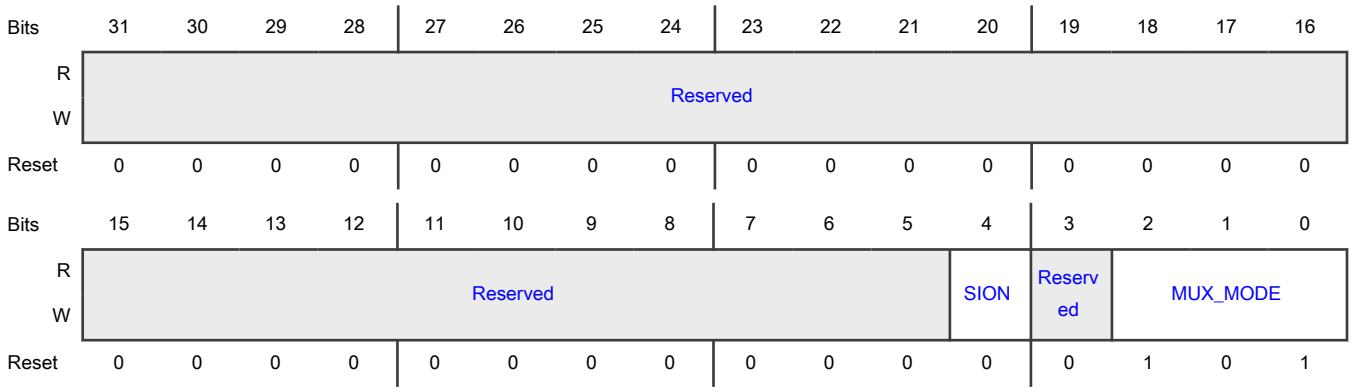
27.3.1.39 SW_MUX_CTL_PAD_CCM_CLKO4 SW MUX Control Register (SW_MUX_CTL_PAD_CCM_CLKO4)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_CCM_CLKO4	94h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad CCM_CLKO4
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: CCM_CLKO4. 000 - Select mux mode: ALT0 mux port: CCMSRCGPCMIX_CLKO4 of instance: ccmsrcgpcmix 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO29 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO29 of instance: gpio4

27.3.1.40 SW_MUX_CTL_PAD_ENET1_MDC SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_MDC)

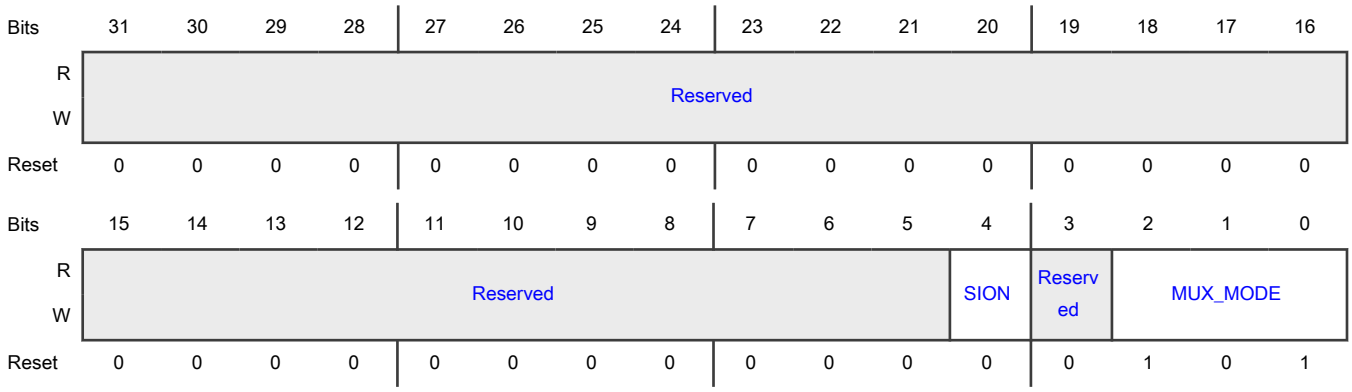
NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC in the registers below.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_MDC	98h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_MDC
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET1_MDC. 000 - Select mux mode: ALT0 mux port: ENET_QOS_MDC of instance: enet_qos 001 - Select mux mode: ALT1 mux port: LPUART3_DCD_B of instance: lpuart3 010 - Select mux mode: ALT2 mux port: I3C2_SCL of instance: i3c2 011 - Select mux mode: ALT3 mux port: HSIOMIX_OTG_ID1 of instance: hsiomix 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO00 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO00 of instance: gpio4

27.3.1.41 SW_MUX_CTL_PAD_ENET1_MDIO SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_MDIO)

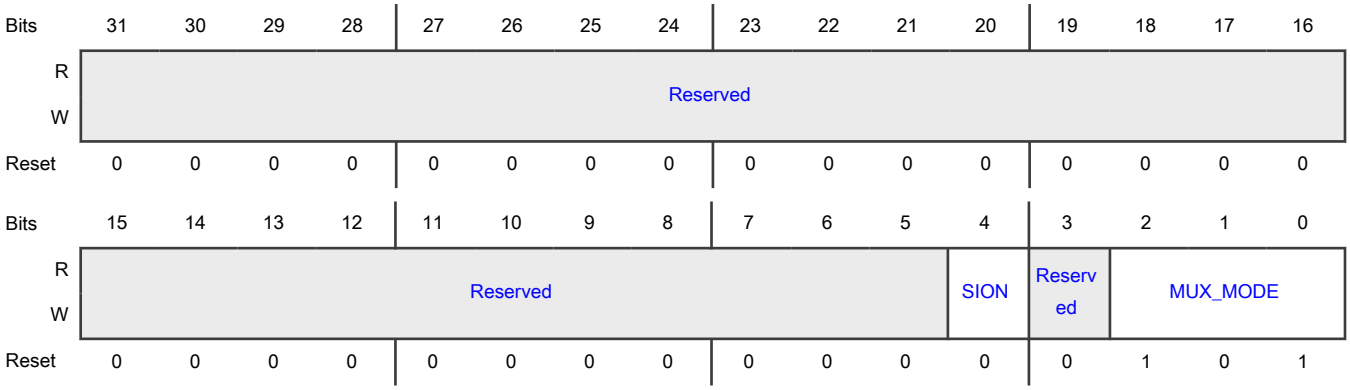
NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC in the registers below.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_MDIO	9Ch

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_MDIO
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET1_MDIO. 000 - Select mux mode: ALT0 mux port: ENET_QOS_MDIO of instance: enet_qos 001 - Select mux mode: ALT1 mux port: LPUART3_RIN_B of instance: lpuart3 010 - Select mux mode: ALT2 mux port: I3C2_SDA of instance: i3c2

Table continues on the next page...

Table continued from the previous page...

Field	Description
	011 - Select mux mode: ALT3 mux port: HSIOMIX_OTG_PWR1 of instance: hsiomix 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO01 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO01 of instance: gpio4

27.3.1.42 SW_MUX_CTL_PAD_ENET1_TD3 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_TD3)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_TD3	A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_TD3
3 —	- Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET1_TD3. 000 - Select mux mode: ALT0 mux port: ENET_QOS_RGMII_TD3 of instance: enet_qos 010 - Select mux mode: ALT2 mux port: CAN2_TX of instance: can2 011 - Select mux mode: ALT3 mux port: HSIOMIX_OTG_ID2 of instance: hsiomix 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO02 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO02 of instance: gpio4

27.3.1.43 SW_MUX_CTL_PAD_ENET1_TD2 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_TD2)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_TD2	A4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_TD2
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET1_TD2. 000 - Select mux mode: ALT0 mux port: ENET_QOS_RGMII_TD2 of instance: enet_qos 001 - Select mux mode: ALT1 mux port: CCM_ENET_QOS_CLOCK_GENERATE_REF_CLK of instance: ccm_enet_qos_clock_generate 010 - Select mux mode: ALT2 mux port: CAN2_RX of instance: can2 011 - Select mux mode: ALT3 mux port: HSIOMIX_OTG_OC2 of instance: hsiomix 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO03 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO03 of instance: gpio4

27.3.1.44 SW_MUX_CTL_PAD_ENET1_TD1 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_TD1)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_TD1	A8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_TD1
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: ENET1_TD1. 000 - Select mux mode: ALT0 mux port: ENET_QOS_RGMII_TD1 of instance: enet_qos 001 - Select mux mode: ALT1 mux port: LPUART3_RTS_B of instance: lpuart3 010 - Select mux mode: ALT2 mux port: I3C2_PUR of instance: i3c2 011 - Select mux mode: ALT3 mux port: HSIOMIX_OTG_OC1 of instance: hsiomix 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO04 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO04 of instance: gpio4 110 - Select mux mode: ALT6 mux port: I3C2_PUR_B of instance: i3c2

27.3.1.45 SW_MUX_CTL_PAD_ENET1_TD0 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_TD0)

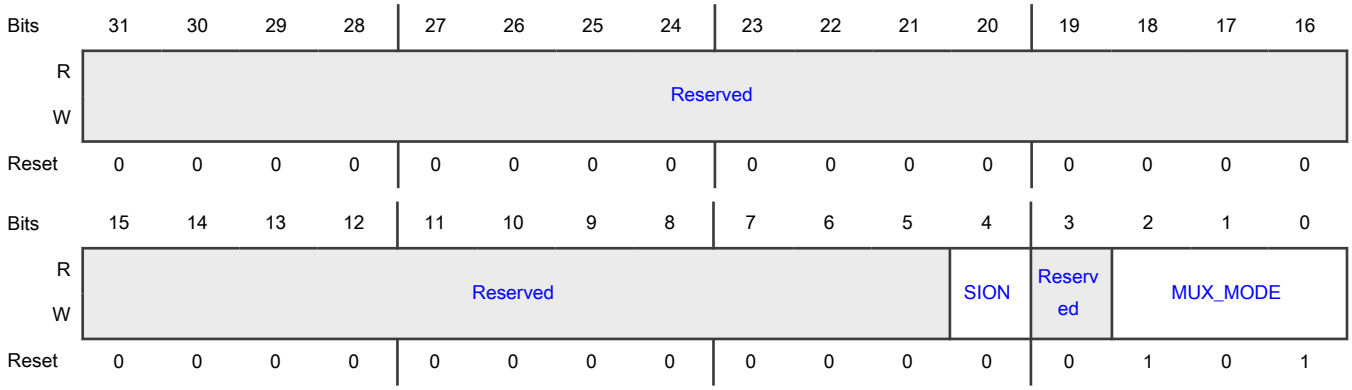
NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_TD0	ACh

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_TD0
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: ENET1_TD0. 000 - Select mux mode: ALT0 mux port: ENET_QOS_RGMII_TD0 of instance: enet_qos 001 - Select mux mode: ALT1 mux port: LPUART3_TX of instance: lpuart3 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO05 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO05 of instance: gpio4

27.3.1.46 SW_MUX_CTL_PAD_ENET1_TX_CTL SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_TX_CTL)

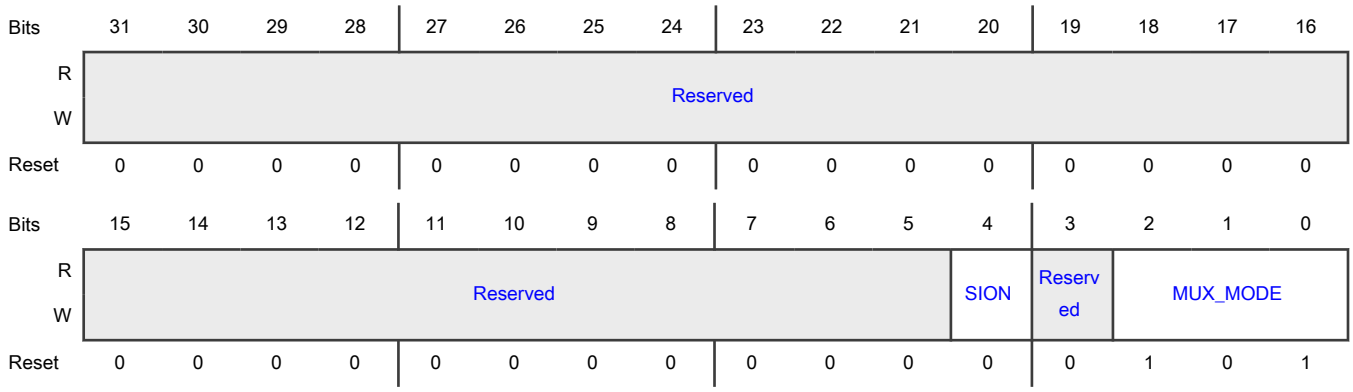
NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_TX_CTL	B0h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_TX_CTL
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: ENET1_TX_CTL. 000 - Select mux mode: ALT0 mux port: ENET_QOS_RGMII_TX_CTL of instance: enet_qos 001 - Select mux mode: ALT1 mux port: LPUART3_DTR_B of instance: lpuart3 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO06 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO06 of instance: gpio4

27.3.1.47 SW_MUX_CTL_PAD_ENET1_TXC SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_TXC)

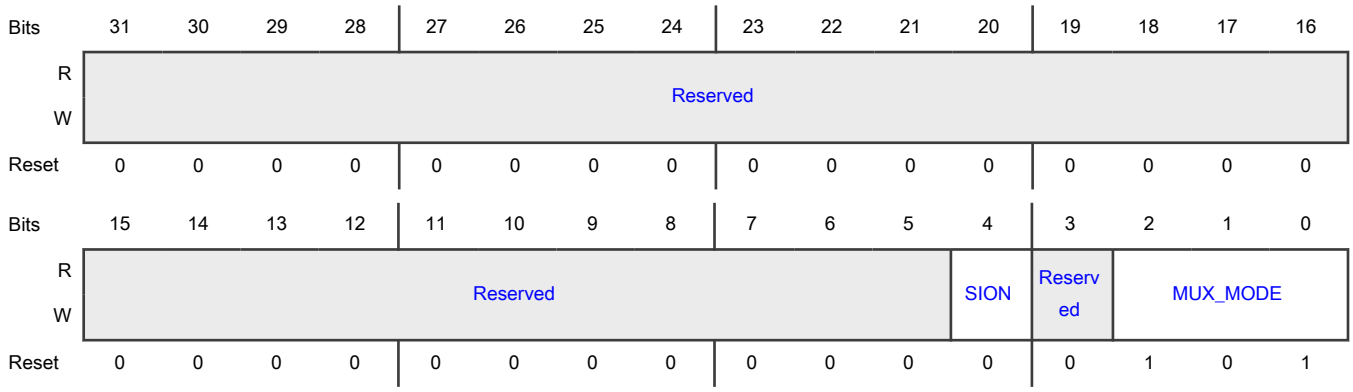
NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_TXC	B4h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_TXC
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: ENET1_TXC. 000 - Select mux mode: ALT0 mux port: CCM_ENET_QOS_CLOCK_GENERATE_TX_CLK of instance: ccm_enet_qos_clock_generate 001 - Select mux mode: ALT1 mux port: ENET_QOS_TX_ER of instance: enet_qos 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO07 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO07 of instance: gpio4

27.3.1.48 SW_MUX_CTL_PAD_ENET1_RX_CTL SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_RX_CTL)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_RX_CTL	B8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_RX_CTL
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET1_RX_CTL. 000 - Select mux mode: ALT0 mux port: ENET_QOS_RGMII_RX_CTL of instance: enet_qos 001 - Select mux mode: ALT1 mux port: LPUART3_DSR_B of instance: lpuart3 011 - Select mux mode: ALT3 mux port: HSIOMIX_OTG_PWR2 of instance: hsiomix

Table continues on the next page...

Table continued from the previous page...

Field	Description
	100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO08 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO08 of instance: gpio4

27.3.1.49 SW_MUX_CTL_PAD_ENET1_RXC SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_RXC)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_RXC	BCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_RXC
3 —	- Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: ENET1_RXC. 000 - Select mux mode: ALT0 mux port: CCM_ENET_QOS_CLOCK_GENERATE_RX_CLK of instance: ccm_enet_qos_clock_generate 001 - Select mux mode: ALT1 mux port: ENET_QOS_RX_ER of instance: enet_qos 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO09 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO09 of instance: gpio4

27.3.1.50 SW_MUX_CTL_PAD_ENET1_RD0 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_RD0)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_RD0	C0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_RD0
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: ENET1_RD0. 000 - Select mux mode: ALT0 mux port: ENET_QOS_RGMII_RD0 of instance: enet_qos 001 - Select mux mode: ALT1 mux port: LPUART3_RX of instance: lpuart3 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO10 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO10 of instance: gpio4

27.3.1.51 SW_MUX_CTL_PAD_ENET1_RD1 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_RD1)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_RD1	C4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_RD1
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET1_RD1. 000 - Select mux mode: ALT0 mux port: ENET_QOS_RGMII_RD1 of instance: enet_qos 001 - Select mux mode: ALT1 mux port: LPUART3_CTS_B of instance: lpuart3 011 - Select mux mode: ALT3 mux port: LPTMR2_ALT1 of instance: lptmr2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO11 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO11 of instance: gpio4

27.3.1.52 SW_MUX_CTL_PAD_ENET1_RD2 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_RD2)

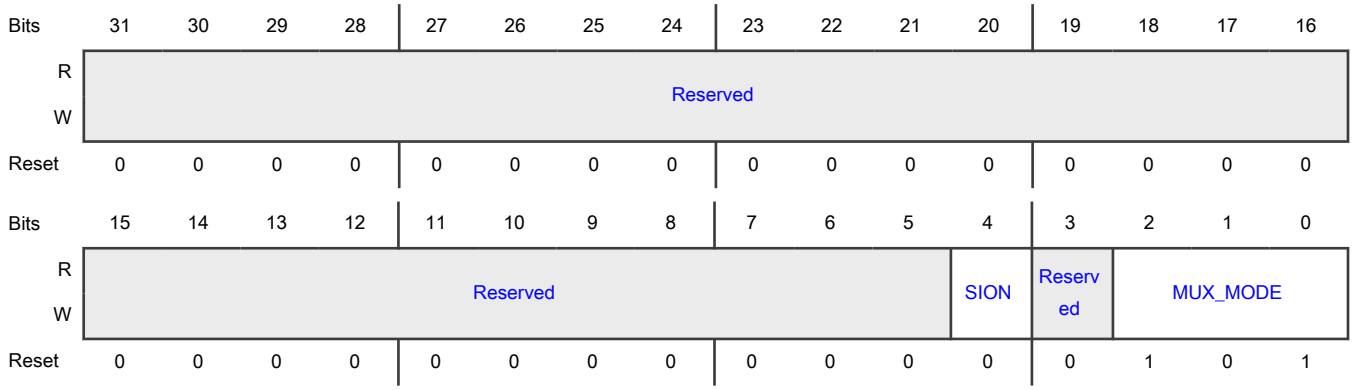
NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_RD2	C8h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_RD2
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: ENET1_RD2. 000 - Select mux mode: ALT0 mux port: ENET_QOS_RGMII_RD2 of instance: enet_qos 011 - Select mux mode: ALT3 mux port: LPTMR2_ALT2 of instance: lptmr2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO12 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO12 of instance: gpio4

27.3.1.53 SW_MUX_CTL_PAD_ENET1_RD3 SW MUX Control Register (SW_MUX_CTL_PAD_ENET1_RD3)

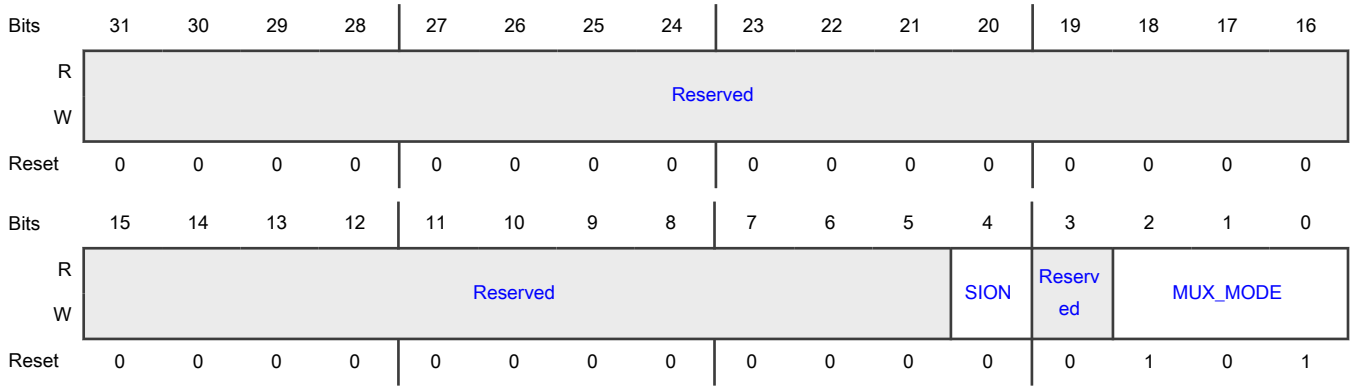
NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET1_RD3	CCh

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET1_RD3
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET1_RD3. 000 - Select mux mode: ALT0 mux port: ENET_QOS_RGMII_RD3 of instance: enet_qos 011 - Select mux mode: ALT3 mux port: LPTMR2_ALT3 of instance: lptmr2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO13 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO13 of instance: gpio4

27.3.1.54 SW_MUX_CTL_PAD_ENET2_MDC SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_MDC)

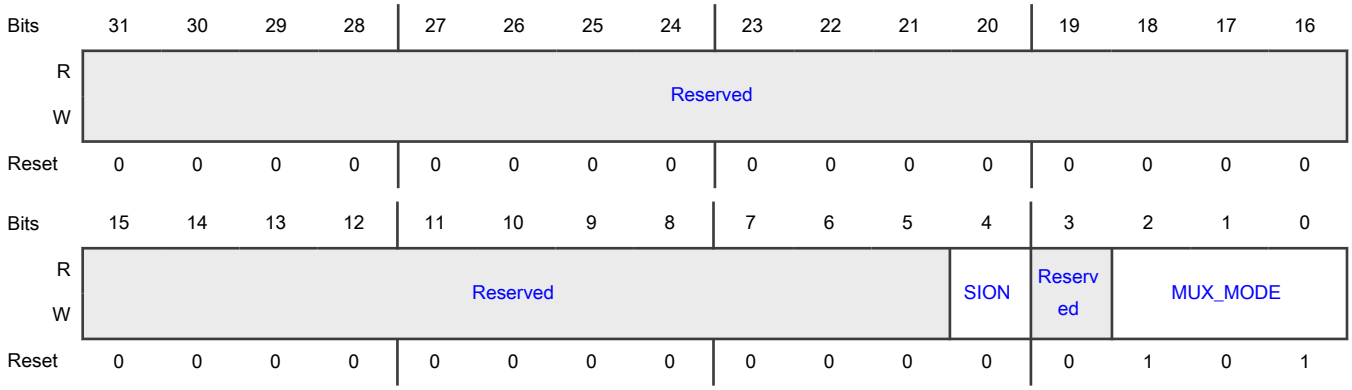
NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC in the registers below.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_MDC	D0h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_MDC
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET2_MDC. 000 - Select mux mode: ALT0 mux port: ENET1_MDC of instance: enet1 001 - Select mux mode: ALT1 mux port: LPUART4_DCD_B of instance: lpuart4 010 - Select mux mode: ALT2 mux port: SAI2_RX_SYNC of instance: sai2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO14 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO14 of instance: gpio4

27.3.1.55 SW_MUX_CTL_PAD_ENET2_MDIO SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_MDIO)

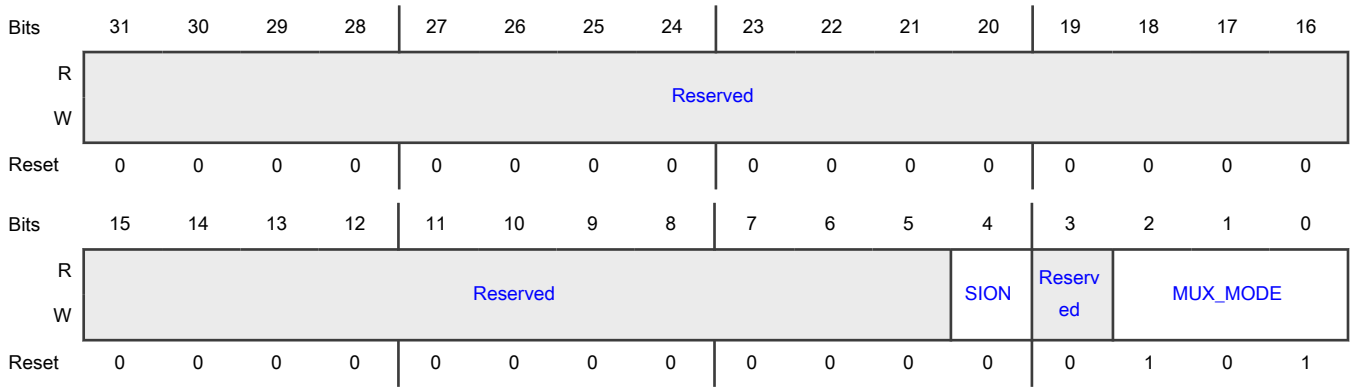
NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_MDIO	D4h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_MDIO
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET2_MDIO. 000 - Select mux mode: ALT0 mux port: ENET1_MDIO of instance: enet1 001 - Select mux mode: ALT1 mux port: LPUART4_RIN_B of instance: lpuart4 010 - Select mux mode: ALT2 mux port: SAI2_RX_BCLK of instance: sai2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO15 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO15 of instance: gpio4

27.3.1.56 SW_MUX_CTL_PAD_ENET2_TD3 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_TD3)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_TD3	D8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_TD3
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: ENET2_TD3. 000 - Select mux mode: ALT0 mux port: ENET1_RGMII_TD3 of instance: enet1 010 - Select mux mode: ALT2 mux port: SAI2_RX_DATA00 of instance: sai2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO16 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO16 of instance: gpio4

27.3.1.57 SW_MUX_CTL_PAD_ENET2_TD2 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_TD2)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_TD2	DCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_TD2
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET2_TD2. 000 - Select mux mode: ALT0 mux port: ENET1_RGMII_TD2 of instance: enet1 001 - Select mux mode: ALT1 mux port: ENET1_TX_CLK of instance: enet1 010 - Select mux mode: ALT2 mux port: SAI2_RX_DATA01 of instance: sai2

Table continues on the next page...

Table continued from the previous page...

Field	Description
	100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO17 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO17 of instance: gpio4

27.3.1.58 SW_MUX_CTL_PAD_ENET2_TD1 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_TD1)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_TD1	E0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_TD1
3 —	- Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET2_TD1. 000 - Select mux mode: ALT0 mux port: ENET1_RGMII_TD1 of instance: enet1 001 - Select mux mode: ALT1 mux port: LPUART4_RTS_B of instance: lpuart4 010 - Select mux mode: ALT2 mux port: SAI2_RX_DATA02 of instance: sai2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO18 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO18 of instance: gpio4

27.3.1.59 SW_MUX_CTL_PAD_ENET2_TD0 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_TD0)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_TD0	E4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_TD0
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET2_TD0. 000 - Select mux mode: ALT0 mux port: ENET1_RGMII_TD0 of instance: enet1 001 - Select mux mode: ALT1 mux port: LPUART4_TX of instance: lpuart4 010 - Select mux mode: ALT2 mux port: SAI2_RX_DATA03 of instance: sai2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO19 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO19 of instance: gpio4

27.3.1.60 SW_MUX_CTL_PAD_ENET2_TX_CTL SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_TX_CTL)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_TX_CTL	E8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_TX_CTL
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET2_TX_CTL. 000 - Select mux mode: ALT0 mux port: ENET1_RGMII_TX_CTL of instance: enet1 001 - Select mux mode: ALT1 mux port: LPUART4_DTR_B of instance: lpuart4 010 - Select mux mode: ALT2 mux port: SAI2_TX_SYNC of instance: sai2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO20 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO20 of instance: gpio4

27.3.1.61 SW_MUX_CTL_PAD_ENET2_TXC SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_TXC)

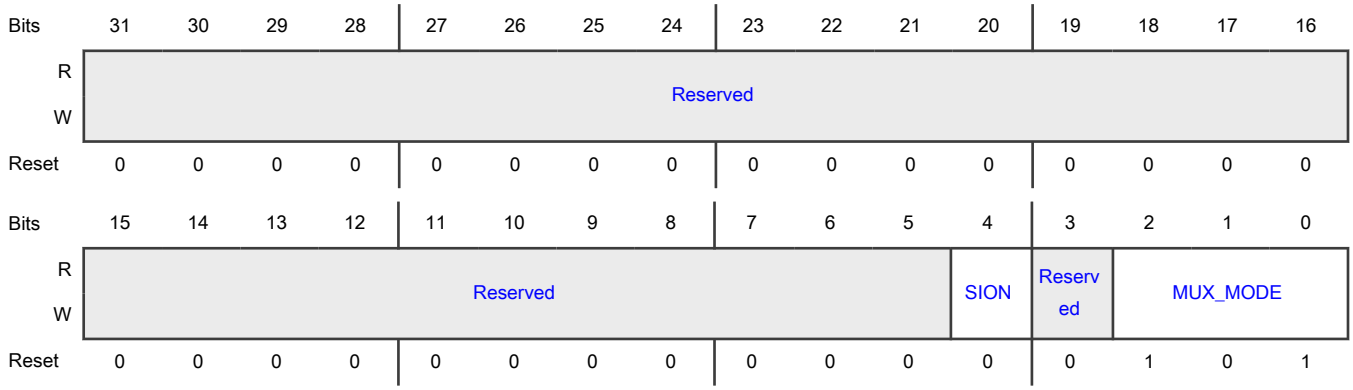
NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_TXC	ECh

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_TXC
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET2_TXC. 000 - Select mux mode: ALT0 mux port: ENET1_RGMII_TXC of instance: enet1 001 - Select mux mode: ALT1 mux port: ENET1_TX_ER of instance: enet1 010 - Select mux mode: ALT2 mux port: SAI2_TX_BCLK of instance: sai2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO21 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO21 of instance: gpio4

27.3.1.62 SW_MUX_CTL_PAD_ENET2_RX_CTL SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_RX_CTL)

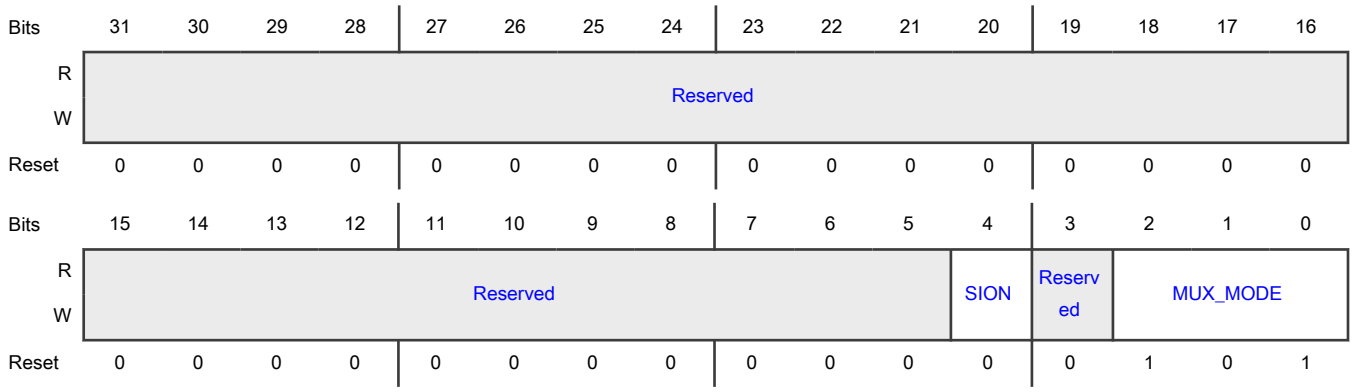
NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_RX_CTL	F0h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_RX_CTL
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET2_RX_CTL. 000 - Select mux mode: ALT0 mux port: ENET1_RGMII_RX_CTL of instance: enet1 001 - Select mux mode: ALT1 mux port: LPUART4_DSR_B of instance: lpuart4 010 - Select mux mode: ALT2 mux port: SAI2_TX_DATA00 of instance: sai2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO22 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO22 of instance: gpio4

27.3.1.63 SW_MUX_CTL_PAD_ENET2_RXC SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_RXC)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_RXC	F4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_RXC
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET2_RXC. 000 - Select mux mode: ALT0 mux port: ENET1_RGMII_RXC of instance: enet1 001 - Select mux mode: ALT1 mux port: ENET1_RX_ER of instance: enet1 010 - Select mux mode: ALT2 mux port: SAI2_TX_DATA01 of instance: sai2

Table continues on the next page...

Table continued from the previous page...

Field	Description
	100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO23 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO23 of instance: gpio4

27.3.1.64 SW_MUX_CTL_PAD_ENET2_RD0 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_RD0)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_RD0	F8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_RD0
3 —	- Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET2_RD0. 000 - Select mux mode: ALT0 mux port: ENET1_RGMII_RD0 of instance: enet1 001 - Select mux mode: ALT1 mux port: LPUART4_RX of instance: lpuart4 010 - Select mux mode: ALT2 mux port: SAI2_TX_DATA02 of instance: sai2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO24 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO24 of instance: gpio4

27.3.1.65 SW_MUX_CTL_PAD_ENET2_RD1 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_RD1)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_RD1	FCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_RD1
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET2_RD1. 000 - Select mux mode: ALT0 mux port: ENET1_RGMII_RD1 of instance: enet1 001 - Select mux mode: ALT1 mux port: SPDIF_IN of instance: spdif 010 - Select mux mode: ALT2 mux port: SAI2_TX_DATA03 of instance: sai2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO25 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO25 of instance: gpio4

27.3.1.66 SW_MUX_CTL_PAD_ENET2_RD2 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_RD2)

NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_RD2	100h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_RD2
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET2_RD2. 000 - Select mux mode: ALT0 mux port: ENET1_RGMII_RD2 of instance: enet1 001 - Select mux mode: ALT1 mux port: LPUART4_CTS_B of instance: lpuart4 010 - Select mux mode: ALT2 mux port: SAI2_MCLK of instance: sai2 011 - Select mux mode: ALT3 mux port: MQS2_RIGHT of instance: mqs2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO26 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO26 of instance: gpio4

27.3.1.67 SW_MUX_CTL_PAD_ENET2_RD3 SW MUX Control Register (SW_MUX_CTL_PAD_ENET2_RD3)

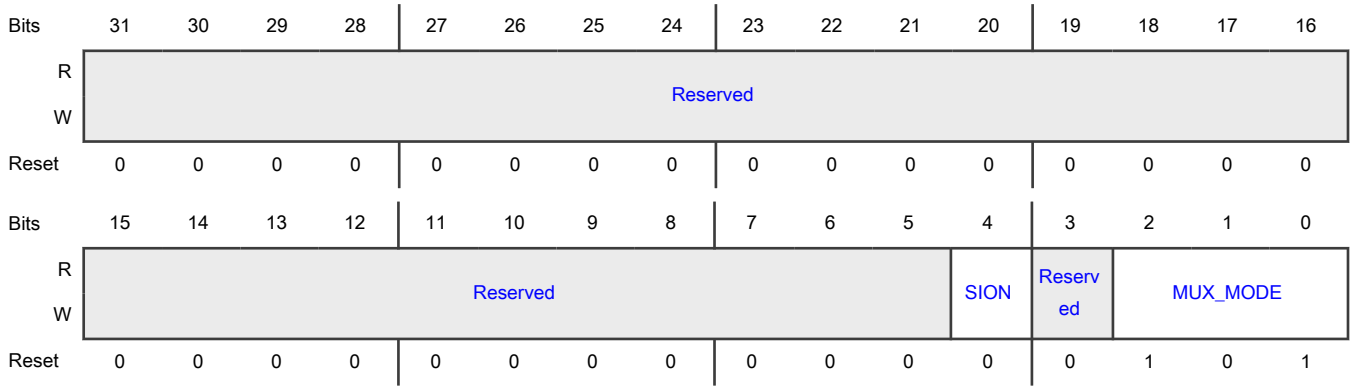
NOTE

ENET1 is Ethernet QoS with TSN and ENET2 is Ethernet MAC.

Offset

Register	Offset
SW_MUX_CTL_PAD_ENET2_RD3	104h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad ENET2_RD3
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET2_RD3. 000 - Select mux mode: ALT0 mux port: ENET1_RGMII_RD3 of instance: enet1 001 - Select mux mode: ALT1 mux port: SPDIF_OUT of instance: spdif 010 - Select mux mode: ALT2 mux port: SPDIF_IN of instance: spdif 011 - Select mux mode: ALT3 mux port: MQS2_LEFT of instance: mqs2 100 - Select mux mode: ALT4 mux port: FLEXIO2_FLEXIO27 of instance: flexio2 101 - Select mux mode: ALT5 mux port: GPIO4_IO27 of instance: gpio4

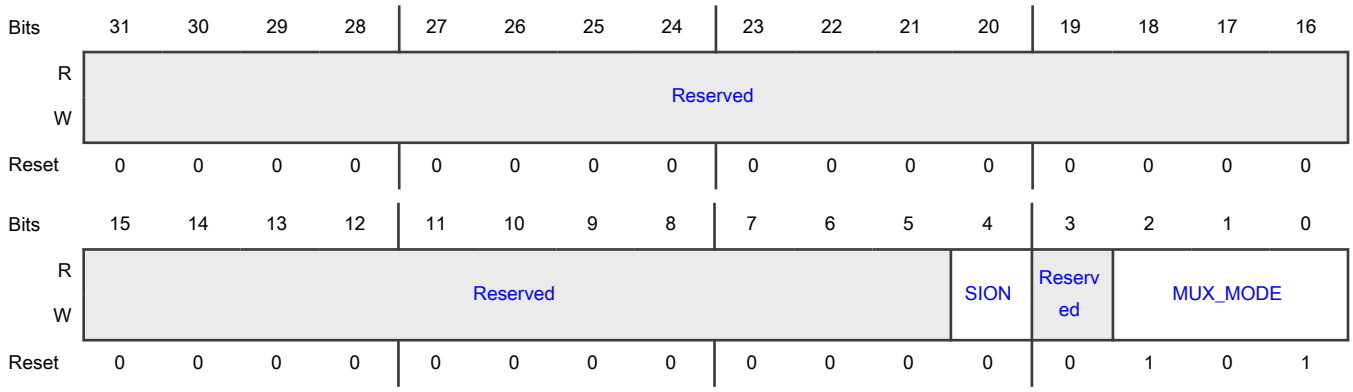
27.3.1.68 SW_MUX_CTL_PAD_SD1_CLK SW MUX Control Register (SW_MUX_CTL_PAD_SD1_CLK)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD1_CLK	108h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD1_CLK
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD1_CLK. 000 - Select mux mode: ALT0 mux port: USDHC1_CLK of instance: usdhc1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO08 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO08 of instance: gpio3

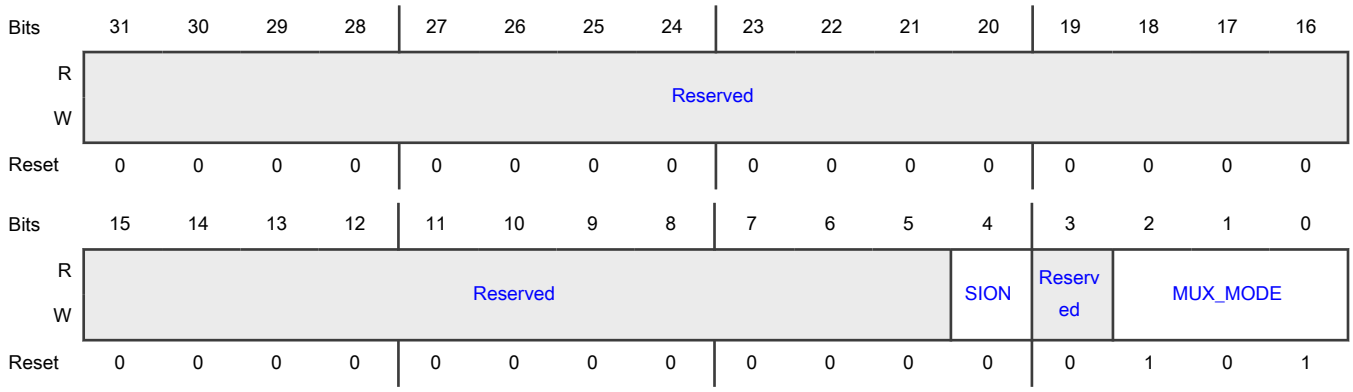
27.3.1.69 SW_MUX_CTL_PAD_SD1_CMD SW MUX Control Register (SW_MUX_CTL_PAD_SD1_CMD)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD1_CMD	10Ch

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD1_CMD
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD1_CMD. 000 - Select mux mode: ALT0 mux port: USDHC1_CMD of instance: usdhc1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO09 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO09 of instance: gpio3

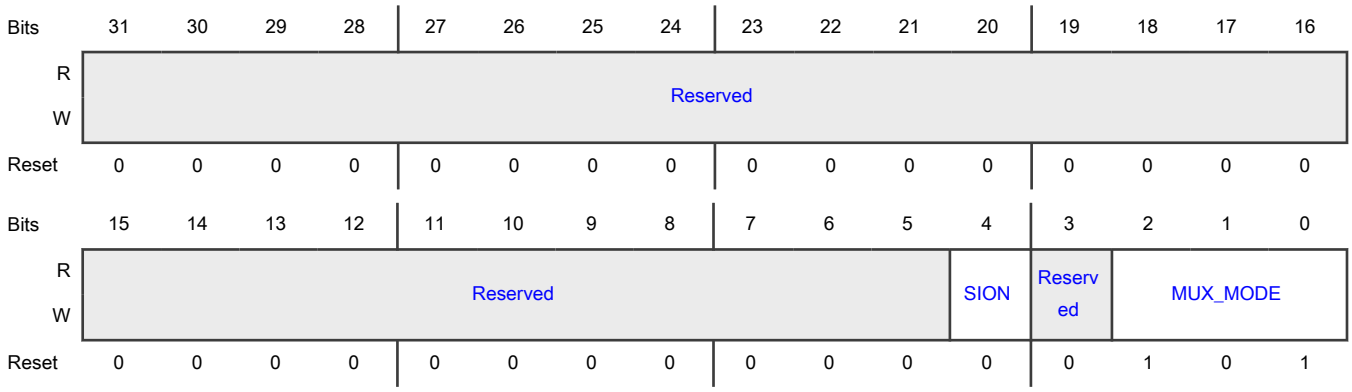
27.3.1.70 SW_MUX_CTL_PAD_SD1_DATA0 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA0)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD1_DATA0	110h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD1_DATA0
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD1_DATA0. 000 - Select mux mode: ALT0 mux port: USDHC1_DATA0 of instance: usdhc1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO10 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO10 of instance: gpio3

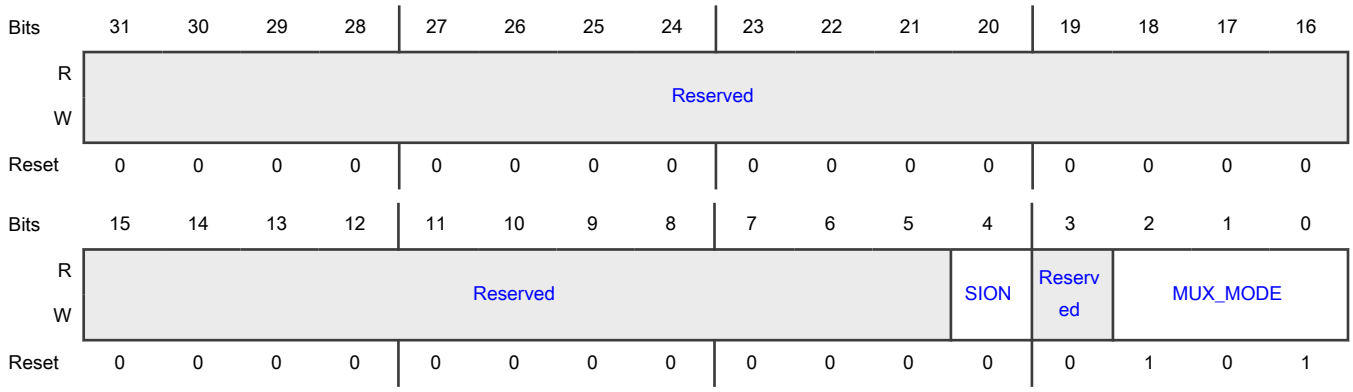
27.3.1.71 SW_MUX_CTL_PAD_SD1_DATA1 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA1)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD1_DATA1	114h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD1_DATA1
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD1_DATA1. 000 - Select mux mode: ALT0 mux port: USDHC1_DATA1 of instance: usdhc1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO11 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO11 of instance: gpio3 110 - Reserved

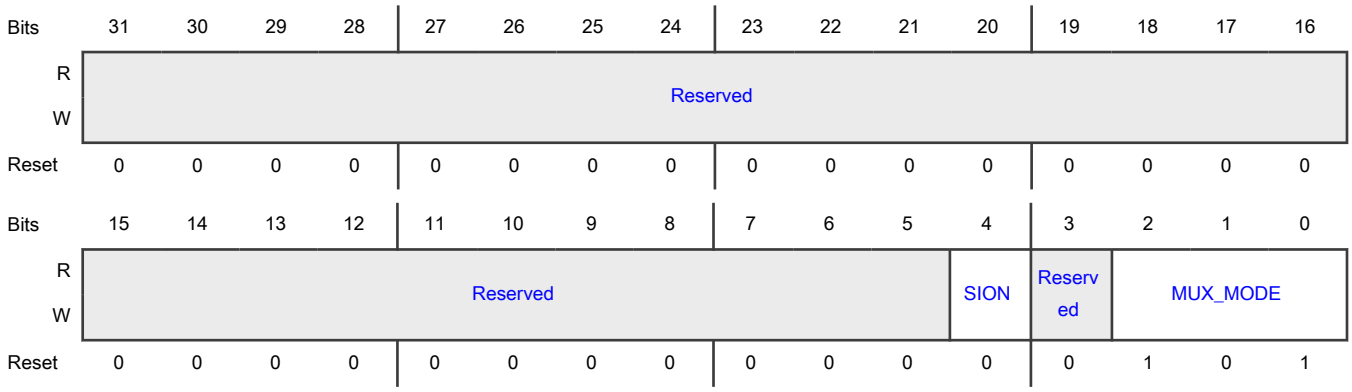
27.3.1.72 SW_MUX_CTL_PAD_SD1_DATA2 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA2)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD1_DATA2	118h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD1_DATA2
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD1_DATA2. 000 - Select mux mode: ALT0 mux port: USDHC1_DATA2 of instance: usdhc1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO12 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO12 of instance: gpio3 110 - Reserved

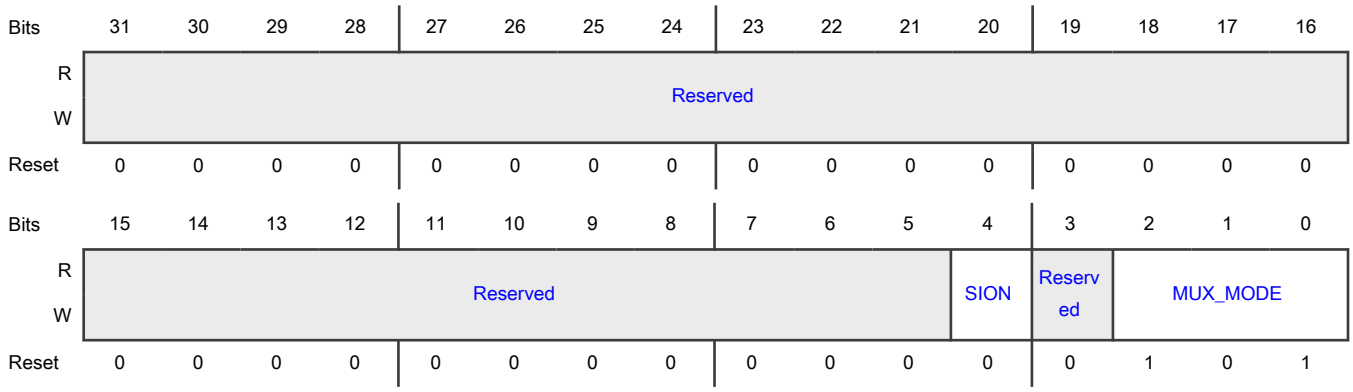
27.3.1.73 SW_MUX_CTL_PAD_SD1_DATA3 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA3)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD1_DATA3	11Ch

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD1_DATA3
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD1_DATA3. 000 - Select mux mode: ALT0 mux port: USDHC1_DATA3 of instance: usdhc1 001 - Select mux mode: ALT1 mux port: FLEXSPI1_A_SS1_B of instance: flexspi1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO13 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO13 of instance: gpio3

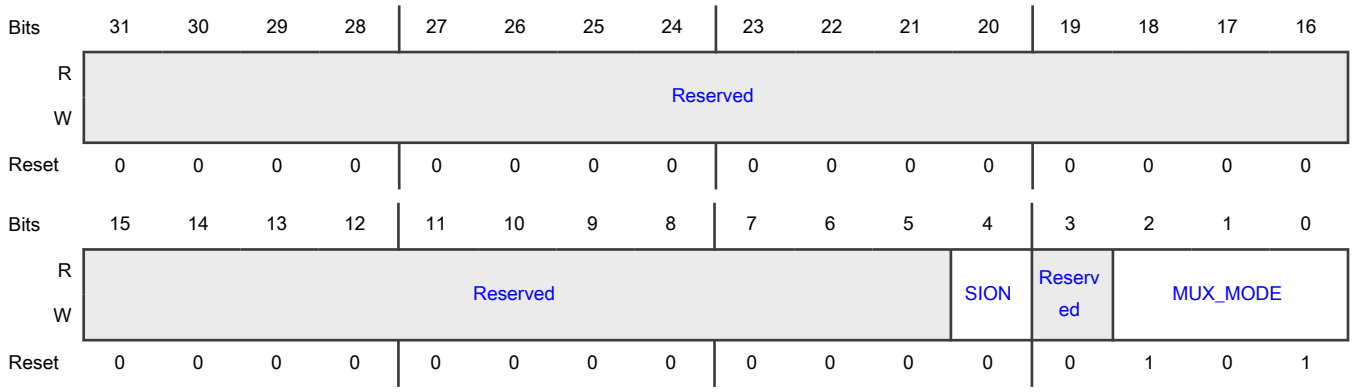
27.3.1.74 SW_MUX_CTL_PAD_SD1_DATA4 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA4)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD1_DATA4	120h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD1_DATA4
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD1_DATA4. 000 - Select mux mode: ALT0 mux port: USDHC1_DATA4 of instance: usdhc1 001 - Select mux mode: ALT1 mux port: FLEXSPI1_A_DATA04 of instance: flexspi1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO14 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO14 of instance: gpio3

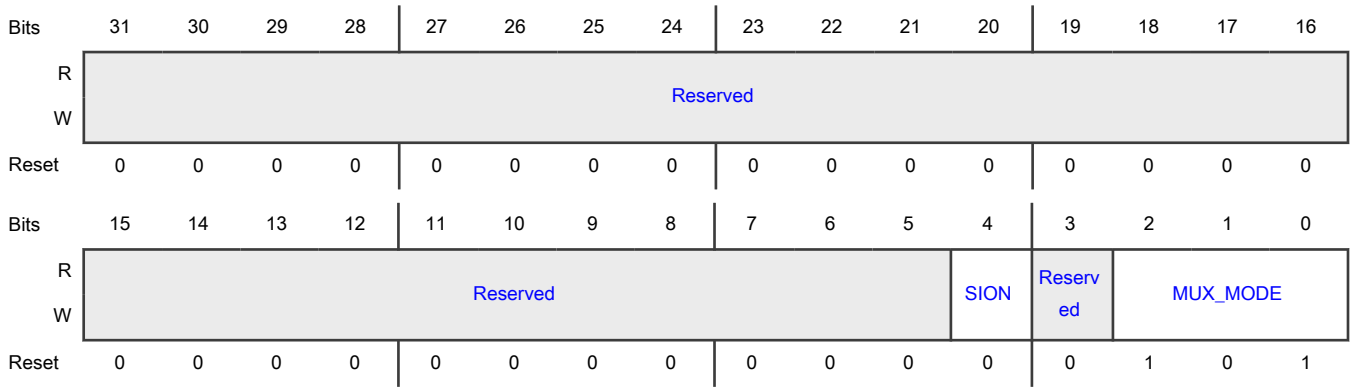
27.3.1.75 SW_MUX_CTL_PAD_SD1_DATA5 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA5)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD1_DATA5	124h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD1_DATA5
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD1_DATA5. 000 - Select mux mode: ALT0 mux port: USDHC1_DATA5 of instance: usdhc1 001 - Select mux mode: ALT1 mux port: FLEXSPI1_A_DATA05 of instance: flexspi1Select mux 010 - Select mux mode: ALT2 mux port: USDHC1_RESET_B of instance: usdhc1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO15 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO15 of instance: gpio3

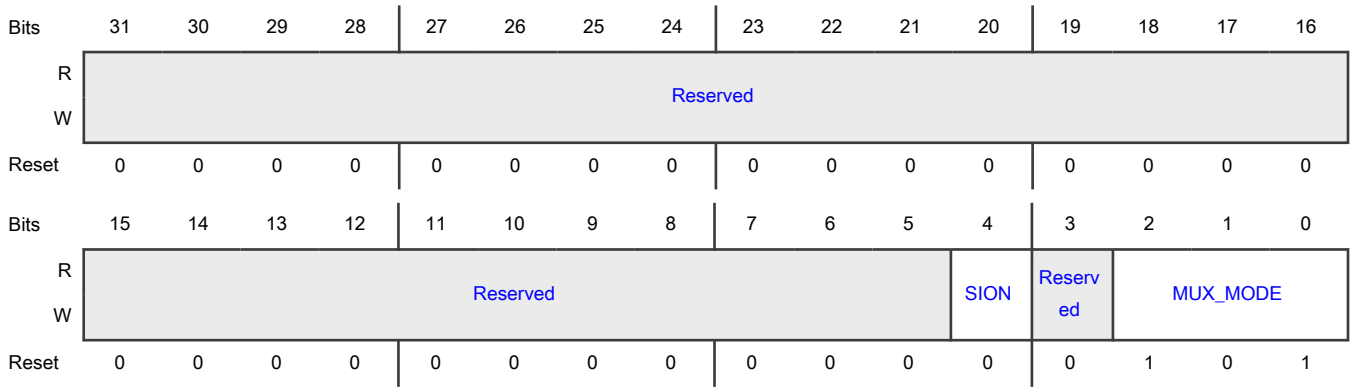
27.3.1.76 SW_MUX_CTL_PAD_SD1_DATA6 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA6)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD1_DATA6	128h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD1_DATA6
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD1_DATA6. 000 - Select mux mode: ALT0 mux port: USDHC1_DATA6 of instance: usdhc1 001 - Select mux mode: ALT1 mux port: FLEXSPI1_A_DATA06 of instance: flexspi1 010 - Select mux mode: ALT2 mux port: USDHC1_CD_B of instance: usdhc1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO16 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO16 of instance: gpio3

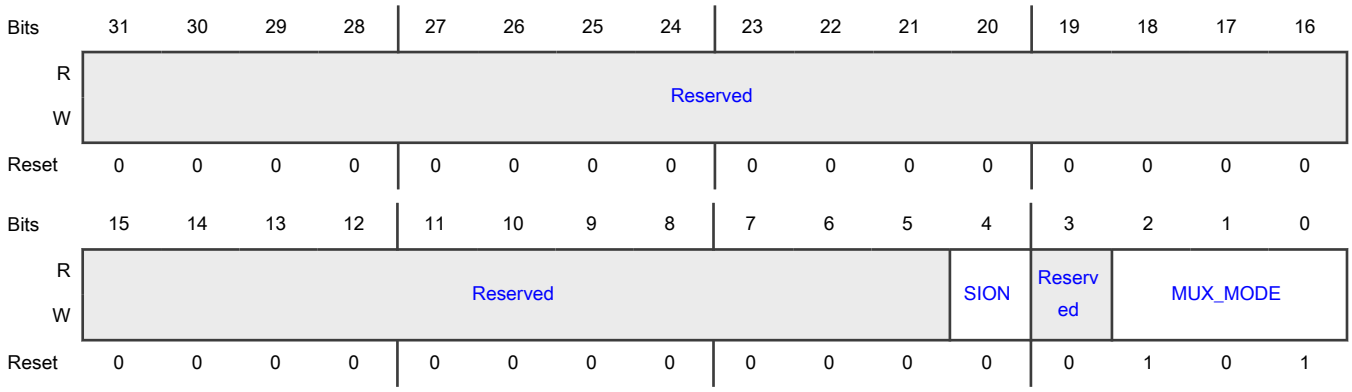
27.3.1.77 SW_MUX_CTL_PAD_SD1_DATA7 SW MUX Control Register (SW_MUX_CTL_PAD_SD1_DATA7)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD1_DATA7	12Ch

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD1_DATA7
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD1_DATA7. 000 - Select mux mode: ALT0 mux port: USDHC1_DATA7 of instance: usdhc1 001 - Select mux mode: ALT1 mux port: FLEXSPI1_A_DATA07 of instance: flexspi1 010 - Select mux mode: ALT2 mux port: USDHC1_WP of instance: usdhc1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO17 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO17 of instance: gpio3

27.3.1.78 SW_MUX_CTL_PAD_SD1_STROBE SW MUX Control Register (SW_MUX_CTL_PAD_SD1_STROBE)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD1_STROBE	130h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD1_STROBE
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD1_STROBE. 000 - Select mux mode: ALT0 mux port: USDHC1_STROBE of instance: usdhc1 001 - Select mux mode: ALT1 mux port: FLEXSPI1_A_DQS of instance: flexspi1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO18 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO18 of instance: gpio3

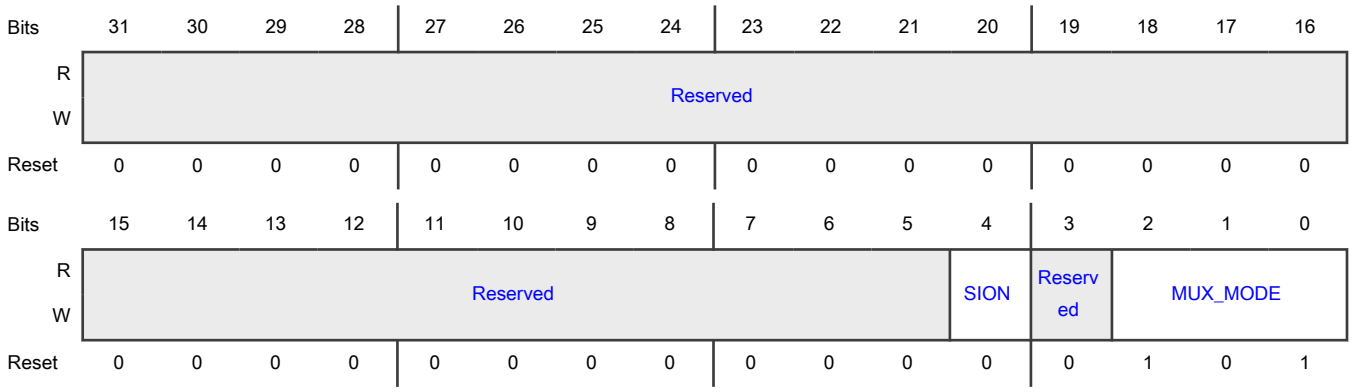
27.3.1.79 SW_MUX_CTL_PAD_SD2_VSELECT SW MUX Control Register (SW_MUX_CTL_PAD_SD2_VSELECT)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD2_VSELECT	134h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD2_VSELECT
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_VSELECT. 000 - Select mux mode: ALT0 mux port: USDHC2_VSELECT of instance: usdhc2 001 - Select mux mode: ALT1 mux port: USDHC2_WP of instance: usdhc2 010 - Select mux mode: ALT2 mux port: LPTMR2_ALT3 of instance: lptmr2 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO19 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO19 of instance: gpio3 110 - Reserved

27.3.1.80 SW_MUX_CTL_PAD_SD3_CLK SW MUX Control Register (SW_MUX_CTL_PAD_SD3_CLK)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD3_CLK	138h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD3_CLK
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD3_CLK. 000 - Select mux mode: ALT0 mux port: USDHC3_CLK of instance: usdhc3 001 - Select mux mode: ALT1 mux port: FLEXSPI1_A_SCLK of instance: flexspi1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO20 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO20 of instance: gpio3

27.3.1.81 SW_MUX_CTL_PAD_SD3_CMD SW MUX Control Register (SW_MUX_CTL_PAD_SD3_CMD)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD3_CMD	13Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD3_CMD
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD3_CMD. 000 - Select mux mode: ALT0 mux port: USDHC3_CMD of instance: usdhc3 001 - Select mux mode: ALT1 mux port: FLEXSPI1_A_SS0_B of instance: flexspi1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO21 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO21 of instance: gpio3

27.3.1.82 SW_MUX_CTL_PAD_SD3_DATA0 SW MUX Control Register (SW_MUX_CTL_PAD_SD3_DATA0)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD3_DATA0	140h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD3_DATA0
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD3_DATA0. 000 - Select mux mode: ALT0 mux port: USDHC3_DATA0 of instance: usdhc3 001 - Select mux mode: ALT1 mux port: FLEXSPI1_A_DATA00 of instance: flexspi1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO22 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO22 of instance: gpio3

27.3.1.83 SW_MUX_CTL_PAD_SD3_DATA1 SW MUX Control Register (SW_MUX_CTL_PAD_SD3_DATA1)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD3_DATA1	144h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD3_DATA1
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD3_DATA1. 000 - Select mux mode: ALT0 mux port: USDHC3_DATA1 of instance: usdhc3 001 - Select mux mode: ALT1 mux port: FLEXSPI1_A_DATA01 of instance: flexspi1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO23 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO23 of instance: gpio3

27.3.1.84 SW_MUX_CTL_PAD_SD3_DATA2 SW MUX Control Register (SW_MUX_CTL_PAD_SD3_DATA2)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD3_DATA2	148h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD3_DATA2
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD3_DATA2. 000 - Select mux mode: ALT0 mux port: USDHC3_DATA2 of instance: usdhc3 001 - Select mux mode: ALT1 mux port: FLEXSPI1_A_DATA02 of instance: flexspi1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO24 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO24 of instance: gpio3

27.3.1.85 SW_MUX_CTL_PAD_SD3_DATA3 SW MUX Control Register (SW_MUX_CTL_PAD_SD3_DATA3)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD3_DATA3	14Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD3_DATA3
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD3_DATA3. 000 - Select mux mode: ALT0 mux port: USDHC3_DATA3 of instance: usdhc3 001 - Select mux mode: ALT1 mux port: FLEXSPI1_A_DATA03 of instance: flexspi1 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO25 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO25 of instance: gpio3

27.3.1.86 SW_MUX_CTL_PAD_SD2_CD_B SW MUX Control Register (SW_MUX_CTL_PAD_SD2_CD_B)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD2_CD_B	150h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD2_CD_B
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD2_CD_B. 000 - Select mux mode: ALT0 mux port: USDHC2_CD_B of instance: usdhc2 001 - Select mux mode: ALT1 mux port: ENET_QOS_1588_EVENT0_IN of instance: enet_qos 010 - Select mux mode: ALT2 mux port: I3C2_SCL of instance: i3c2 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO00 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO00 of instance: gpio3

27.3.1.87 SW_MUX_CTL_PAD_SD2_CLK SW MUX Control Register (SW_MUX_CTL_PAD_SD2_CLK)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD2_CLK	154h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD2_CLK
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_CLK. 000 - Select mux mode: ALT0 mux port: USDHC2_CLK of instance: usdhc2 001 - Select mux mode: ALT1 mux port: ENET_QOS_1588_EVENT0_OUT of instance: enet_qos 010 - Select mux mode: ALT2 mux port: I3C2_SDA of instance: i3c2 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO01 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO01 of instance: gpio3

Table continues on the next page...

Table continued from the previous page...

Field	Description
	110 - Reserved

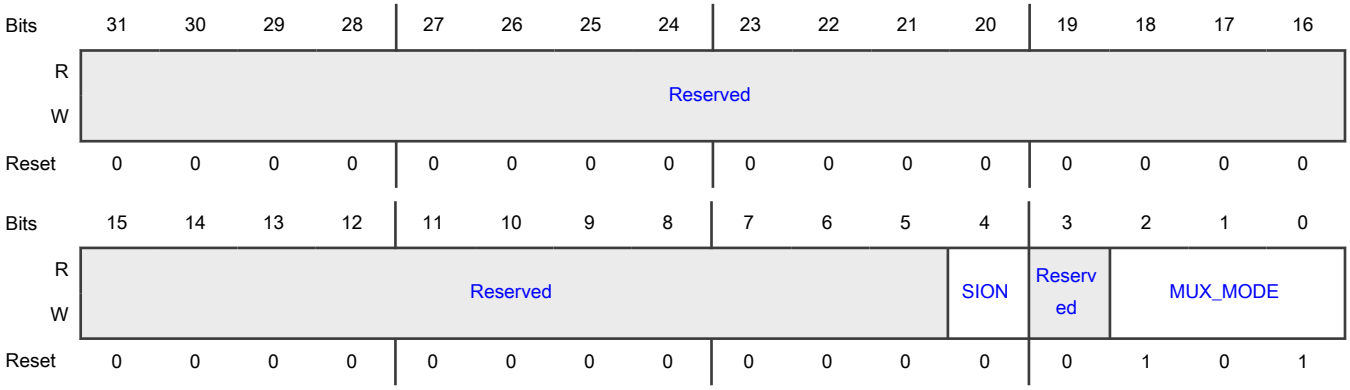
27.3.1.88 SW_MUX_CTL_PAD_SD2_CMD SW MUX Control Register (SW_MUX_CTL_PAD_SD2_CMD)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD2_CMD	158h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD2_CMD
3 —	- Reserved
2-0	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: SD2_CMD.

Table continues on the next page...

Table continued from the previous page...

Field	Description
MUX_MODE	000 - Select mux mode: ALT0 mux port: USDHC2_CMD of instance: usdhc2 001 - Select mux mode: ALT1 mux port: ENET1_1588_EVENT0_IN of instance: enet1 010 - Select mux mode: ALT2 mux port: I3C2_PUR of instance: i3c2 011 - Select mux mode: ALT3 mux port: I3C2_PUR_B of instance: i3c2 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO02 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO02 of instance: gpio3 110 - Reserved

27.3.1.89 SW_MUX_CTL_PAD_SD2_DATA0 SW MUX Control Register (SW_MUX_CTL_PAD_SD2_DATA0)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD2_DATA0	15Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Input Path is determined by functionality 1 - Force input path of pad SD2_DATA0
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_DATA0. 000 - Select mux mode: ALT0 mux port: USDHC2_DATA0 of instance: usdhc2 001 - Select mux mode: ALT1 mux port: ENET1_1588_EVENT0_OUT of instance: enet1 010 - Select mux mode: ALT2 mux port: CAN2_TX of instance: can2 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO03 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO03 of instance: gpio3 110 - Reserved

27.3.1.90 SW_MUX_CTL_PAD_SD2_DATA1 SW MUX Control Register (SW_MUX_CTL_PAD_SD2_DATA1)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD2_DATA1	160h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD2_DATA1
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_DATA1. 000 - Select mux mode: ALT0 mux port: USDHC2_DATA1 of instance: usdhc2 001 - Select mux mode: ALT1 mux port: ENET1_1588_EVENT1_IN of instance: enet1 010 - Select mux mode: ALT2 mux port: CAN2_RX of instance: can2 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO04 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO04 of instance: gpio3 110 - Reserved

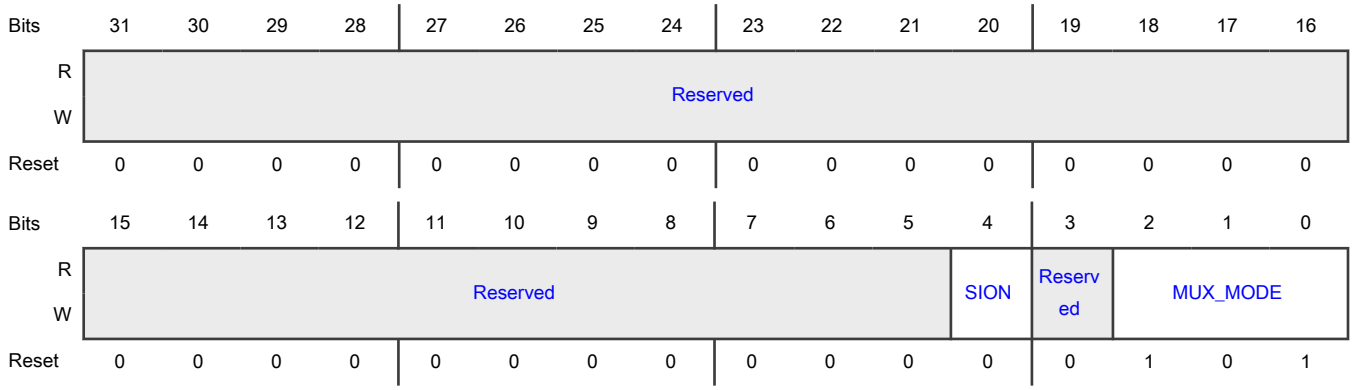
27.3.1.91 SW_MUX_CTL_PAD_SD2_DATA2 SW MUX Control Register (SW_MUX_CTL_PAD_SD2_DATA2)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD2_DATA2	164h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD2_DATA2
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_DATA2. 000 - Select mux mode: ALT0 mux port: USDHC2_DATA2 of instance: usdhc2 001 - Select mux mode: ALT1 mux port: ENET1_1588_EVENT1_OUT of instance: enet1 010 - Select mux mode: ALT2 mux port: MQS2_RIGHT of instance: mqs2 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO05 of instance: flexio1 101 - Select mux mode: GPIO3_IO05 of instance: gpio3 110 - Reserved

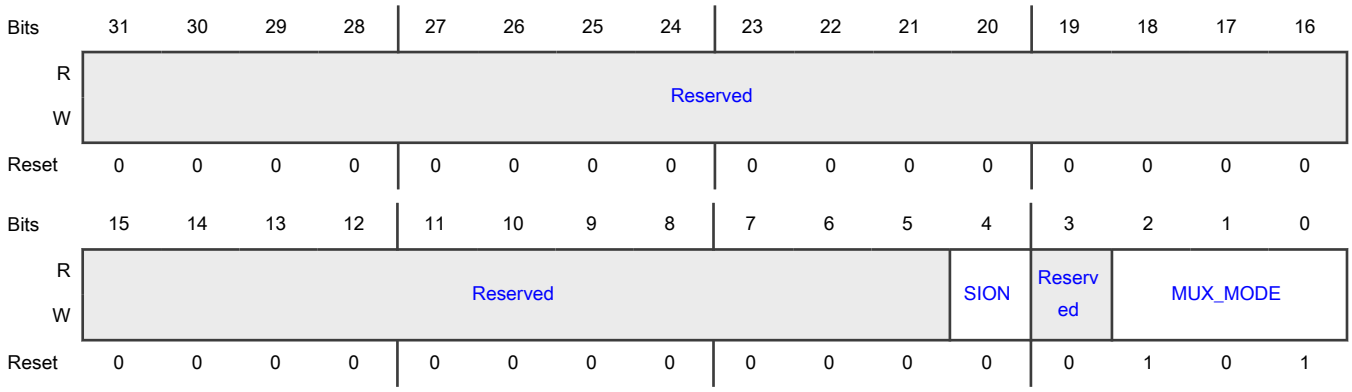
27.3.1.92 SW_MUX_CTL_PAD_SD2_DATA3 SW MUX Control Register (SW_MUX_CTL_PAD_SD2_DATA3)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD2_DATA3	168h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD2_DATA3
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_DATA3. 000 - Select mux mode: ALT0 mux port: USDHC2_DATA3 of instance: usdhc2 001 - Select mux mode: ALT1 mux port: LPTMR2_ALT1 of instance: lptmr2 010 - Select mux mode: ALT2 mux port: MQS2_LEFT of instance: mqs2 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO06 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO06 of instance: gpio3 110 - Reserved

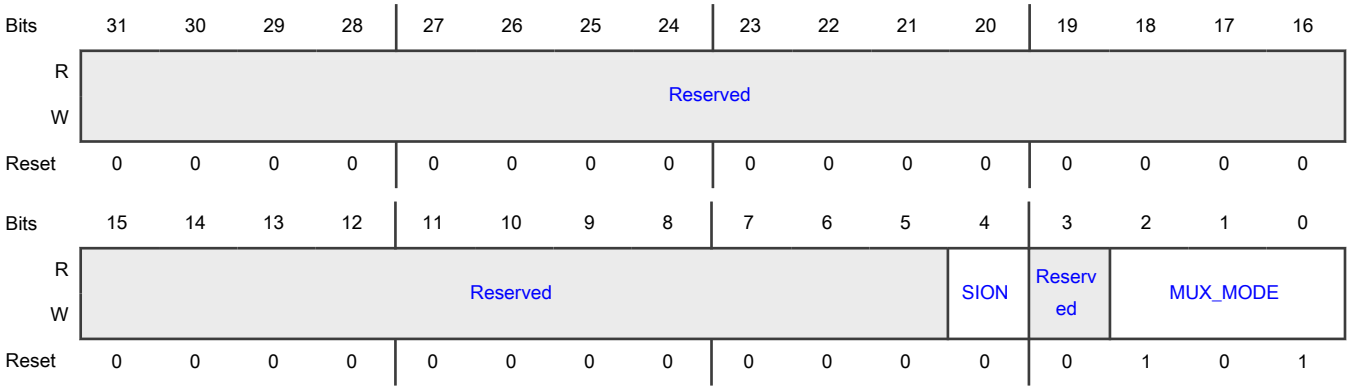
27.3.1.93 SW_MUX_CTL_PAD_SD2_RESET_B SW MUX Control Register (SW_MUX_CTL_PAD_SD2_RESET_B)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SD2_RESET_B	16Ch

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SD2_RESET_B
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD2_RESET_B. 000 - Select mux mode: ALT0 mux port: USDHC2_RESET_B of instance: usdhc2 001 - Select mux mode: ALT1 mux port: LPTMR2_ALT2 of instance: lptmr2 100 - Select mux mode: ALT4 mux port: FLEXIO1_FLEXIO07 of instance: flexio1 101 - Select mux mode: ALT5 mux port: GPIO3_IO07 of instance: gpio3 110 - Reserved

27.3.1.94 SW_MUX_CTL_PAD_I2C1_SCL SW MUX Control Register (SW_MUX_CTL_PAD_I2C1_SCL)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_I2C1_SCL	170h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad I2C1_SCL
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: I2C1_SCL. 000 - Select mux mode: ALT0 mux port: LPI2C1_SCL of instance: lpi2c1 001 - Select mux mode: ALT1 mux port: I3C1_SCL of instance: i3c1 010 - Select mux mode: ALT2 mux port: LPUART1_DCD_B of instance: lpuart1 011 - Select mux mode: ALT3 mux port: TPM2_CH0 of instance: tpm2 101 - Select mux mode: ALT5 mux port: GPIO1_IO00 of instance: gpio1

27.3.1.95 SW_MUX_CTL_PAD_I2C1_SDA SW MUX Control Register (SW_MUX_CTL_PAD_I2C1_SDA)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_I2C1_SDA	174h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad I2C1_SDA
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: I2C1_SDA. 000 - Select mux mode: ALT0 mux port: LPI2C1_SDA of instance: lpi2c1 001 - Select mux mode: ALT1 mux port: I3C1_SDA of instance: i3c1 010 - Select mux mode: ALT2 mux port: LPUART1_RIN_B of instance: lpuart1 011 - Select mux mode: ALT3 mux port: TPM2_CH1 of instance: tpm2 101 - Select mux mode: ALT5 mux port: GPIO1_IO01 of instance: gpio1

27.3.1.96 SW_MUX_CTL_PAD_I2C2_SCL SW MUX Control Register (SW_MUX_CTL_PAD_I2C2_SCL)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_I2C2_SCL	178h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad I2C2_SCL
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: I2C2_SCL. 000 - Select mux mode: ALT0 mux port: LPI2C2_SCL of instance: lpi2c2 001 - Select mux mode: ALT1 mux port: I3C1_PUR of instance: i3c1 010 - Select mux mode: ALT2 mux port: LPUART2_DCD_B of instance: lpuart2 011 - Select mux mode: ALT3 mux port: TPM2_CH2 of instance: tpm2 100 - Select mux mode: ALT4 mux port: SAI1_RX_SYNC of instance: sai1

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - Select mux mode: ALT5 mux port: GPIO1_IO02 of instance: gpio1 110 - Select mux mode: ALT6 mux port: I3C1_PUR_B of instance: i3c1

27.3.1.97 SW_MUX_CTL_PAD_I2C2_SDA SW MUX Control Register (SW_MUX_CTL_PAD_I2C2_SDA)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_I2C2_SDA	17Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad I2C2_SDA
3 —	- Reserved
2-0	MUX Mode Select Field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
MUX_MODE	Select 1 of 5 iomux modes to be used for pad: I2C2_SDA. 000 - Select mux mode: ALT0 mux port: LPI2C2_SDA of instance: lpi2c2 010 - Select mux mode: ALT2 mux port: LPUART2_RIN_B of instance: lpuart2 011 - Select mux mode: ALT3 mux port: TPM2_CH3 of instance: tpm2 100 - Select mux mode: ALT4 mux port: SAI1_RX_BCLK of instance: sai1 101 - Select mux mode: ALT5 mux port: GPIO1_IO03 of instance: gpio1

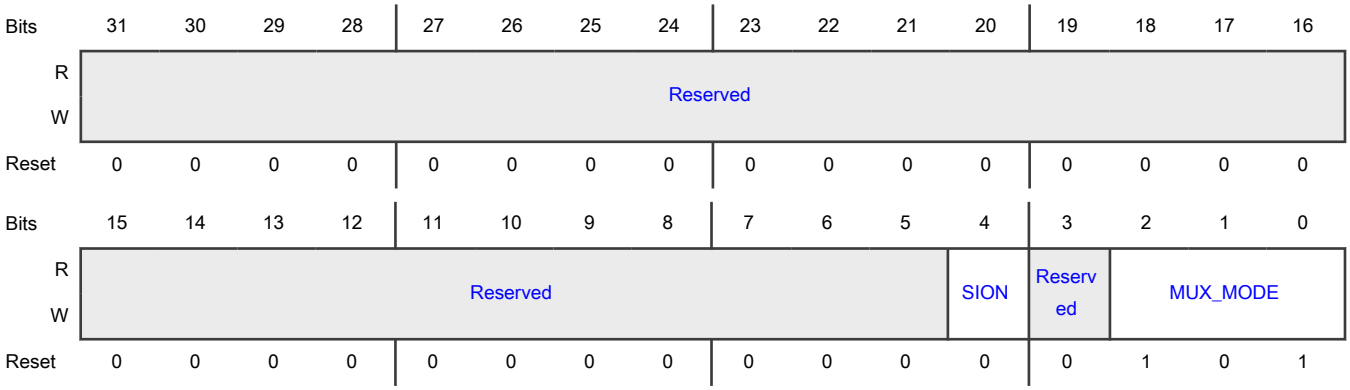
27.3.1.98 SW_MUX_CTL_PAD_UART1_RXD SW MUX Control Register (SW_MUX_CTL_PAD_UART1_RXD)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_UART1_RXD	180h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Force input path of pad UART1_RXD
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: UART1_RXD. 000 - Select mux mode: ALT0 mux port: LPUART1_RX of instance: lpuart1 001 - Select mux mode: ALT1 mux port: ELE_UART_RX of instance: ELE 010 - Select mux mode: ALT2 mux port: LPSPI2_SIN of instance: lpspi2 011 - Select mux mode: ALT3 mux port: TPM1_CH0 of instance: tpm1 101 - Select mux mode: ALT5 mux port: GPIO1_IO04 of instance: gpio1

27.3.1.99 SW_MUX_CTL_PAD_UART1_TXD SW MUX Control Register (SW_MUX_CTL_PAD_UART1_TXD)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_UART1_TXD	184h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5	-

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad UART1_TXD
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: UART1_TXD. 000 - Select mux mode: ALT0 mux port: LPUART1_TX of instance: lpuart1 001 - Select mux mode: ALT1 mux port: ELE_UART_TX of instance: ELE 010 - Select mux mode: ALT2 mux port: LPSPI2_PCS0 of instance: lpspi2 011 - Select mux mode: ALT3 mux port: TPM1_CH1 of instance: tpm1 101 - Select mux mode: ALT5 mux port: GPIO1_IO05 of instance: gpio1

27.3.1.100 SW_MUX_CTL_PAD_UART2_RXD SW MUX Control Register (SW_MUX_CTL_PAD_UART2_RXD)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_UART2_RXD	188h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad UART2_RXD
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: UART2_RXD. 000 - Select mux mode: ALT0 mux port: LPUART2_RX of instance: lpuart2 001 - Select mux mode: ALT1 mux port: LPUART1_CTS_B of instance: lpuart1 010 - Select mux mode: ALT2 mux port: LPSPI2_SOUT of instance: lpspi2 011 - Select mux mode: ALT3 mux port: TPM1_CH2 of instance: tpm1 100 - Select mux mode: ALT4 mux port: SAI1_MCLK of instance: sai1 101 - Select mux mode: ALT5 mux port: GPIO1_IO06 of instance: gpio1

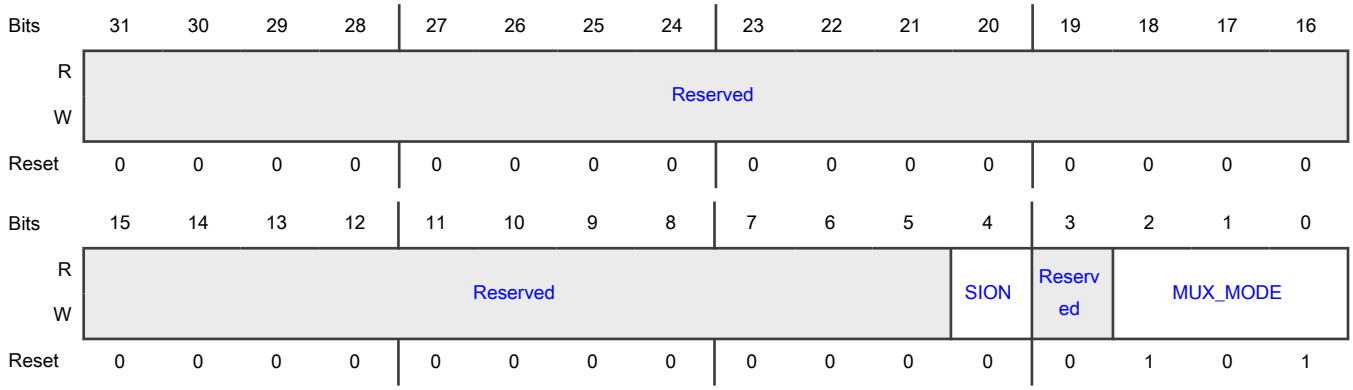
27.3.1.101 SW_MUX_CTL_PAD_UART2_TXD SW MUX Control Register (SW_MUX_CTL_PAD_UART2_TXD)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_UART2_TXD	18Ch

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad UART2_TXD
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: UART2_TXD. 000 - Select mux mode: ALT0 mux port: LPUART2_TX of instance: lpuart2 001 - Select mux mode: ALT1 mux port: LPUART1_RTS_B of instance: lpuart1 010 - Select mux mode: ALT2 mux port: LPSPI2_SCK of instance: lpspi2 011 - Select mux mode: ALT3 mux port: TPM1_CH3 of instance: tpm1 101 - Select mux mode: ALT5 mux port: GPIO1_IO07 of instance: gpio1

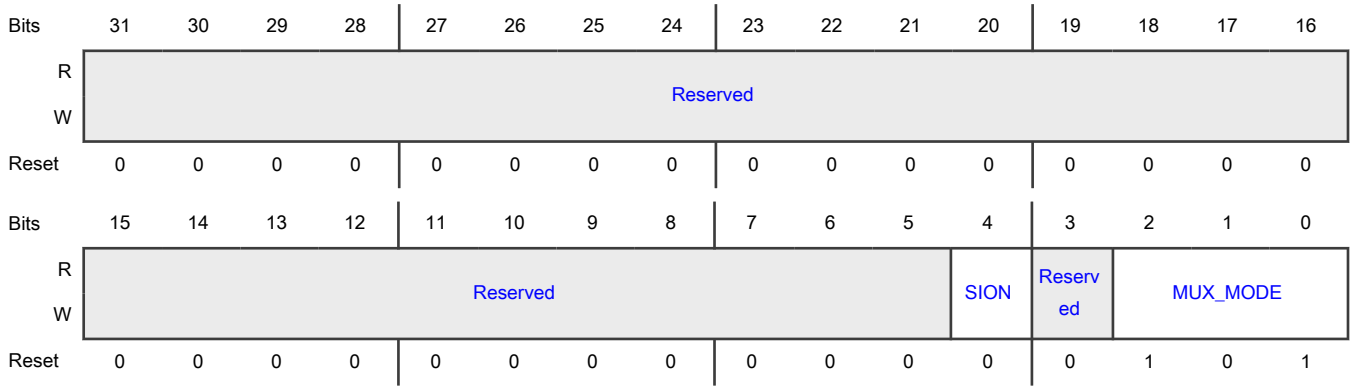
27.3.1.102 SW_MUX_CTL_PAD_PDM_CLK SW MUX Control Register (SW_MUX_CTL_PAD_PDM_CLK)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_PDM_CLK	190h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad PDM_CLK
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: PDM_CLK. 000 - Select mux mode: ALT0 mux port: PDM_CLK of instance: pdm 001 - Select mux mode: ALT1 mux port: MQS1_LEFT of instance: mqs1 100 - Select mux mode: ALT4 mux port: LPTMR1_ALT1 of instance: lptmr1 101 - Select mux mode: ALT5 mux port: GPIO1_IO08 of instance: gpio1 110 - Select mux mode: ALT6 mux port: CAN1_TX of instance: can1

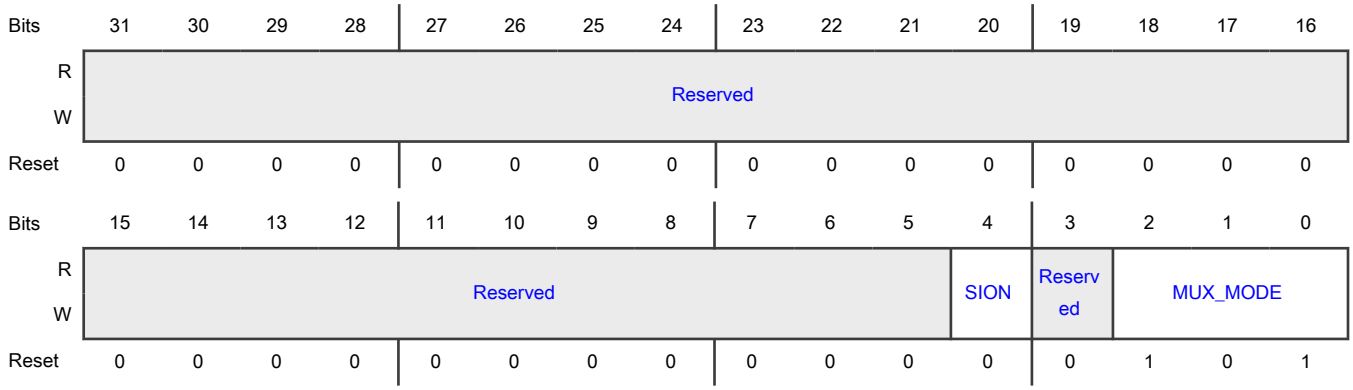
27.3.1.103 SW_MUX_CTL_PAD_PDM_BIT_STREAM0 SW MUX Control Register (SW_MUX_CTL_PAD_PDM_BIT_STREAM0)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_PDM_BIT_STREAM0	194h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad PDM_BIT_STREAM0
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: PDM_BIT_STREAM0. 000 - Select mux mode: ALT0 mux port: PDM_BIT_STREAM00 of instance: pdm 001 - Select mux mode: ALT1 mux port: MQS1_RIGHT of instance: mqs1 010 - Select mux mode: ALT2 mux port: LPSPI1_PCS1 of instance: lpspi1 011 - Select mux mode: ALT3 mux port: TPM1_EXTCLK of instance: tpm1 100 - Select mux mode: ALT4 mux port: LPTMR1_ALT2 of instance: lptmr1 101 - Select mux mode: ALT5 mux port: GPIO1_IO09 of instance: gpio1 110 - Select mux mode: ALT6 mux port: CAN1_TX of instance: can1

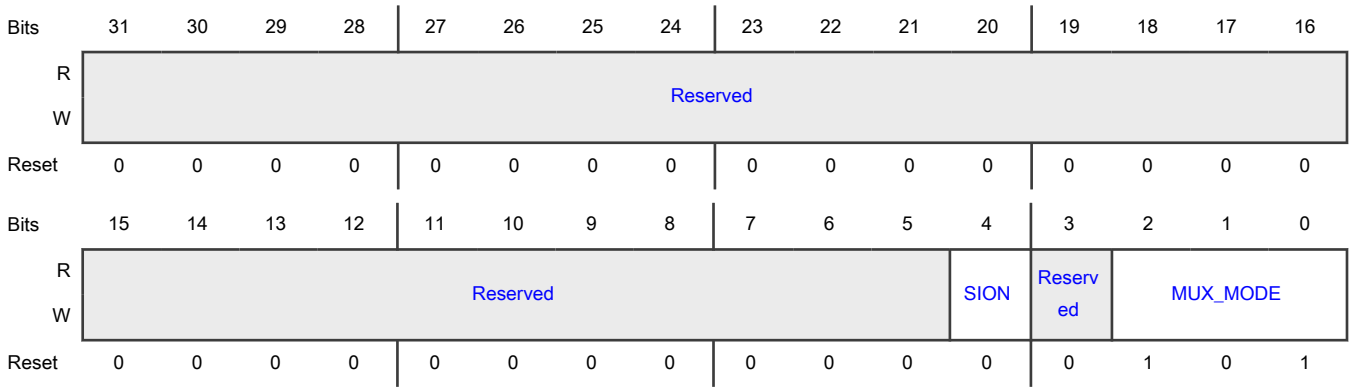
27.3.1.104 SW_MUX_CTL_PAD_PDM_BIT_STREAM1 SW MUX Control Register (SW_MUX_CTL_PAD_PDM_BIT_STREAM1)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_PDM_BIT_STREAM1	198h

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad PDM_BIT_STREAM1
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: PDM_BIT_STREAM1. 000 - Select mux mode: ALT0 mux port: PDM_BIT_STREAM01 of instance: pdm 001 - Select mux mode: ALT1 mux port: NMI_GLUE_NMI of instance: nmi_glue 010 - Select mux mode: ALT2 mux port: LPSPI2_PCS1 of instance: lpspi2 011 - Select mux mode: ALT3 mux port: TPM2_EXTCLK of instance: tpm2 100 - Select mux mode: ALT4 mux port: LPTMR1_ALT3 of instance: lptmr1 101 - Select mux mode: ALT5 mux port: GPIO1_IO10 of instance: gpio1 110 - Reserved

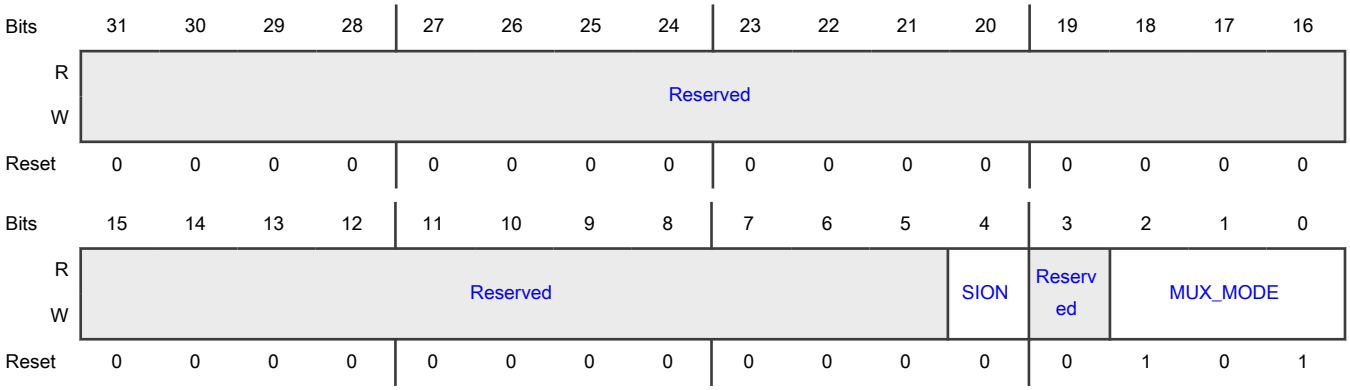
27.3.1.105 SW_MUX_CTL_PAD_SAI1_TXFS SW MUX Control Register (SW_MUX_CTL_PAD_SAI1_TXFS)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SAI1_TXFS	19Ch

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SAI1_TXFS
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SAI1_TXFS. 000 - Select mux mode: ALT0 mux port: SAI1_TX_SYNC of instance: sai1 001 - Select mux mode: ALT1 mux port: SAI1_TX_DATA01 of instance: sai1 010 - Select mux mode: ALT2 mux port: LPSPI1_PCS0 of instance: lpspi1 011 - Select mux mode: ALT3 mux port: LPUART2_DTR_B of instance: lpuart2 100 - Select mux mode: ALT4 mux port: MQS1_LEFT of instance: mqs1

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - Select mux mode: ALT5 mux port: GPIO1_IO11 of instance: gpio1

27.3.1.106 SW_MUX_CTL_PAD_SAI1_TXC SW MUX Control Register (SW_MUX_CTL_PAD_SAI1_TXC)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SAI1_TXC	1A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SAI1_TXC
3 —	- Reserved
2-0	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SAI1_TXC.

Table continues on the next page...

Table continued from the previous page...

Field	Description
MUX_MODE	000 - Select mux mode: ALT0 mux port: SAI1_TX_BCLK of instance: sai1 001 - Select mux mode: ALT1 mux port: LPUART2_CTS_B of instance: lpuart2 010 - Select mux mode: ALT2 mux port: LPSPI1_SIN of instance: lpspi1 011 - Select mux mode: ALT3 mux port: LPUART1_DSR_B of instance: lpuart1 100 - Select mux mode: ALT4 mux port: CAN1_RX of instance: can1 101 - Select mux mode: ALT5 mux port: GPIO1_IO12 of instance: gpio1

27.3.1.107 SW_MUX_CTL_PAD_SAI1_TXD0 SW MUX Control Register (SW_MUX_CTL_PAD_SAI1_TXD0)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SAI1_TXD0	1A4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Force input path of pad SAI1_TXD0
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SAI1_TXD0. 000 - Select mux mode: ALT0 mux port: SAI1_TX_DATA00 of instance: sai1 001 - Select mux mode: ALT1 mux port: LPUART2_RTS_B of instance: lpuart2 010 - Select mux mode: ALT2 mux port: LPSPI1_SCK of instance: lpspi1 011 - Select mux mode: ALT3 mux port: LPUART1_DTR_B of instance: lpuart1 100 - Select mux mode: ALT4 mux port: CAN1_TX of instance: can1 101 - Select mux mode: ALT5 mux port: GPIO1_IO13 of instance: gpio1

27.3.1.108 SW_MUX_CTL_PAD_SAI1_RXD0 SW MUX Control Register (SW_MUX_CTL_PAD_SAI1_RXD0)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_SAI1_RXD0	1A8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad SAI1_RXD0
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SAI1_RXD0. 000 - Select mux mode: ALT0 mux port: SAI1_RX_DATA00 of instance: sai1 001 - Select mux mode: ALT1 mux port: SAI1_MCLK of instance: sai1 010 - Select mux mode: ALT2 mux port: LPSPI1_SOUT of instance: lpspi1 011 - Select mux mode: ALT3 mux port: LPUART2_DSR_B of instance: lpuart2 100 - Select mux mode: ALT4 mux port: MQS1_RIGHT of instance: mqs1 101 - Select mux mode: ALT5 mux port: GPIO1_IO14 of instance: gpio1

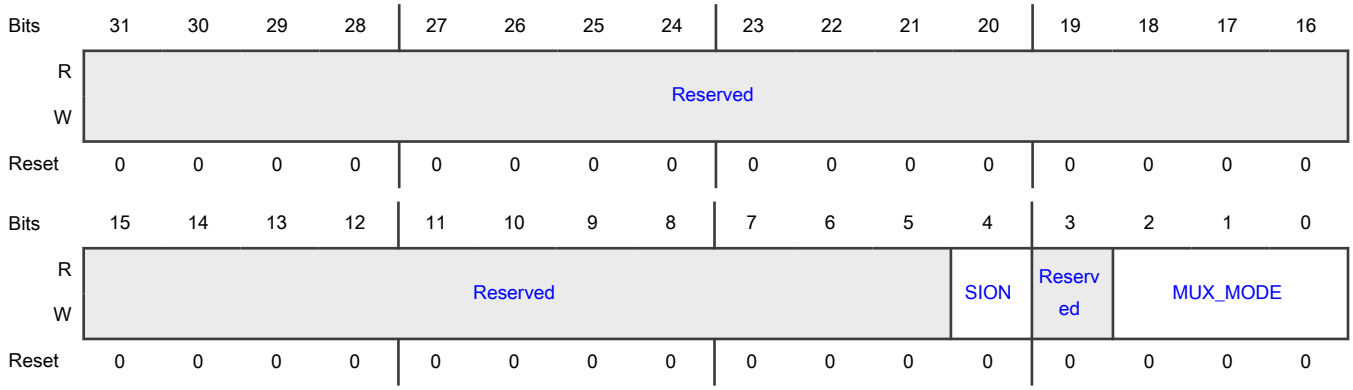
27.3.1.109 SW_MUX_CTL_PAD_WDOG_ANY SW MUX Control Register (SW_MUX_CTL_PAD_WDOG_ANY)

SW_MUX_CTL Register

Offset

Register	Offset
SW_MUX_CTL_PAD_WDOG_ANY	1ACh

Diagram



Fields

Field	Description
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0 - Input Path is determined by functionality 1 - Force input path of pad WDOG_ANY
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: WDOG_ANY. 000 - Select mux mode: ALT0 mux port: WDOG1_WDOG_ANY of instance: wdog1 101 - Select mux mode: ALT5 mux port: GPIO1_IO15 of instance: gpio1

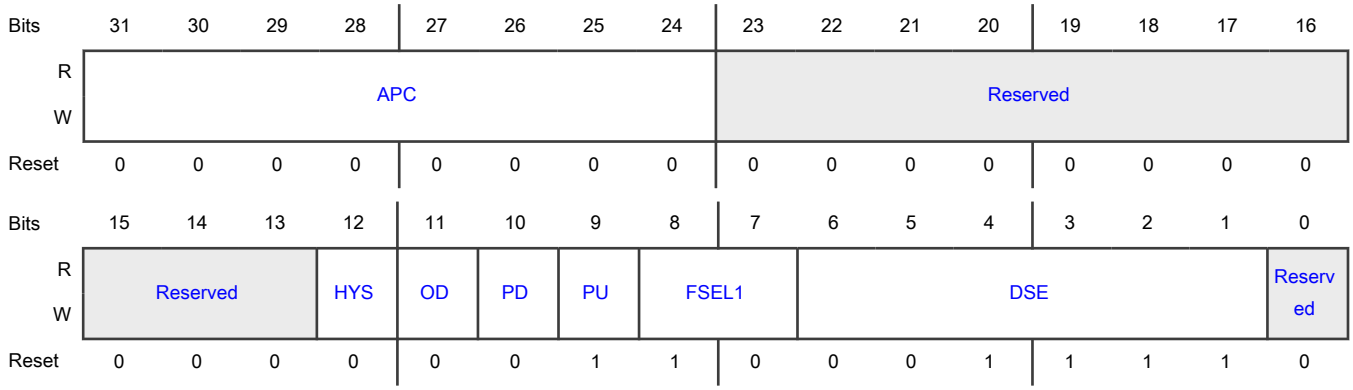
27.3.1.110 SW_PAD_CTL_PAD_DAP_TDI SW PAD Control Register (SW_PAD_CTL_PAD_DAP_TDI)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_DAP_TDI	1B0h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: DAP_TDI 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: DAP_TDI 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: DAP_TDI 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: DAP_TDI 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: DAP_TDI 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: DAP_TDI 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

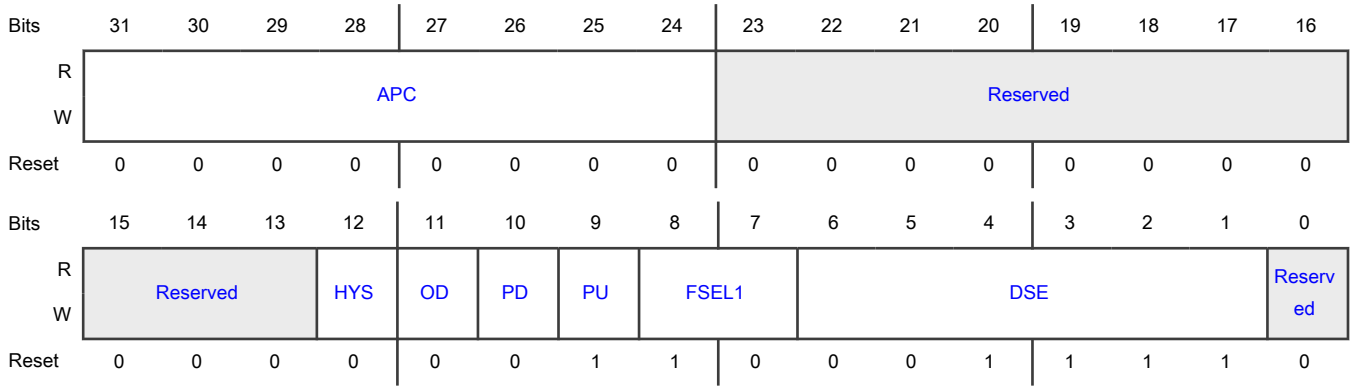
27.3.1.111 SW_PAD_CTL_PAD_DAP_TMS_SWDIO SW PAD Control Register (SW_PAD_CTL_PAD_DAP_TMS_SWDIO)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_DAP_TMS_SWDIO	1B4h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: DAP_TMS_SWDIO 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: DAP_TMS_SWDIO 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: DAP_TMS_SWDIO 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: DAP_TMS_SWDIO 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: DAP_TMS_SWDIO 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: DAP_TMS_SWDIO 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

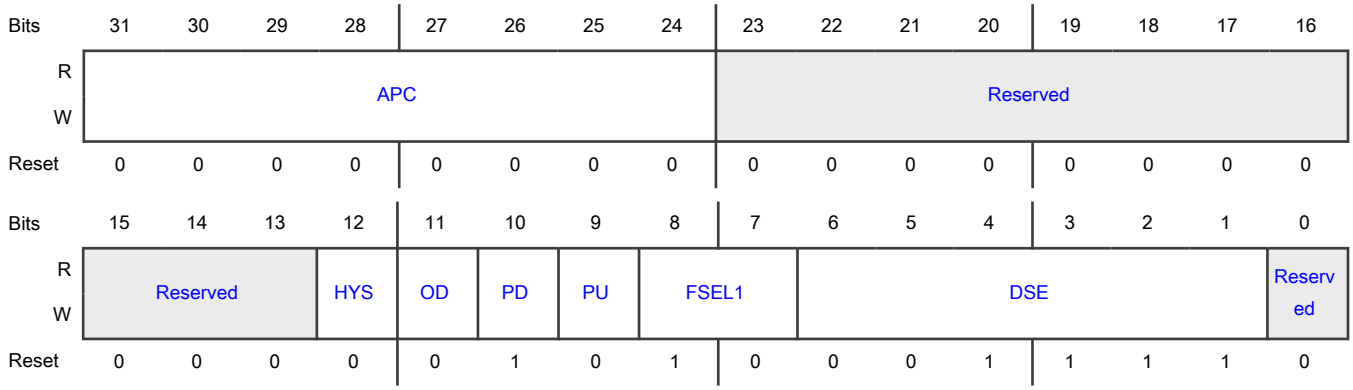
27.3.1.112 SW_PAD_CTL_PAD_DAP_TCLK_SWCLK SW PAD Control Register (SW_PAD_CTL_PAD_DAP_TCLK_SWCLK)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_DAP_TCLK_SWCLK	1B8h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: DAP_TCLK_SWCLK 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: DAP_TCLK_SWCLK 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: DAP_TCLK_SWCLK 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: DAP_TCLK_SWCLK 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: DAP_TCLK_SWCLK 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: DAP_TCLK_SWCLK 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

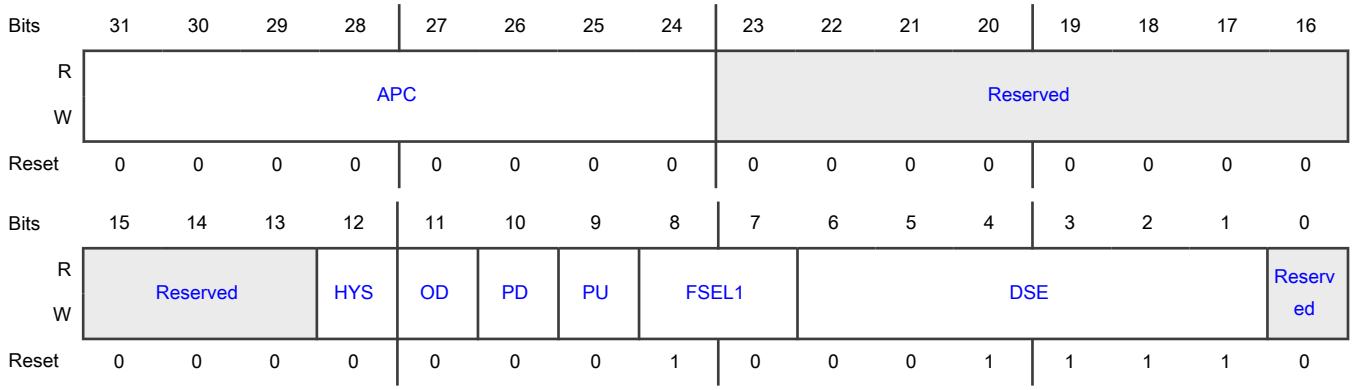
27.3.1.113 SW_PAD_CTL_PAD_DAP_TDO_TRACESWO SW PAD Control Register (SW_PAD_CTL_PAD_DAP_TDO_TRACESWO)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_DAP_TDO_TRACESWO	1BCh

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: DAP_TDO_TRACESWO 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: DAP_TDO_TRACESWO 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: DAP_TDO_TRACESWO 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: DAP_TDO_TRACESWO 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: DAP_TDO_TRACESWO 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: DAP_TDO_TRACESWO 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

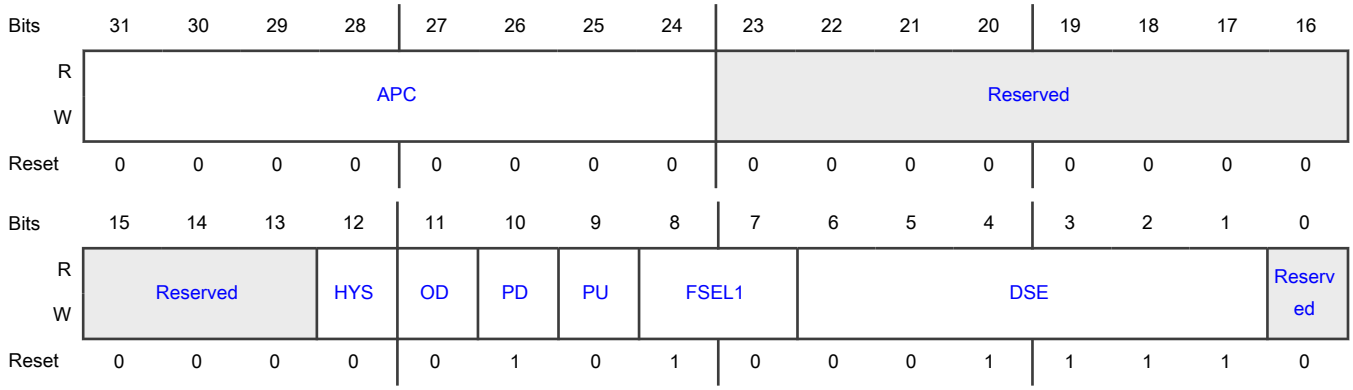
27.3.1.114 SW_PAD_CTL_PAD_GPIO_IO00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO00)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO00	1C0h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO00 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO00 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO00 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO00 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO00 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO00 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

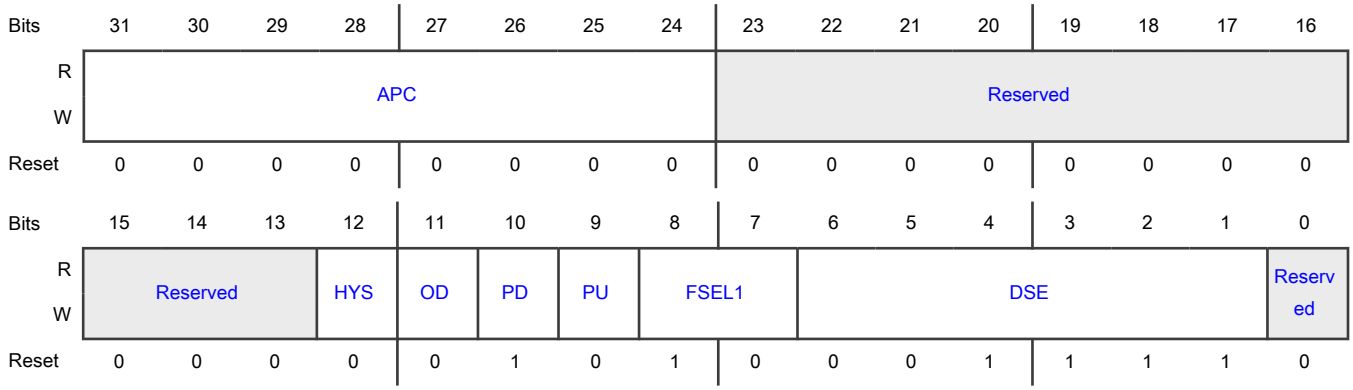
27.3.1.115 SW_PAD_CTL_PAD_GPIO_IO01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO01)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO01	1C4h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO01 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO01 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO01 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO01 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO01 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO01 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

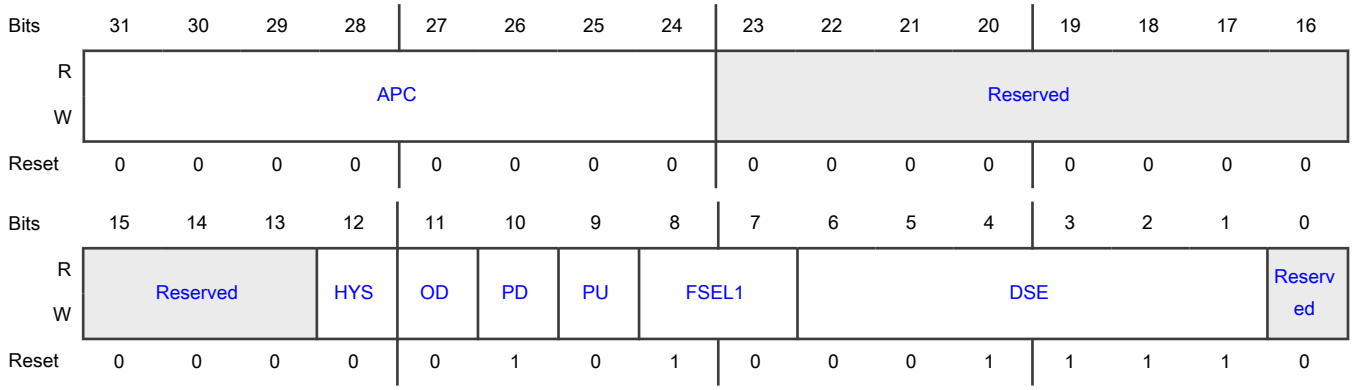
27.3.1.116 SW_PAD_CTL_PAD_GPIO_IO02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO02)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO02	1C8h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO02 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO02 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO02 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO02 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO02 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO02 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

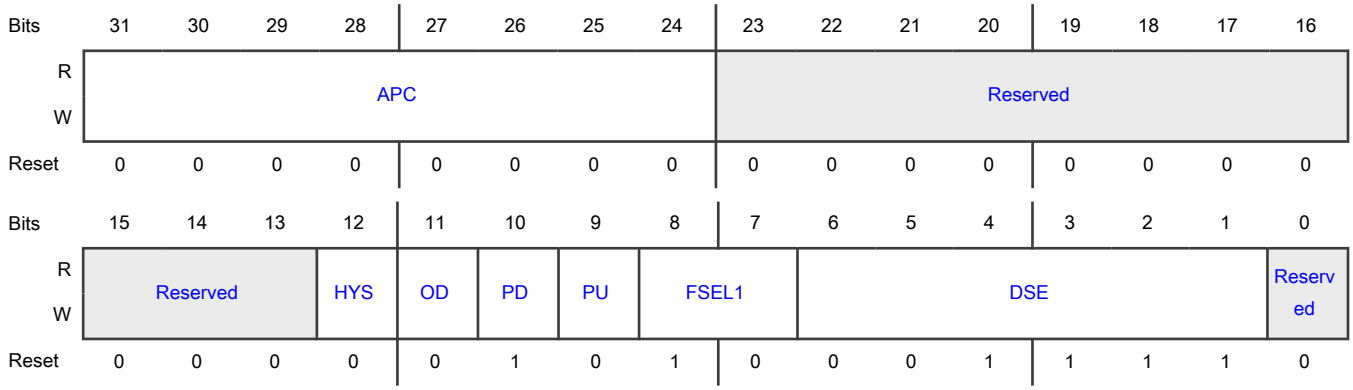
27.3.1.117 SW_PAD_CTL_PAD_GPIO_IO03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO03)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO03	1CCh

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO03 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO03 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO03 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO03 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO03 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO03 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

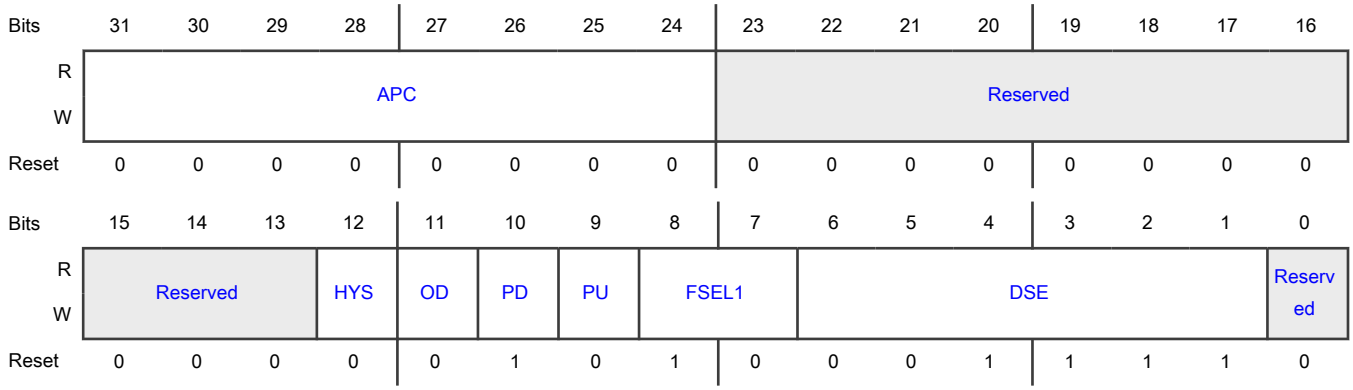
27.3.1.118 SW_PAD_CTL_PAD_GPIO_IO04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO04)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO04	1D0h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO04 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO04 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO04 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO04 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO04 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO04 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

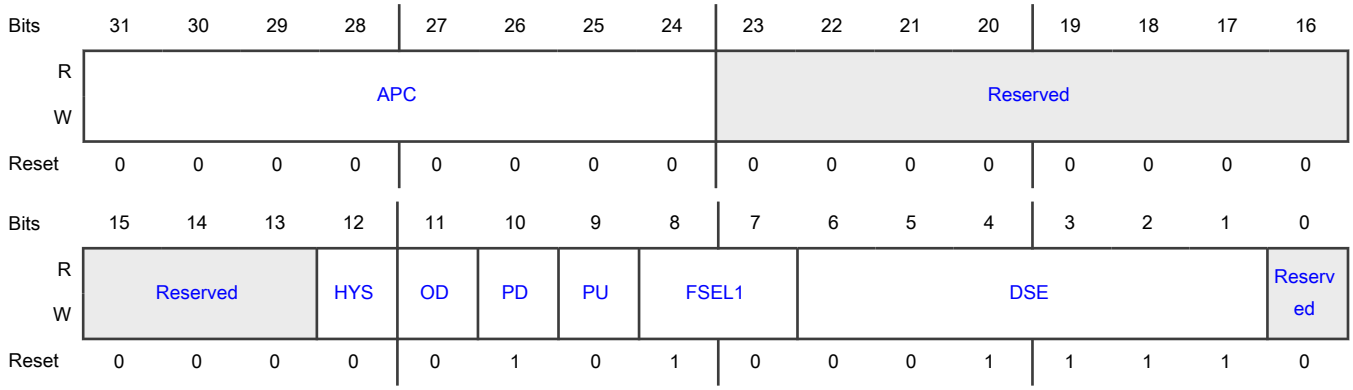
27.3.1.119 SW_PAD_CTL_PAD_GPIO_IO05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO05)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO05	1D4h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO05 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO05 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO05 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO05 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO05 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO05 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

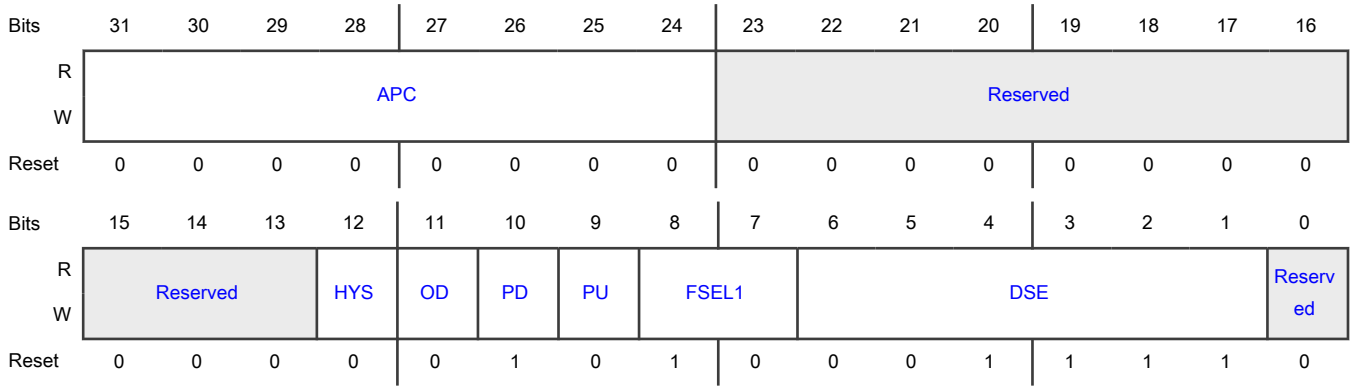
27.3.1.120 SW_PAD_CTL_PAD_GPIO_IO06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO06)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO06	1D8h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO06 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO06 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO06 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO06 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO06 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO06 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

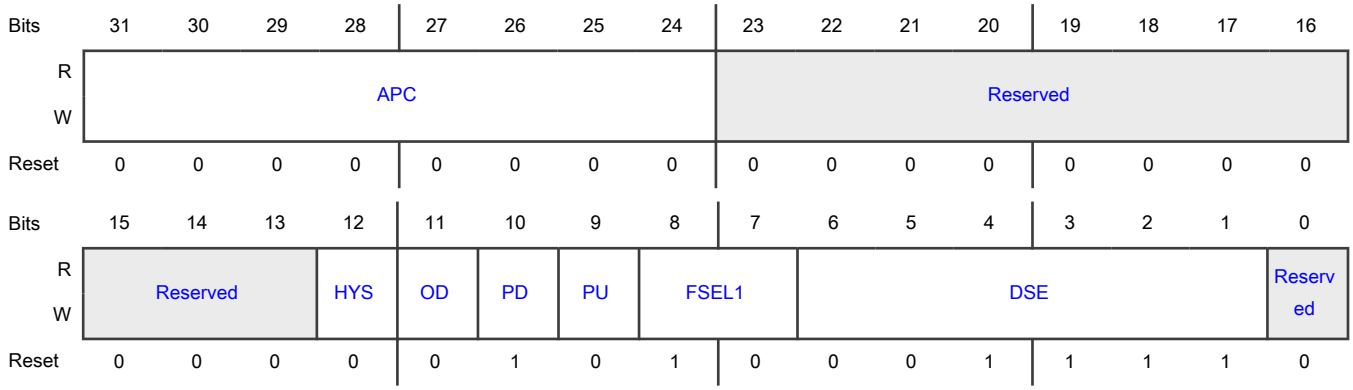
27.3.1.121 SW_PAD_CTL_PAD_GPIO_IO07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO07)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO07	1DCh

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO07 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO07 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO07 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO07 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO07 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO07 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

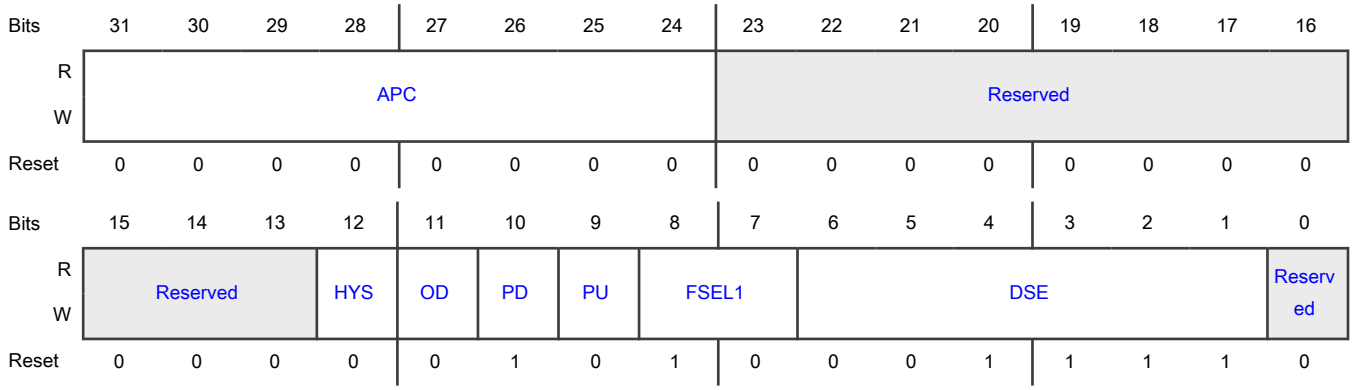
27.3.1.122 SW_PAD_CTL_PAD_GPIO_IO08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO08)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO08	1E0h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO08 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO08 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO08 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO08 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO08 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO08 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

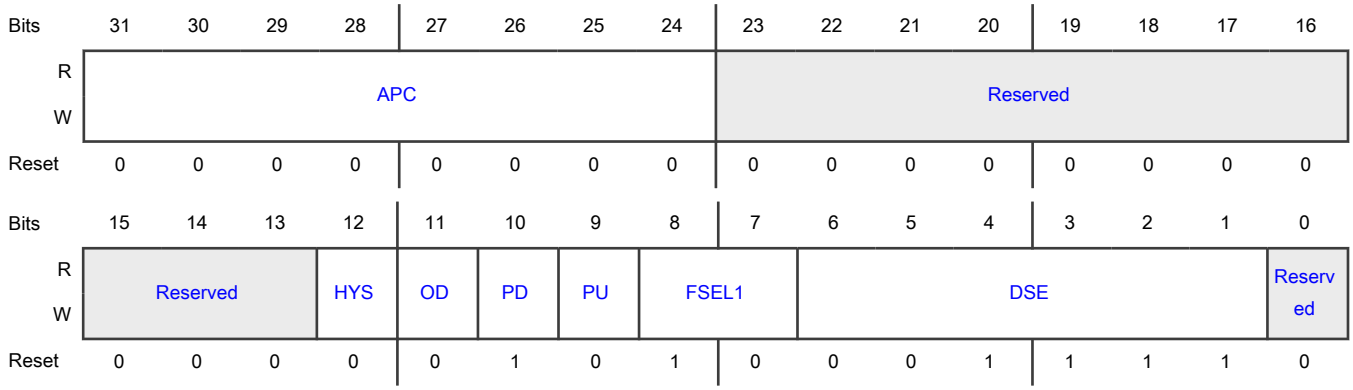
27.3.1.123 SW_PAD_CTL_PAD_GPIO_IO09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO09)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO09	1E4h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO09 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO09 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO09 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO09 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO09 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO09 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

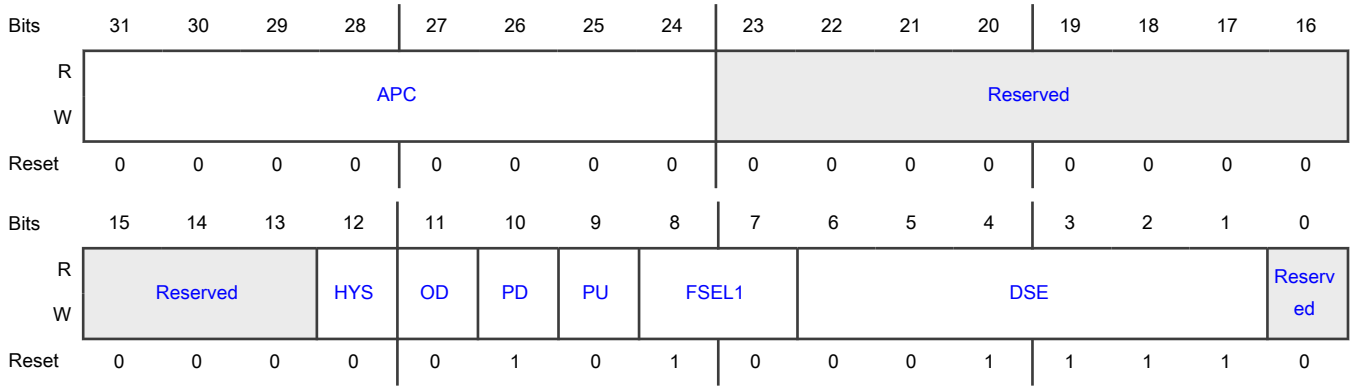
27.3.1.124 SW_PAD_CTL_PAD_GPIO_IO10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO10)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO10	1E8h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO10 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO10 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO10 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO10 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO10 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO10 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

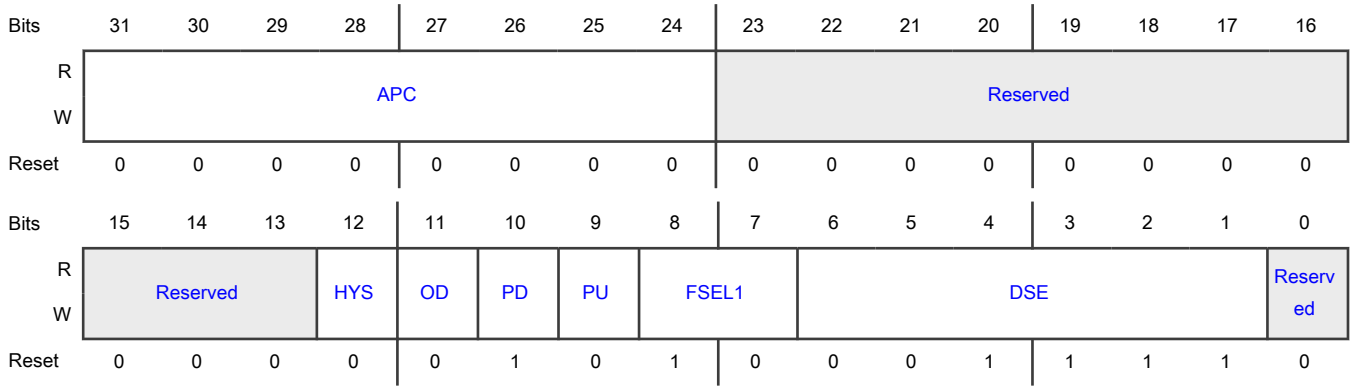
27.3.1.125 SW_PAD_CTL_PAD_GPIO_IO11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO11)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO11	1ECh

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO11 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO11 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO11 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO11 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO11 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO11 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

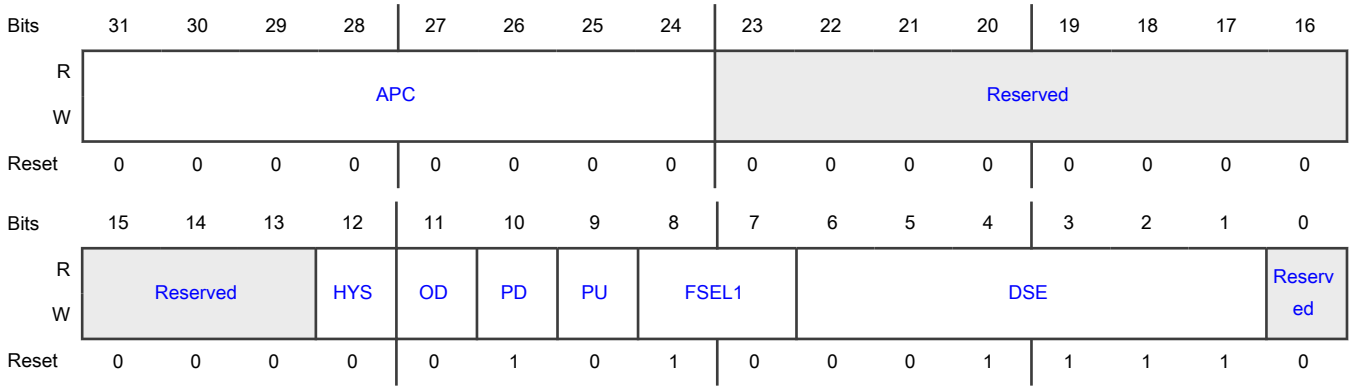
27.3.1.126 SW_PAD_CTL_PAD_GPIO_IO12 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO12)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO12	1F0h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO12 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO12 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO12 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO12 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO12 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO12 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

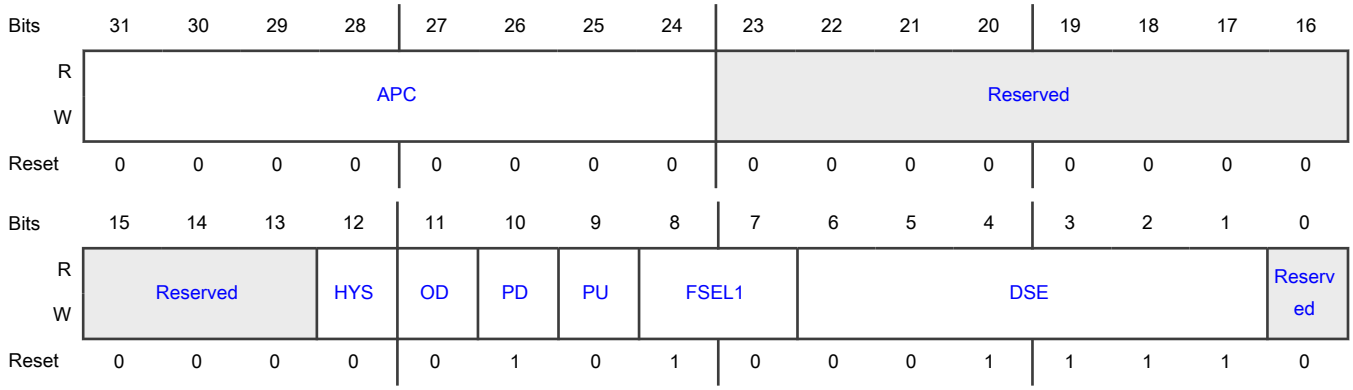
27.3.1.127 SW_PAD_CTL_PAD_GPIO_IO13 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO13)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO13	1F4h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO13 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO13 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO13 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO13 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO13 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO13 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

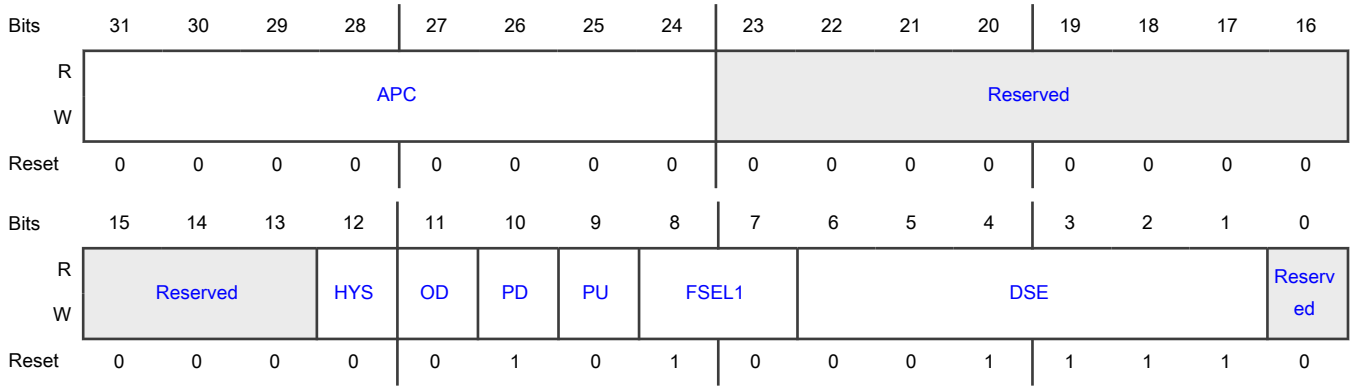
27.3.1.128 SW_PAD_CTL_PAD_GPIO_IO14 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO14)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO14	1F8h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO14 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO14 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO14 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO14 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO14 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO14 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

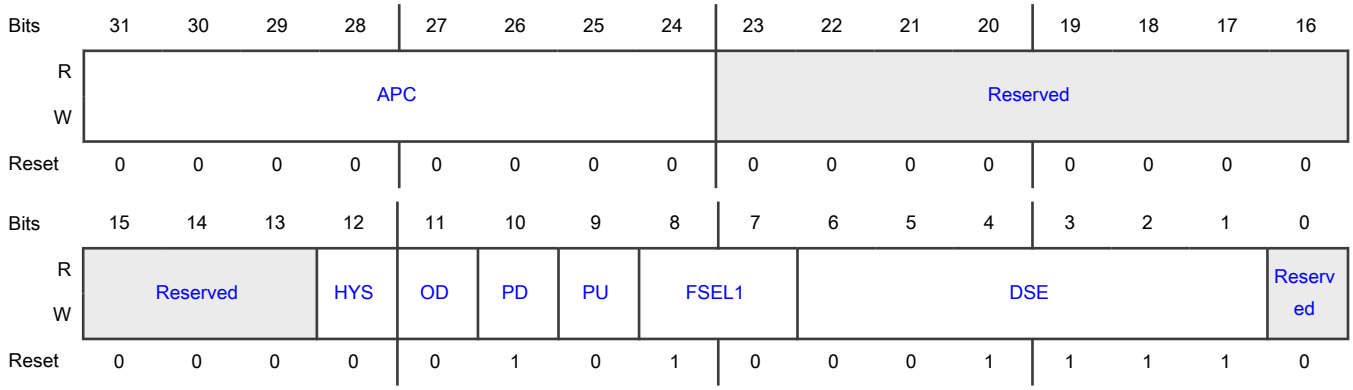
27.3.1.129 SW_PAD_CTL_PAD_GPIO_IO15 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO15)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO15	1FCh

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO15 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO15 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO15 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO15 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO15 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO15 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

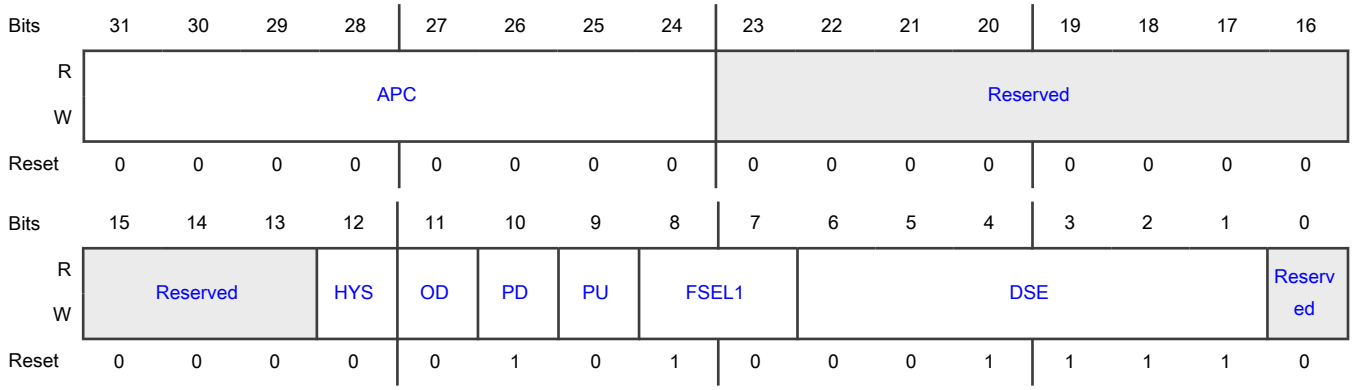
27.3.1.130 SW_PAD_CTL_PAD_GPIO_IO16 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO16)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO16	200h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO16 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO16 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO16 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO16 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO16 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO16 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

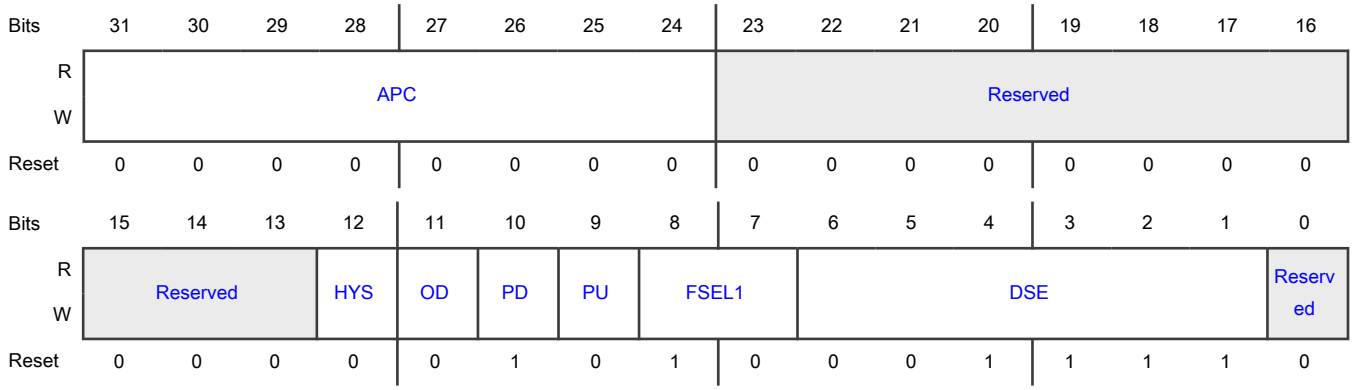
27.3.1.131 SW_PAD_CTL_PAD_GPIO_IO17 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO17)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO17	204h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO17 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO17 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO17 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO17 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO17 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO17 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

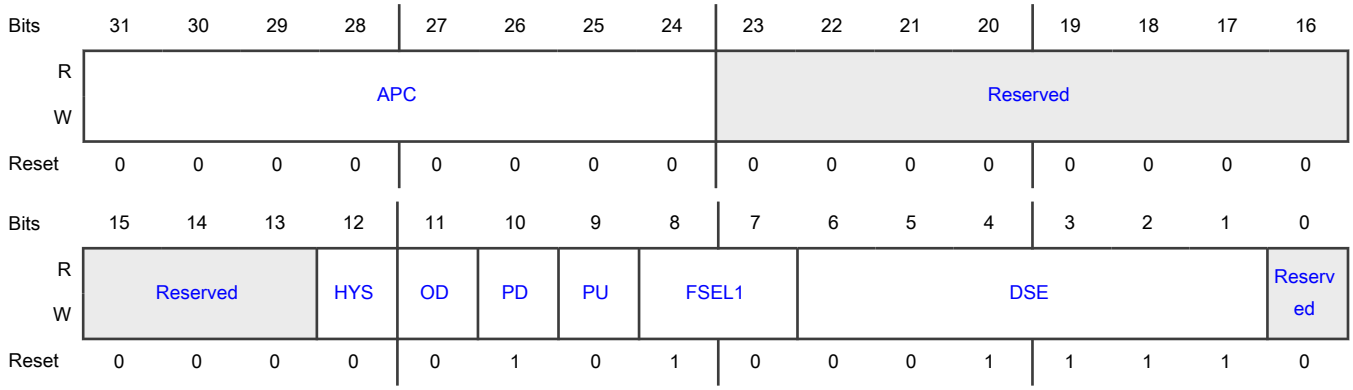
27.3.1.132 SW_PAD_CTL_PAD_GPIO_IO18 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO18)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO18	208h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO18 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO18 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO18 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO18 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO18 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO18 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

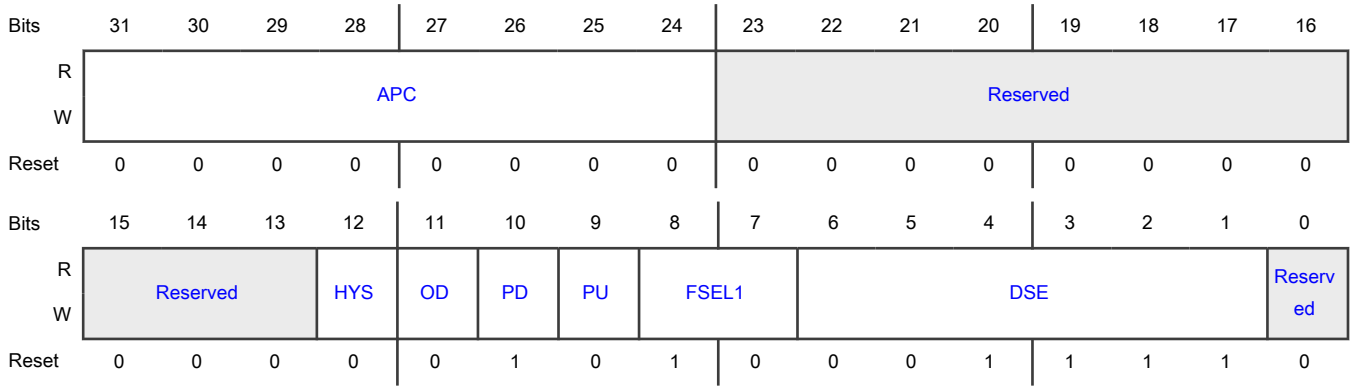
27.3.1.133 SW_PAD_CTL_PAD_GPIO_IO19 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO19)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO19	20Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO19 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO19 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO19 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO19 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO19 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO19 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

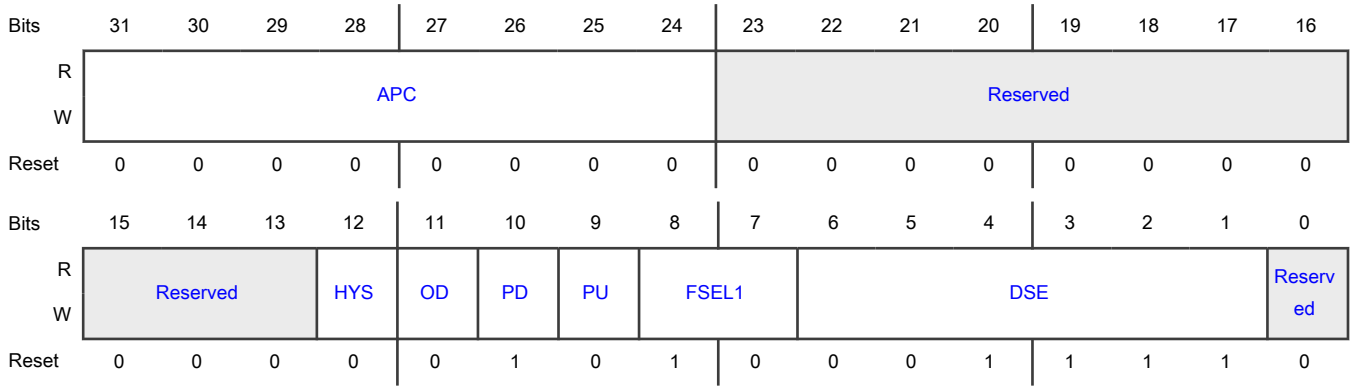
27.3.1.134 SW_PAD_CTL_PAD_GPIO_IO20 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO20)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO20	210h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO20 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO20 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO20 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO20 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO20 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO20 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

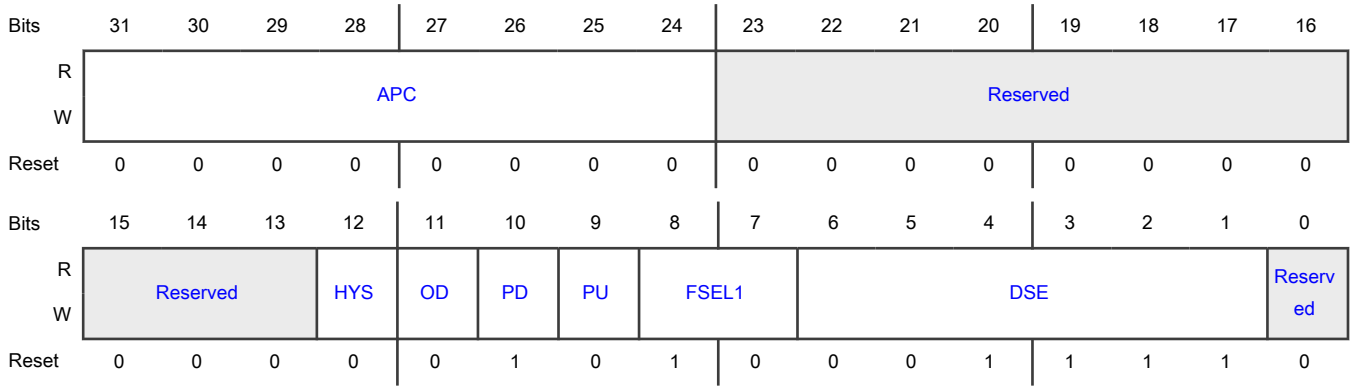
27.3.1.135 SW_PAD_CTL_PAD_GPIO_IO21 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO21)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO21	214h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO21 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO21 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO21 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO21 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO21 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO21 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

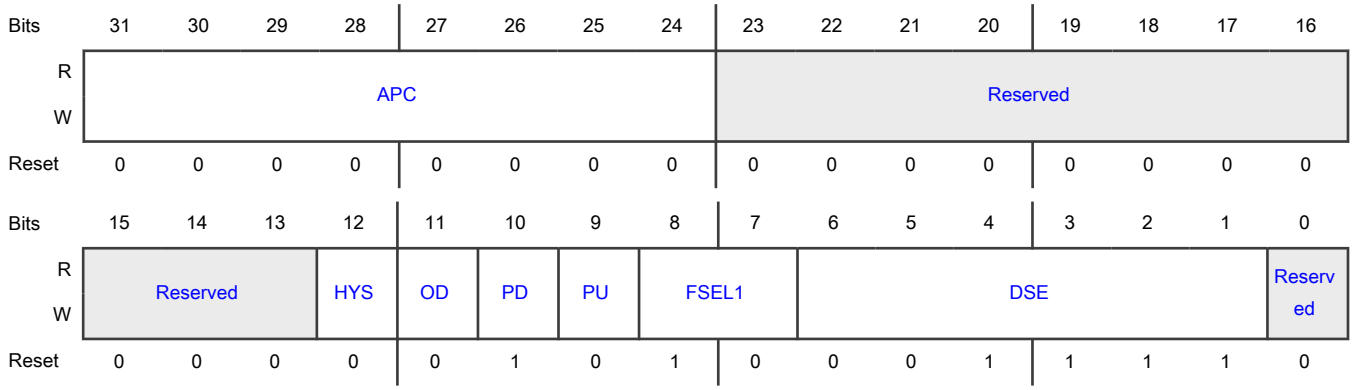
27.3.1.136 SW_PAD_CTL_PAD_GPIO_IO22 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO22)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO22	218h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO22 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO22 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO22 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO22 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO22 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO22 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

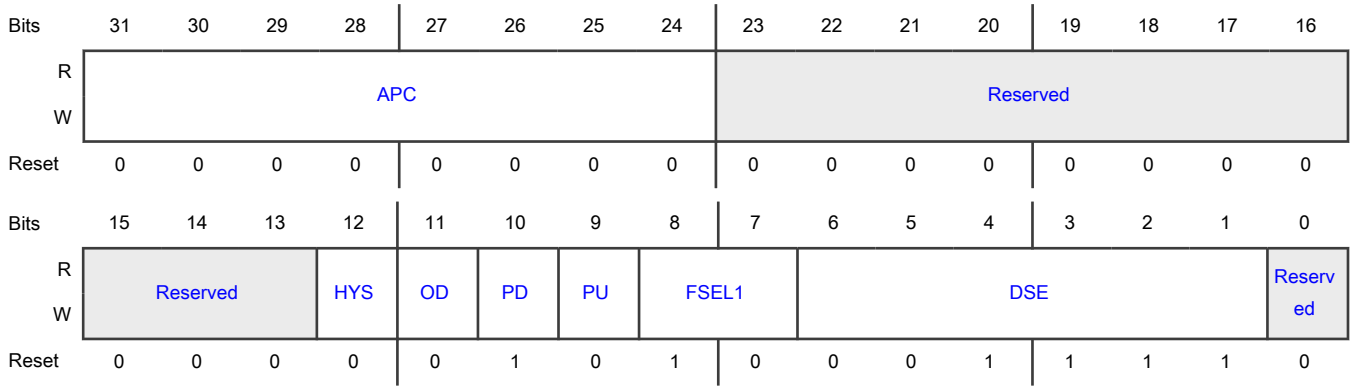
27.3.1.137 SW_PAD_CTL_PAD_GPIO_IO23 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO23)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO23	21Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO23 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO23 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO23 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO23 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO23 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO23 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

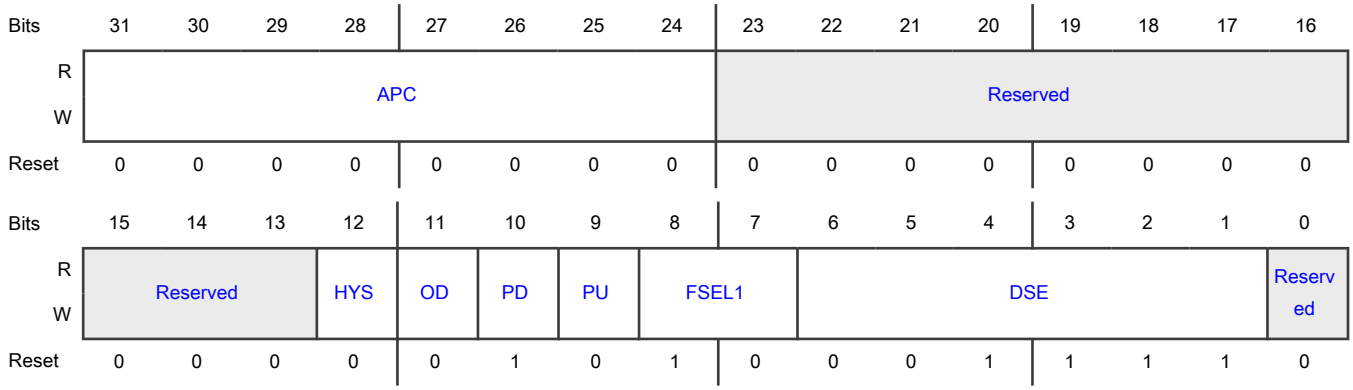
27.3.1.138 SW_PAD_CTL_PAD_GPIO_IO24 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO24)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO24	220h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO24 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO24 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO24 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO24 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO24 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO24 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

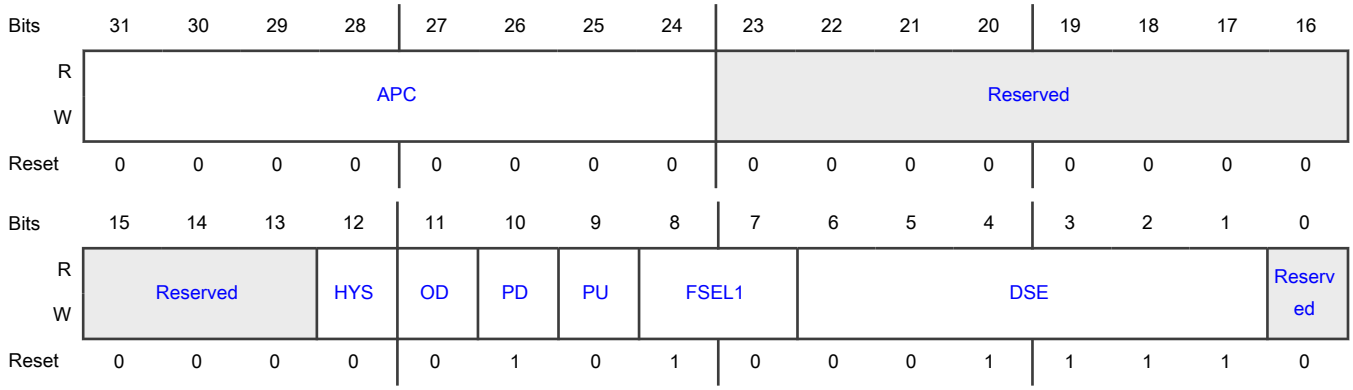
27.3.1.139 SW_PAD_CTL_PAD_GPIO_IO25 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO25)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO25	224h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO25 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO25 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO25 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO25 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO25 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO25 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

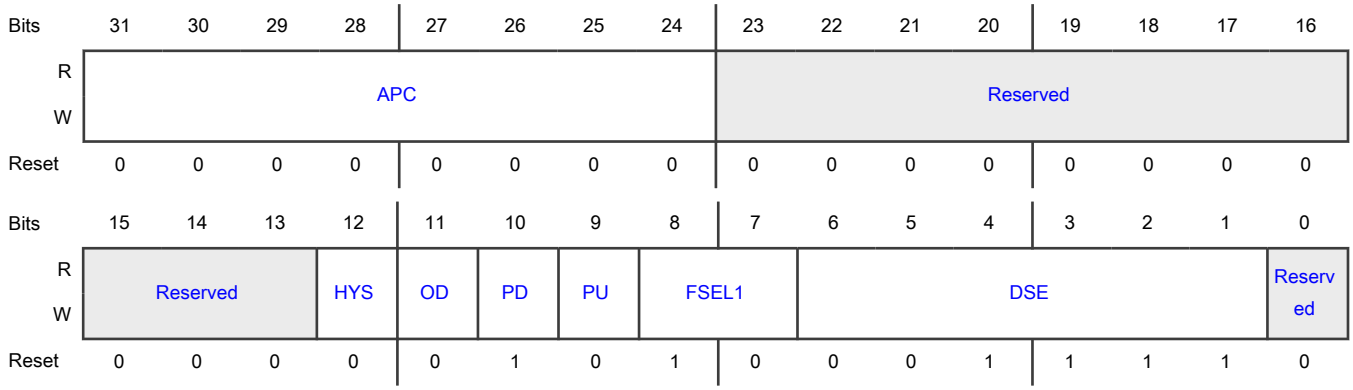
27.3.1.140 SW_PAD_CTL_PAD_GPIO_IO26 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO26)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO26	228h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO26 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO26 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO26 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO26 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO26 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO26 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

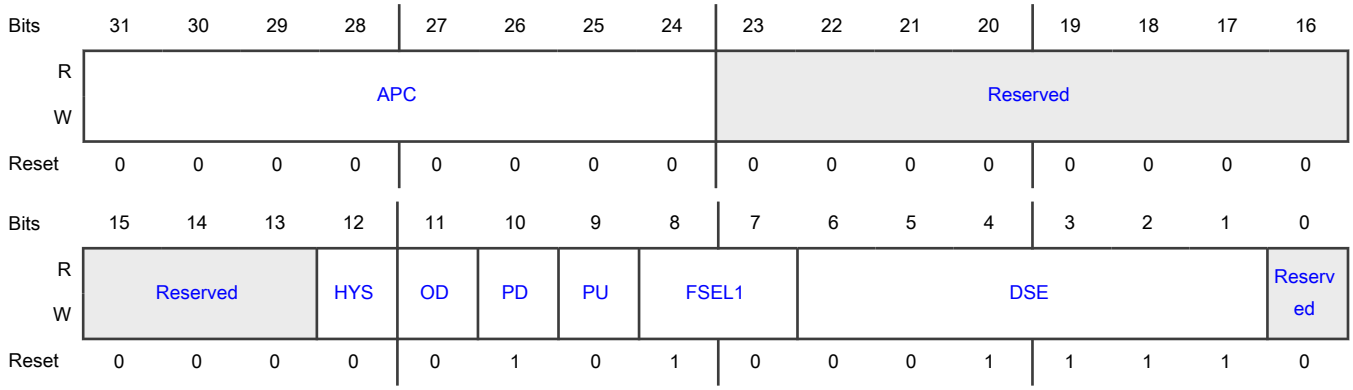
27.3.1.141 SW_PAD_CTL_PAD_GPIO_IO27 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO27)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO27	22Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO27 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO27 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO27 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO27 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO27 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO27 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

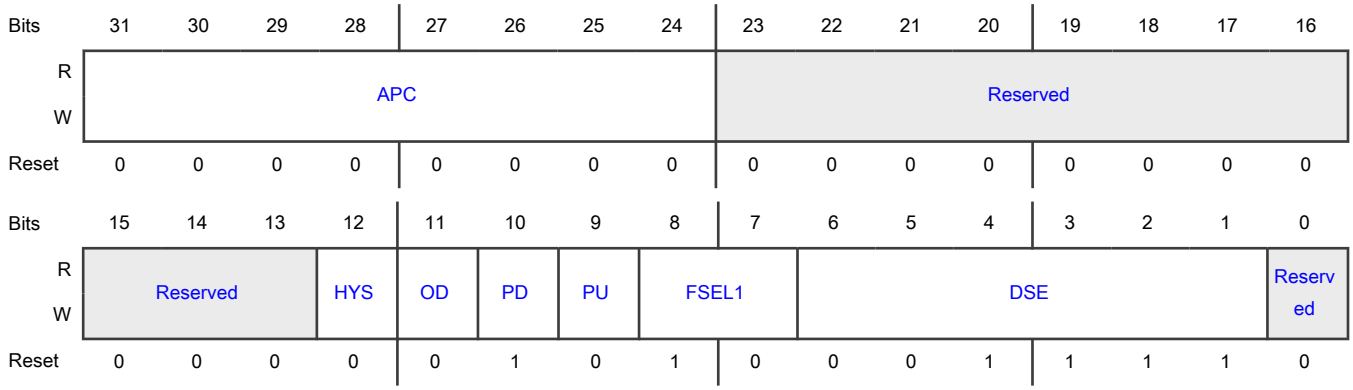
27.3.1.142 SW_PAD_CTL_PAD_GPIO_IO28 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO28)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO28	230h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO28 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO28 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO28 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO28 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO28 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO28 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

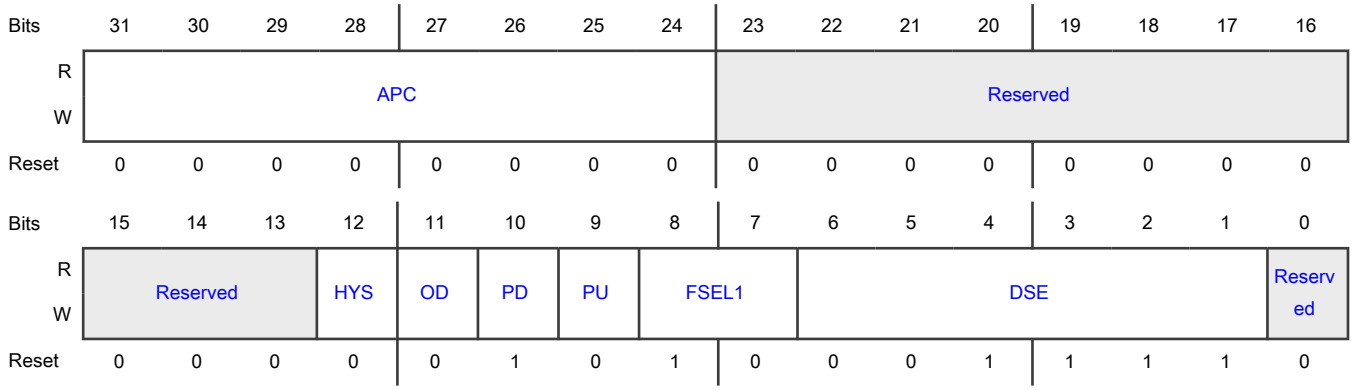
27.3.1.143 SW_PAD_CTL_PAD_GPIO_IO29 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_IO29)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_IO29	234h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: GPIO_IO29 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: GPIO_IO29 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: GPIO_IO29 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: GPIO_IO29 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: GPIO_IO29 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_IO29 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

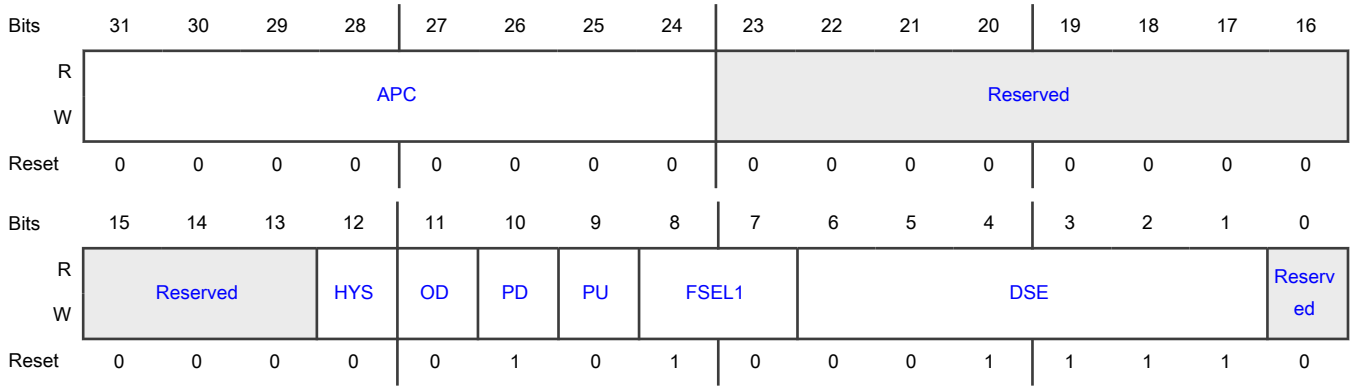
27.3.1.144 SW_PAD_CTL_PAD_CCM_CLKO1 SW PAD Control Register (SW_PAD_CTL_PAD_CCM_CLKO1)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_CCM_CLKO1	238h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: CCM_CLKO1 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: CCM_CLKO1 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: CCM_CLKO1 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: CCM_CLKO1 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: CCM_CLKO1 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: CCM_CLKO1 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

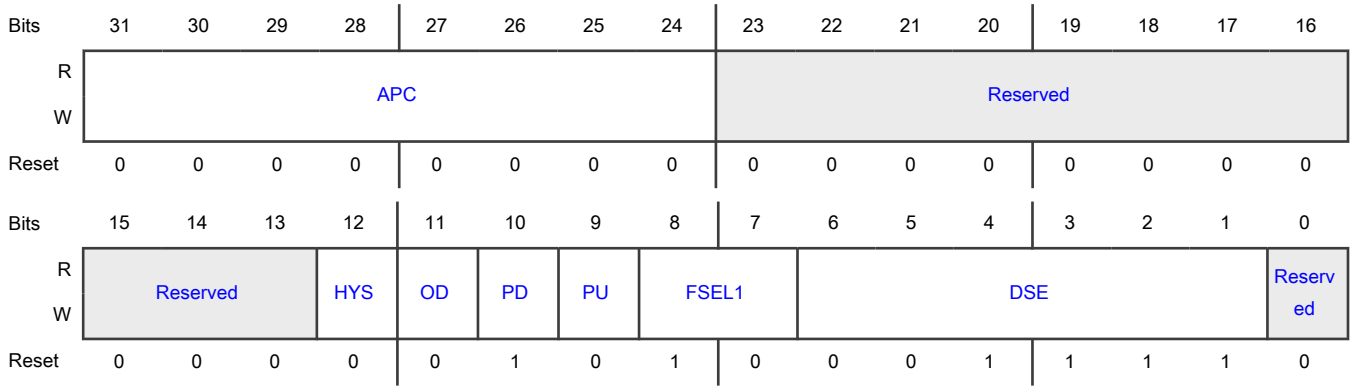
27.3.1.145 SW_PAD_CTL_PAD_CCM_CLKO2 SW PAD Control Register (SW_PAD_CTL_PAD_CCM_CLKO2)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_CCM_CLKO2	23Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: CCM_CLKO2 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: CCM_CLKO2 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: CCM_CLKO2 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: CCM_CLKO2 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: CCM_CLKO2 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: CCM_CLKO2 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

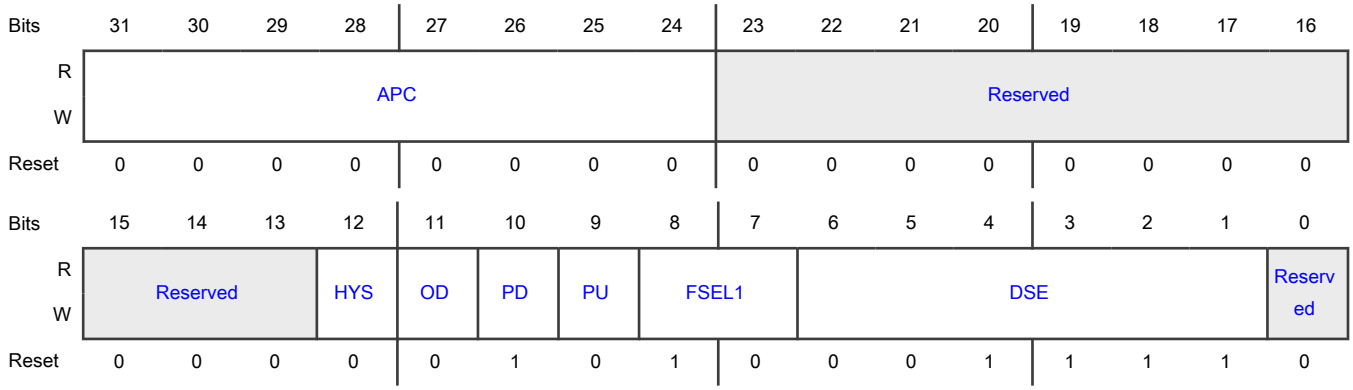
27.3.1.146 SW_PAD_CTL_PAD_CCM_CLKO3 SW PAD Control Register (SW_PAD_CTL_PAD_CCM_CLKO3)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_CCM_CLKO3	240h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: CCM_CLKO3 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: CCM_CLKO3 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: CCM_CLKO3 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: CCM_CLKO3 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: CCM_CLKO3 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: CCM_CLKO3 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

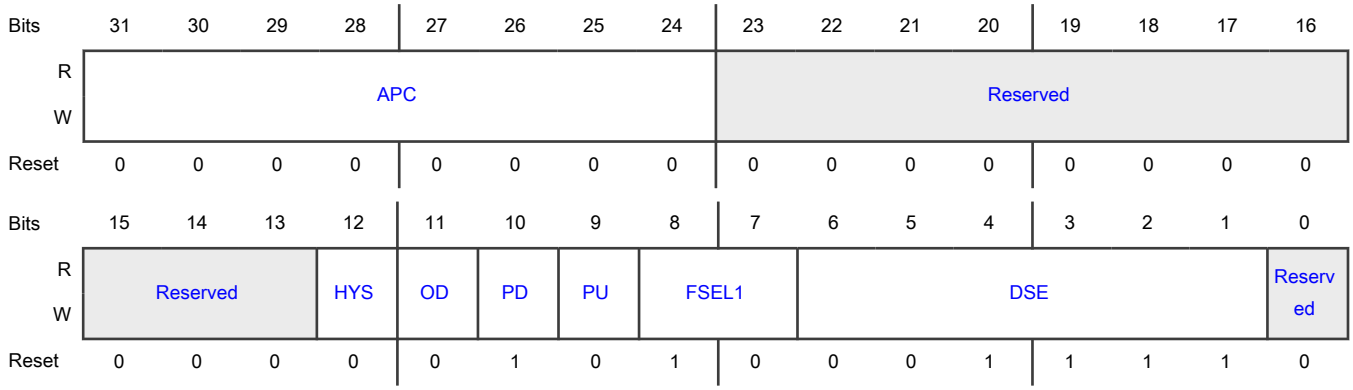
27.3.1.147 SW_PAD_CTL_PAD_CCM_CLKO4 SW PAD Control Register (SW_PAD_CTL_PAD_CCM_CLKO4)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_CCM_CLKO4	244h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: CCM_CLKO4 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: CCM_CLKO4 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: CCM_CLKO4 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: CCM_CLKO4 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: CCM_CLKO4 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: CCM_CLKO4 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

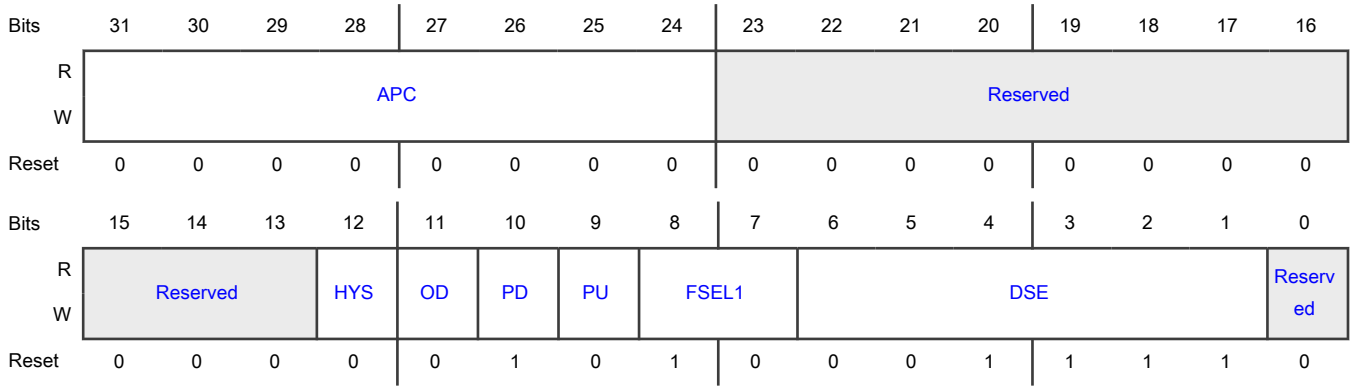
27.3.1.148 SW_PAD_CTL_PAD_ENET1_MDC SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_MDC)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_MDC	248h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_MDC 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_MDC 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_MDC 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_MDC 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_MDC 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_MDC 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

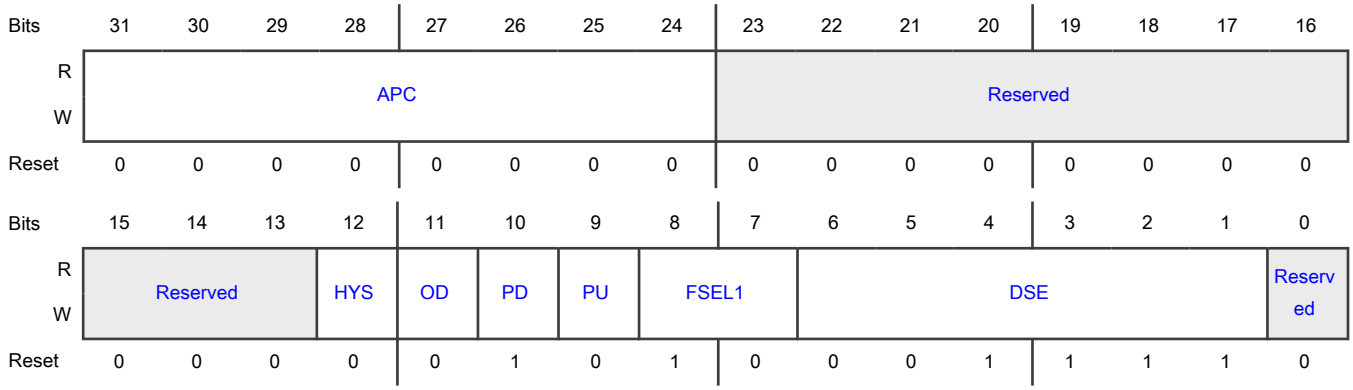
27.3.1.149 SW_PAD_CTL_PAD_ENET1_MDIO SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_MDIO)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_MDIO	24Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_MDIO 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_MDIO 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_MDIO 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_MDIO 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_MDIO 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_MDIO 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

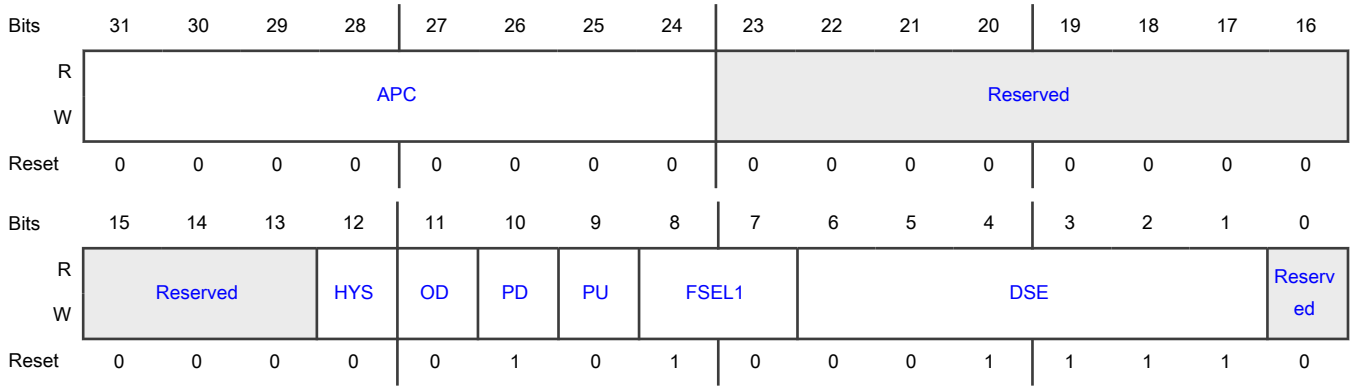
27.3.1.150 SW_PAD_CTL_PAD_ENET1_TD3 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_TD3)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_TD3	250h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_TD3 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_TD3 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_TD3 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_TD3 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_TD3 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_TD3 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

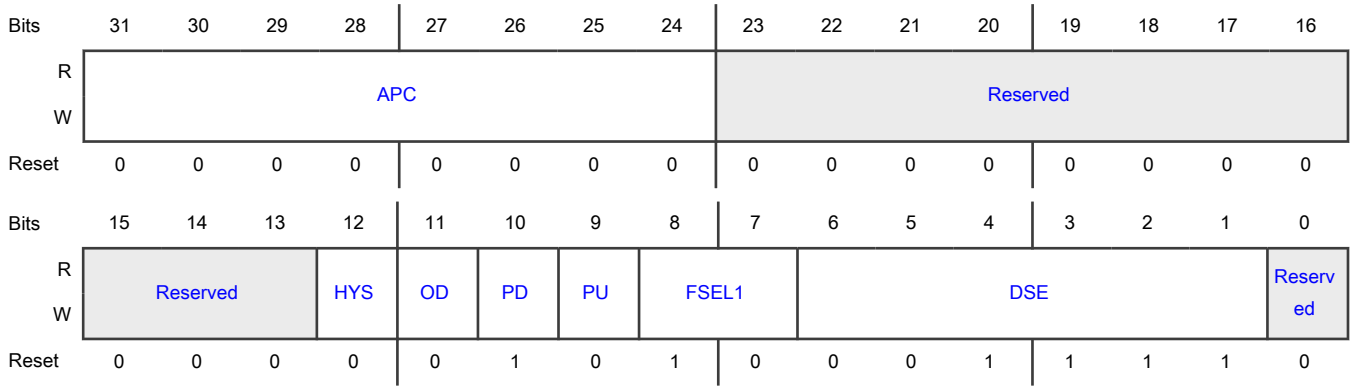
27.3.1.151 SW_PAD_CTL_PAD_ENET1_TD2 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_TD2)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_TD2	254h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_TD2 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_TD2 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_TD2 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_TD2 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_TD2 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_TD2 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

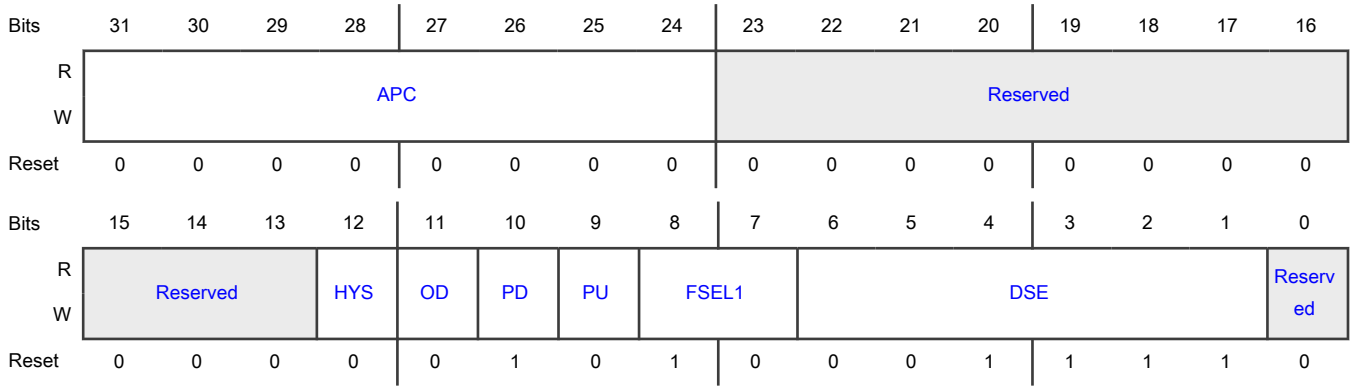
27.3.1.152 SW_PAD_CTL_PAD_ENET1_TD1 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_TD1)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_TD1	258h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_TD1 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_TD1 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_TD1 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_TD1 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_TD1 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_TD1 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

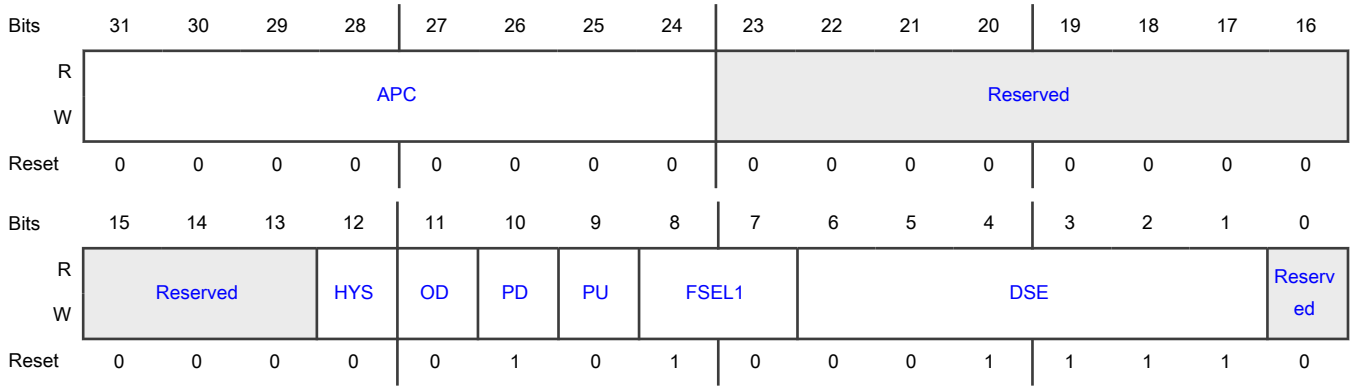
27.3.1.153 SW_PAD_CTL_PAD_ENET1_TD0 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_TD0)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_TD0	25Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_TD0 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_TD0 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_TD0 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_TD0 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_TD0 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_TD0 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

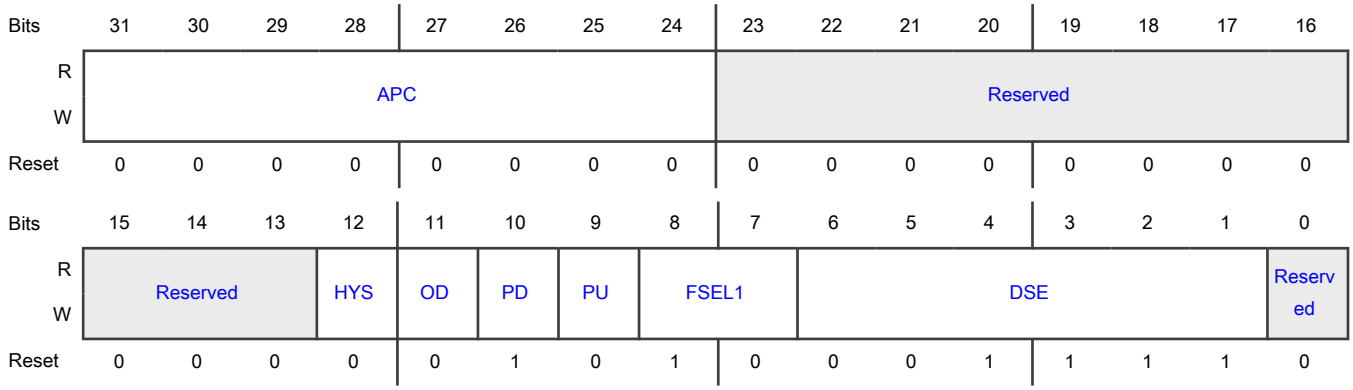
27.3.1.154 SW_PAD_CTL_PAD_ENET1_TX_CTL SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_TX_CTL)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_TX_CTL	260h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_TX_CTL 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_TX_CTL 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_TX_CTL 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_TX_CTL 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_TX_CTL 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_TX_CTL 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

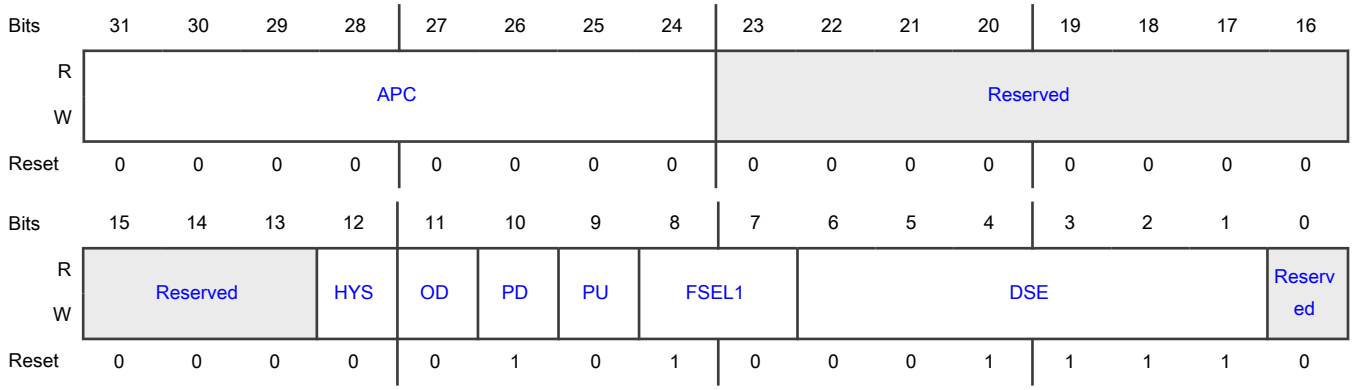
27.3.1.155 SW_PAD_CTL_PAD_ENET1_TXC SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_TXC)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_TXC	264h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_TXC 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_TXC 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_TXC 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_TXC 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_TXC 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_TXC 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

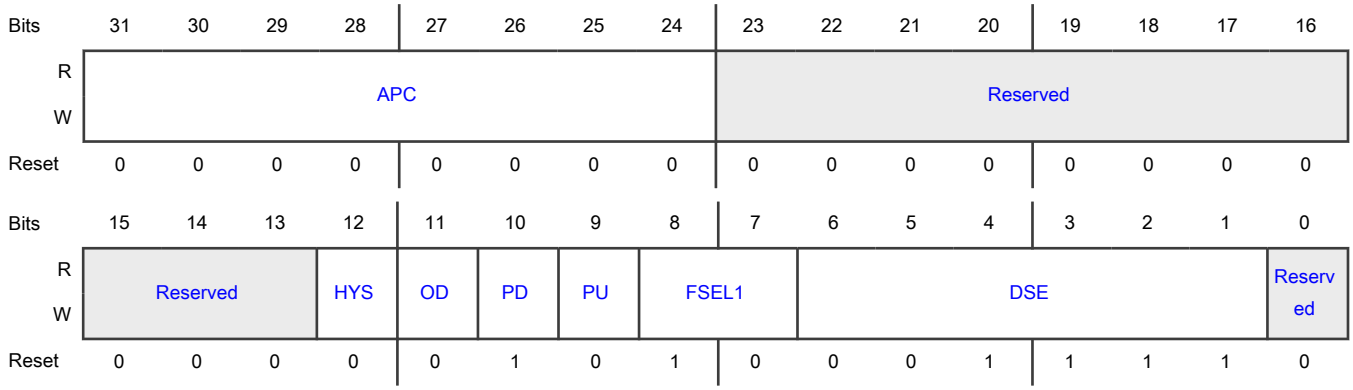
27.3.1.156 SW_PAD_CTL_PAD_ENET1_RX_CTL SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_RX_CTL)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_RX_CTL	268h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_RX_CTL 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_RX_CTL 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_RX_CTL 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_RX_CTL 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_RX_CTL 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_RX_CTL 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

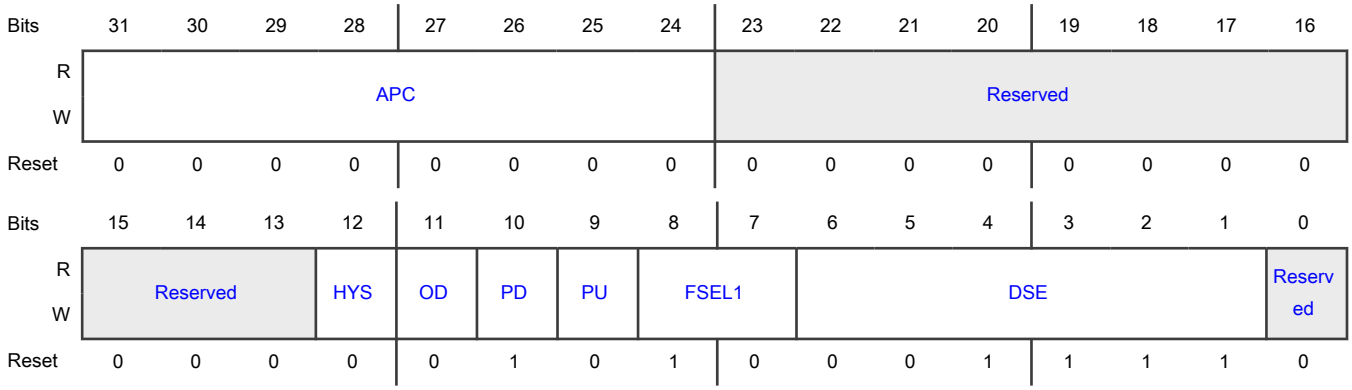
27.3.1.157 SW_PAD_CTL_PAD_ENET1_RXC SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_RXC)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_RXC	26Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_RXC 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_RXC 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_RXC 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_RXC 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_RXC 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_RXC 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

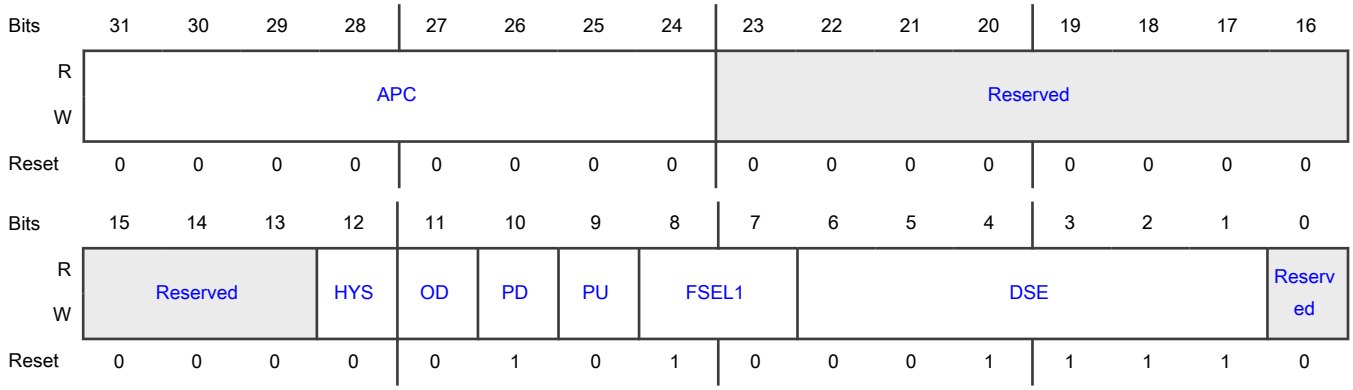
27.3.1.158 SW_PAD_CTL_PAD_ENET1_RD0 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_RD0)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_RD0	270h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_RD0 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_RD0 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_RD0 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_RD0 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_RD0 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_RD0 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

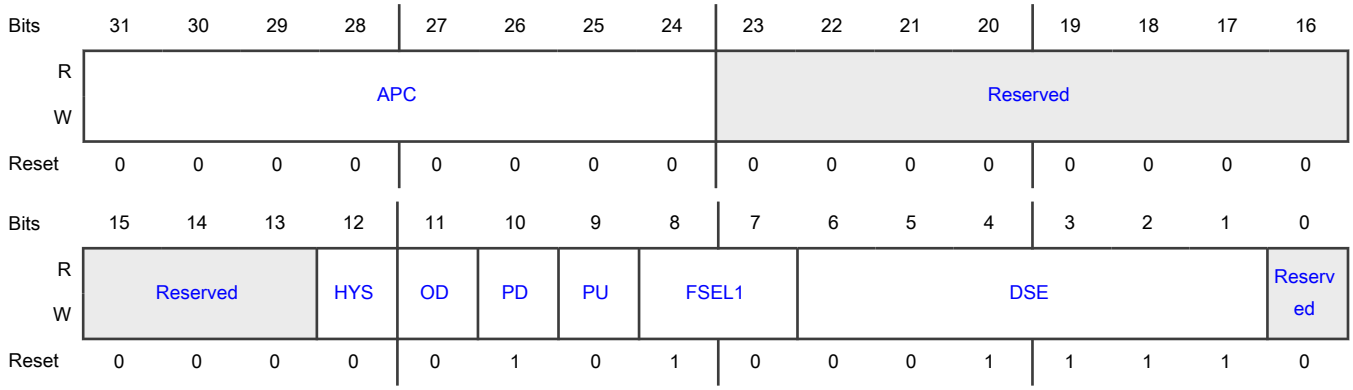
27.3.1.159 SW_PAD_CTL_PAD_ENET1_RD1 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_RD1)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_RD1	274h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_RD1 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_RD1 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_RD1 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_RD1 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_RD1 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_RD1 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

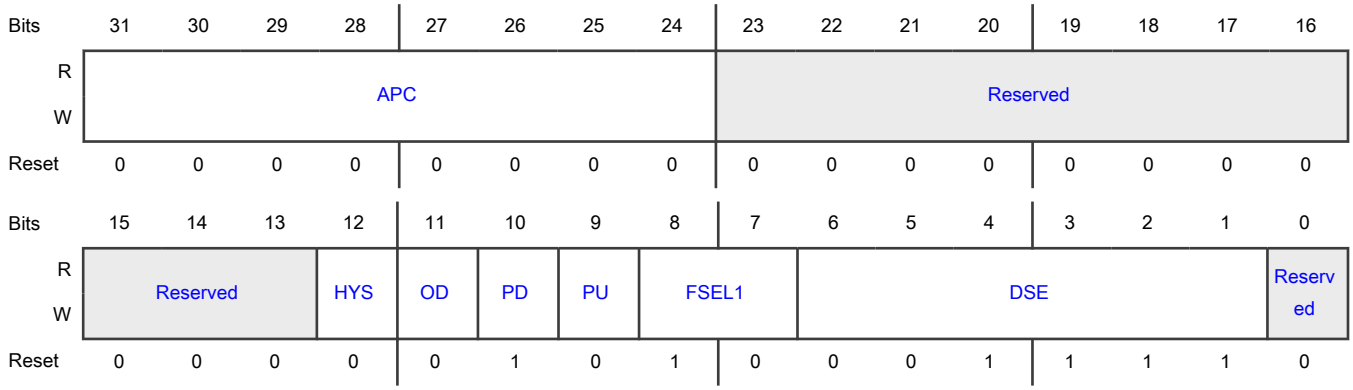
27.3.1.160 SW_PAD_CTL_PAD_ENET1_RD2 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_RD2)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_RD2	278h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_RD2 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_RD2 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_RD2 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_RD2 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_RD2 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_RD2 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

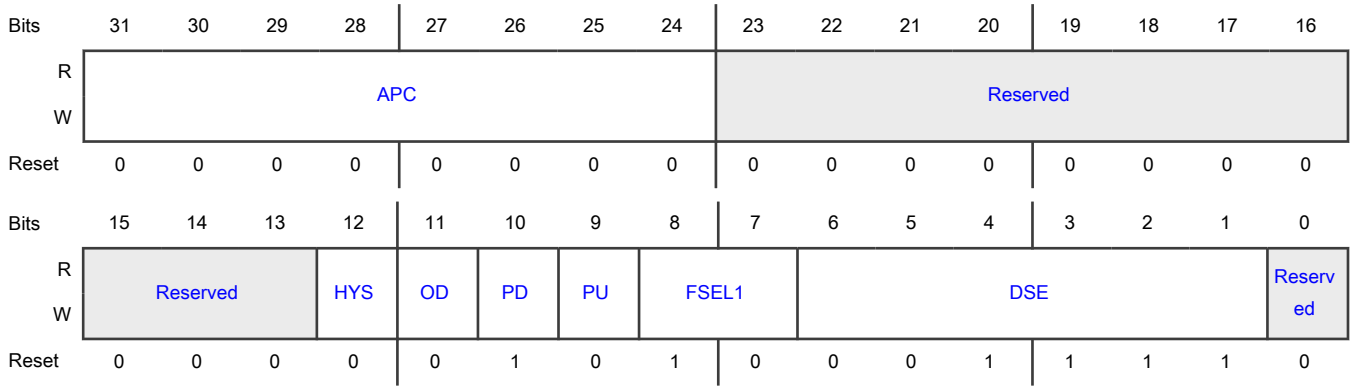
27.3.1.161 SW_PAD_CTL_PAD_ENET1_RD3 SW PAD Control Register (SW_PAD_CTL_PAD_ENET1_RD3)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET1_RD3	27Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET1_RD3 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET1_RD3 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET1_RD3 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET1_RD3 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET1_RD3 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET1_RD3 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

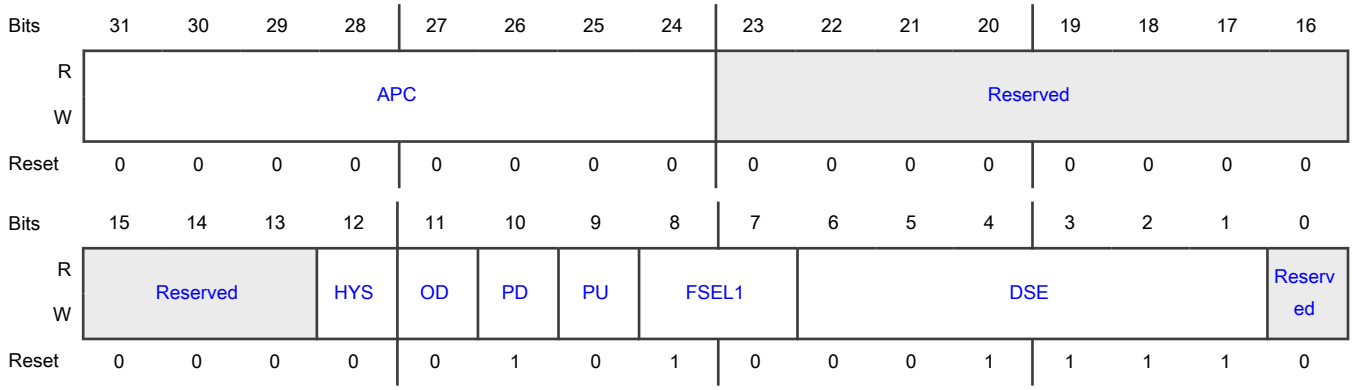
27.3.1.162 SW_PAD_CTL_PAD_ENET2_MDC SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_MDC)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_MDC	280h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_MDC 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_MDC 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_MDC 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_MDC 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_MDC 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_MDC 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

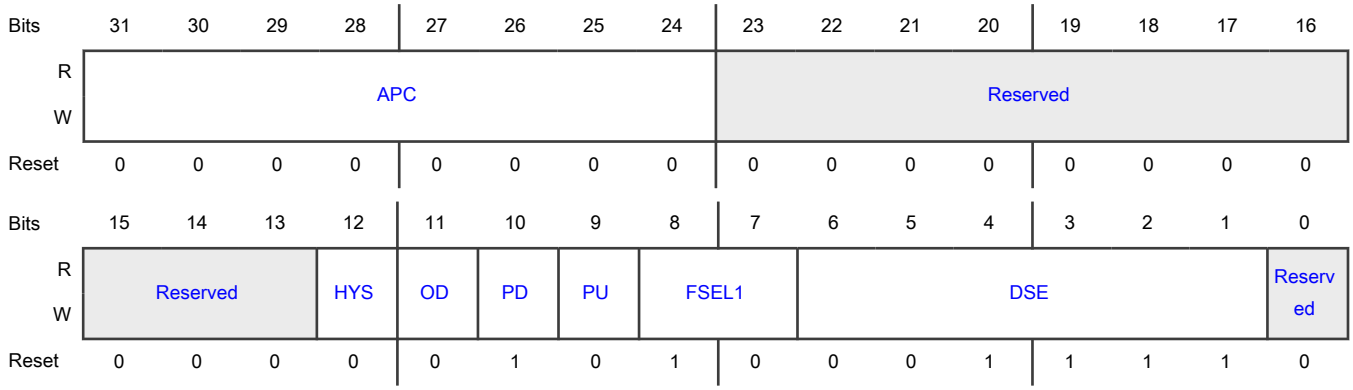
27.3.1.163 SW_PAD_CTL_PAD_ENET2_MDIO SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_MDIO)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_MDIO	284h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_MDIO 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_MDIO 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_MDIO 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_MDIO 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_MDIO 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_MDIO 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

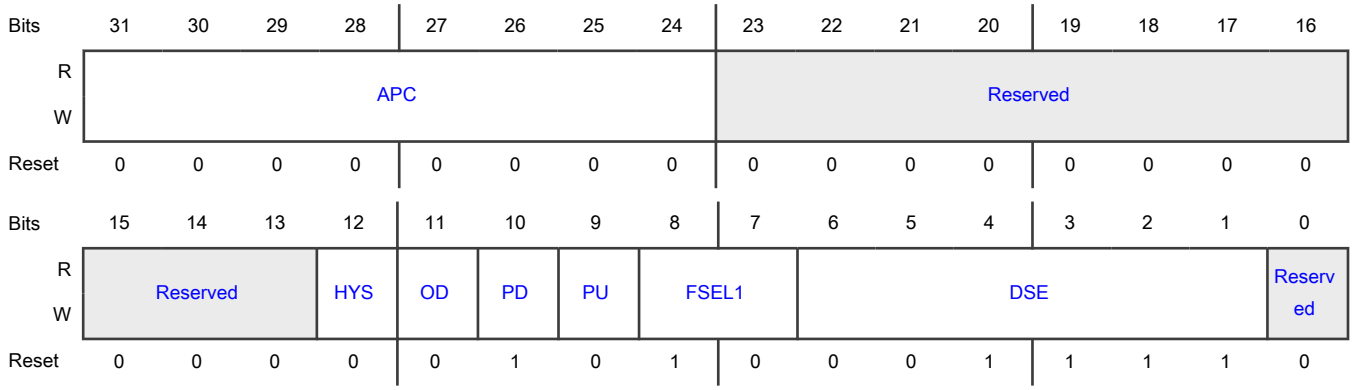
27.3.1.164 SW_PAD_CTL_PAD_ENET2_TD3 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_TD3)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_TD3	288h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_TD3 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_TD3 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_TD3 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_TD3 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_TD3 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_TD3 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

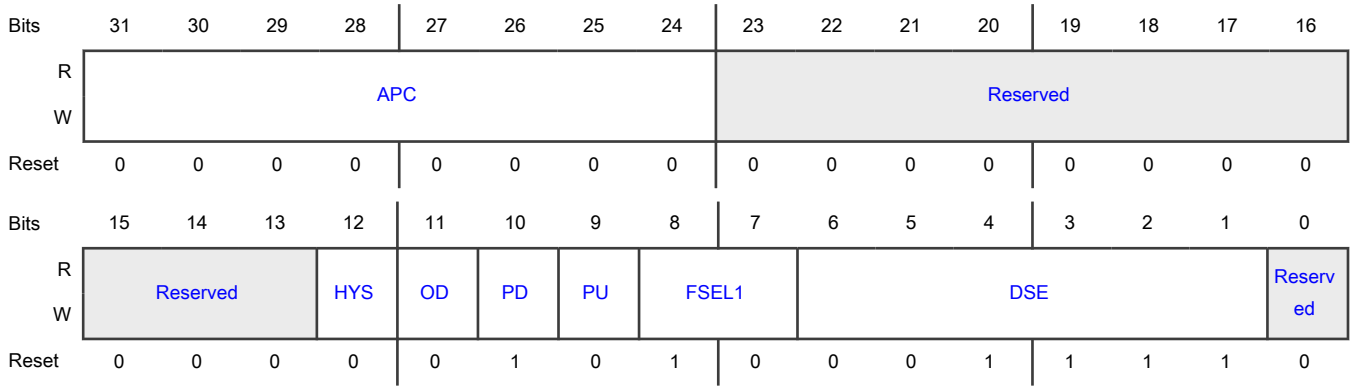
27.3.1.165 SW_PAD_CTL_PAD_ENET2_TD2 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_TD2)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_TD2	28Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_TD2 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_TD2 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_TD2 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_TD2 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_TD2 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_TD2 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

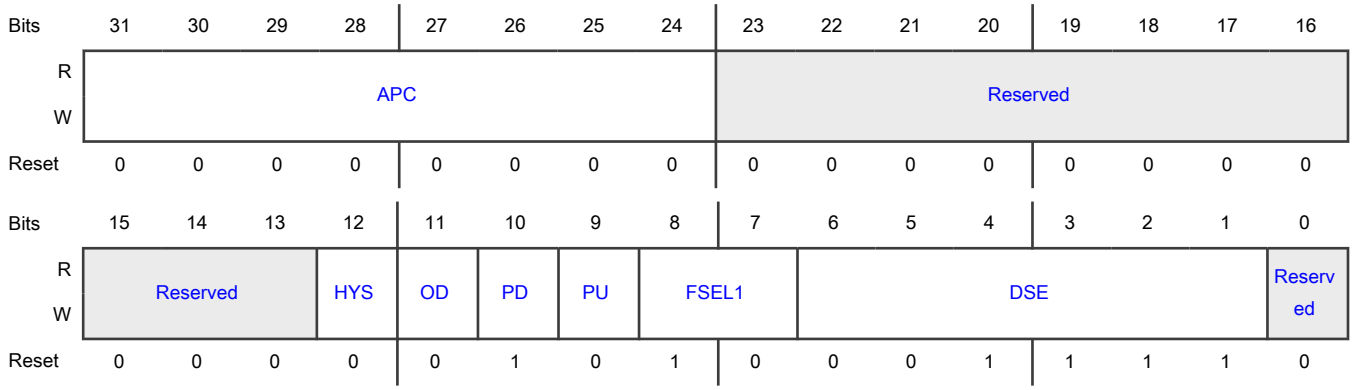
27.3.1.166 SW_PAD_CTL_PAD_ENET2_TD1 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_TD1)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_TD1	290h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_TD1 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_TD1 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_TD1 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_TD1 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_TD1 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_TD1 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

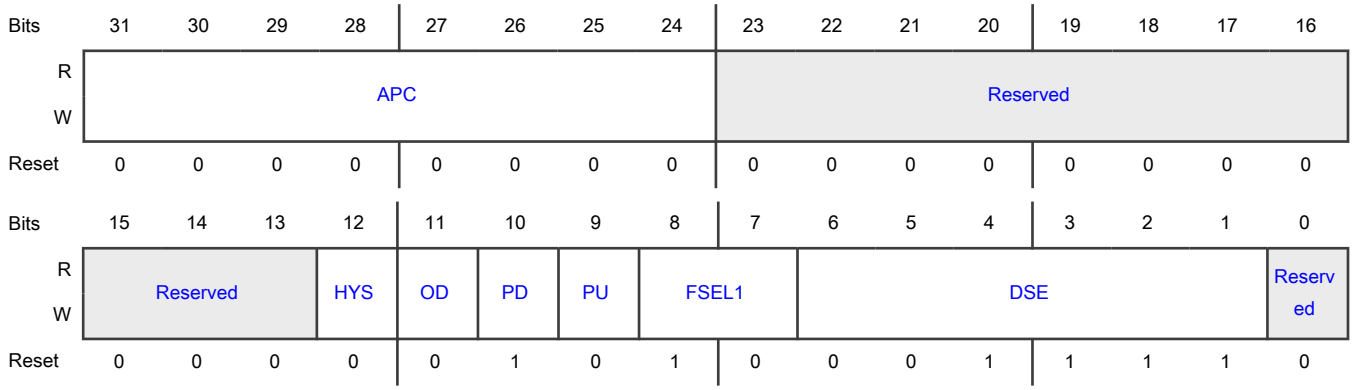
27.3.1.167 SW_PAD_CTL_PAD_ENET2_TD0 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_TD0)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_TD0	294h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_TD0 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_TD0 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_TD0 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_TD0 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_TD0 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_TD0 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

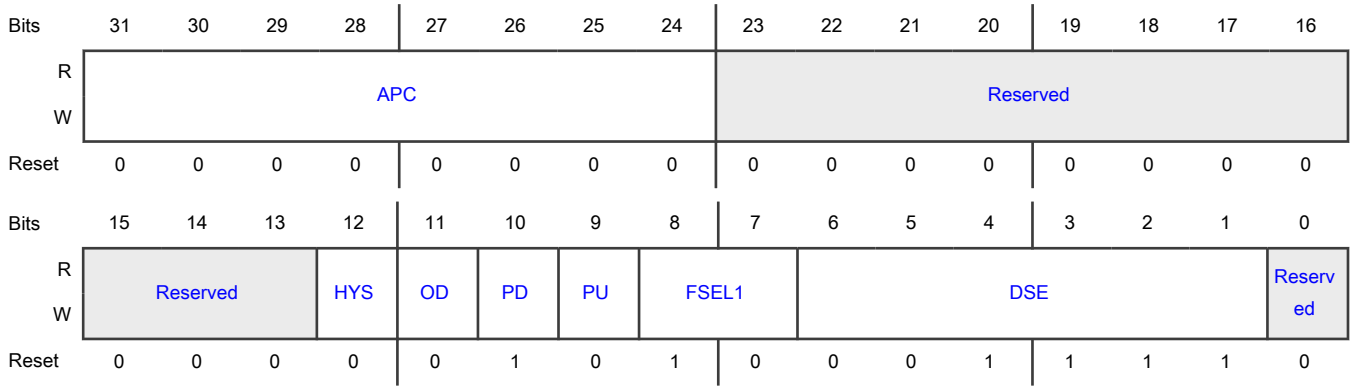
27.3.1.168 SW_PAD_CTL_PAD_ENET2_TX_CTL SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_TX_CTL)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_TX_CTL	298h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_TX_CTL 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_TX_CTL 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_TX_CTL 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_TX_CTL 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_TX_CTL 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_TX_CTL 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

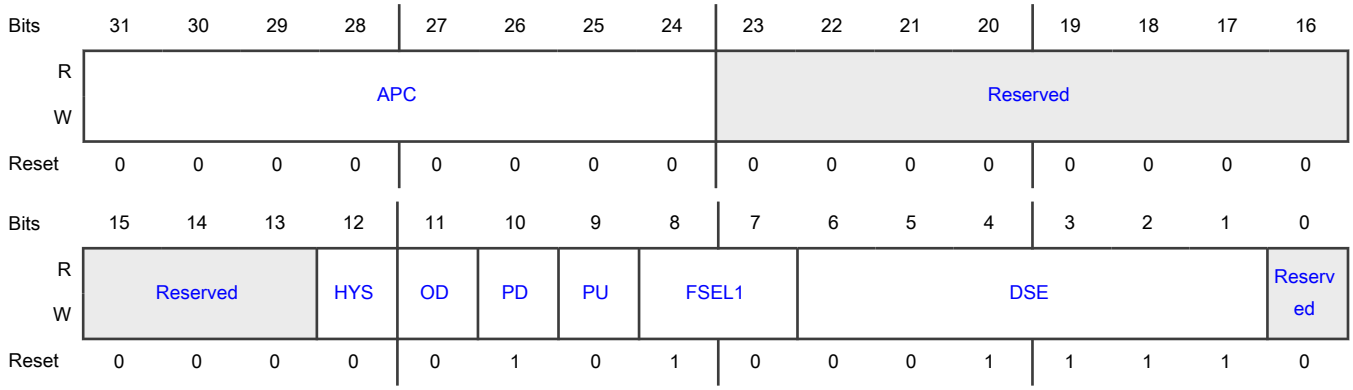
27.3.1.169 SW_PAD_CTL_PAD_ENET2_TXC SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_TXC)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_TXC	29Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_TXC 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_TXC 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_TXC 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_TXC 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_TXC 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_TXC 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

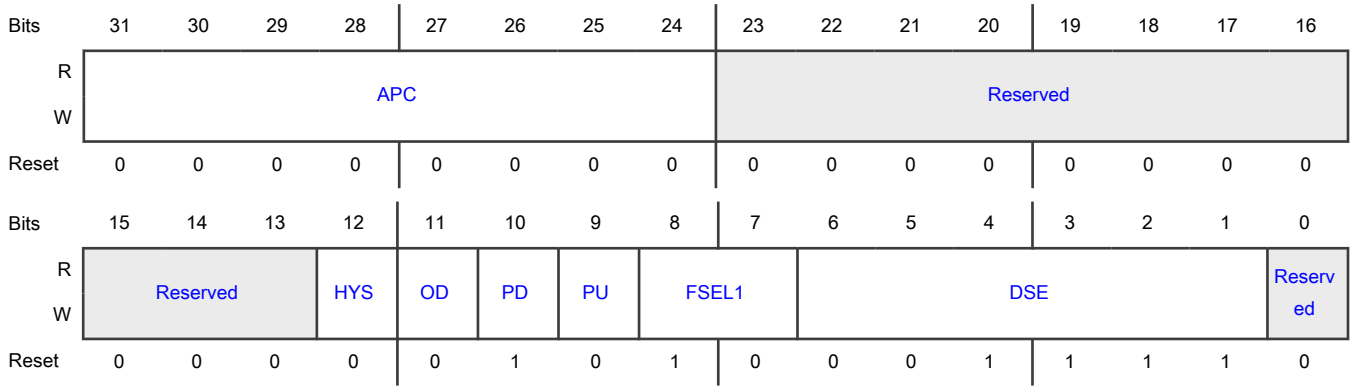
27.3.1.170 SW_PAD_CTL_PAD_ENET2_RX_CTL SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_RX_CTL)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_RX_CTL	2A0h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_RX_CTL 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_RX_CTL 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_RX_CTL 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_RX_CTL 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_RX_CTL 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_RX_CTL 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

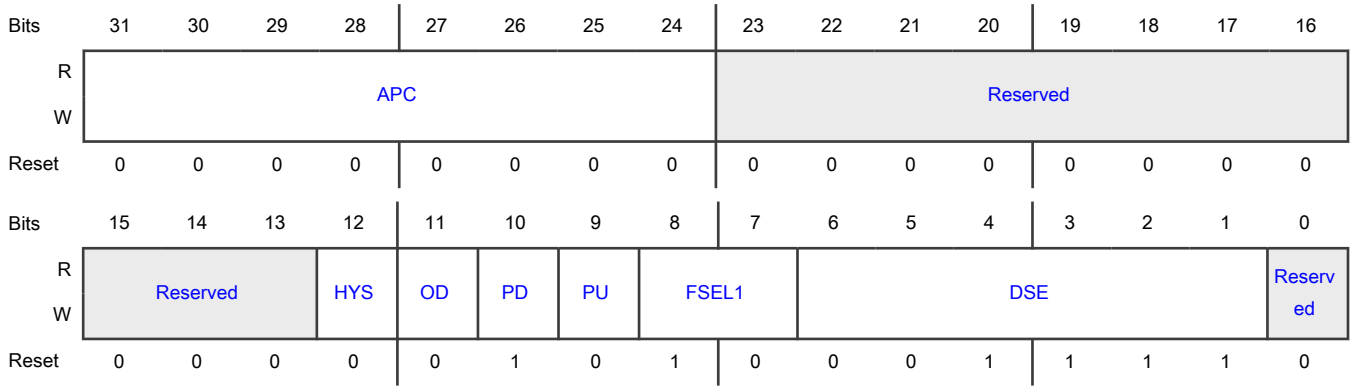
27.3.1.171 SW_PAD_CTL_PAD_ENET2_RXC SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_RXC)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_RXC	2A4h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_RXC 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_RXC 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_RXC 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_RXC 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_RXC 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_RXC 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

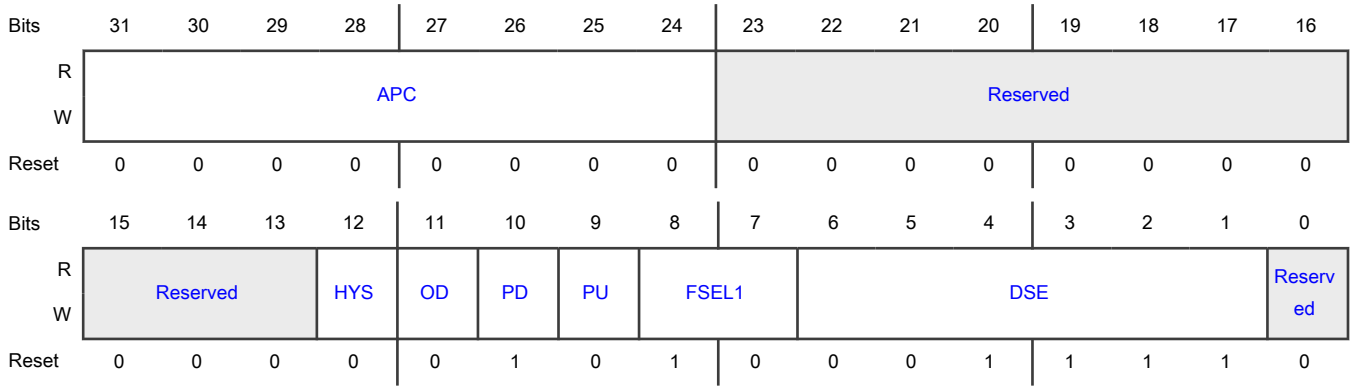
27.3.1.172 SW_PAD_CTL_PAD_ENET2_RD0 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_RD0)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_RD0	2A8h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_RD0 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_RD0 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_RD0 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_RD0 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_RD0 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_RD0 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

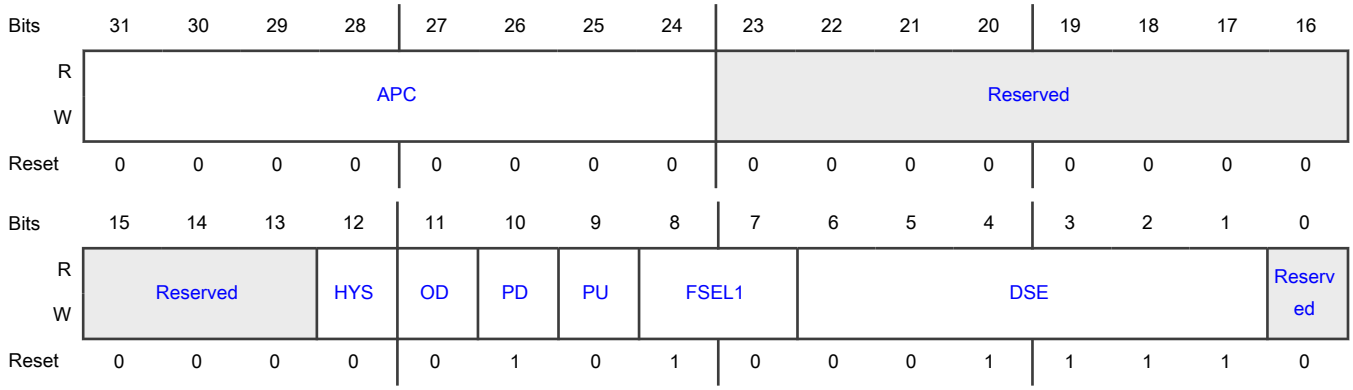
27.3.1.173 SW_PAD_CTL_PAD_ENET2_RD1 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_RD1)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_RD1	2ACh

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_RD1 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_RD1 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_RD1 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_RD1 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_RD1 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_RD1 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

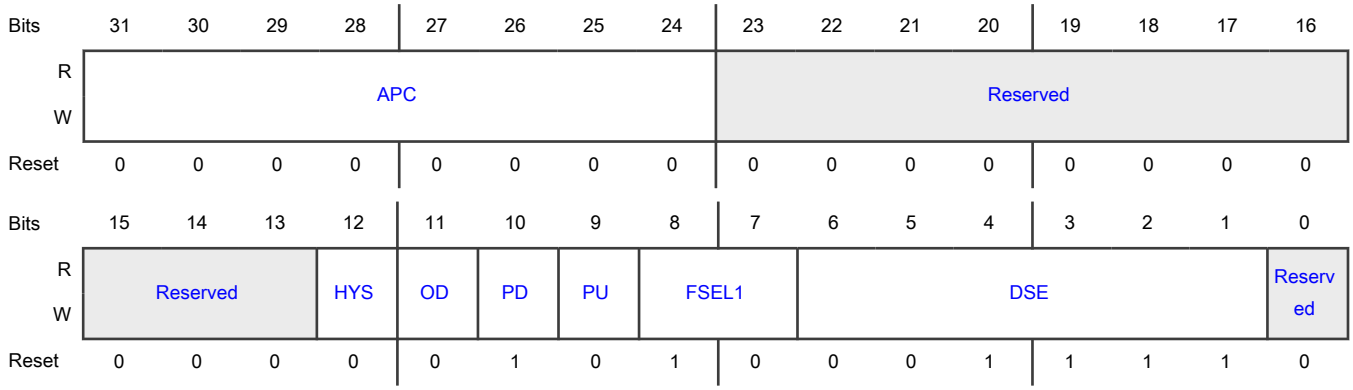
27.3.1.174 SW_PAD_CTL_PAD_ENET2_RD2 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_RD2)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_RD2	2B0h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_RD2 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_RD2 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_RD2 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_RD2 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_RD2 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_RD2 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

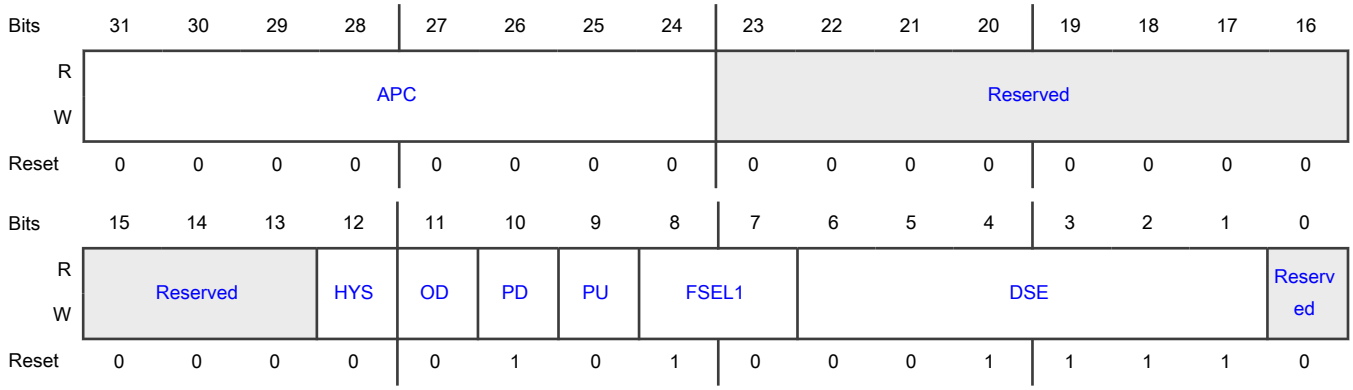
27.3.1.175 SW_PAD_CTL_PAD_ENET2_RD3 SW PAD Control Register (SW_PAD_CTL_PAD_ENET2_RD3)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_ENET2_RD3	2B4h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: ENET2_RD3 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: ENET2_RD3 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: ENET2_RD3 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: ENET2_RD3 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: ENET2_RD3 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: ENET2_RD3 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

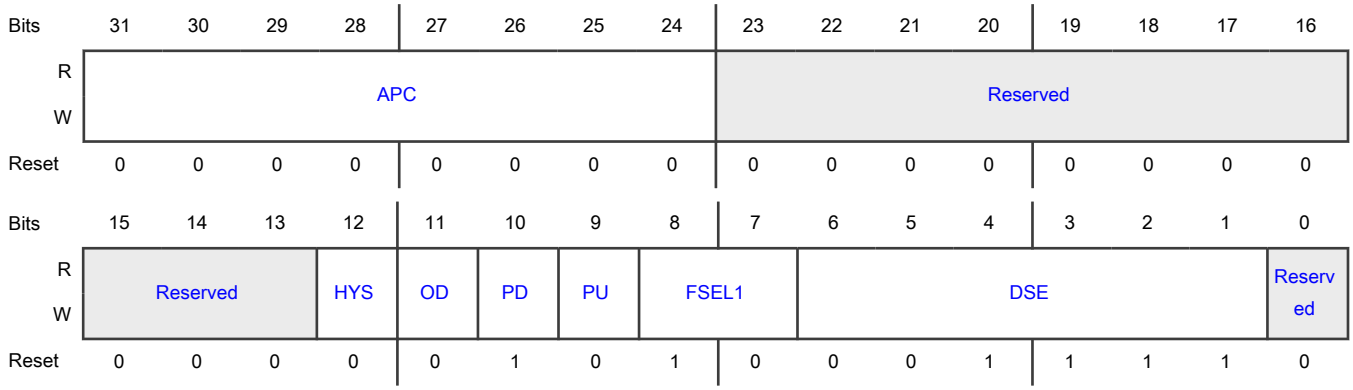
27.3.1.176 SW_PAD_CTL_PAD_SD1_CLK SW PAD Control Register (SW_PAD_CTL_PAD_SD1_CLK)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD1_CLK	2B8h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD1_CLK 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD1_CLK 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD1_CLK 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD1_CLK 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD1_CLK 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD1_CLK 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

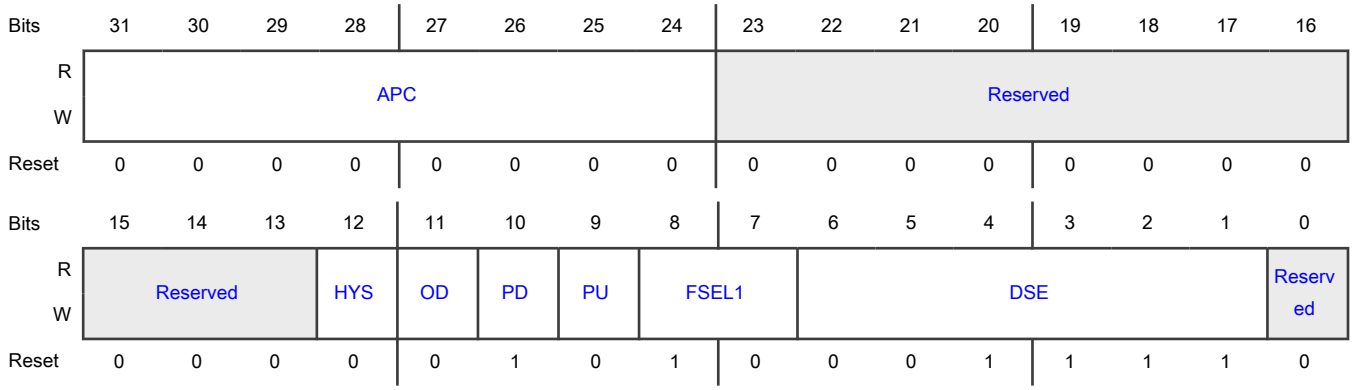
27.3.1.177 SW_PAD_CTL_PAD_SD1_CMD SW PAD Control Register (SW_PAD_CTL_PAD_SD1_CMD)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD1_CMD	2BCh

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD1_CMD 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD1_CMD 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD1_CMD 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD1_CMD 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD1_CMD 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD1_CMD 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

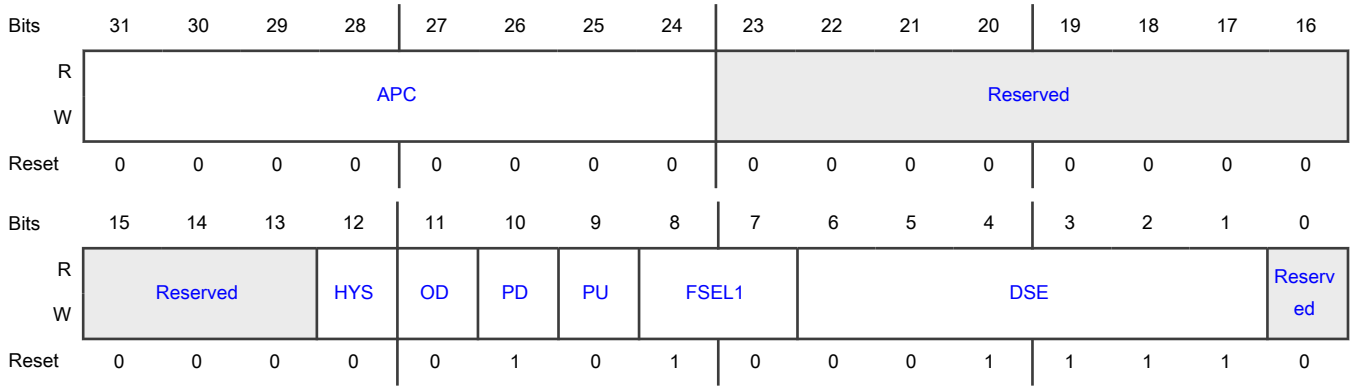
27.3.1.178 SW_PAD_CTL_PAD_SD1_DATA0 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA0)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD1_DATA0	2C0h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD1_DATA0 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD1_DATA0 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD1_DATA0 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD1_DATA0 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD1_DATA0 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA0 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

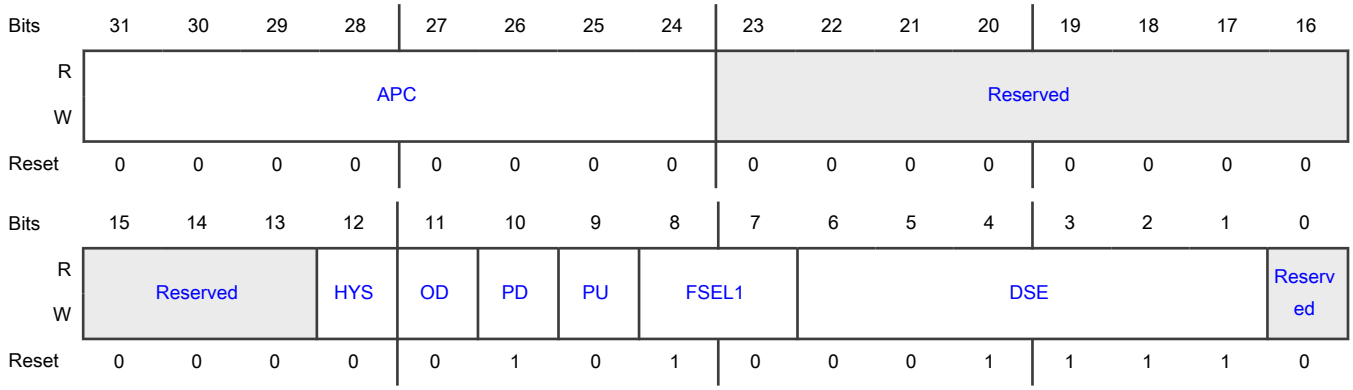
27.3.1.179 SW_PAD_CTL_PAD_SD1_DATA1 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA1)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD1_DATA1	2C4h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD1_DATA1 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD1_DATA1 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD1_DATA1 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD1_DATA1 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD1_DATA1 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA1 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

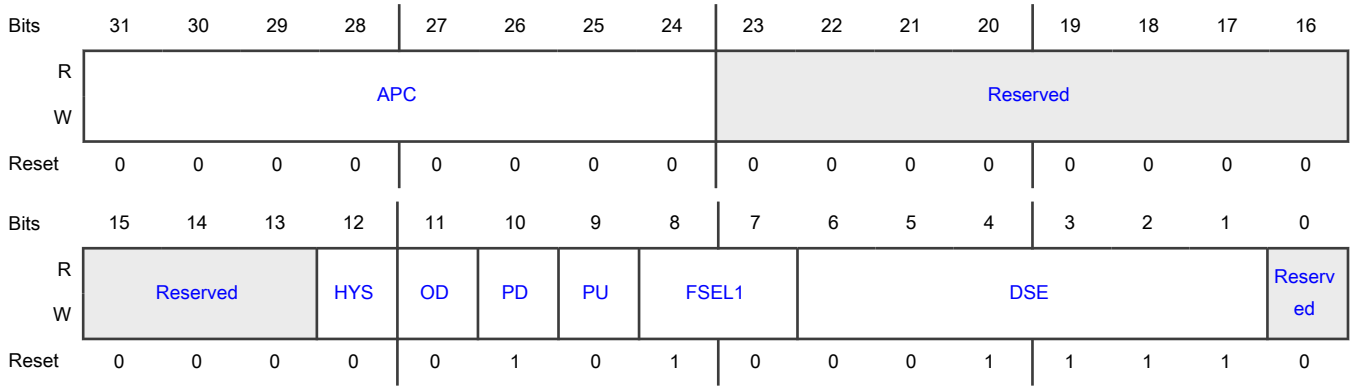
27.3.1.180 SW_PAD_CTL_PAD_SD1_DATA2 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA2)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD1_DATA2	2C8h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD1_DATA2 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD1_DATA2 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD1_DATA2 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD1_DATA2 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD1_DATA2 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA2 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

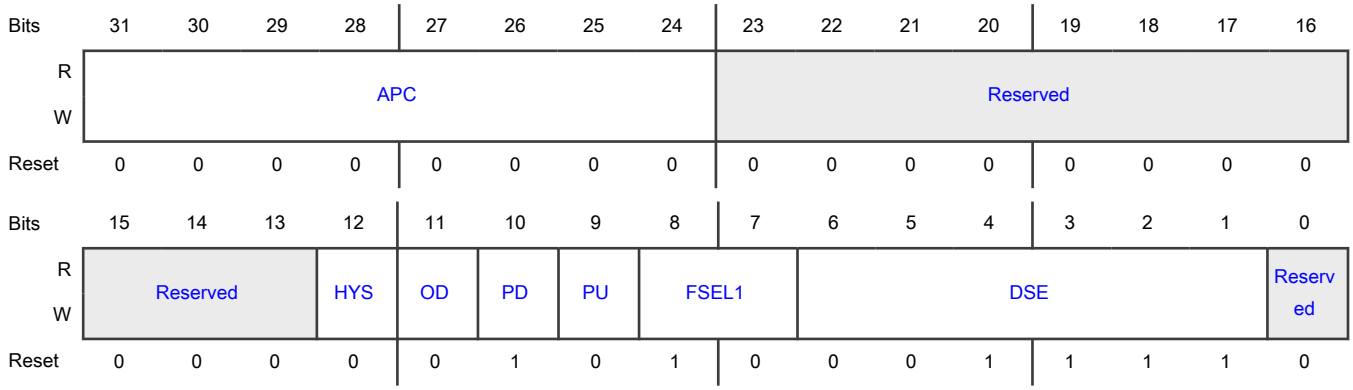
27.3.1.181 SW_PAD_CTL_PAD_SD1_DATA3 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA3)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD1_DATA3	2CCh

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD1_DATA3 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD1_DATA3 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD1_DATA3 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD1_DATA3 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD1_DATA3 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA3 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

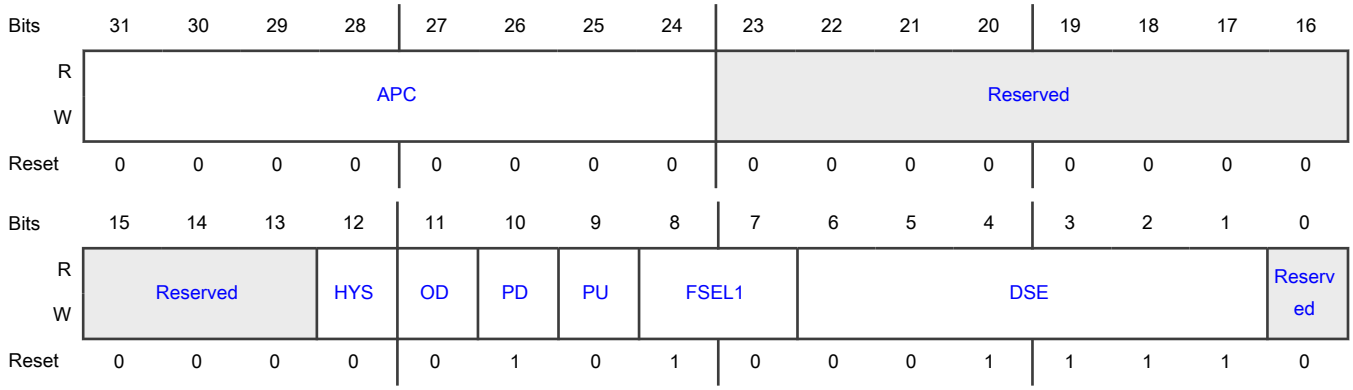
27.3.1.182 SW_PAD_CTL_PAD_SD1_DATA4 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA4)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD1_DATA4	2D0h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD1_DATA4 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD1_DATA4 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD1_DATA4 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD1_DATA4 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD1_DATA4 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA4 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

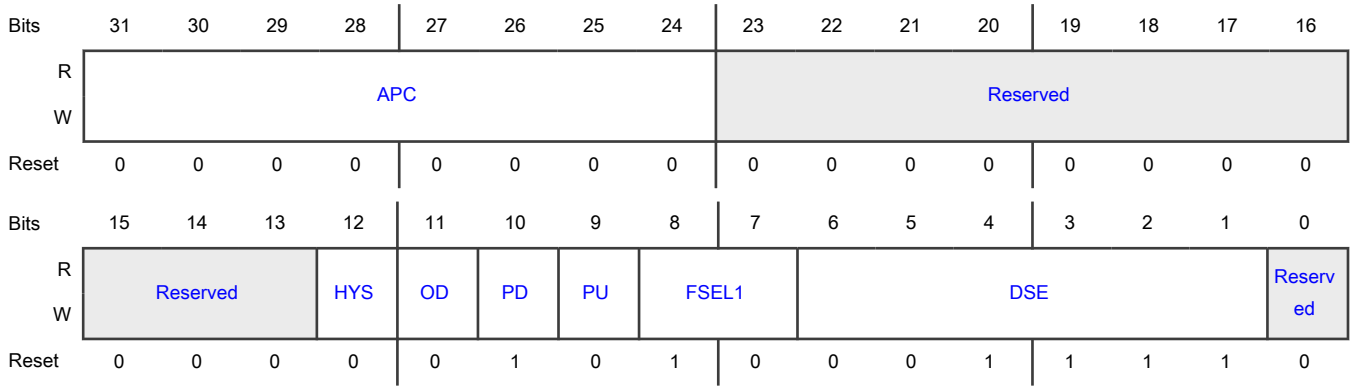
27.3.1.183 SW_PAD_CTL_PAD_SD1_DATA5 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA5)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD1_DATA5	2D4h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD1_DATA5 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD1_DATA5 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD1_DATA5 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD1_DATA5 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD1_DATA5 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA5 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

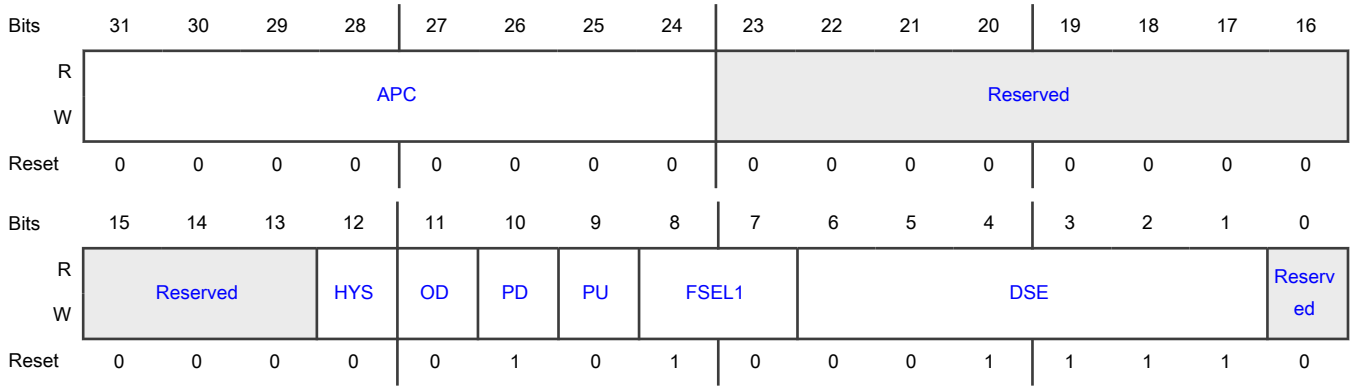
27.3.1.184 SW_PAD_CTL_PAD_SD1_DATA6 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA6)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD1_DATA6	2D8h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD1_DATA6 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD1_DATA6 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD1_DATA6 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD1_DATA6 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD1_DATA6 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA6 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

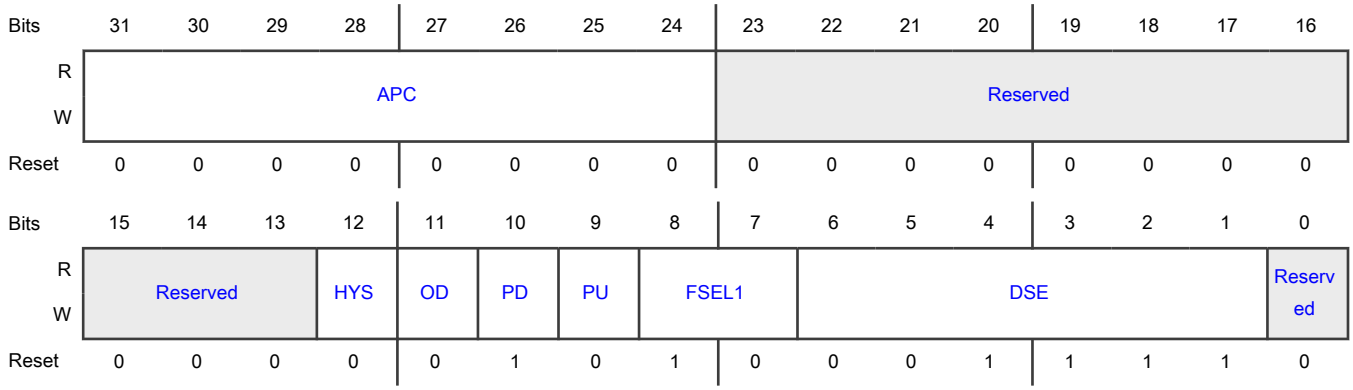
27.3.1.185 SW_PAD_CTL_PAD_SD1_DATA7 SW PAD Control Register (SW_PAD_CTL_PAD_SD1_DATA7)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD1_DATA7	2DCh

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD1_DATA7 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD1_DATA7 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD1_DATA7 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD1_DATA7 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD1_DATA7 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA7 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

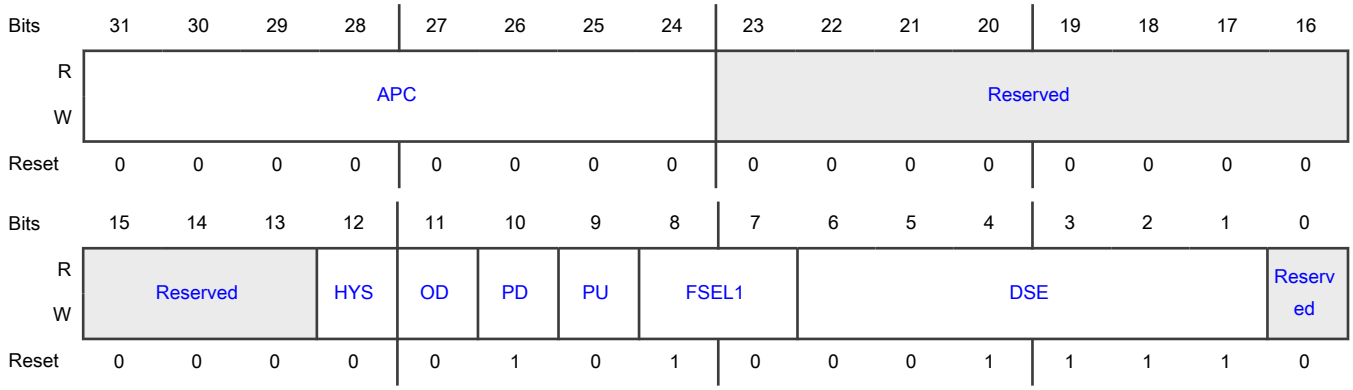
27.3.1.186 SW_PAD_CTL_PAD_SD1_STROBE SW PAD Control Register (SW_PAD_CTL_PAD_SD1_STROBE)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD1_STROBE	2E0h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD1_STROBE 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD1_STROBE 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD1_STROBE 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD1_STROBE 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD1_STROBE 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD1_STROBE 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

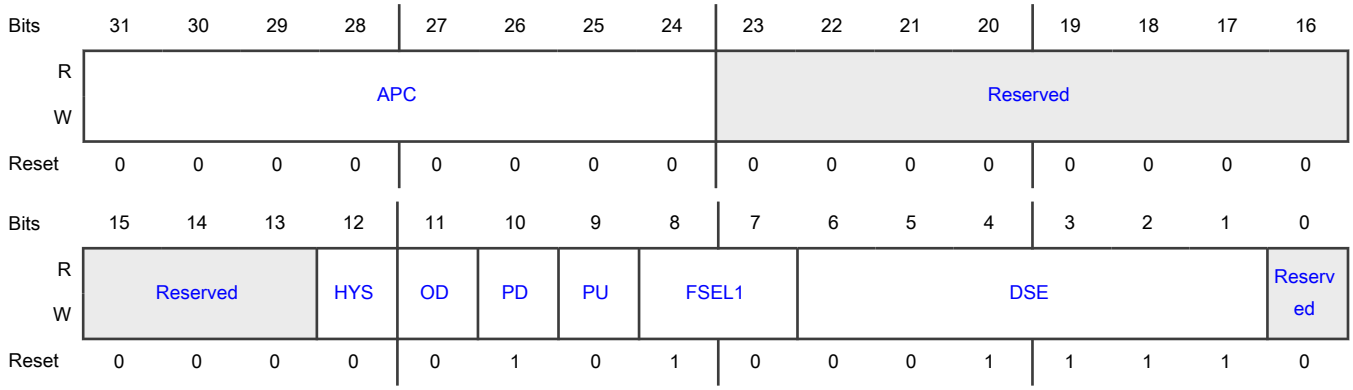
27.3.1.187 SW_PAD_CTL_PAD_SD2_VSELECT SW PAD Control Register (SW_PAD_CTL_PAD_SD2_VSELECT)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD2_VSELECT	2E4h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD2_VSELECT 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD2_VSELECT 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD2_VSELECT 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD2_VSELECT 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD2_VSELECT 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD2_VSELECT 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

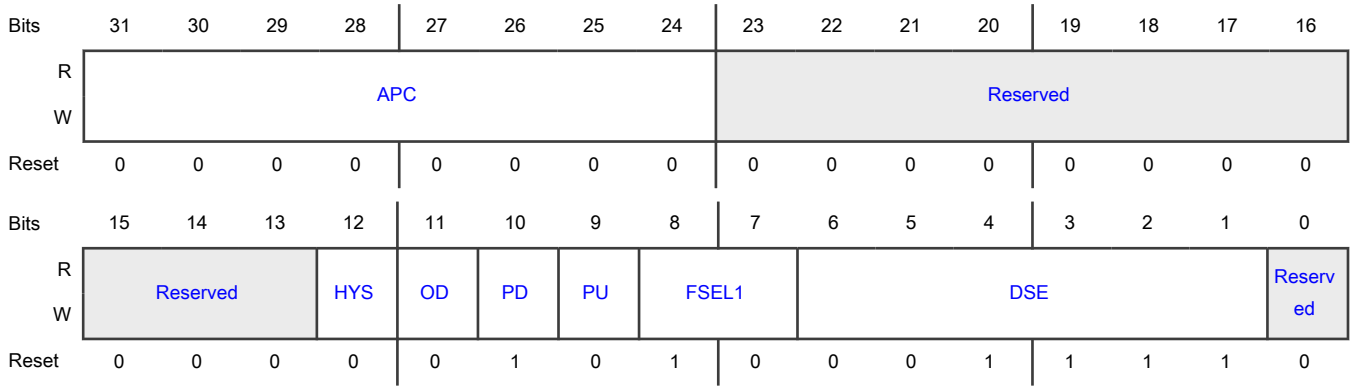
27.3.1.188 SW_PAD_CTL_PAD_SD3_CLK SW PAD Control Register (SW_PAD_CTL_PAD_SD3_CLK)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD3_CLK	2E8h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD3_CLK 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD3_CLK 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD3_CLK 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD3_CLK 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD3_CLK 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD3_CLK 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

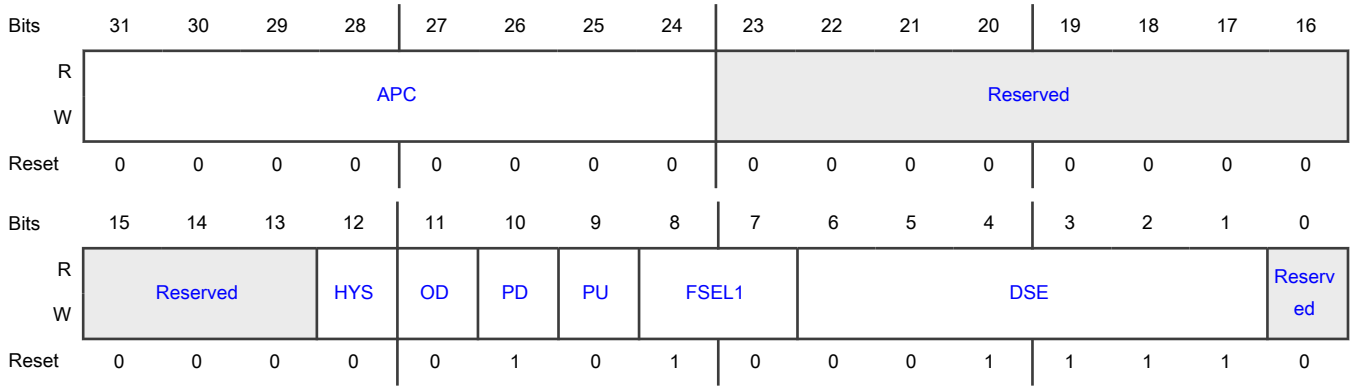
27.3.1.189 SW_PAD_CTL_PAD_SD3_CMD SW PAD Control Register (SW_PAD_CTL_PAD_SD3_CMD)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD3_CMD	2ECh

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD3_CMD 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD3_CMD 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD3_CMD 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD3_CMD 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD3_CMD 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD3_CMD 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

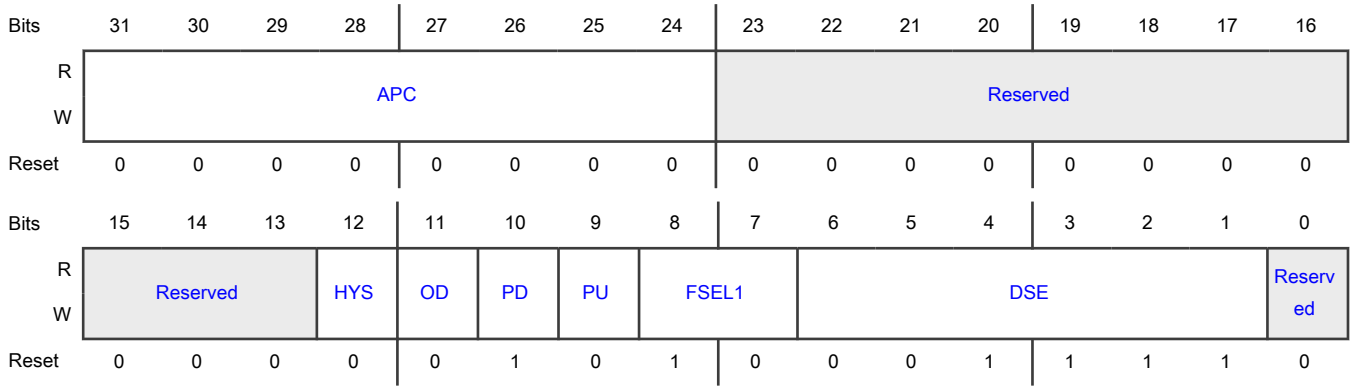
27.3.1.190 SW_PAD_CTL_PAD_SD3_DATA0 SW PAD Control Register (SW_PAD_CTL_PAD_SD3_DATA0)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD3_DATA0	2F0h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD3_DATA0 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD3_DATA0 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD3_DATA0 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD3_DATA0 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD3_DATA0 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD3_DATA0 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

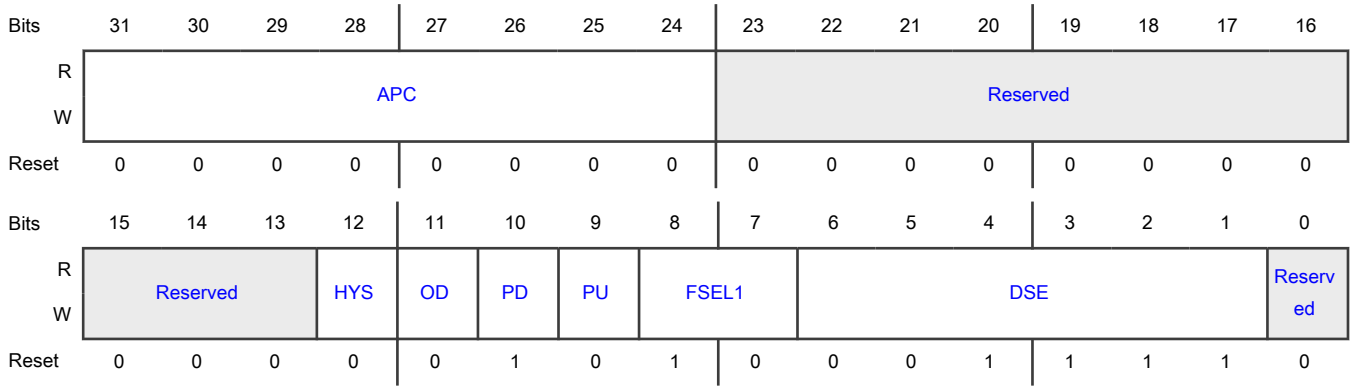
27.3.1.191 SW_PAD_CTL_PAD_SD3_DATA1 SW PAD Control Register (SW_PAD_CTL_PAD_SD3_DATA1)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD3_DATA1	2F4h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD3_DATA1 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD3_DATA1 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD3_DATA1 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD3_DATA1 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD3_DATA1 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD3_DATA1 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

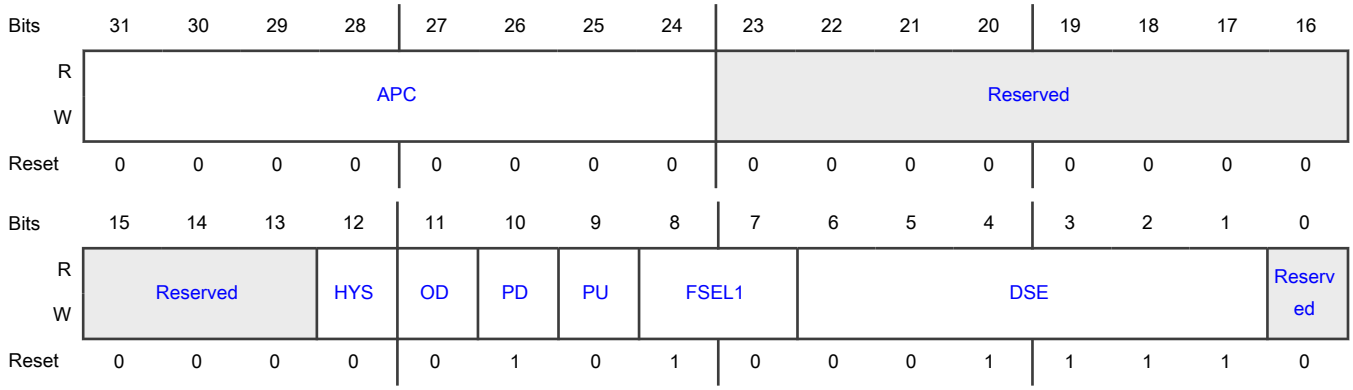
27.3.1.192 SW_PAD_CTL_PAD_SD3_DATA2 SW PAD Control Register (SW_PAD_CTL_PAD_SD3_DATA2)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD3_DATA2	2F8h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD3_DATA2 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD3_DATA2 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD3_DATA2 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD3_DATA2 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD3_DATA2 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD3_DATA2 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

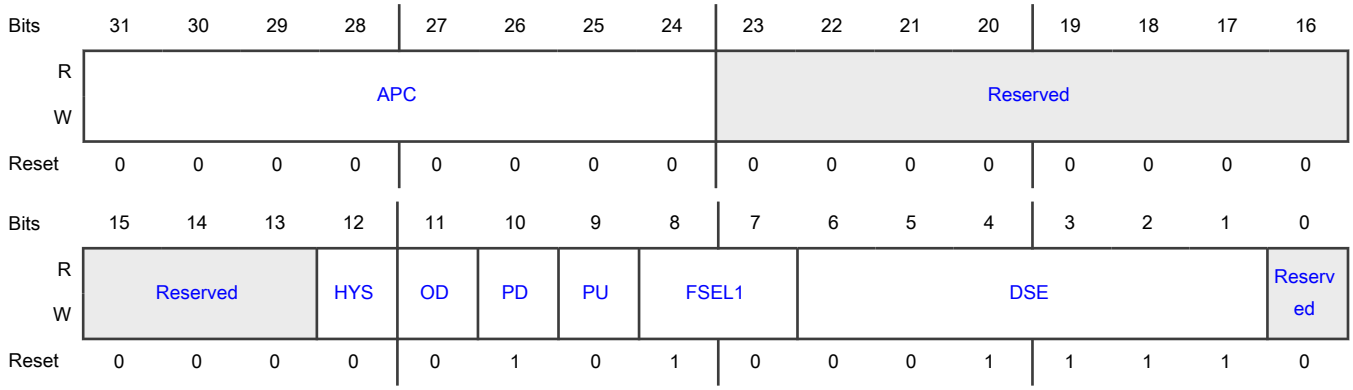
27.3.1.193 SW_PAD_CTL_PAD_SD3_DATA3 SW PAD Control Register (SW_PAD_CTL_PAD_SD3_DATA3)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD3_DATA3	2FCh

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD3_DATA3 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD3_DATA3 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD3_DATA3 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD3_DATA3 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD3_DATA3 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD3_DATA3 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

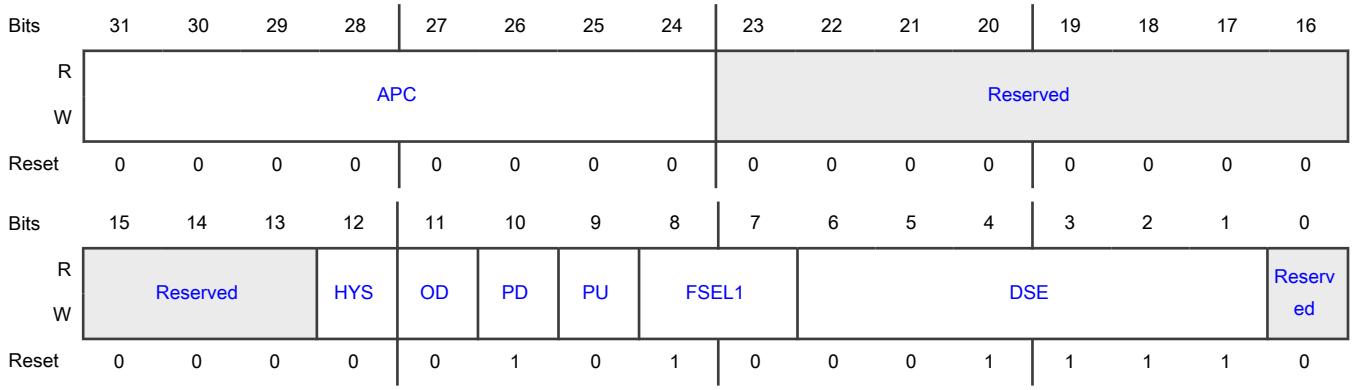
27.3.1.194 SW_PAD_CTL_PAD_SD2_CD_B SW PAD Control Register (SW_PAD_CTL_PAD_SD2_CD_B)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD2_CD_B	300h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD2_CD_B 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD2_CD_B 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD2_CD_B 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD2_CD_B 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD2_CD_B 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD2_CD_B 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

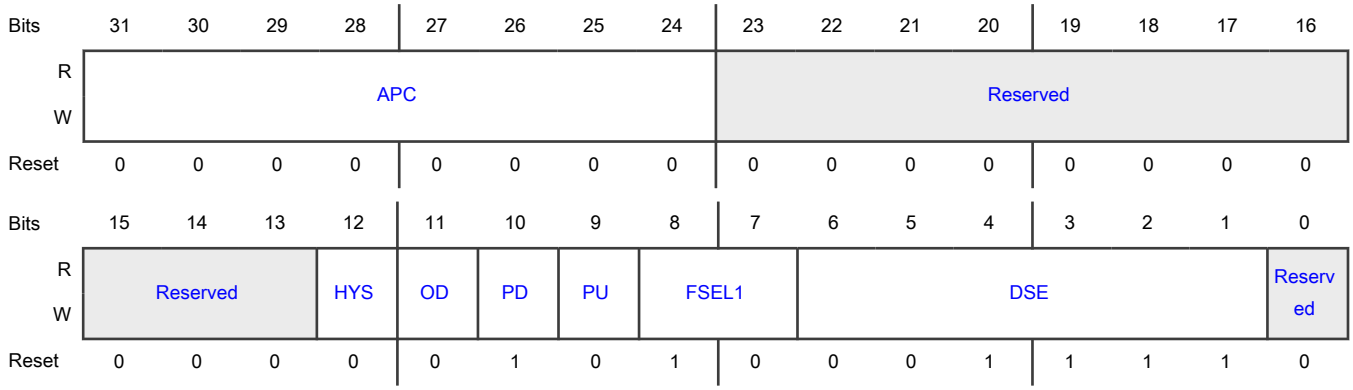
27.3.1.195 SW_PAD_CTL_PAD_SD2_CLK SW PAD Control Register (SW_PAD_CTL_PAD_SD2_CLK)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD2_CLK	304h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD2_CLK 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD2_CLK 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD2_CLK 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD2_CLK 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD2_CLK 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD2_CLK 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

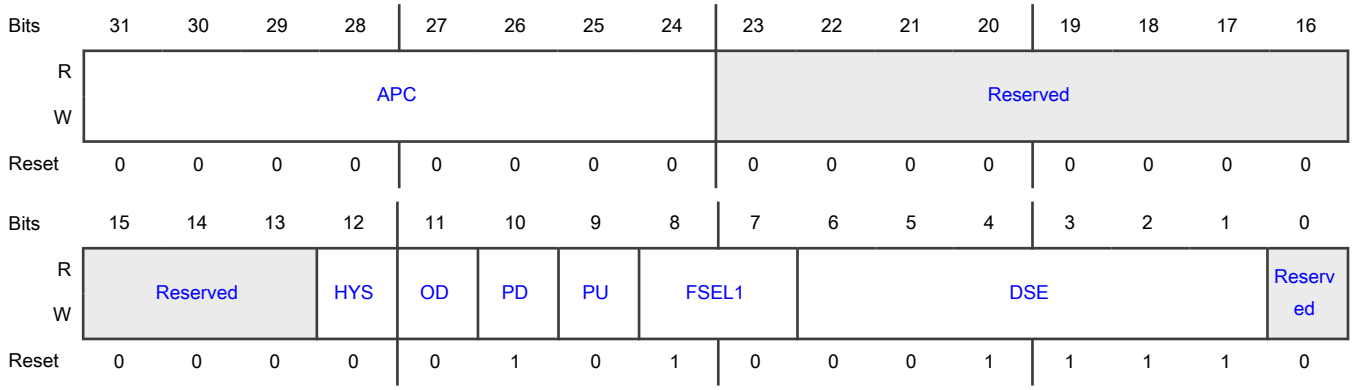
27.3.1.196 SW_PAD_CTL_PAD_SD2_CMD SW PAD Control Register (SW_PAD_CTL_PAD_SD2_CMD)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD2_CMD	308h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD2_CMD 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD2_CMD 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD2_CMD 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD2_CMD 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD2_CMD 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD2_CMD 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

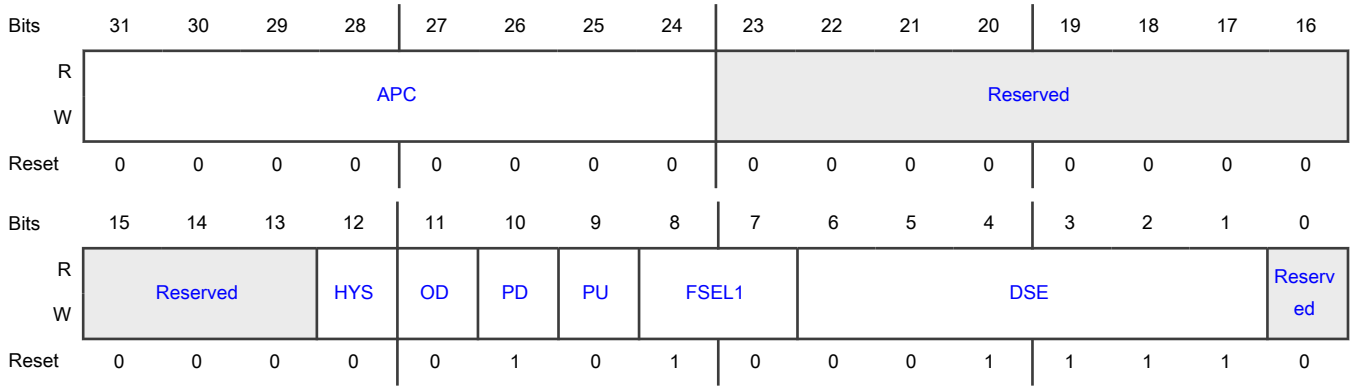
27.3.1.197 SW_PAD_CTL_PAD_SD2_DATA0 SW PAD Control Register (SW_PAD_CTL_PAD_SD2_DATA0)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD2_DATA0	30Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD2_DATA0 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD2_DATA0 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD2_DATA0 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD2_DATA0 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD2_DATA0 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA0 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

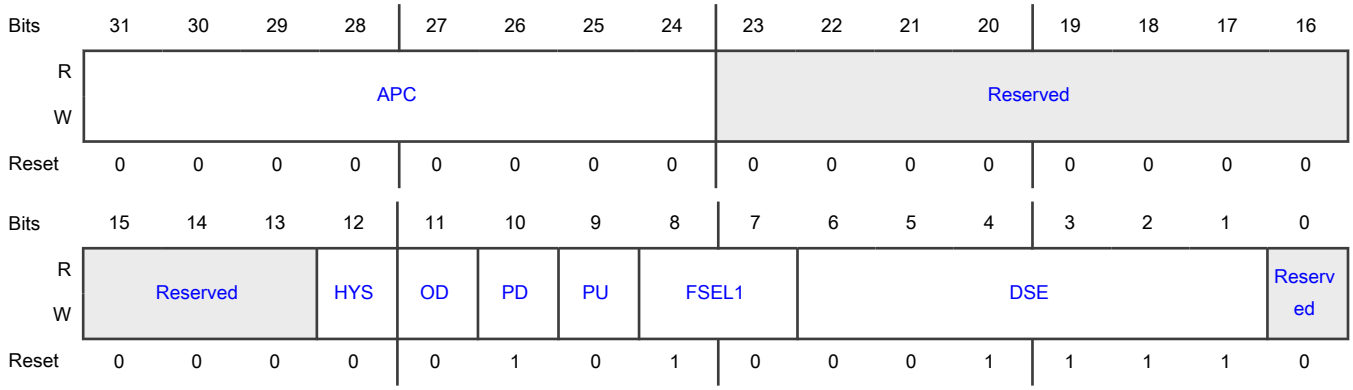
27.3.1.198 SW_PAD_CTL_PAD_SD2_DATA1 SW PAD Control Register (SW_PAD_CTL_PAD_SD2_DATA1)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD2_DATA1	310h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD2_DATA1 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD2_DATA1 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD2_DATA1 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD2_DATA1 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD2_DATA1 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA1 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

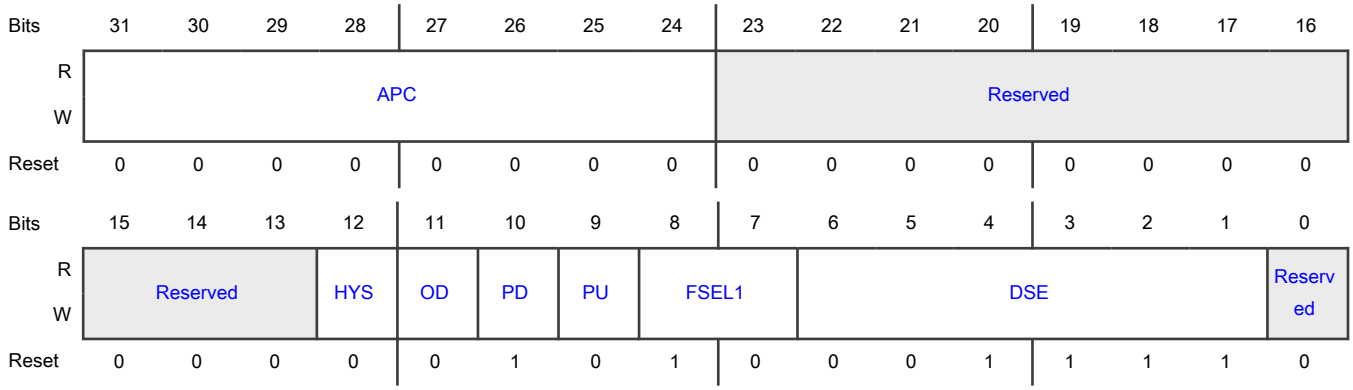
27.3.1.199 SW_PAD_CTL_PAD_SD2_DATA2 SW PAD Control Register (SW_PAD_CTL_PAD_SD2_DATA2)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD2_DATA2	314h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD2_DATA2 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD2_DATA2 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD2_DATA2 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD2_DATA2 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD2_DATA2 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA2 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

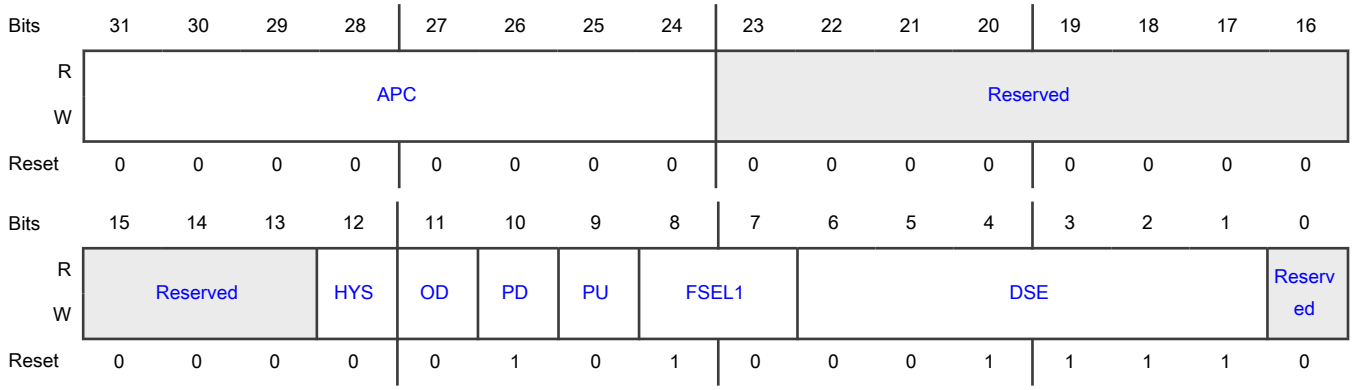
27.3.1.200 SW_PAD_CTL_PAD_SD2_DATA3 SW PAD Control Register (SW_PAD_CTL_PAD_SD2_DATA3)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD2_DATA3	318h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD2_DATA3 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD2_DATA3 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD2_DATA3 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD2_DATA3 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD2_DATA3 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA3 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

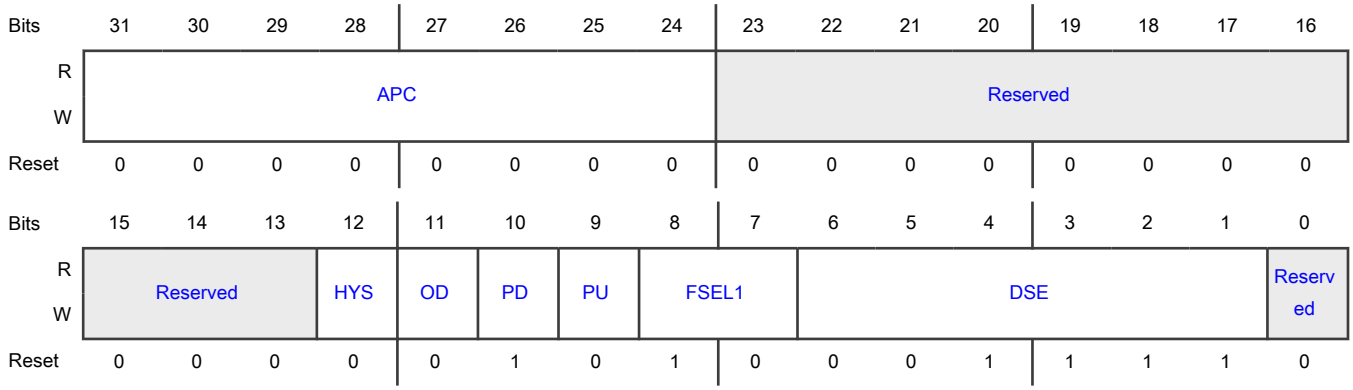
27.3.1.201 SW_PAD_CTL_PAD_SD2_RESET_B SW PAD Control Register (SW_PAD_CTL_PAD_SD2_RESET_B)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SD2_RESET_B	31Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SD2_RESET_B 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SD2_RESET_B 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SD2_RESET_B 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SD2_RESET_B 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SD2_RESET_B 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SD2_RESET_B 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

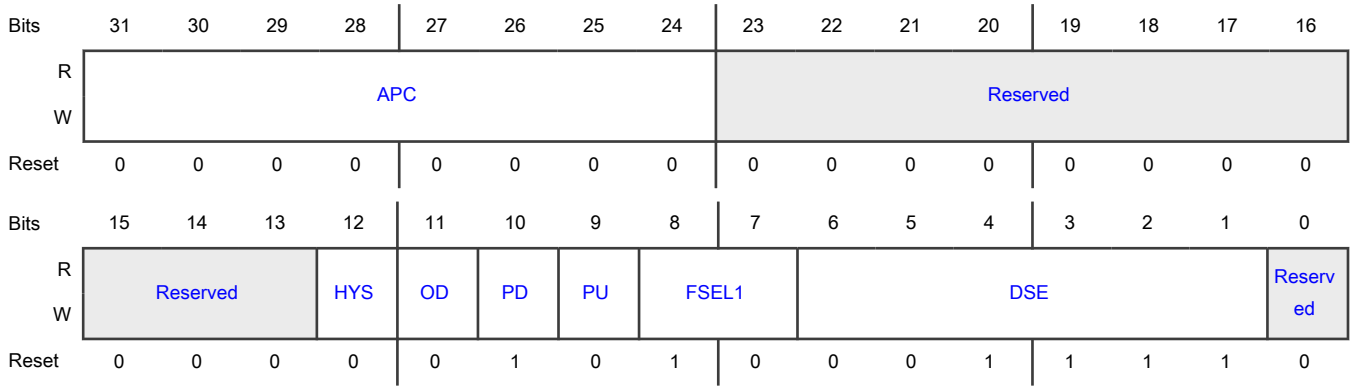
27.3.1.202 SW_PAD_CTL_PAD_I2C1_SCL SW PAD Control Register (SW_PAD_CTL_PAD_I2C1_SCL)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_I2C1_SCL	320h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: I2C1_SCL 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: I2C1_SCL 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: I2C1_SCL 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: I2C1_SCL 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: I2C1_SCL 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: I2C1_SCL 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

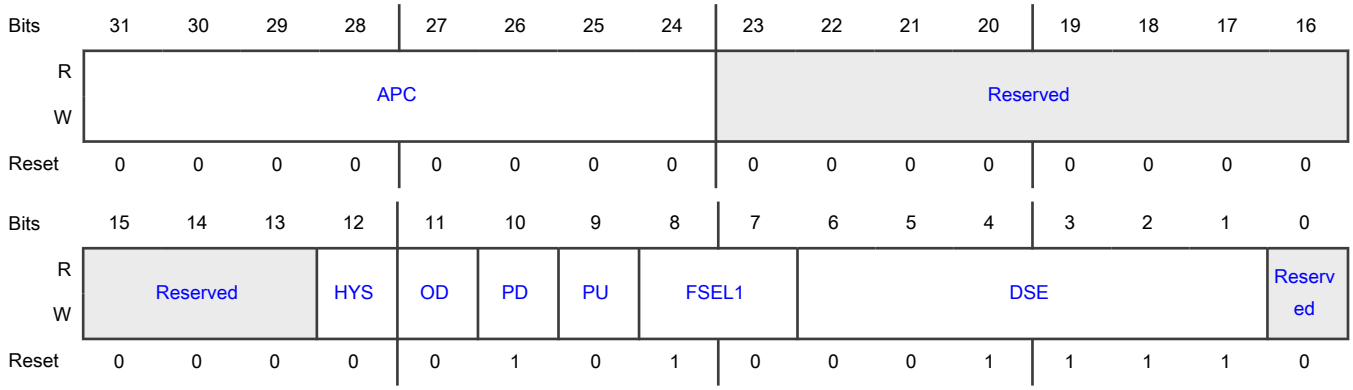
27.3.1.203 SW_PAD_CTL_PAD_I2C1_SDA SW PAD Control Register (SW_PAD_CTL_PAD_I2C1_SDA)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_I2C1_SDA	324h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: I2C1_SDA 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: I2C1_SDA 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: I2C1_SDA 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: I2C1_SDA 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: I2C1_SDA 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: I2C1_SDA 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

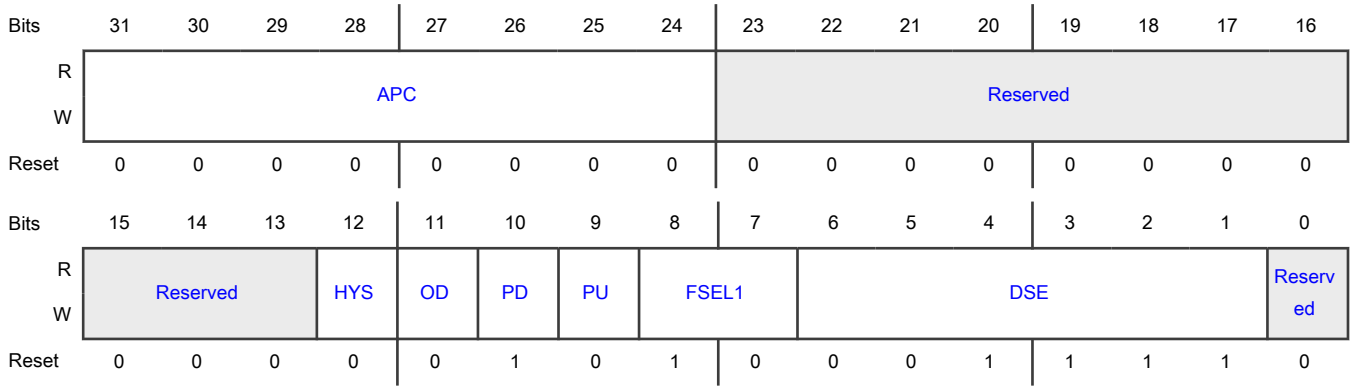
27.3.1.204 SW_PAD_CTL_PAD_I2C2_SCL SW PAD Control Register (SW_PAD_CTL_PAD_I2C2_SCL)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_I2C2_SCL	328h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: I2C2_SCL 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: I2C2_SCL 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: I2C2_SCL 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: I2C2_SCL 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: I2C2_SCL 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: I2C2_SCL 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

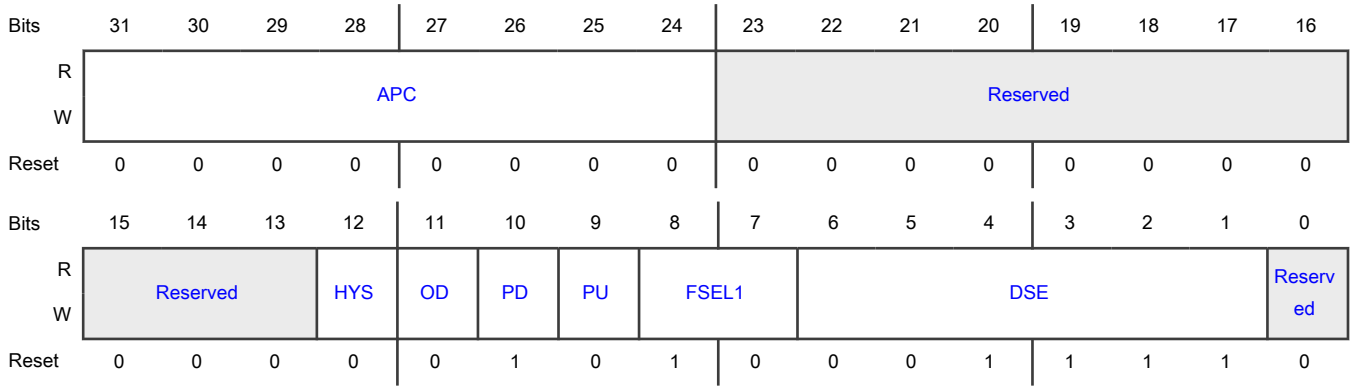
27.3.1.205 SW_PAD_CTL_PAD_I2C2_SDA SW PAD Control Register (SW_PAD_CTL_PAD_I2C2_SDA)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_I2C2_SDA	32Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: I2C2_SDA 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: I2C2_SDA 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: I2C2_SDA 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: I2C2_SDA 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: I2C2_SDA 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: I2C2_SDA 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

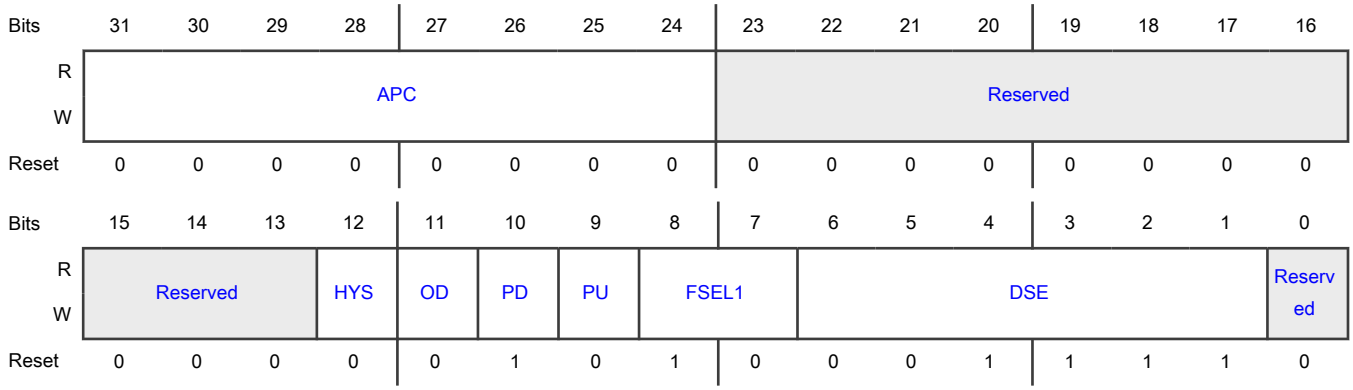
27.3.1.206 SW_PAD_CTL_PAD_UART1_RXD SW PAD Control Register (SW_PAD_CTL_PAD_UART1_RXD)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_UART1_RXD	330h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: UART1_RXD 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: UART1_RXD 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: UART1_RXD 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: UART1_RXD 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: UART1_RXD 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: UART1_RXD 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

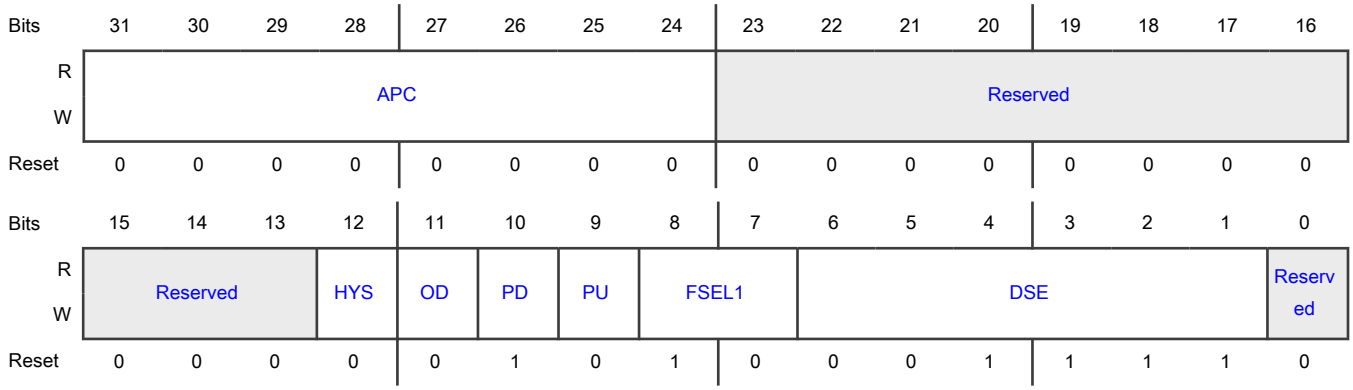
27.3.1.207 SW_PAD_CTL_PAD_UART1_TXD SW PAD Control Register (SW_PAD_CTL_PAD_UART1_TXD)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_UART1_TXD	334h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: UART1_TXD 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: UART1_TXD 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: UART1_TXD 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: UART1_TXD 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: UART1_TXD 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: UART1_TXD 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

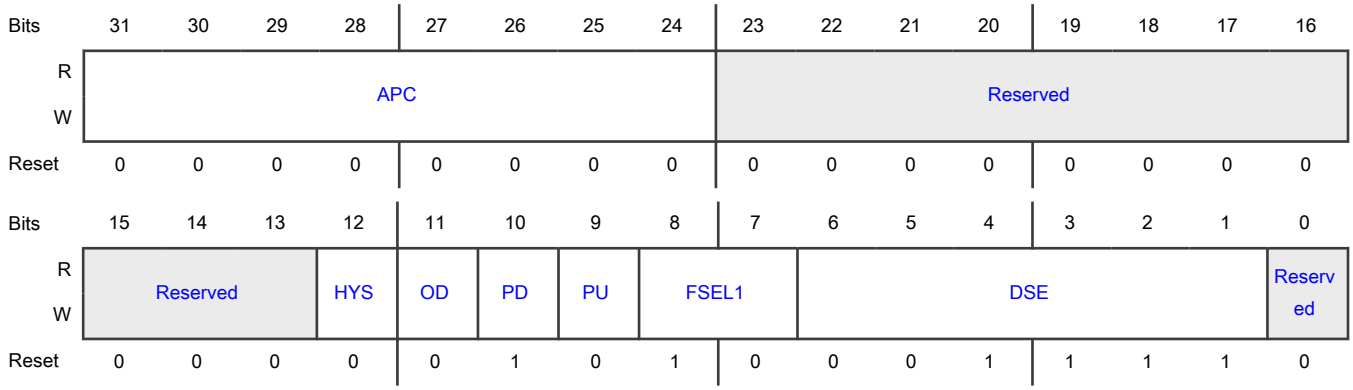
27.3.1.208 SW_PAD_CTL_PAD_UART2_RXD SW PAD Control Register (SW_PAD_CTL_PAD_UART2_RXD)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_UART2_RXD	338h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: UART2_RXD 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: UART2_RXD 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: UART2_RXD 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: UART2_RXD 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: UART2_RXD 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: UART2_RXD 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

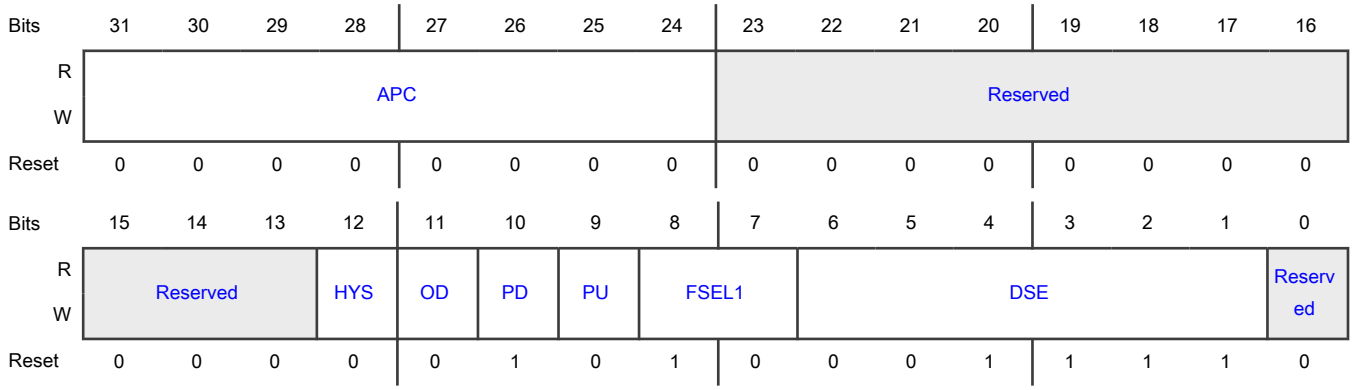
27.3.1.209 SW_PAD_CTL_PAD_UART2_TXD SW PAD Control Register (SW_PAD_CTL_PAD_UART2_TXD)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_UART2_TXD	33Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: UART2_TXD 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: UART2_TXD 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: UART2_TXD 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: UART2_TXD 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: UART2_TXD 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: UART2_TXD 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

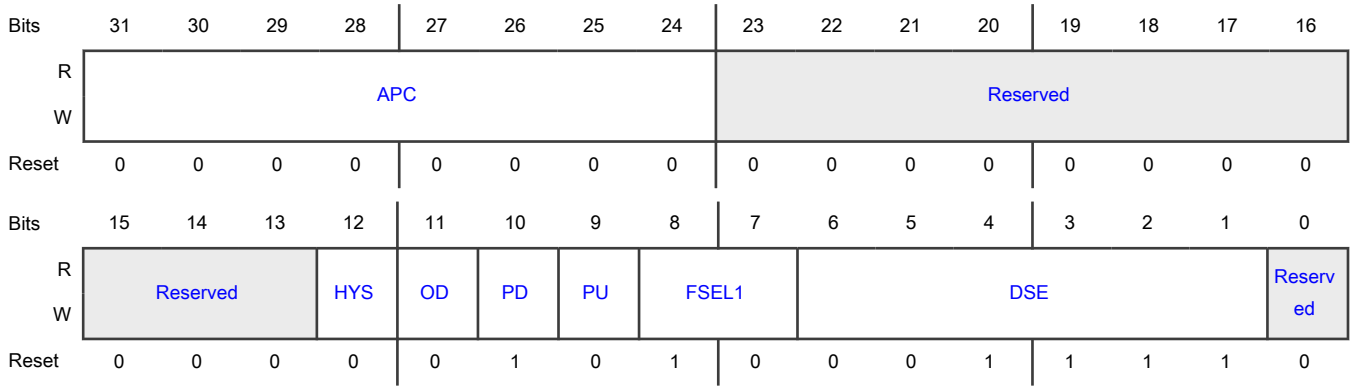
27.3.1.210 SW_PAD_CTL_PAD_PDM_CLK SW PAD Control Register (SW_PAD_CTL_PAD_PDM_CLK)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_PDM_CLK	340h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: PDM_CLK 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: PDM_CLK 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: PDM_CLK 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: PDM_CLK 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: PDM_CLK 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: PDM_CLK 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

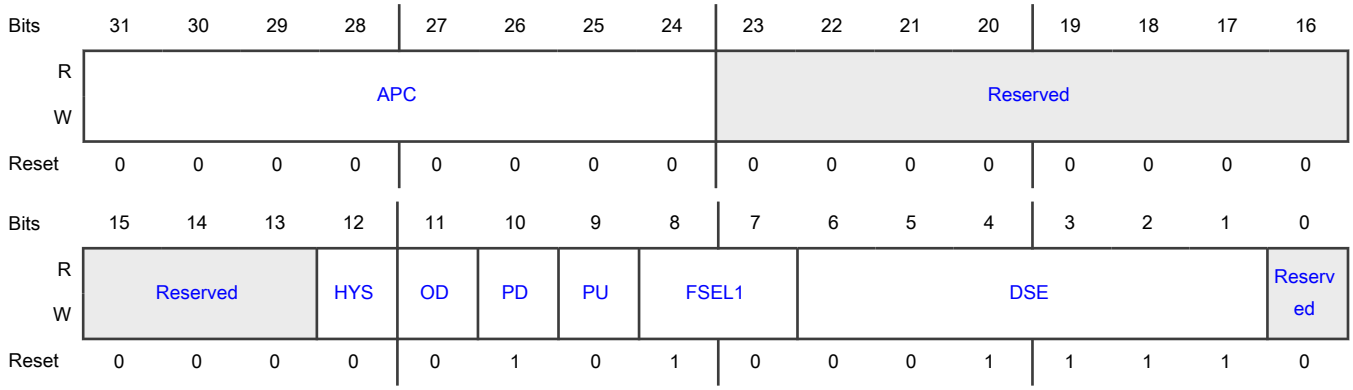
27.3.1.211 SW_PAD_CTL_PAD_PDM_BIT_STREAM0 SW PAD Control Register (SW_PAD_CTL_PAD_PDM_BIT_STREAM0)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_PDM_BIT_STREAM0	344h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: PDM_BIT_STREAM0 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: PDM_BIT_STREAM0 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: PDM_BIT_STREAM0 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: PDM_BIT_STREAM0 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: PDM_BIT_STREAM0 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: PDM_BIT_STREAM0 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

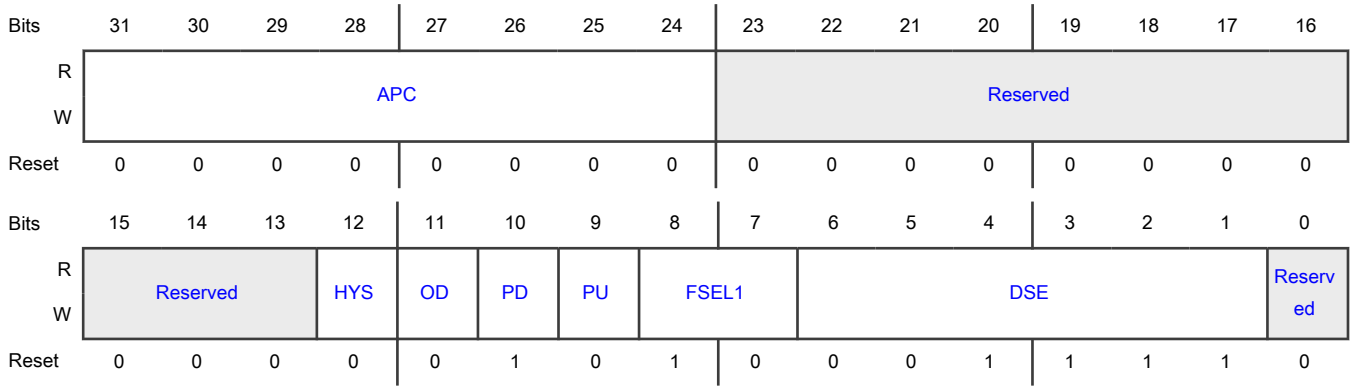
27.3.1.212 SW_PAD_CTL_PAD_PDM_BIT_STREAM1 SW PAD Control Register (SW_PAD_CTL_PAD_PDM_BIT_STREAM1)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_PDM_BIT_STREAM1	348h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: PDM_BIT_STREAM1 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: PDM_BIT_STREAM1 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: PDM_BIT_STREAM1 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: PDM_BIT_STREAM1 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: PDM_BIT_STREAM1 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: PDM_BIT_STREAM1 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

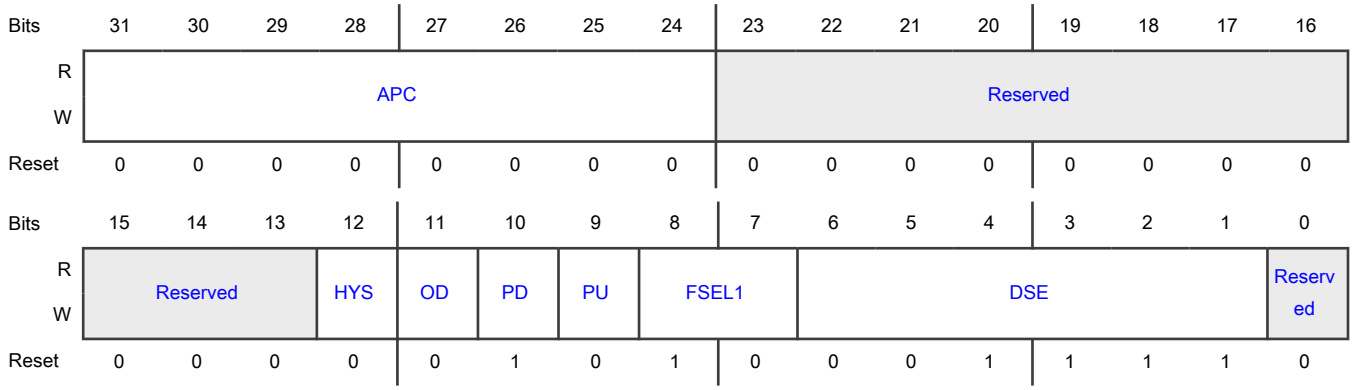
27.3.1.213 SW_PAD_CTL_PAD_SAI1_TXFS SW PAD Control Register (SW_PAD_CTL_PAD_SAI1_TXFS)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SAI1_TXFS	34Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SAI1_TXFS 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SAI1_TXFS 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SAI1_TXFS 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SAI1_TXFS 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SAI1_TXFS 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXFS 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

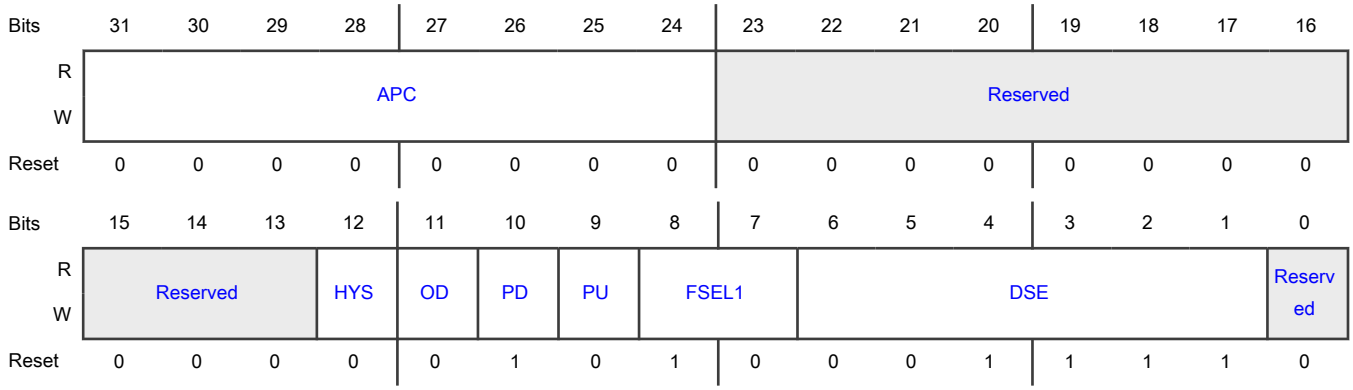
27.3.1.214 SW_PAD_CTL_PAD_SAI1_TXC SW PAD Control Register (SW_PAD_CTL_PAD_SAI1_TXC)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SAI1_TXC	350h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SAI1_TXC 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SAI1_TXC 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SAI1_TXC 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SAI1_TXC 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SAI1_TXC 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXC 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

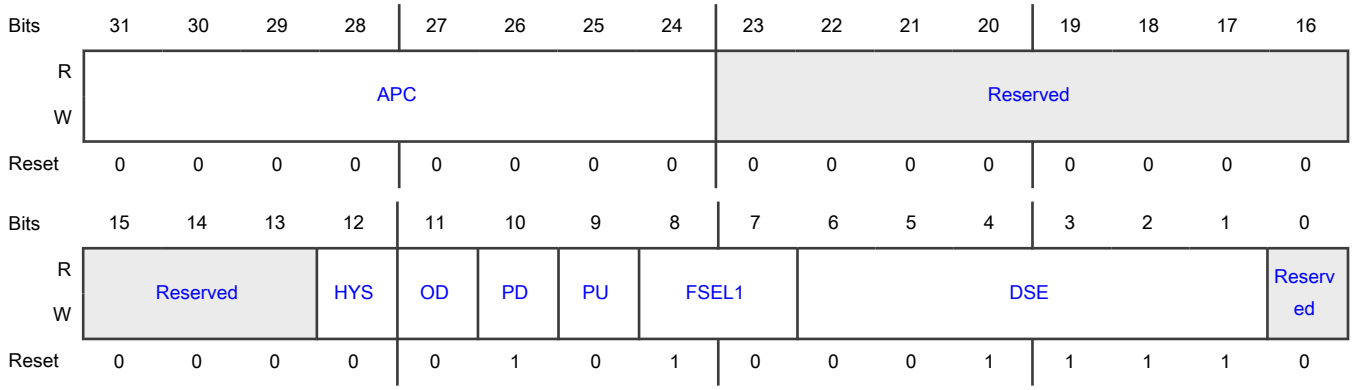
27.3.1.215 SW_PAD_CTL_PAD_SAI1_TXD0 SW PAD Control Register (SW_PAD_CTL_PAD_SAI1_TXD0)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SAI1_TXD0	354h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SAI1_TXD0 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SAI1_TXD0 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SAI1_TXD0 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SAI1_TXD0 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SAI1_TXD0 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD0 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

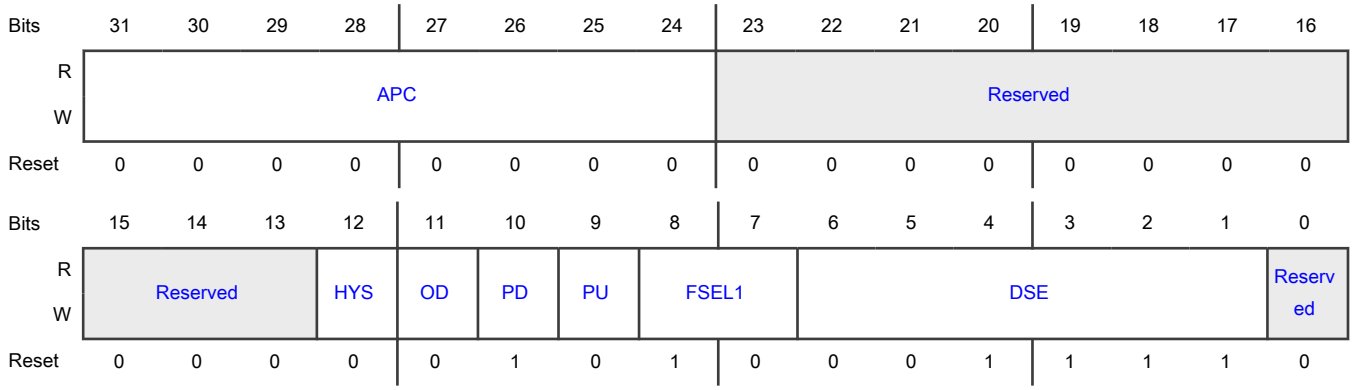
27.3.1.216 SW_PAD_CTL_PAD_SAI1_RXD0 SW PAD Control Register (SW_PAD_CTL_PAD_SAI1_RXD0)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_SAI1_RXD0	358h

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: SAI1_RXD0 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: SAI1_RXD0 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: SAI1_RXD0 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: SAI1_RXD0 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: SAI1_RXD0 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD0 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

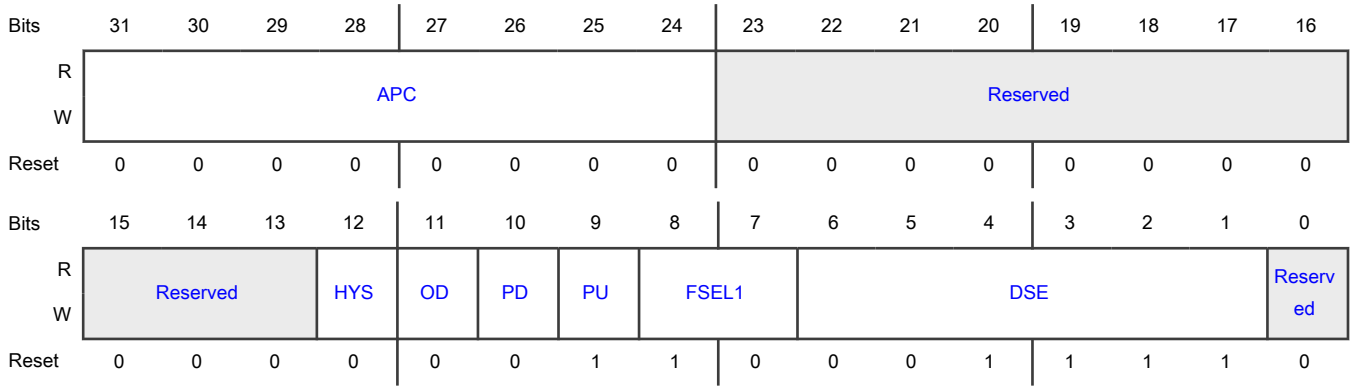
27.3.1.217 SW_PAD_CTL_PAD_WDOG_ANY SW PAD Control Register (SW_PAD_CTL_PAD_WDOG_ANY)

SW_PAD_CTL Register

Offset

Register	Offset
SW_PAD_CTL_PAD_WDOG_ANY	35Ch

Diagram



Fields

Field	Description
31-24 APC	Domain Access Field For APC, the upper 4 bits are lock bits and the lower 4 bits are domain control bits. The higher 4 bits control whether the lower 4 bits can be changed, respectively. If the lock bit is set to 1, the corresponding domain control bit cannot be changed. The higher 4 bits are sticky registers, which means that once they are set to 1, they cannot be set to 0 unless reset. The domain control bits control whether the corresponding domain can access the IOMUX registers. When set to 1, the corresponding domain cannot access the IOMUX register. When set to 0, the corresponding domain can access the IOMUX register. Domain control bit 3 controls access for domain 3, bit 2 controls access for domain 2, and bit 1 controls access for domain 1. Domain control bit 0 controls whether other domains can access the IOMUX register.
23-13 —	- Reserved
12 HYS	Schmitt trigger Field Select one out of next values for pad: WDOG_ANY 0 - No Schmitt input 1 - Schmitt input
11 OD	Open Drain Field Select one out of next values for pad: WDOG_ANY 0 - Open Drain Disable 1 - Open Drain Enable
10 PD	Pull Down Field Select one out of next values for pad: WDOG_ANY 0 - Not pull down 1 - Pull down

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 PU	Pull Up Field Select one out of next values for pad: WDOG_ANY 0 - No pull up 1 - Pull up
8-7 FSEL1	Slew Rate Field 2'b10 and 2'b11 are identical. Select one out of next values for pad: WDOG_ANY 00 - Reserved 01 - Reserved 10 - Fast Slew Rate 11 - Fast Slew Rate
6-1 DSE	Drive Strength Field Select one out of next values for pad: WDOG_ANY 00_0000 - No drive 00_0001 - X1 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0011 - X2 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_0111 - X3 (1.8V and 3.3V I/O supply for NVCC_XXX) 00_1111 - X4 (1.8V and 3.3V I/O supply for NVCC_XXX) 01_1111 - X5 (1.8V I/O supply for NVCC_XXX) 11_1111 - X6 (1.8V I/O supply for NVCC_XXX)
0 —	- Reserved

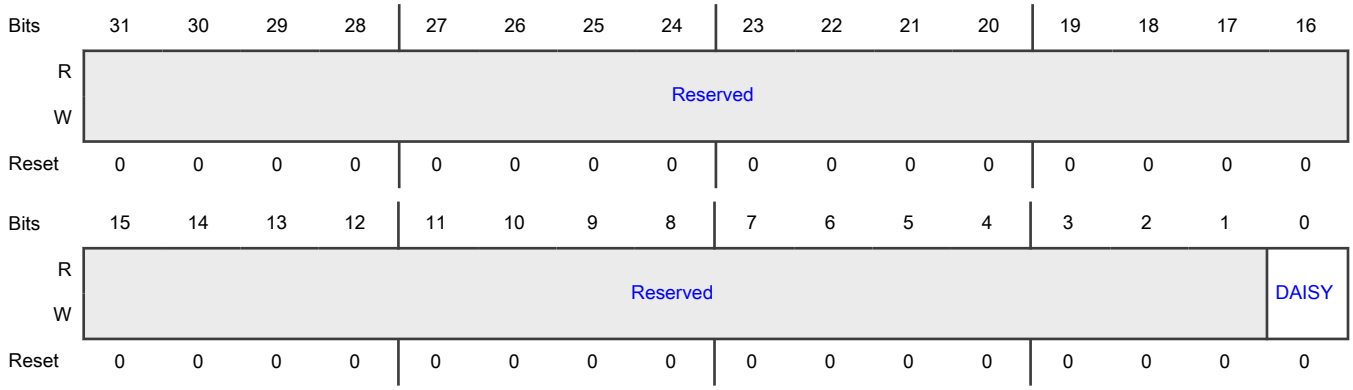
27.3.1.218 CAN1_IPP_IND_CANRX_SELECT_INPUT DAISY Register (CAN1_IPP_IND_CANRX_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
CAN1_IPP_IND_CANRX_SELECT_INPUT	360h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: can1, In Pin: ipp_ind_canrx 0 - Selecting Pad: PDM_BIT_STREAM0 for Mode: ALT6 1 - Selecting Pad: SAI1_TXC for Mode: ALT4

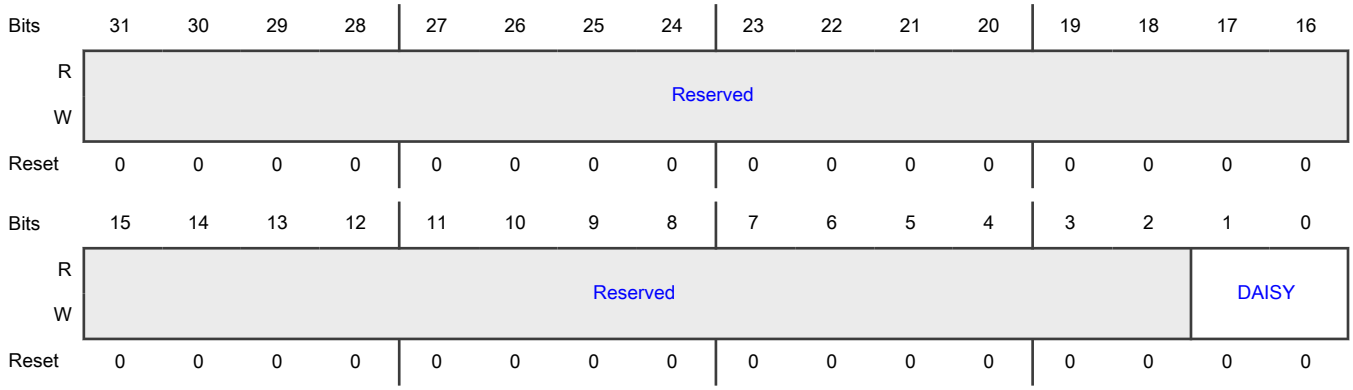
27.3.1.219 CAN2_IPP_IND_CANRX_SELECT_INPUT DAISY Register (CAN2_IPP_IND_CANRX_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
CAN2_IPP_IND_CANRX_SELECT_INPUT	364h

Diagram



Fields

Field	Description
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: can2, In Pin: ipp_ind_canrx 00 - Selecting Pad: DAP_TDO_TRACESWO for Mode: ALT3 01 - Selecting Pad: GPIO_IO27 for Mode: ALT2 10 - Selecting Pad: ENET1_TD2 for Mode: ALT2 11 - Selecting Pad: SD2_DATA1 for Mode: ALT2

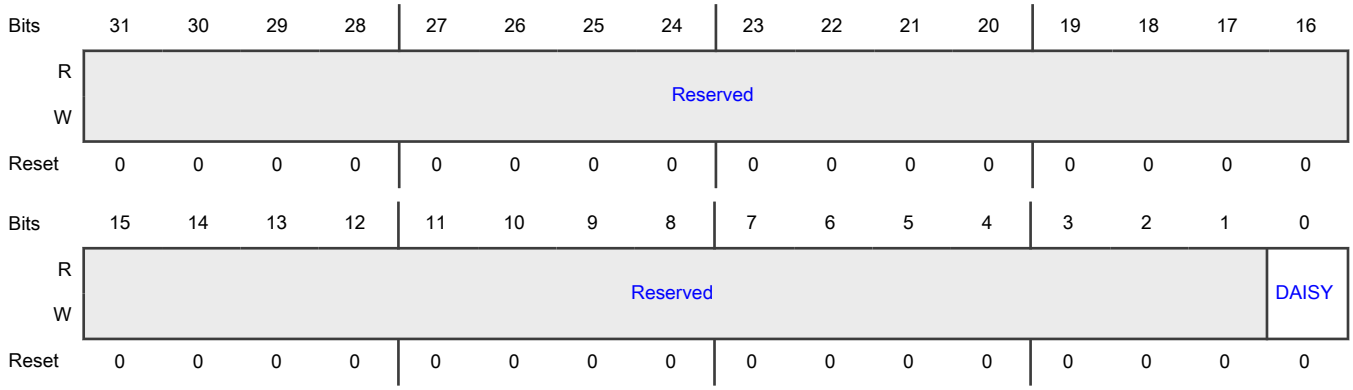
27.3.1.220 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_0 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_0)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_0	36Ch

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[0] 0 - Selecting Pad: GPIO_IO00 for Mode: ALT7 1 - Selecting Pad: SD2_CD_B for Mode: ALT4

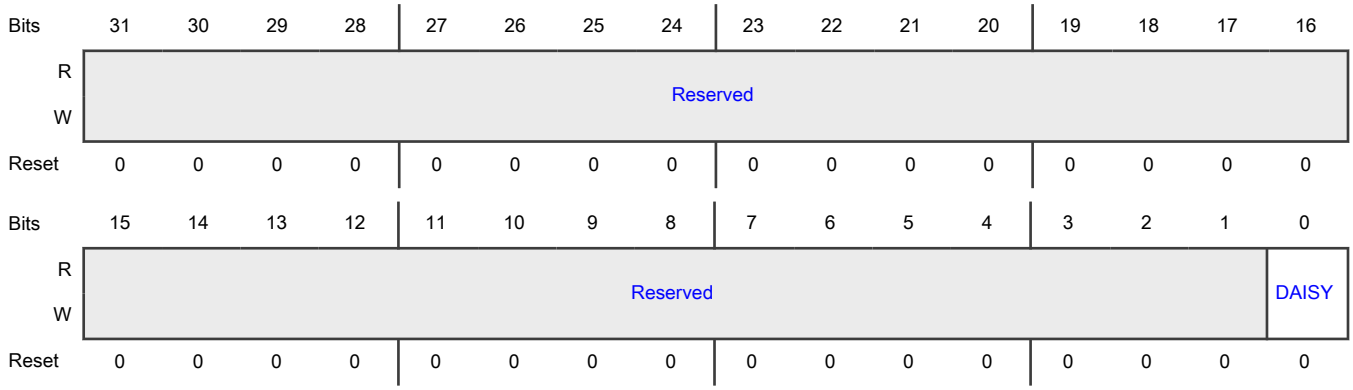
27.3.1.221 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_1 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_1)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_1	370h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[1] 0 - Selecting Pad: GPIO_IO01 for Mode: ALT7 1 - Selecting Pad: SD2_CLK for Mode: ALT4

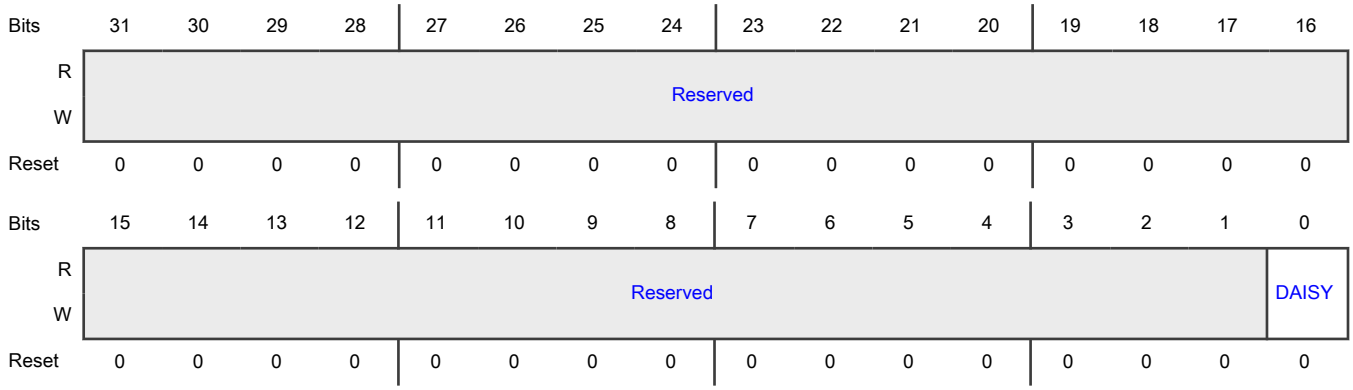
27.3.1.222 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_2 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_2)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_2	374h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[2] 0 - Selecting Pad: GPIO_IO02 for Mode: ALT7 1 - Selecting Pad: SD2_CMD for Mode: ALT4

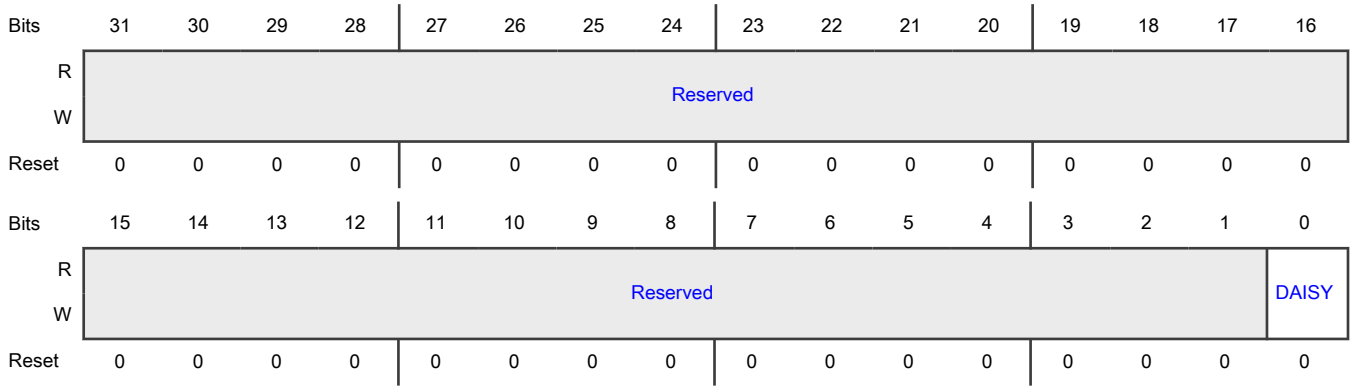
27.3.1.223 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_3 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_3)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_3	378h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[3] 0 - Selecting Pad: GPIO_IO03 for Mode: ALT7 1 - Selecting Pad: SD2_DATA0 for Mode: ALT4

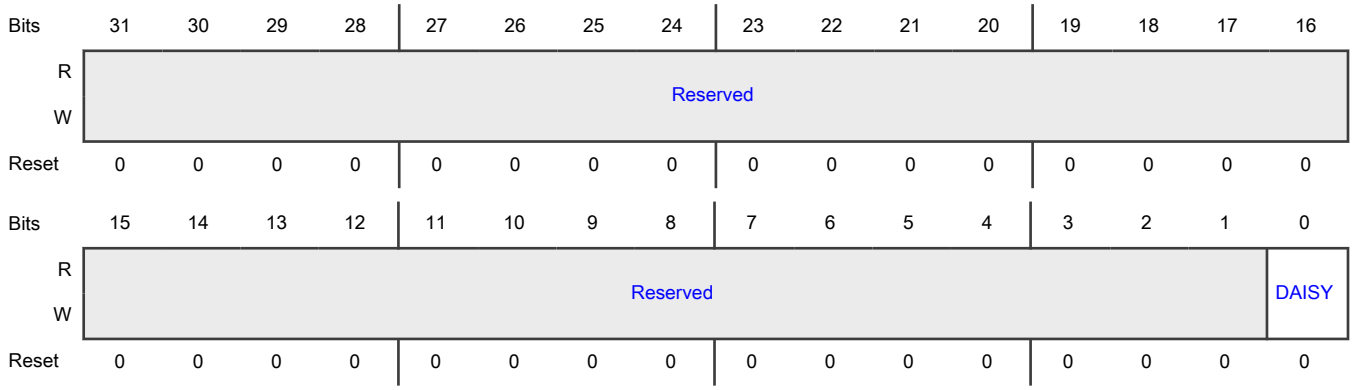
27.3.1.224 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_4 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_4)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_4	37Ch

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[4] 0 - Selecting Pad: GPIO_IO04 for Mode: ALT7 1 - Selecting Pad: SD2_DATA1 for Mode: ALT4

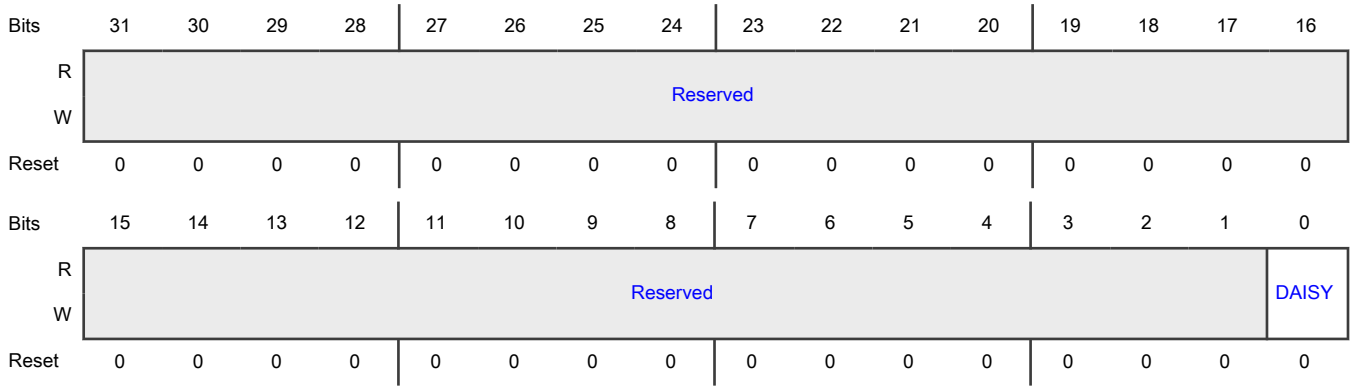
27.3.1.225 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_5 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_5)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_5	380h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[5] 0 - Selecting Pad: GPIO_IO05 for Mode: ALT7 1 - Selecting Pad: SD2_DATA2 for Mode: ALT4

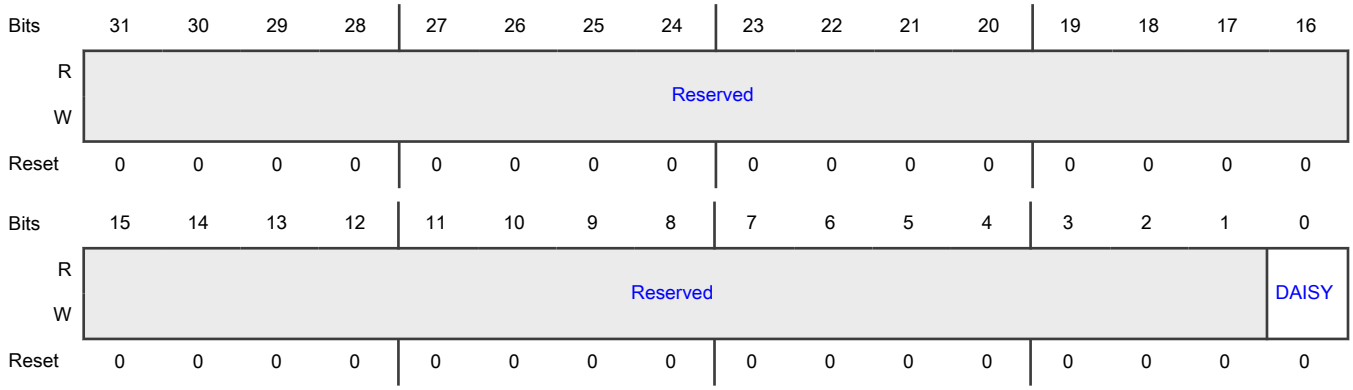
27.3.1.226 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_6 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_6)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_6	384h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[6] 0 - Selecting Pad: GPIO_IO06 for Mode: ALT7 1 - Selecting Pad: SD2_DATA3 for Mode: ALT4

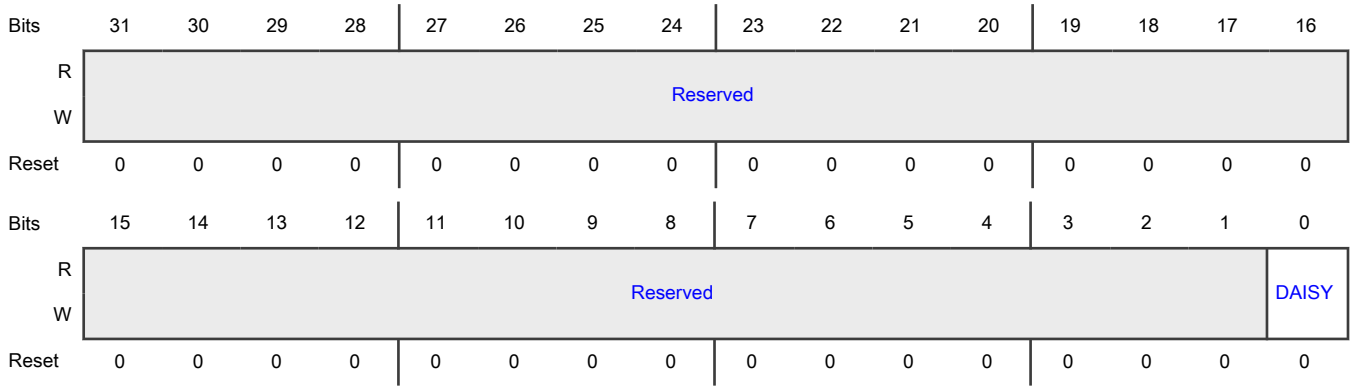
27.3.1.227 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_7 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_7)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_7	388h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[7] 0 - Selecting Pad: GPIO_IO07 for Mode: ALT7 1 - Selecting Pad: SD2_RESET_B for Mode: ALT4

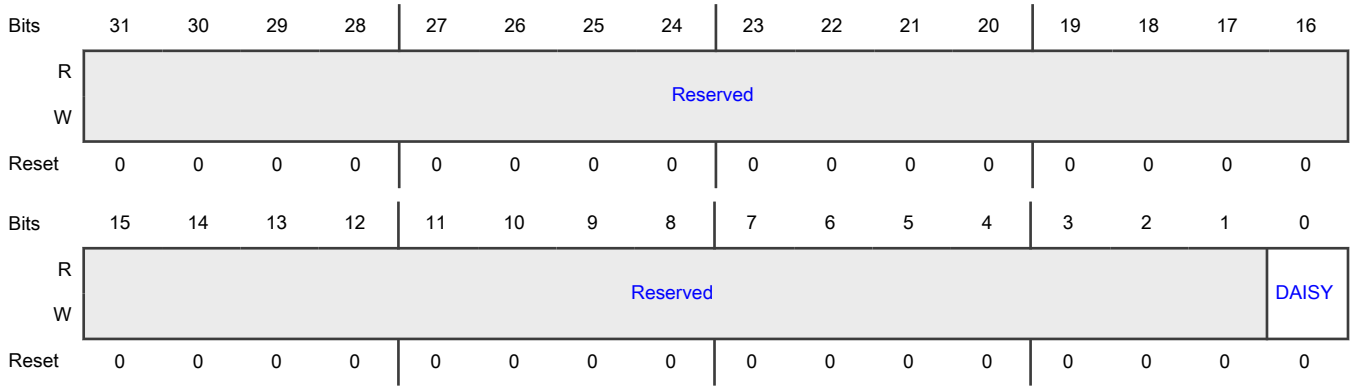
27.3.1.228 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_8 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_8)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_8	38Ch

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[8] 0 - Selecting Pad: GPIO_IO08 for Mode: ALT7 1 - Selecting Pad: SD1_CLK for Mode: ALT4

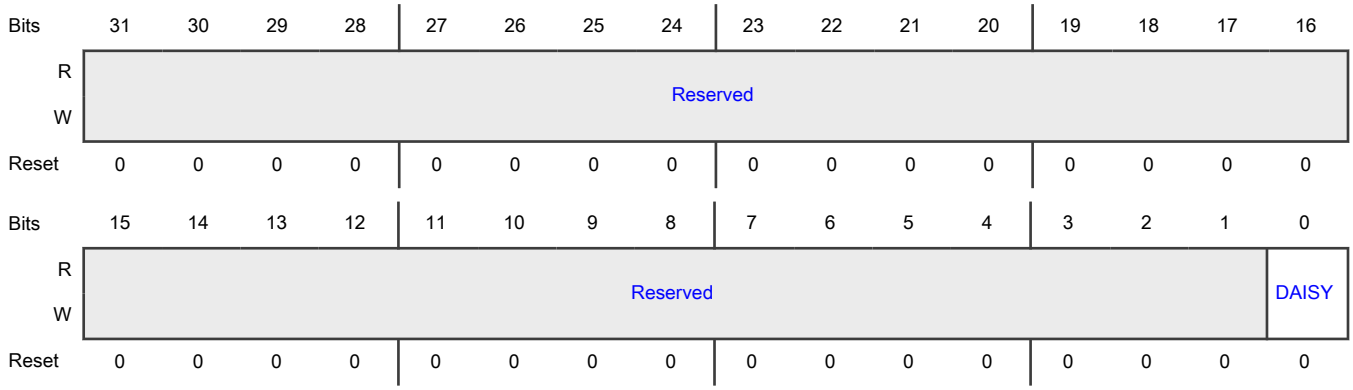
27.3.1.229 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_9 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_9)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_9	390h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[9] 0 - Selecting Pad: GPIO_IO09 for Mode: ALT7 1 - Selecting Pad: SD1_CMD for Mode: ALT4

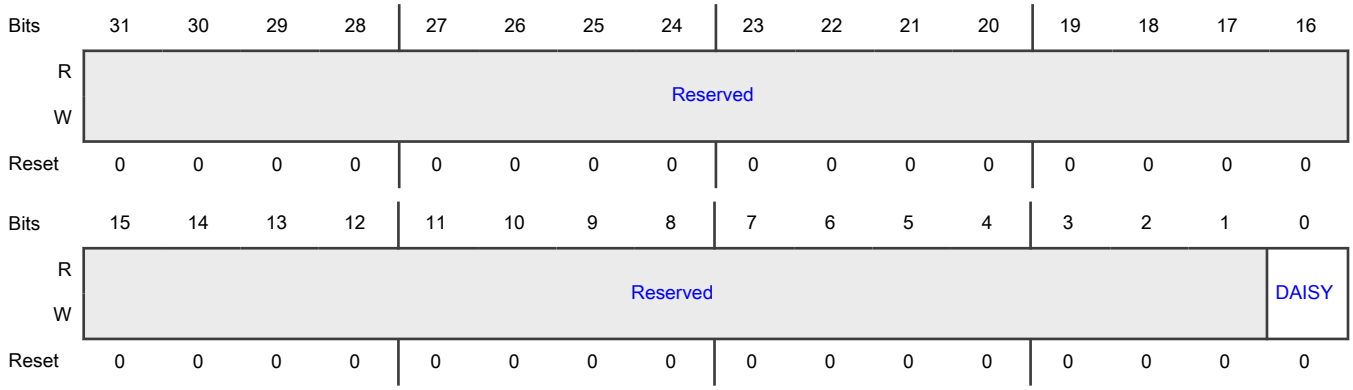
27.3.1.230 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_10 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_10)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_10	394h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[10] 0 - Selecting Pad: GPIO_IO10 for Mode: ALT7 1 - Selecting Pad: SD1_DATA0 for Mode: ALT4

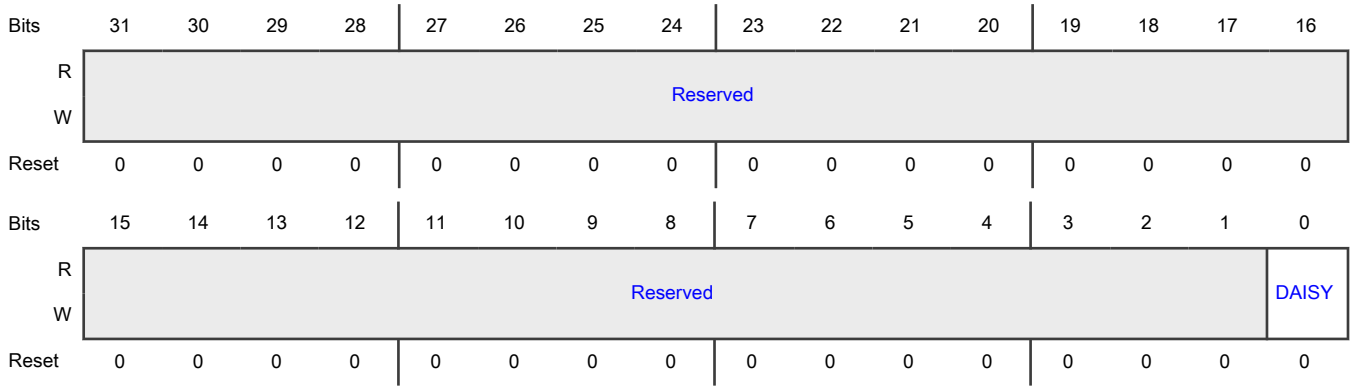
27.3.1.231 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_11 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_11)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_11	398h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[11] 0 - Selecting Pad: GPIO_IO11 for Mode: ALT7 1 - Selecting Pad: SD1_DATA1 for Mode: ALT4

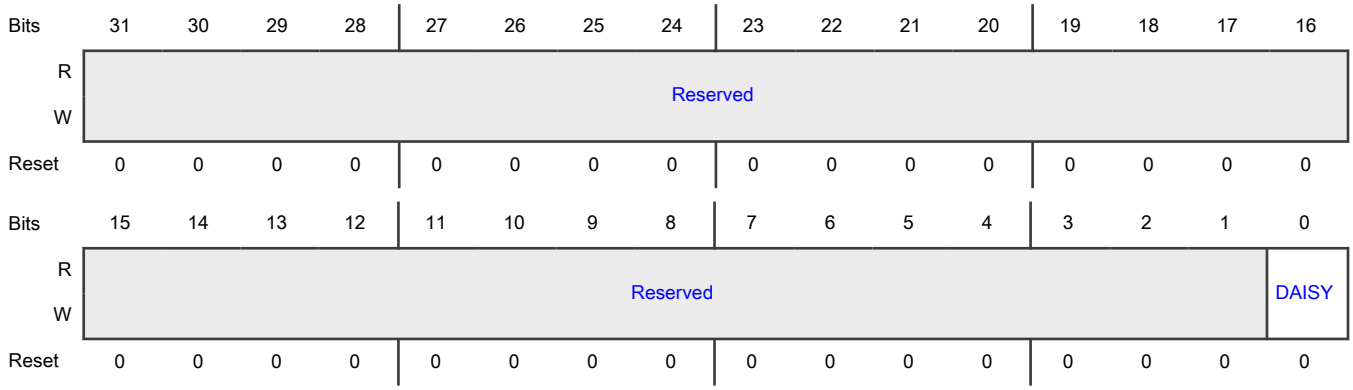
27.3.1.232 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_13 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_13)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_13	39Ch

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[13] 0 - Selecting Pad: GPIO_IO13 for Mode: ALT7 1 - Selecting Pad: SD1_DATA3 for Mode: ALT4

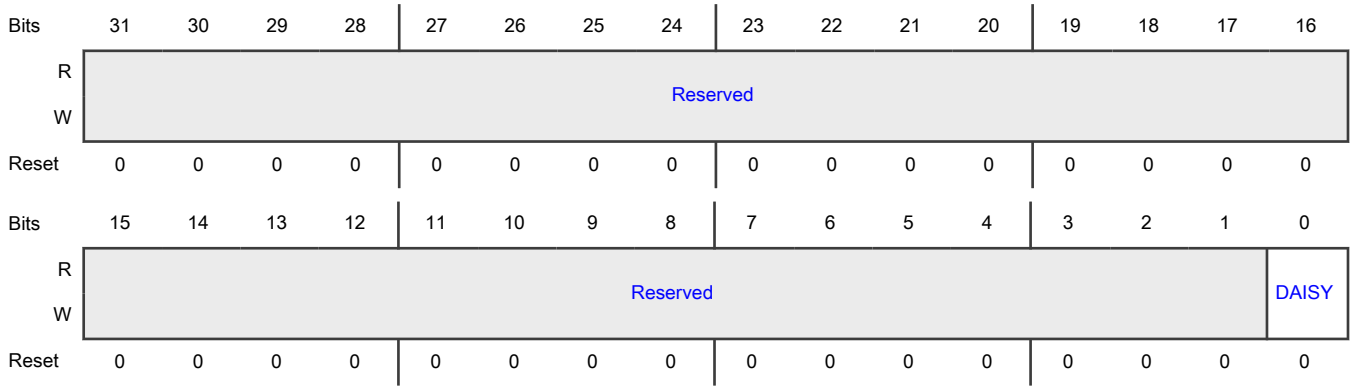
27.3.1.233 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_14 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_14)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_14	3A0h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[14] 0 - Selecting Pad: GPIO_IO14 for Mode: ALT7 1 - Selecting Pad: SD1_DATA4 for Mode: ALT4

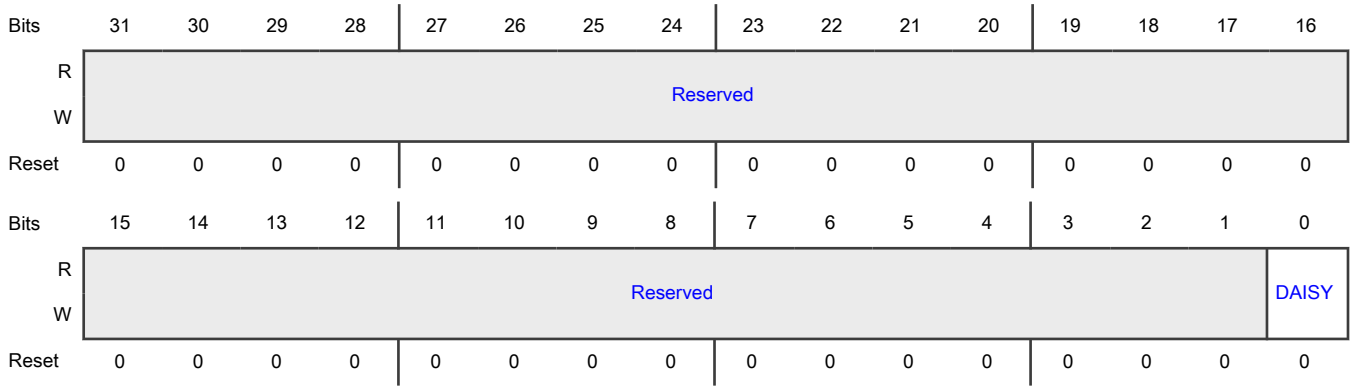
27.3.1.234 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_15 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_15)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_15	3A4h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[15] 0 - Selecting Pad: GPIO_IO15 for Mode: ALT7 1 - Selecting Pad: SD1_DATA5 for Mode: ALT4

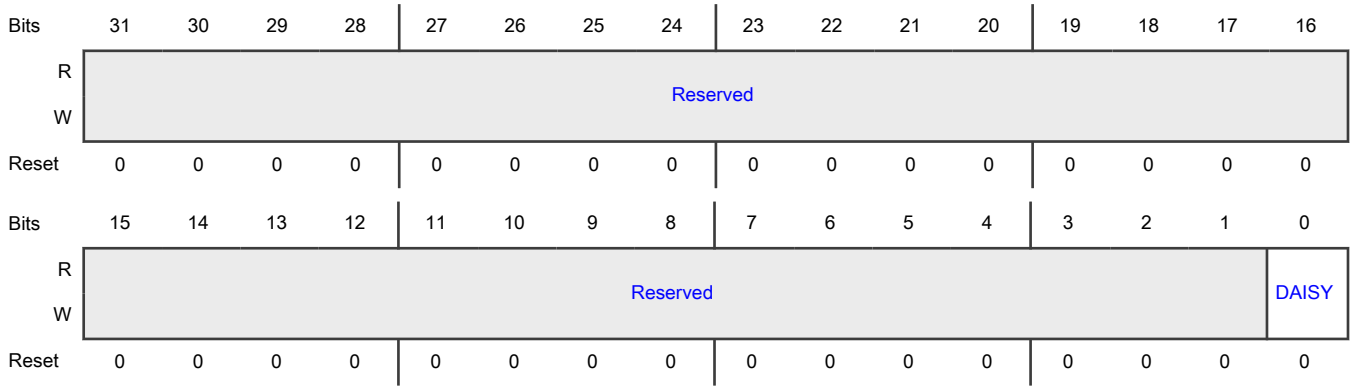
27.3.1.235 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_16 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_16)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_16	3A8h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[16] 0 - Selecting Pad: GPIO_IO16 for Mode: ALT7 1 - Selecting Pad: SD1_DATA6 for Mode: ALT4

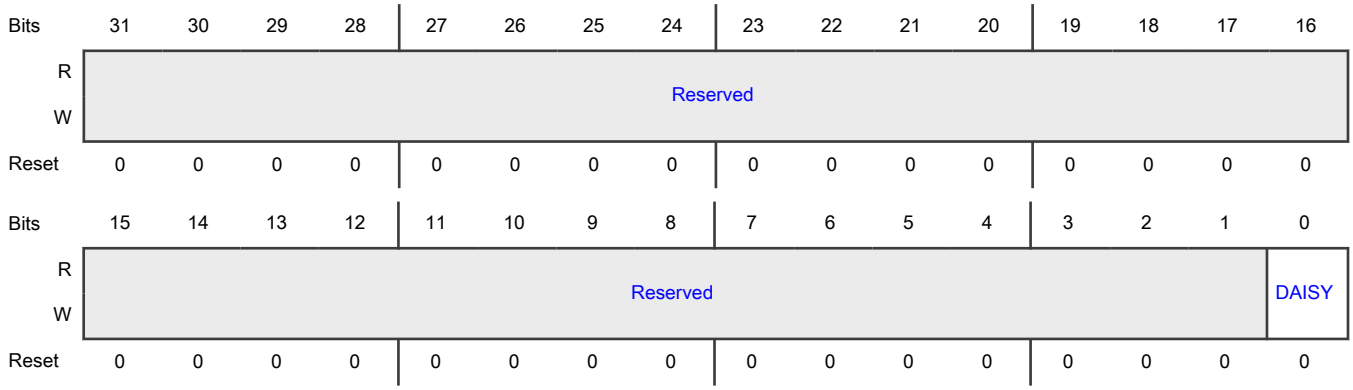
27.3.1.236 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_17 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_17)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_17	3ACh

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[17] 0 - Selecting Pad: GPIO_IO17 for Mode: ALT7 1 - Selecting Pad: SD1_DATA7 for Mode: ALT4

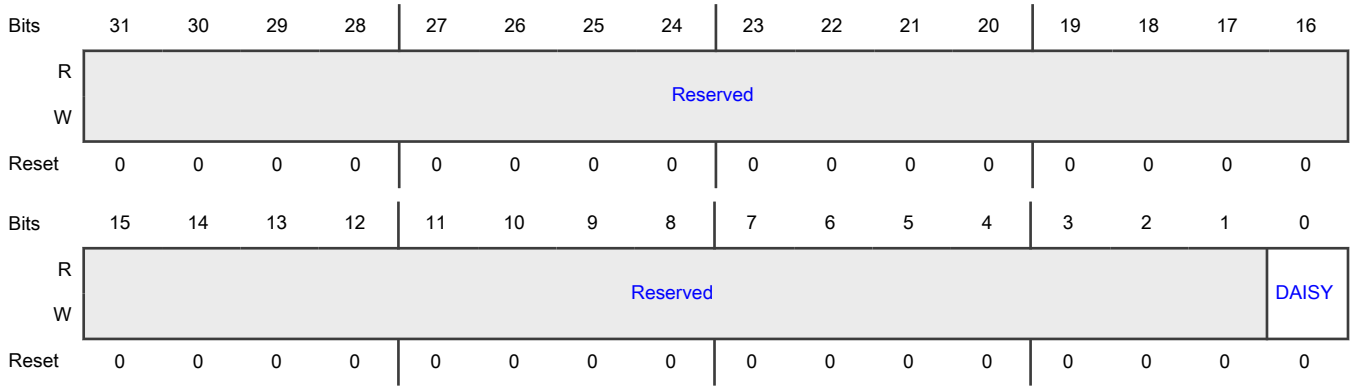
27.3.1.237 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_18 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_18)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_18	3B0h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[18] 0 - Selecting Pad: GPIO_IO18 for Mode: ALT7 1 - Selecting Pad: SD1_STROBE for Mode: ALT4

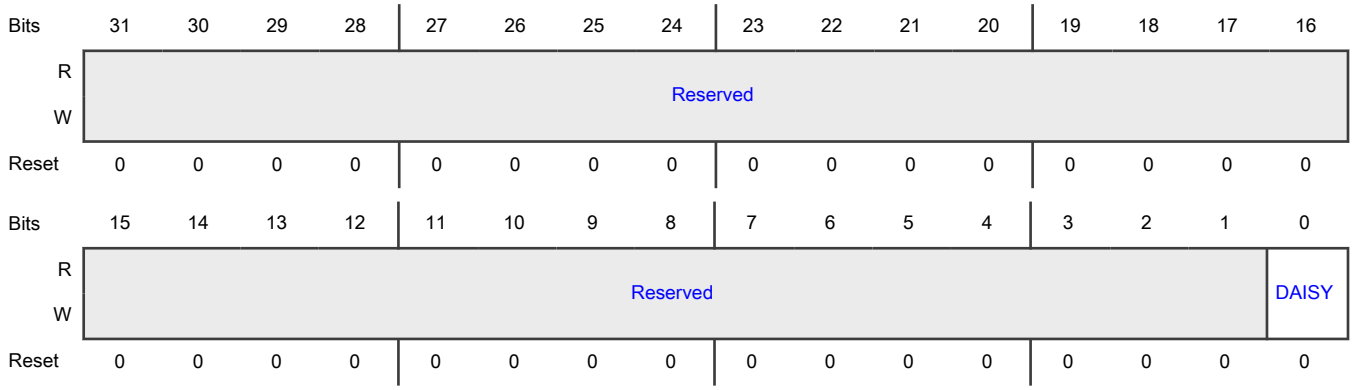
27.3.1.238 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_20 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_20)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_20	3B4h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[20] 0 - Selecting Pad: GPIO_IO20 for Mode: ALT7 1 - Selecting Pad: SD3_CLK for Mode: ALT4

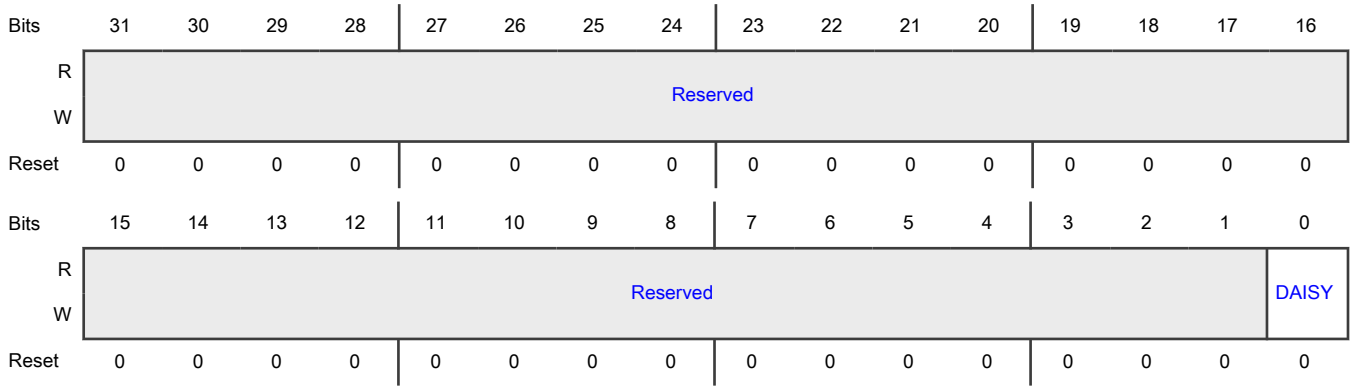
27.3.1.239 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_22 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_22)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_22	3B8h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[22] 0 - Selecting Pad: GPIO_IO22 for Mode: ALT7 1 - Selecting Pad: SD3_DATA0 for Mode: ALT4

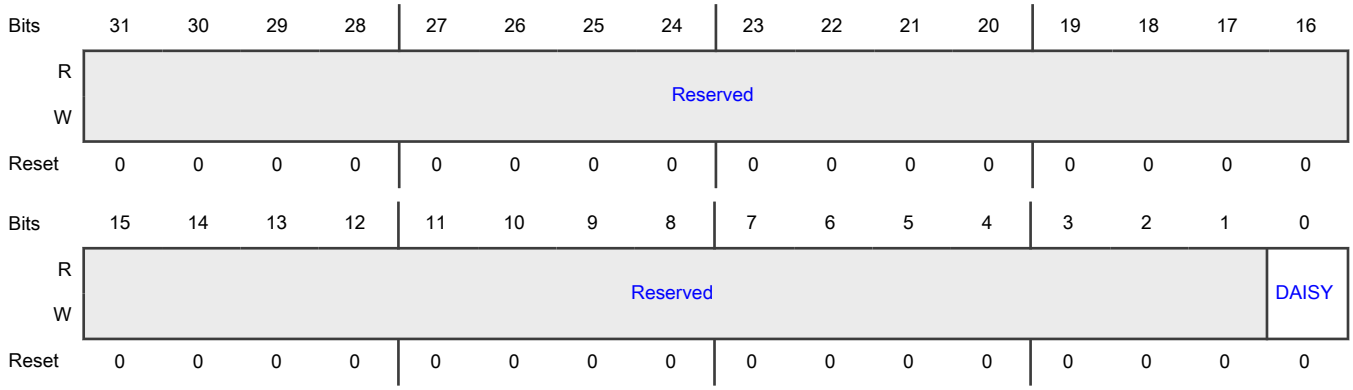
27.3.1.240 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_23 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_23)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_23	3BCh

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[23] 0 - Selecting Pad: GPIO_IO23 for Mode: ALT7 1 - Selecting Pad: SD3_DATA1 for Mode: ALT4

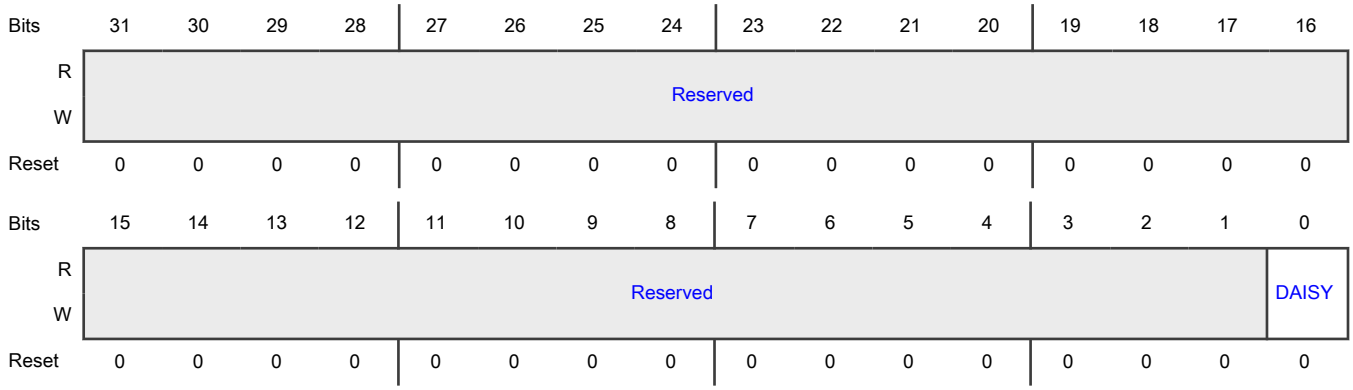
27.3.1.241 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_24 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_24)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_24	3C0h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[24] 0 - Selecting Pad: GPIO_IO24 for Mode: ALT7 1 - Selecting Pad: SD3_DATA2 for Mode: ALT4

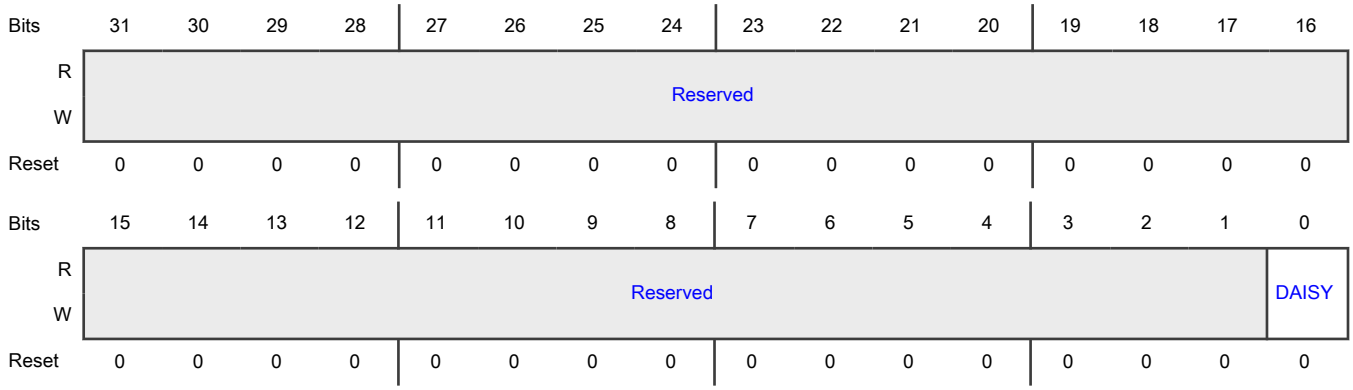
27.3.1.242 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_25 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_25)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_25	3C4h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[25] 0 - Selecting Pad: GPIO_IO25 for Mode: ALT7 1 - Selecting Pad: SD3_DATA3 for Mode: ALT4

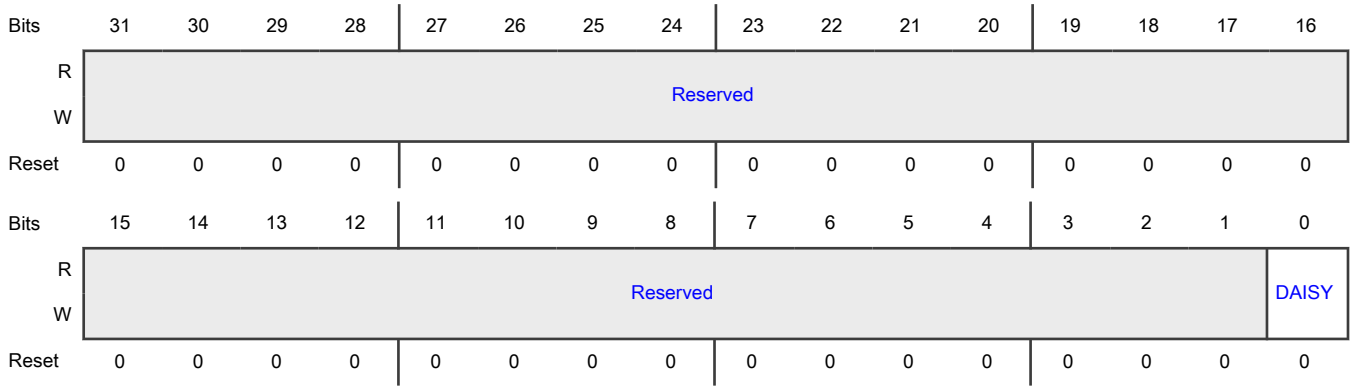
27.3.1.243 FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_27 DAISY Register (FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_27)

DAISY Register

Offset

Register	Offset
FLEXIO1_IPP_IND_FLEXIO_SELECT_INPUT_27	3C8h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexio1, In Pin: ipp_ind_flexio[27] 0 - Selecting Pad: GPIO_IO27 for Mode: ALT7 1 - Selecting Pad: CCM_CLKO2 for Mode: ALT4

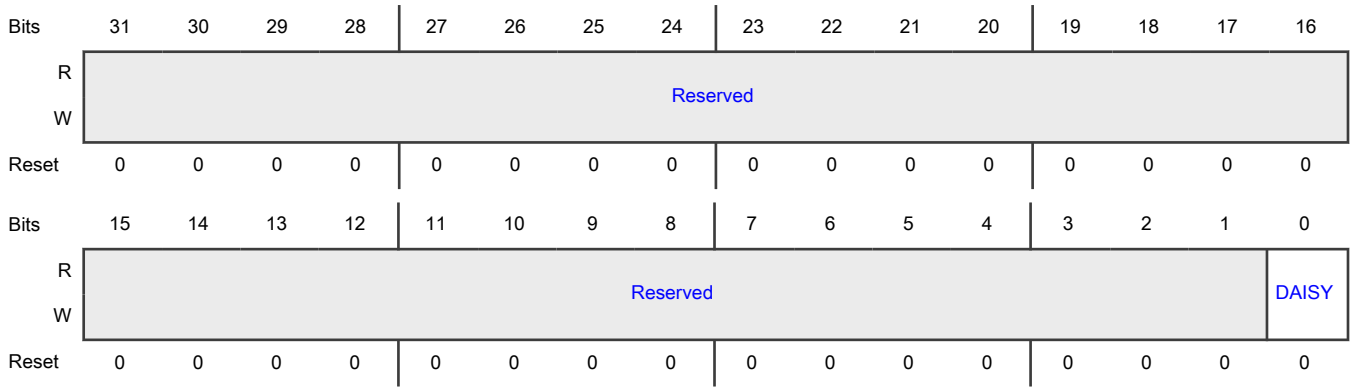
27.3.1.244 I3C2_PIN_SCL_IN_SELECT_INPUT DAISY Register (I3C2_PIN_SCL_IN_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
I3C2_PIN_SCL_IN_SELECT_INPUT	3CCh

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i3c2, In Pin: pin_SCL_in 0 - Selecting Pad: ENET1_MDC for Mode: ALT2 1 - Selecting Pad: SD2_CD_B for Mode: ALT2

27.3.1.245 I3C2_PIN_SDA_IN_SELECT_INPUT DAISY Register (I3C2_PIN_SDA_IN_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
I3C2_PIN_SDA_IN_SELECT_INPUT	3D0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i3c2, In Pin: pin_SDA_in 0 - Selecting Pad: ENET1_MDIO for Mode: ALT2 1 - Selecting Pad: SD2_CLK for Mode: ALT2

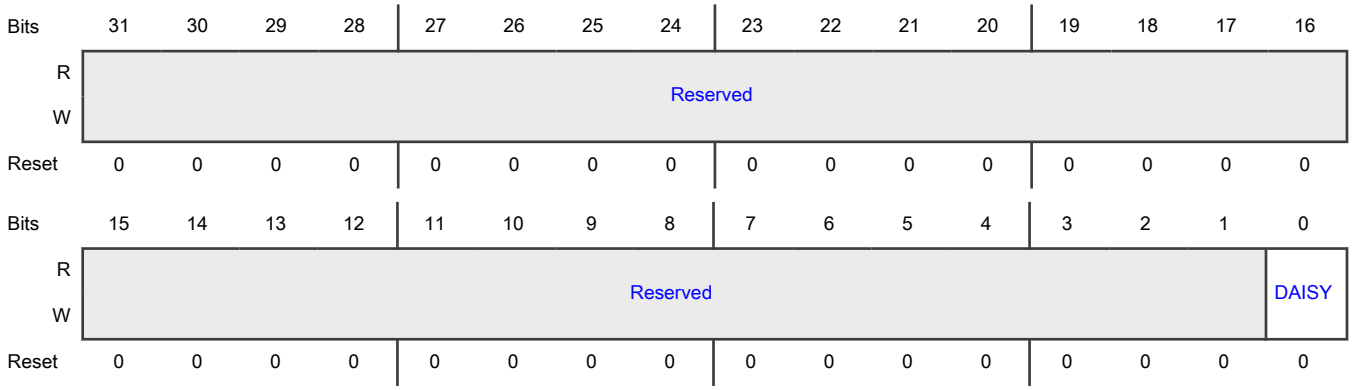
27.3.1.246 JTAG_MUX_TCK_SELECT_INPUT DAISY Register (JTAG_MUX_TCK_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
JTAG_MUX_TCK_SELECT_INPUT	3D4h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: jtag_mux, In Pin: tck 0 - Selecting Pad: DAP_TCLK_SWCLK for Mode: ALT0 1 - Selecting Pad: GPIO_IO25 for Mode: ALT5

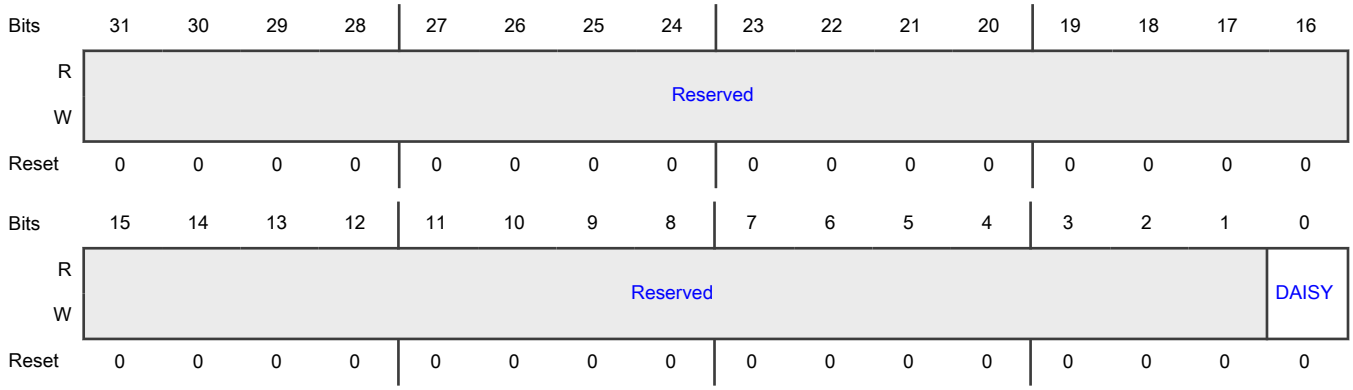
27.3.1.247 JTAG_MUX_TDI_SELECT_INPUT DAISY Register (JTAG_MUX_TDI_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
JTAG_MUX_TDI_SELECT_INPUT	3D8h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: jtag_mux, In Pin: tdi 0 - Selecting Pad: DAP_TDI for Mode: ALT0 1 - Selecting Pad: GPIO_IO26 for Mode: ALT5

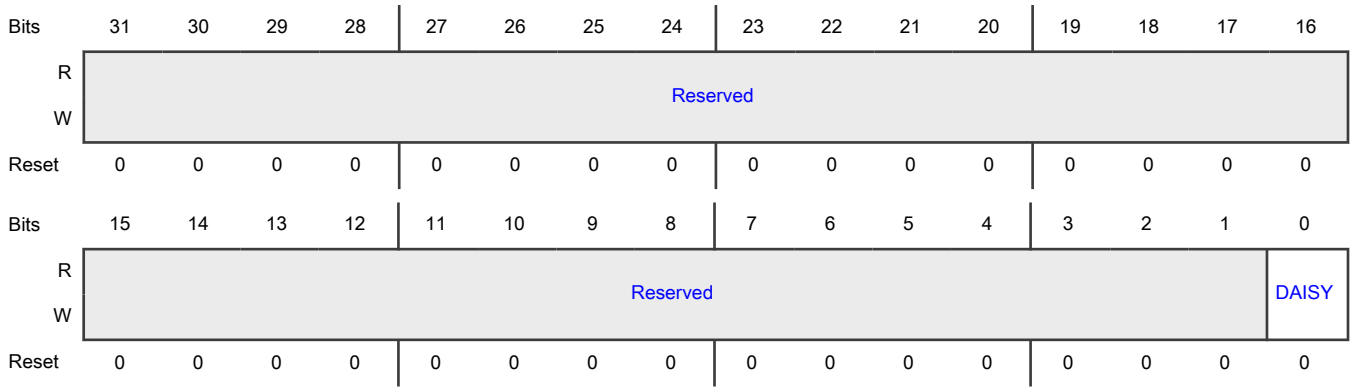
27.3.1.248 JTAG_MUX_TMS_SELECT_INPUT DAISY Register (JTAG_MUX_TMS_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
JTAG_MUX_TMS_SELECT_INPUT	3DCh

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: jtag_mux, In Pin: tms 0 - Selecting Pad: DAP_TMS_SWDIO for Mode: ALT0 1 - Selecting Pad: GPIO_IO27 for Mode: ALT5

27.3.1.249 LPI2C3_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C3_IPP_IND_LPI2C_SCL_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPI2C3_IPP_IND_LPI2C_SCL_SELECT_INPUT	3E0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c3, In Pin: ipp_ind_lpi2c_scl

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Selecting Pad: GPIO_IO01 for Mode: ALT1 1 - Selecting Pad: GPIO_IO29 for Mode: ALT1

27.3.1.250 LPI2C3_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C3_IPP_IND_LPI2C_SDA_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPI2C3_IPP_IND_LPI2C_SDA_SELECT_INPUT	3E4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c3, In Pin: ipp_ind_lpi2c_sda 0 - Selecting Pad: GPIO_IO00 for Mode: ALT1 1 - Selecting Pad: GPIO_IO28 for Mode: ALT1

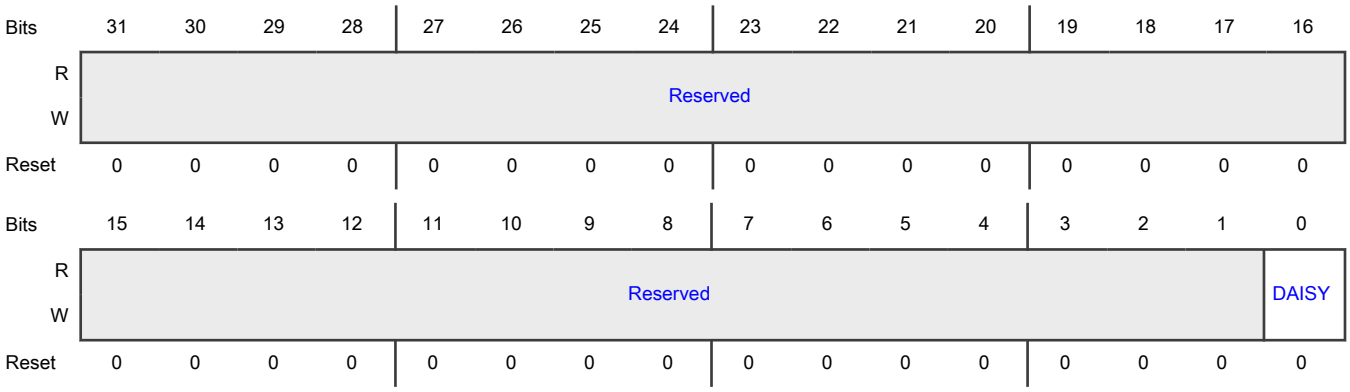
27.3.1.251 LPI2C5_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C5_IPP_IND_LPI2C_SCL_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPI2C5_IPP_IND_LPI2C_SCL_SELECT_INPUT	3E8h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c5, In Pin: ipp_ind_lpi2c_scl 0 - Selecting Pad: GPIO_IO01 for Mode: ALT6 1 - Selecting Pad: GPIO_IO23 for Mode: ALT6

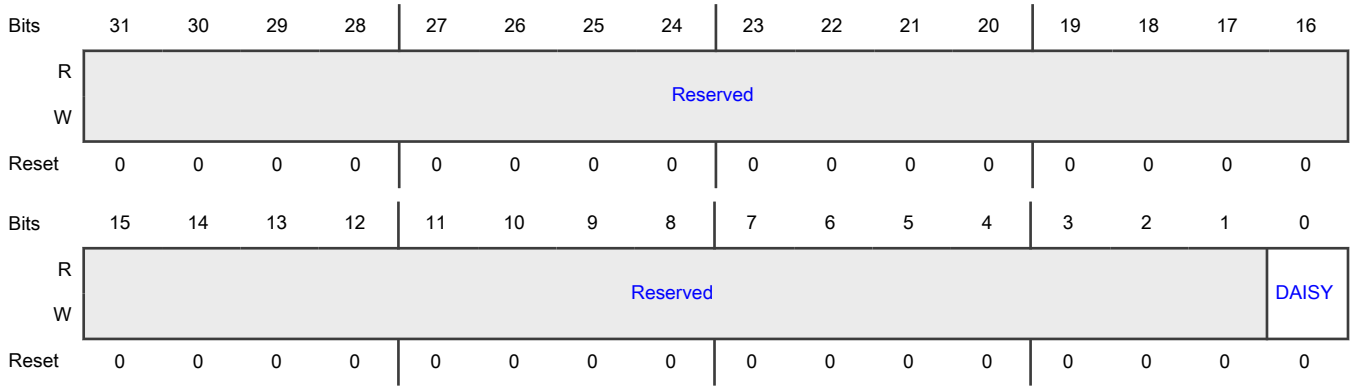
27.3.1.252 LPI2C5_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C5_IPP_IND_LPI2C_SDA_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPI2C5_IPP_IND_LPI2C_SDA_SELECT_INPUT	3ECh

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c5, In Pin: ipp_ind_lpi2c_sda 0 - Selecting Pad: GPIO_IO00 for Mode: ALT6 1 - Selecting Pad: GPIO_IO22 for Mode: ALT6

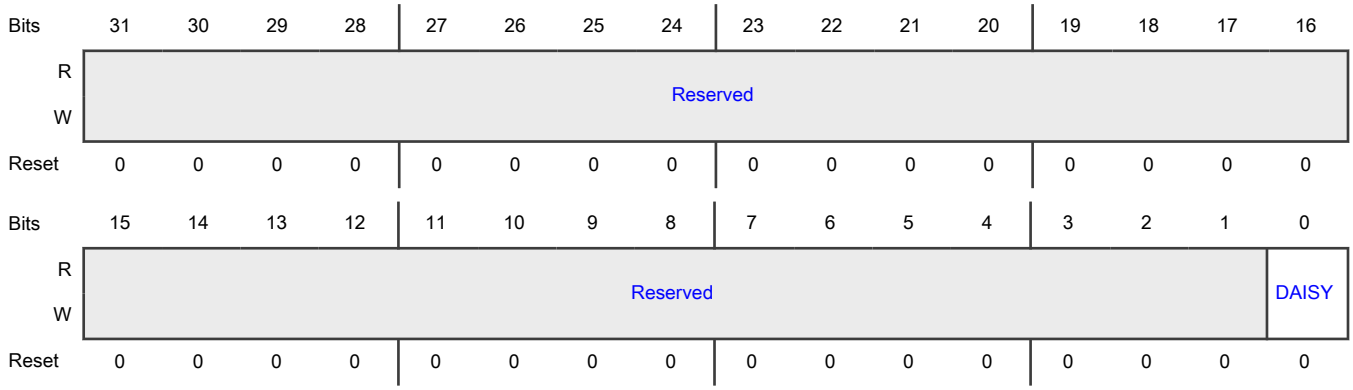
27.3.1.253 LPI2C6_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C6_IPP_IND_LPI2C_SCL_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPI2C6_IPP_IND_LPI2C_SCL_SELECT_INPUT	3F0h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c6, In Pin: ipp_ind_lpi2c_scl 0 - Selecting Pad: GPIO_IO03 for Mode: ALT6 1 - Selecting Pad: GPIO_IO05 for Mode: ALT6

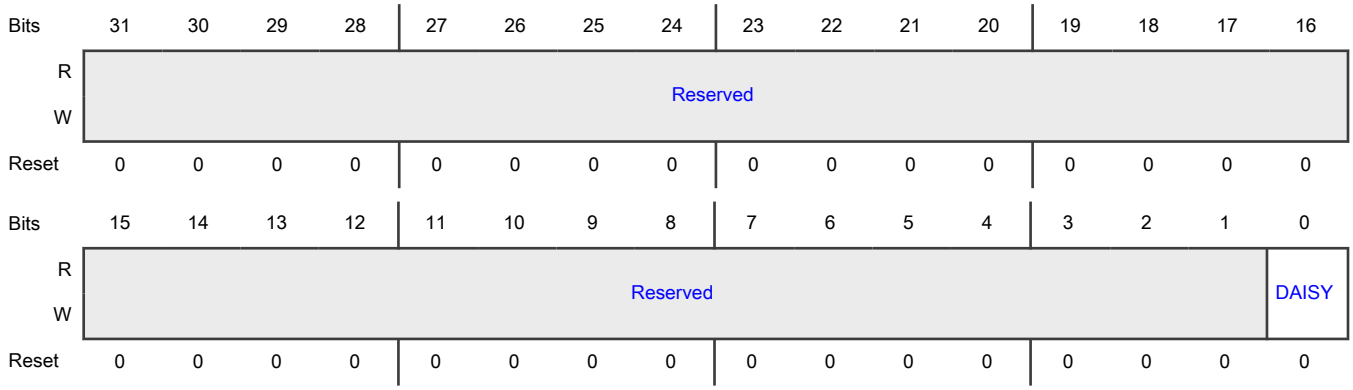
27.3.1.254 LPI2C6_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C6_IPP_IND_LPI2C_SDA_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPI2C6_IPP_IND_LPI2C_SDA_SELECT_INPUT	3F4h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c6, In Pin: ipp_ind_lpi2c_sda 0 - Selecting Pad: GPIO_IO02 for Mode: ALT6 1 - Selecting Pad: GPIO_IO04 for Mode: ALT6

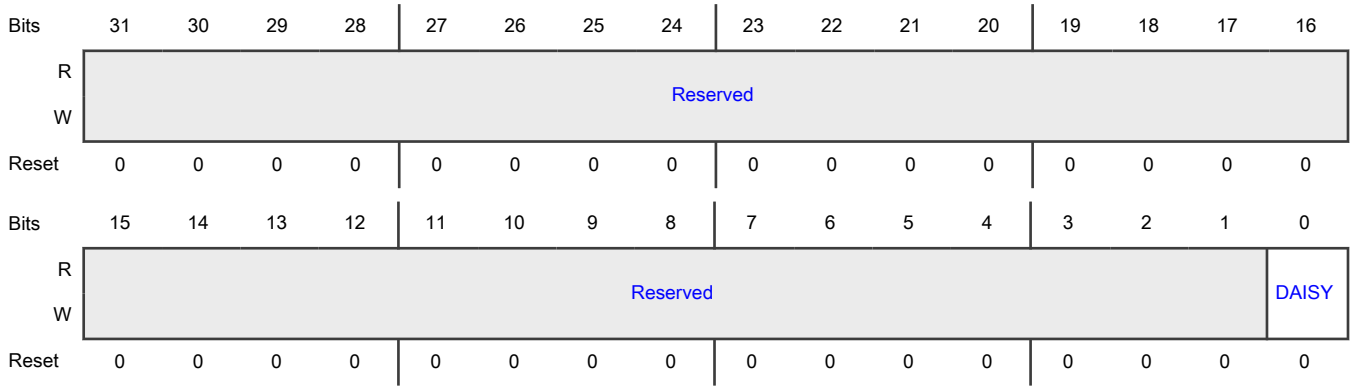
27.3.1.255 LPI2C7_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C7_IPP_IND_LPI2C_SCL_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPI2C7_IPP_IND_LPI2C_SCL_SELECT_INPUT	3F8h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c7, In Pin: ipp_ind_lpi2c_scl 0 - Selecting Pad: GPIO_IO07 for Mode: ALT6 1 - Selecting Pad: GPIO_IO09 for Mode: ALT6

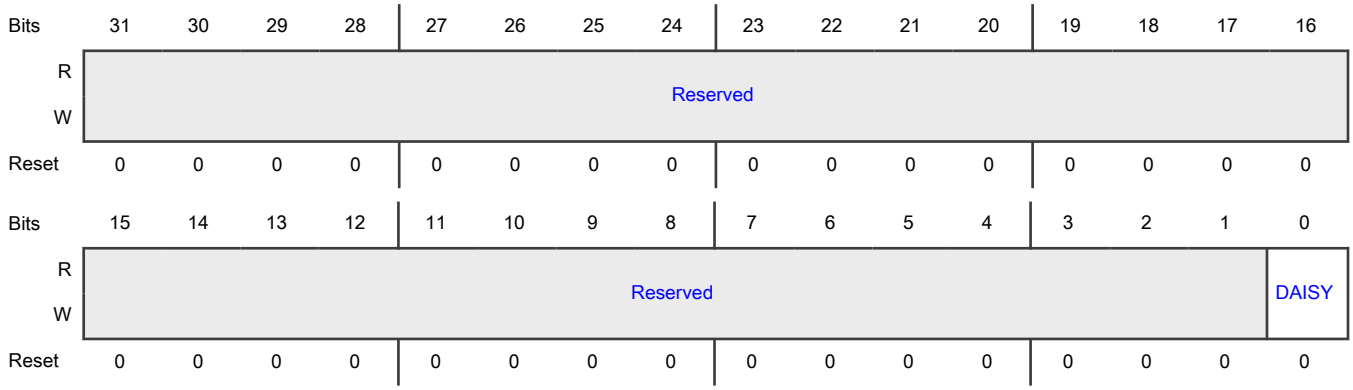
27.3.1.256 LPI2C7_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C7_IPP_IND_LPI2C_SDA_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPI2C7_IPP_IND_LPI2C_SDA_SELECT_INPUT	3FCh

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c7, In Pin: ipp_ind_lpi2c_sda 0 - Selecting Pad: GPIO_IO06 for Mode: ALT6 1 - Selecting Pad: GPIO_IO08 for Mode: ALT6

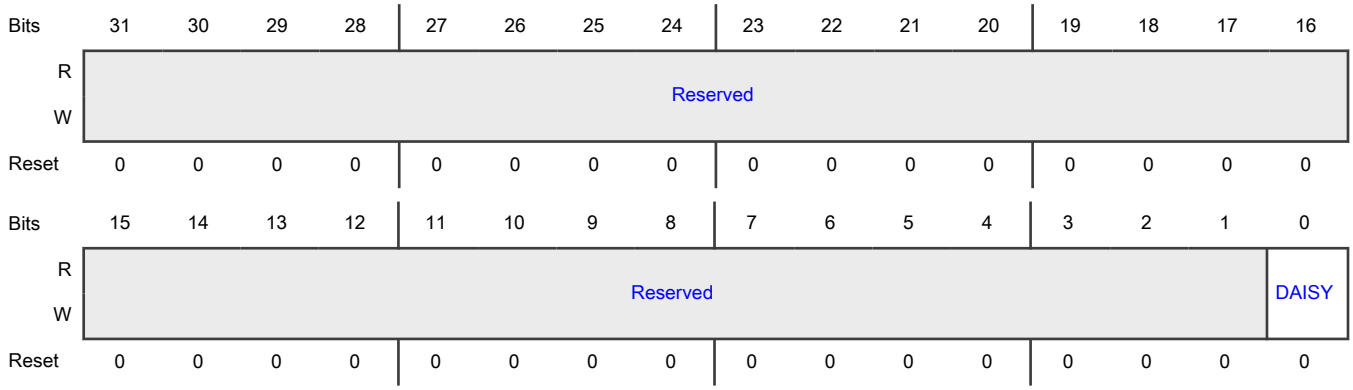
27.3.1.257 LPI2C8_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C8_IPP_IND_LPI2C_SCL_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPI2C8_IPP_IND_LPI2C_SCL_SELECT_INPUT	400h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c8, In Pin: ipp_ind_lpi2c_scl 0 - Selecting Pad: GPIO_IO11 for Mode: ALT6 1 - Selecting Pad: GPIO_IO13 for Mode: ALT6

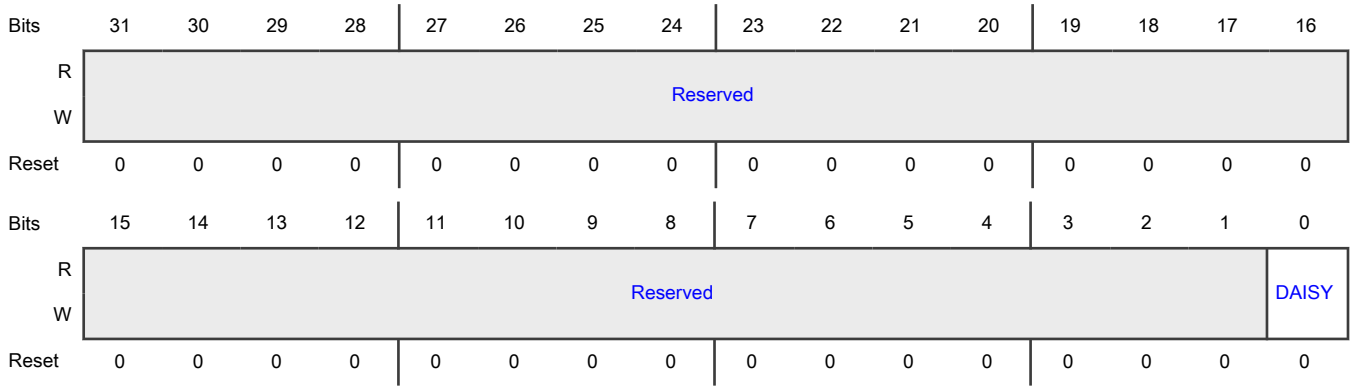
27.3.1.258 LPI2C8_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C8_IPP_IND_LPI2C_SDA_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPI2C8_IPP_IND_LPI2C_SDA_SELECT_INPUT	404h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c8, In Pin: ipp_ind_lpi2c_sda 0 - Selecting Pad: GPIO_IO10 for Mode: ALT6 1 - Selecting Pad: GPIO_IO12 for Mode: ALT6

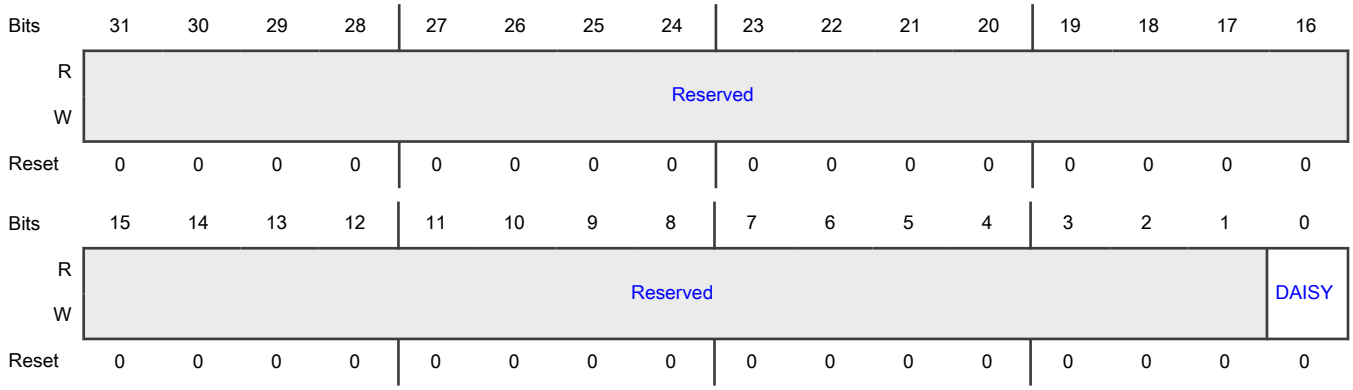
27.3.1.259 LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_0 DAISY Register (LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_0)

DAISY Register

Offset

Register	Offset
LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_0	408h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lptmr2, In Pin: ipp_ind_lptimer[0] 0 - Selecting Pad: ENET1_RD1 for Mode: ALT3 1 - Selecting Pad: SD2_DATA3 for Mode: ALT1

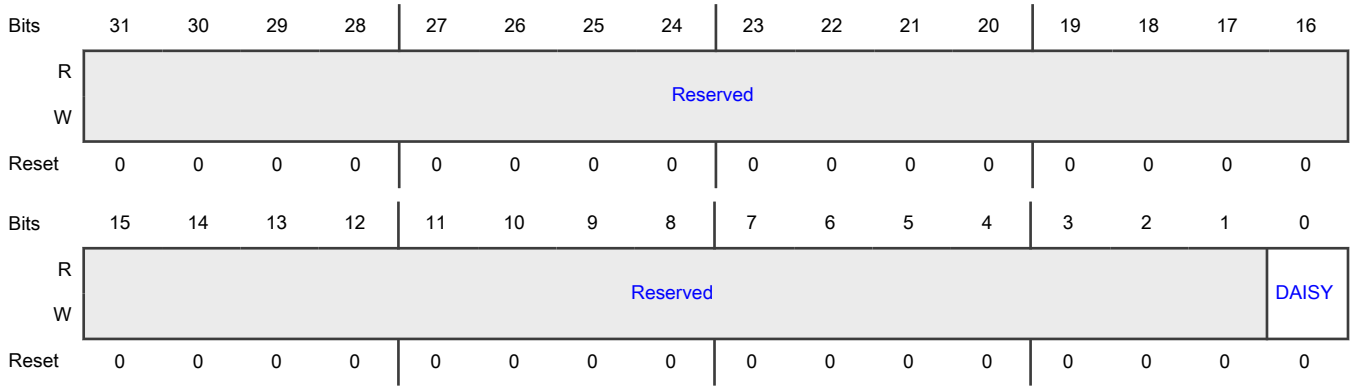
27.3.1.260 LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_1 DAISY Register (LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_1)

DAISY Register

Offset

Register	Offset
LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_1	40Ch

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lptmr2, In Pin: ipp_ind_lptimer[1] 0 - Selecting Pad: ENET1_RD2 for Mode: ALT3 1 - Selecting Pad: SD2_RESET_B for Mode: ALT1

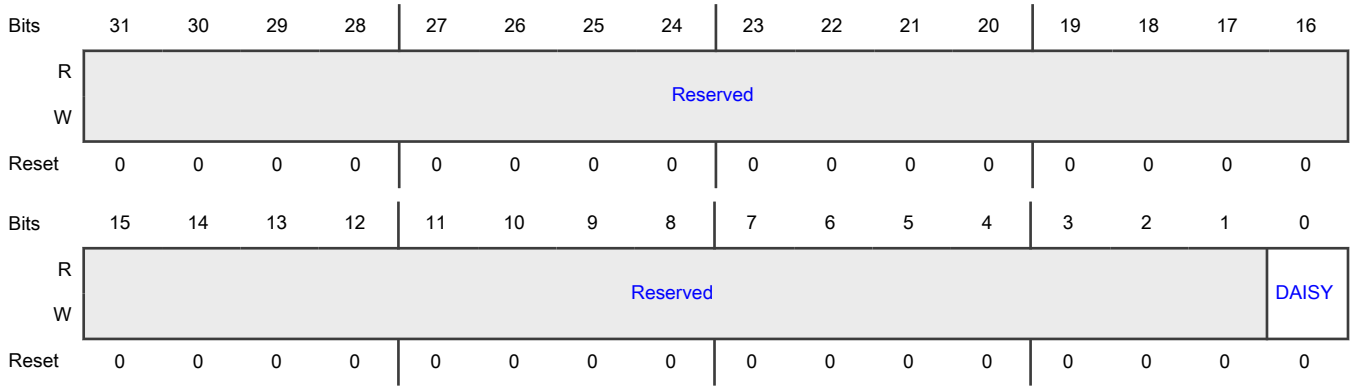
27.3.1.261 LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_2 DAISY Register (LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_2)

DAISY Register

Offset

Register	Offset
LPTMR2_IPP_IND_LPTIMER_SELECT_INPUT_2	410h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lptmr2, In Pin: ipp_ind_lptimer[2] 0 - Selecting Pad: ENET1_RD3 for Mode: ALT3 1 - Selecting Pad: SD2_VSELECT for Mode: ALT2

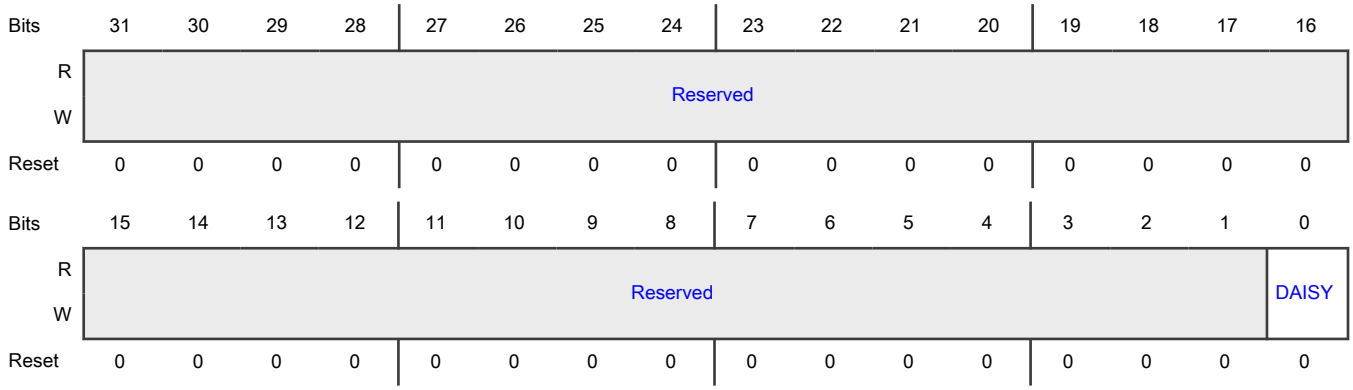
27.3.1.262 LPUART3_IPP_IND_LPUART_CTS_N_SELECT_INPUT DAISY Register (LPUART3_IPP_IND_LPUART_CTS_N_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPUART3_IPP_IND_LPUART_CTS_N_SELECT_INPUT	414h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart3, In Pin: ipp_ind_lpuart_cts_n 0 - Selecting Pad: GPIO_IO16 for Mode: ALT4 1 - Selecting Pad: ENET1_RD1 for Mode: ALT1

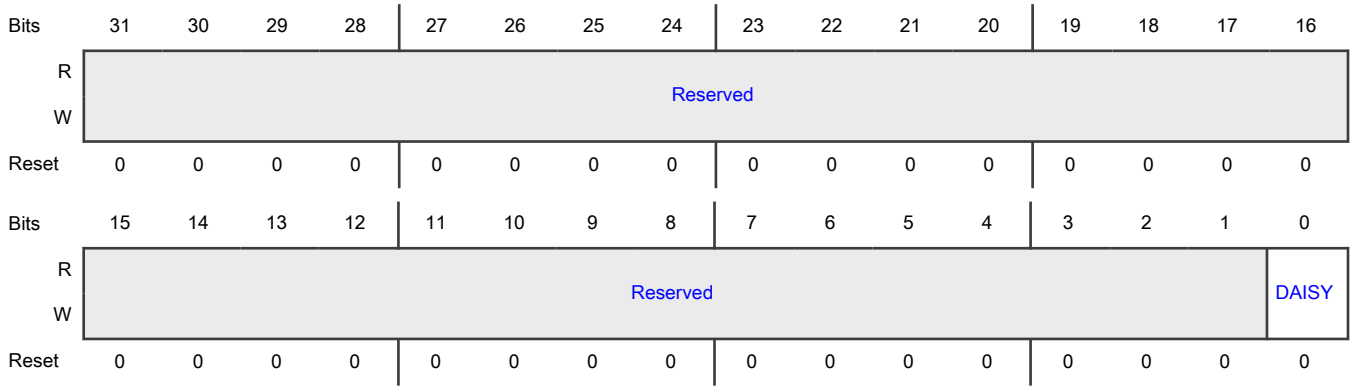
27.3.1.263 LPUART3_IPP_IND_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART3_IPP_IND_LPUART_RXD_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPUART3_IPP_IND_LPUART_RXD_SELECT_INPUT	418h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart3, In Pin: ipp_ind_lpuart_rxd 0 - Selecting Pad: GPIO_IO15 for Mode: ALT1 1 - Selecting Pad: ENET1_RD0 for Mode: ALT1

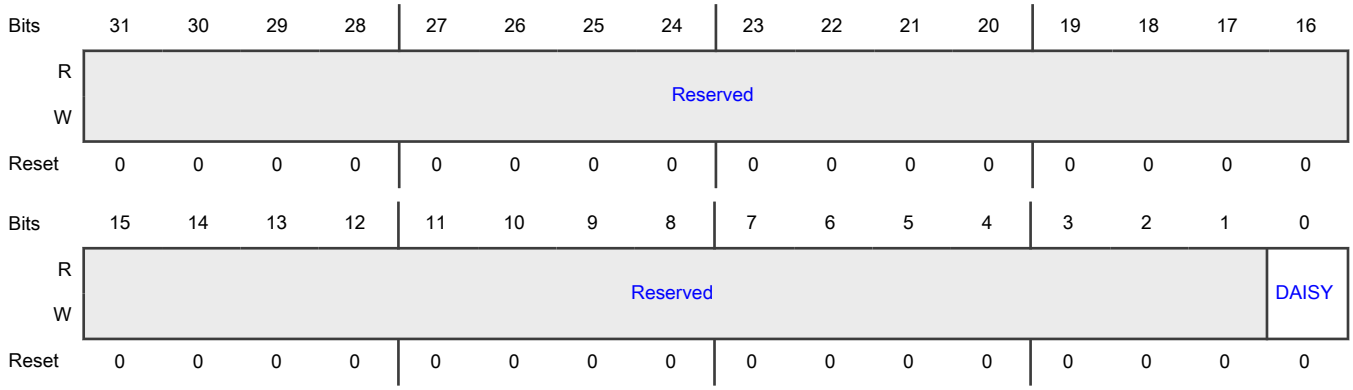
27.3.1.264 LPUART3_IPP_IND_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART3_IPP_IND_LPUART_TXD_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPUART3_IPP_IND_LPUART_TXD_SELECT_INPUT	41Ch

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart3, In Pin: ipp_ind_lpuart_txd 0 - Selecting Pad: GPIO_IO14 for Mode: ALT1 1 - Selecting Pad: ENET1_TD0 for Mode: ALT1

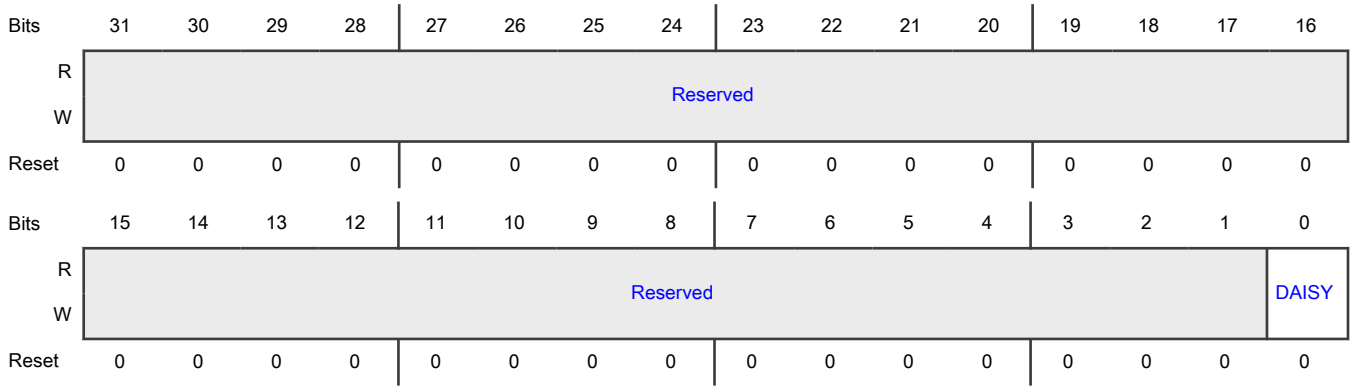
27.3.1.265 LPUART4_IPP_IND_LPUART_CTS_N_SELECT_INPUT DAISY Register (LPUART4_IPP_IND_LPUART_CTS_N_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPUART4_IPP_IND_LPUART_CTS_N_SELECT_INPUT	420h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart4, In Pin: ipp_ind_lpuart_cts_n 0 - Selecting Pad: GPIO_IO16 for Mode: ALT6 1 - Selecting Pad: ENET2_RD2 for Mode: ALT1

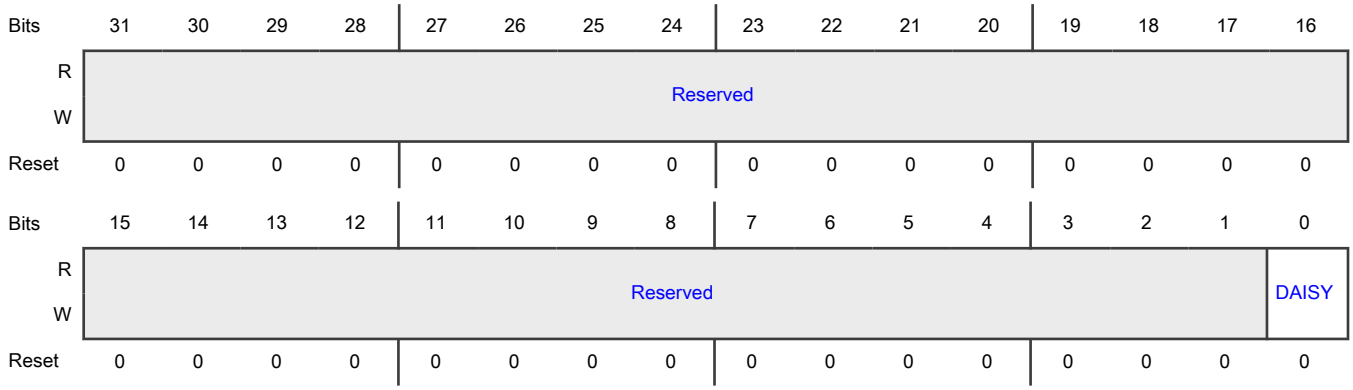
27.3.1.266 LPUART4_IPP_IND_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART4_IPP_IND_LPUART_RXD_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPUART4_IPP_IND_LPUART_RXD_SELECT_INPUT	424h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart4, In Pin: ipp_ind_lpuart_rxd 0 - Selecting Pad: GPIO_IO15 for Mode: ALT6 1 - Selecting Pad: ENET2_RD0 for Mode: ALT1

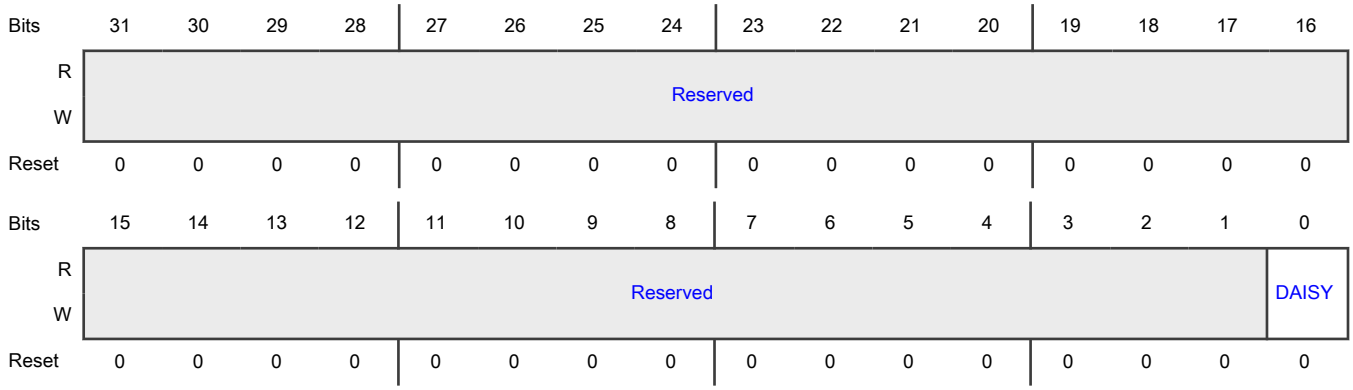
27.3.1.267 LPUART4_IPP_IND_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART4_IPP_IND_LPUART_TXD_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPUART4_IPP_IND_LPUART_TXD_SELECT_INPUT	428h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart4, In Pin: ipp_ind_lpuart_txd 0 - Selecting Pad: GPIO_IO14 for Mode: ALT6 1 - Selecting Pad: ENET2_TD0 for Mode: ALT1

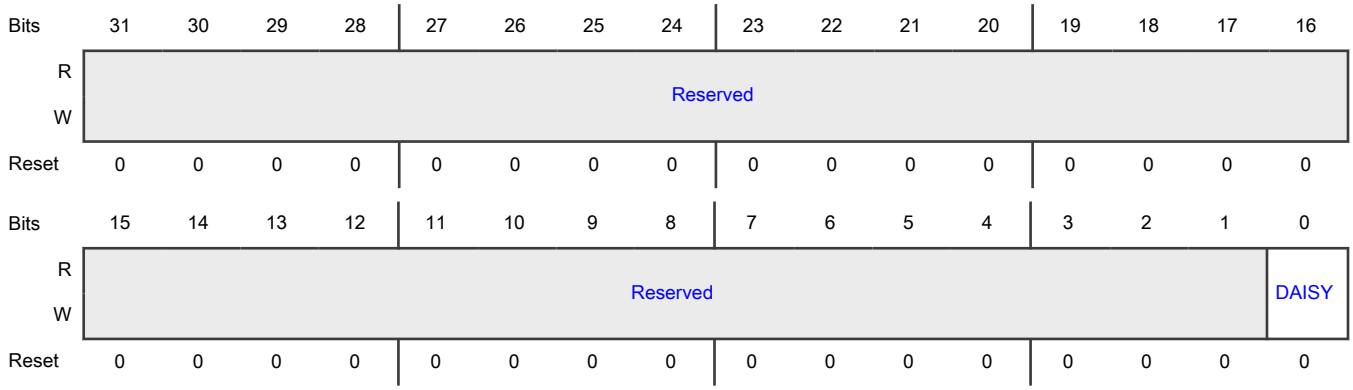
27.3.1.268 LPUART5_IPP_IND_LPUART_CTS_N_SELECT_INPUT DAISY Register (LPUART5_IPP_IND_LPUART_CTS_N_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPUART5_IPP_IND_LPUART_CTS_N_SELECT_INPUT	42Ch

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart5, In Pin: ipp_ind_lpuart_cts_n 0 - Selecting Pad: DAP_TCLK_SWCLK for Mode: ALT6 1 - Selecting Pad: GPIO_IO02 for Mode: ALT5

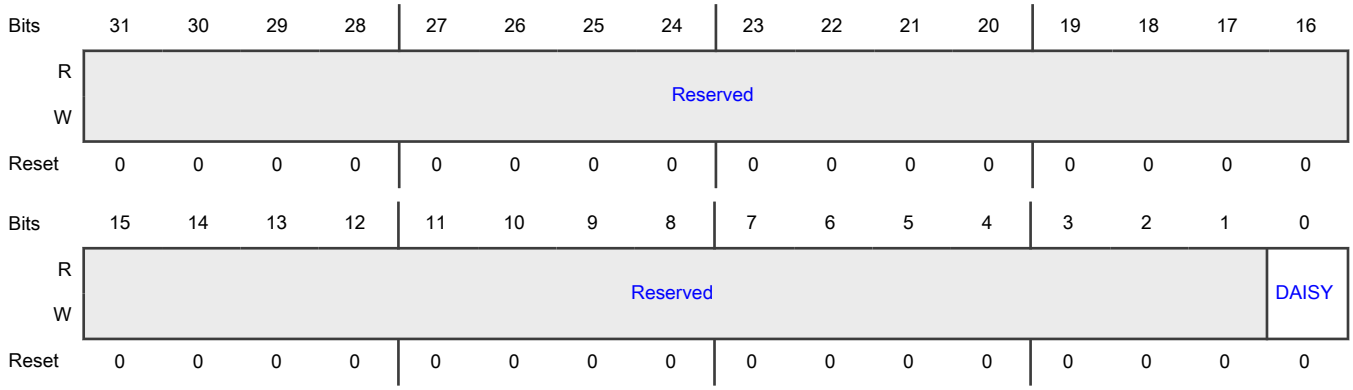
27.3.1.269 LPUART5_IPP_IND_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART5_IPP_IND_LPUART_RXD_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPUART5_IPP_IND_LPUART_RXD_SELECT_INPUT	430h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart5, In Pin: ipp_ind_lpuart_rxd 0 - Selecting Pad: DAP_TDI for Mode: ALT6 1 - Selecting Pad: GPIO_IO01 for Mode: ALT5

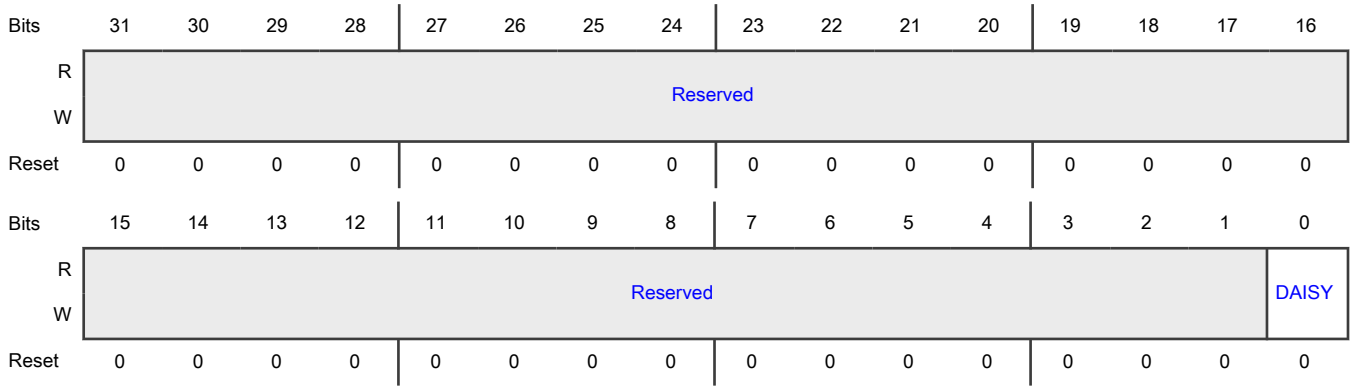
27.3.1.270 LPUART5_IPP_IND_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART5_IPP_IND_LPUART_TXD_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
LPUART5_IPP_IND_LPUART_TXD_SELECT_INPUT	434h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart5, In Pin: ipp_ind_lpuart_txd 0 - Selecting Pad: DAP_TDO_TRACESWO for Mode: ALT6 1 - Selecting Pad: GPIO_IO00 for Mode: ALT5

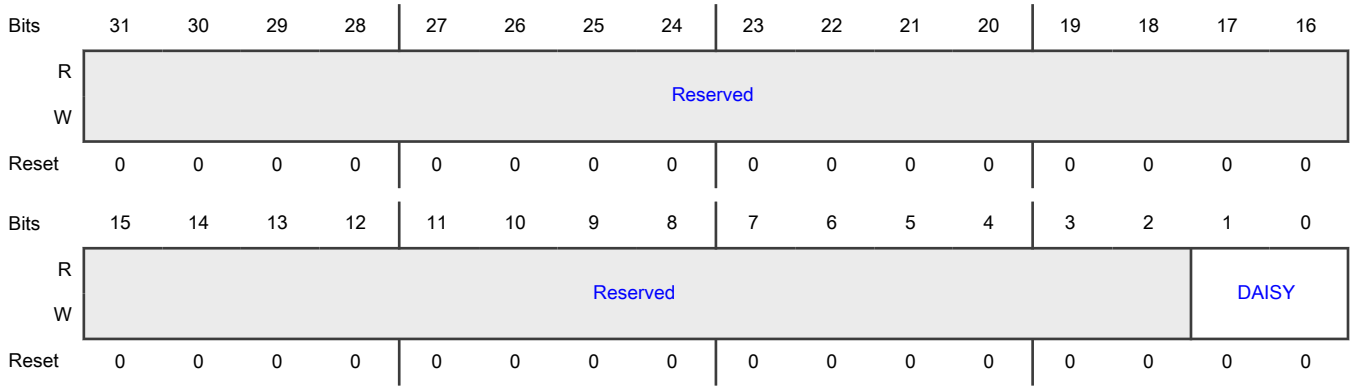
27.3.1.271 PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_0 DAISY Register (PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_0)

DAISY Register

Offset

Register	Offset
PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_0	438h

Diagram



Fields

Field	Description
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: pdm, In Pin: ipp_ind_mic_pdm_bitstream[0] 00 - Selecting Pad: GPIO_IO05 for Mode: ALT2 01 - Selecting Pad: GPIO_IO20 for Mode: ALT2 10 - Selecting Pad: PDM_BIT_STREAM0 for Mode: ALT0

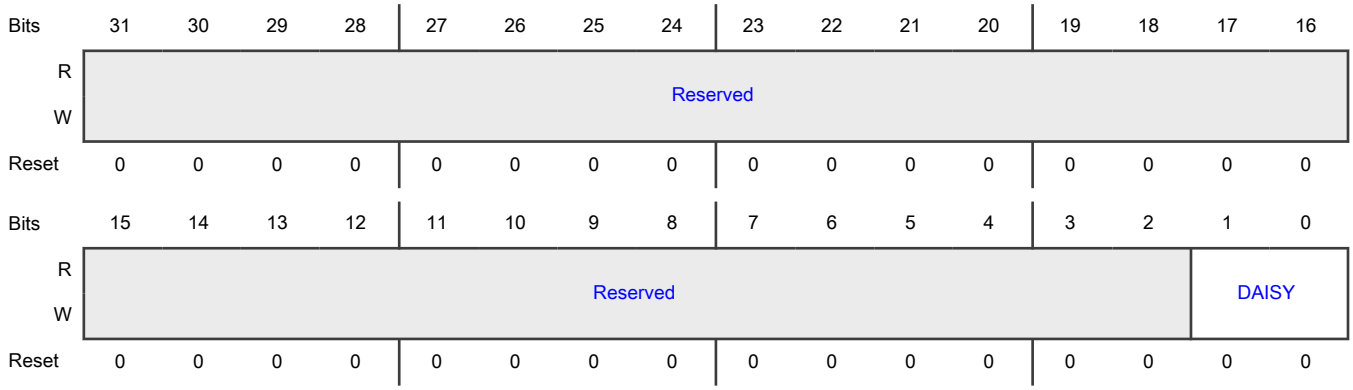
27.3.1.272 PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_1 DAISY Register (PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_1)

DAISY Register

Offset

Register	Offset
PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_1	43Ch

Diagram



Fields

Field	Description
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: pdm, In Pin: ipp_ind_mic_pdm_bitstream[1] 00 - Selecting Pad: GPIO_IO06 for Mode: ALT2 01 - Selecting Pad: GPIO_IO26 for Mode: ALT2 10 - Selecting Pad: PDM_BIT_STREAM1 for Mode: ALT0

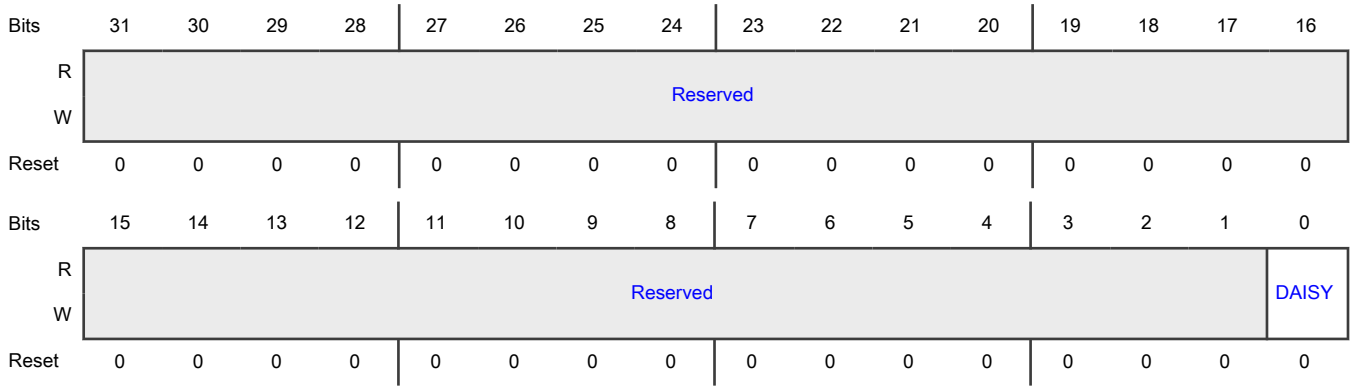
27.3.1.273 PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_2 DAISY Register (PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_2)

DAISY Register

Offset

Register	Offset
PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_2	440h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: pdm, In Pin: ipp_ind_mic_pdm_bitstream[2] 0 - Selecting Pad: GPIO_IO12 for Mode: ALT2 1 - Selecting Pad: GPIO_IO16 for Mode: ALT2

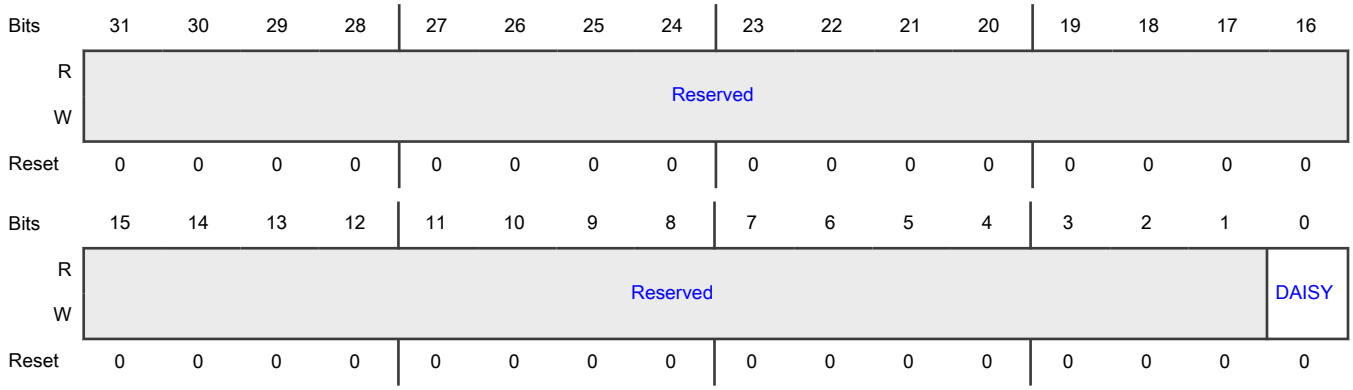
27.3.1.274 PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_3 DAISY Register (PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_3)

DAISY Register

Offset

Register	Offset
PDM_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_3	444h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: pdm, In Pin: ipp_ind_mic_pdm_bitstream[3] 0 - Selecting Pad: GPIO_IO13 for Mode: ALT2 1 - Selecting Pad: GPIO_IO19 for Mode: ALT2

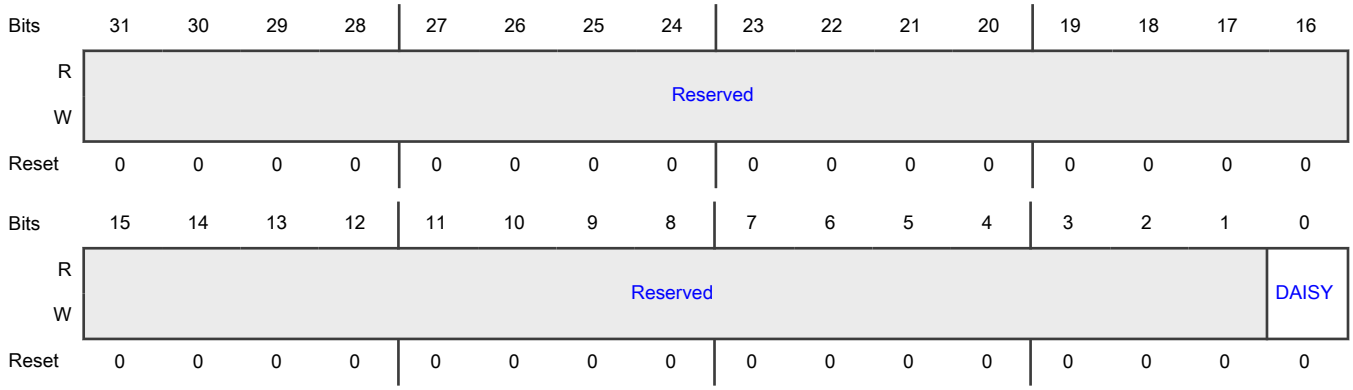
27.3.1.275 SAI1_IPP_IND_SAI_MCLK_SELECT_INPUT DAISY Register (SAI1_IPP_IND_SAI_MCLK_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
SAI1_IPP_IND_SAI_MCLK_SELECT_INPUT	448h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai1, In Pin: ipp_ind_sai_mclk 0 - Selecting Pad: UART2_RXD for Mode: ALT4 1 - Selecting Pad: SAI1_RXD0 for Mode: ALT1

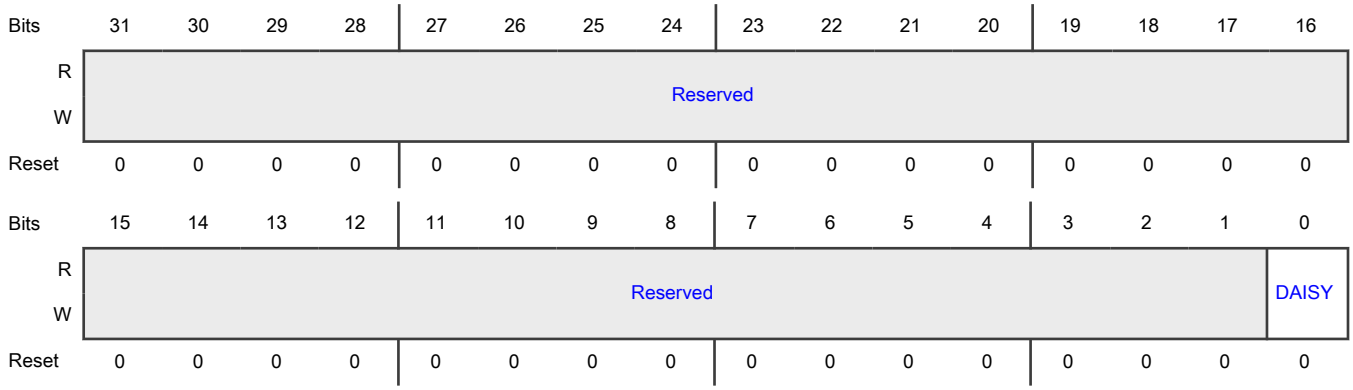
27.3.1.276 SAI3_IPP_IND_SAI_RXBCLK_SELECT_INPUT DAISY Register (SAI3_IPP_IND_SAI_RXBCLK_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
SAI3_IPP_IND_SAI_RXBCLK_SELECT_INPUT	44Ch

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai3, In Pin: ipp_ind_sai_rxbclk 0 - Selecting Pad: GPIO_IO18 for Mode: ALT1 1 - Selecting Pad: GPIO_IO21 for Mode: ALT7

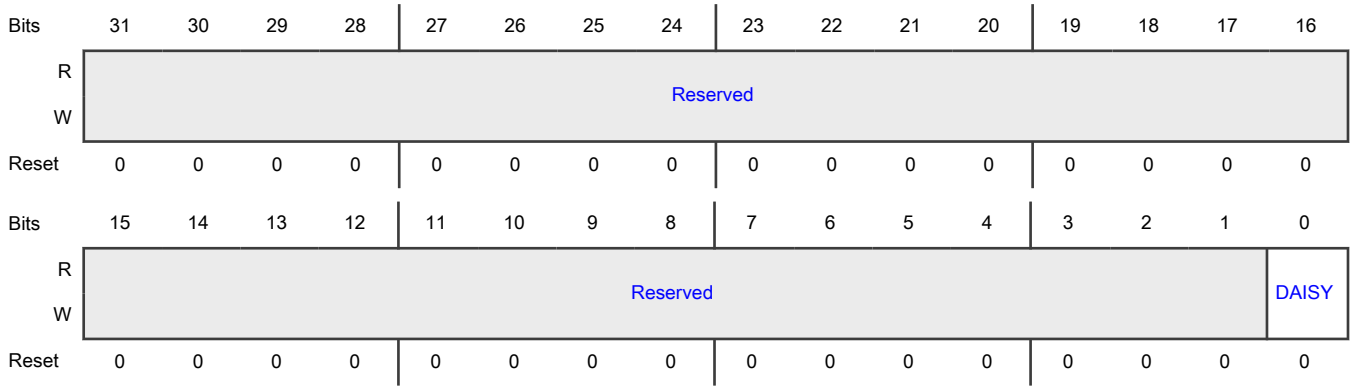
27.3.1.277 SAI3_IPP_IND_SAI_RXSYNC_SELECT_INPUT DAISY Register (SAI3_IPP_IND_SAI_RXSYNC_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
SAI3_IPP_IND_SAI_RXSYNC_SELECT_INPUT	450h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai3, In Pin: ipp_ind_sai_rxsync 0 - Selecting Pad: GPIO_IO12 for Mode: ALT7 1 - Selecting Pad: GPIO_IO19 for Mode: ALT1

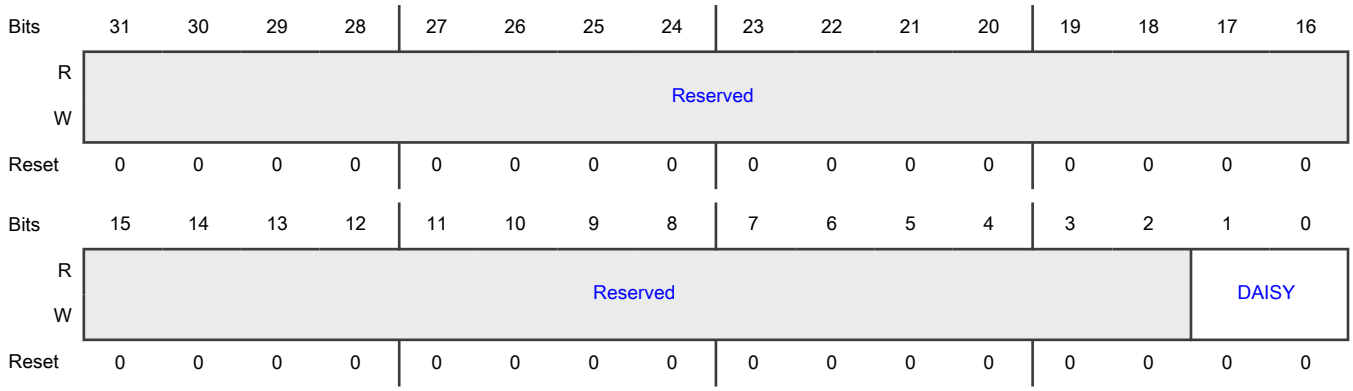
27.3.1.278 SPDIF_SPDIF_I_SELECT_INPUT DAISY Register (SPDIF_SPDIF_I_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
SPDIF_SPDIF_I_SELECT_INPUT	454h

Diagram



Fields

Field	Description
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: spdif, In Pin: spdif_i 00 - Selecting Pad: GPIO_IO22 for Mode: ALT2 01 - Selecting Pad: ENET2_RD1 for Mode: ALT1 10 - Selecting Pad: ENET2_RD3 for Mode: ALT2

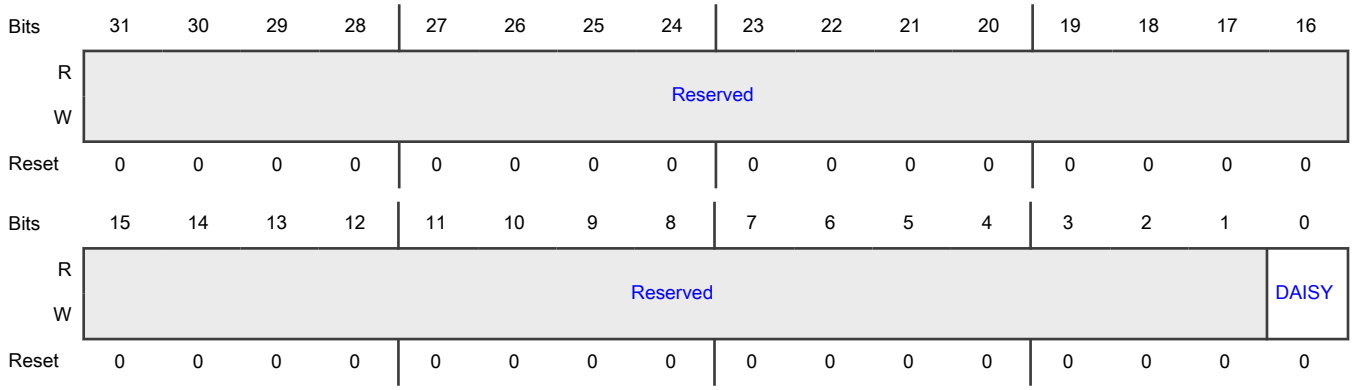
27.3.1.279 USDHC3_IPP_CARD_CLK_IN_SELECT_INPUT DAISY Register (USDHC3_IPP_CARD_CLK_IN_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
USDHC3_IPP_CARD_CLK_IN_SELECT_INPUT	458h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0	Selecting Pads Involved in Daisy Chain.

Table continues on the next page...

Table continued from the previous page...

Field	Description
DAISY	Instance: usdhc3, In Pin: ipp_card_clk_in 0 - Selecting Pad: GPIO_IO22 for Mode: ALT1 1 - Selecting Pad: SD3_CLK for Mode: ALT0

27.3.1.280 USDHC3_IPP_CMD_IN_SELECT_INPUT DAISY Register (USDHC3_IPP_CMD_IN_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
USDHC3_IPP_CMD_IN_SELECT_INPUT	45Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: ipp_cmd_in 0 - Selecting Pad: GPIO_IO23 for Mode: ALT1 1 - Selecting Pad: SD3_CMD for Mode: ALT0

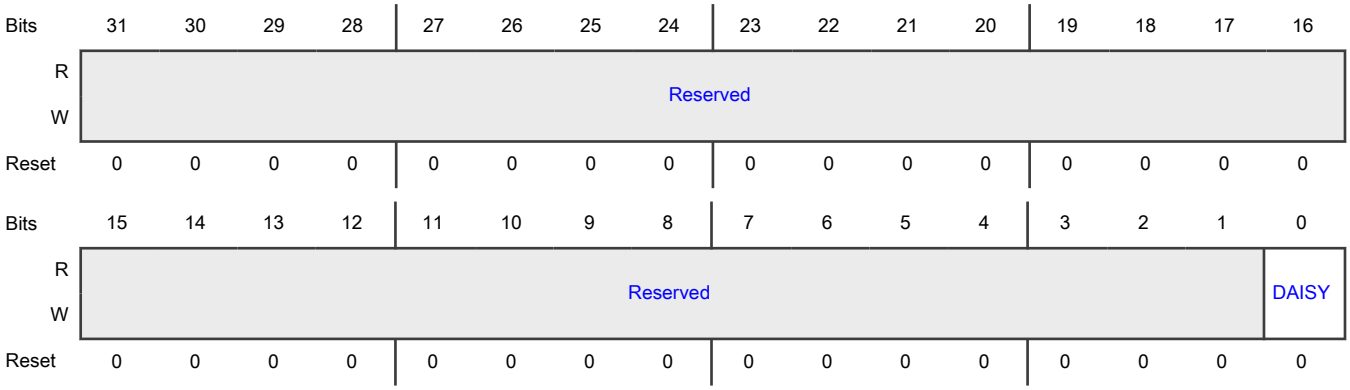
27.3.1.281 USDHC3_IPP_DAT0_IN_SELECT_INPUT DAISY Register (USDHC3_IPP_DAT0_IN_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
USDHC3_IPP_DAT0_IN_SELECT_INPUT	460h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: ipp_dat0_in 0 - Selecting Pad: GPIO_IO24 for Mode: ALT1 1 - Selecting Pad: SD3_DATA0 for Mode: ALT0

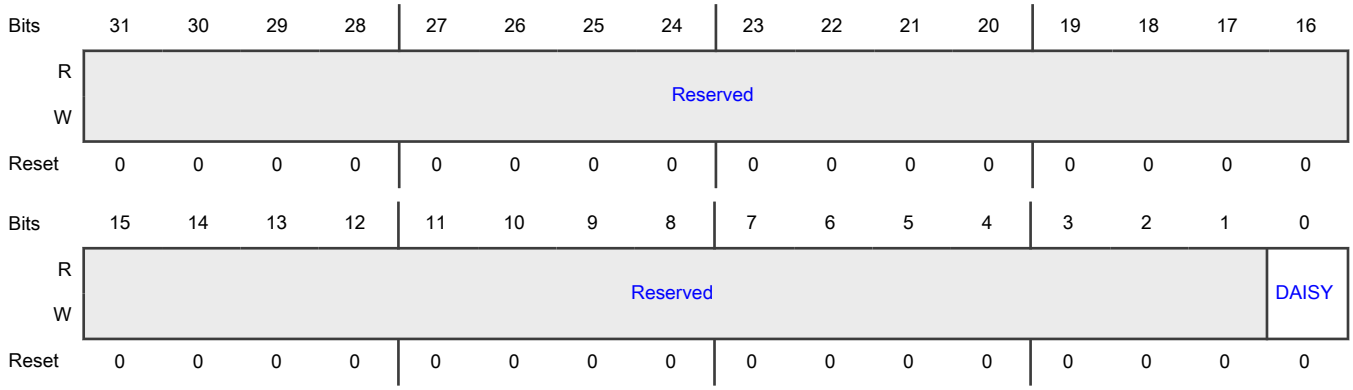
27.3.1.282 USDHC3_IPP_DAT1_IN_SELECT_INPUT DAISY Register (USDHC3_IPP_DAT1_IN_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
USDHC3_IPP_DAT1_IN_SELECT_INPUT	464h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: ipp_dat1_in 0 - Selecting Pad: GPIO_IO25 for Mode: ALT1 1 - Selecting Pad: SD3_DATA1 for Mode: ALT0

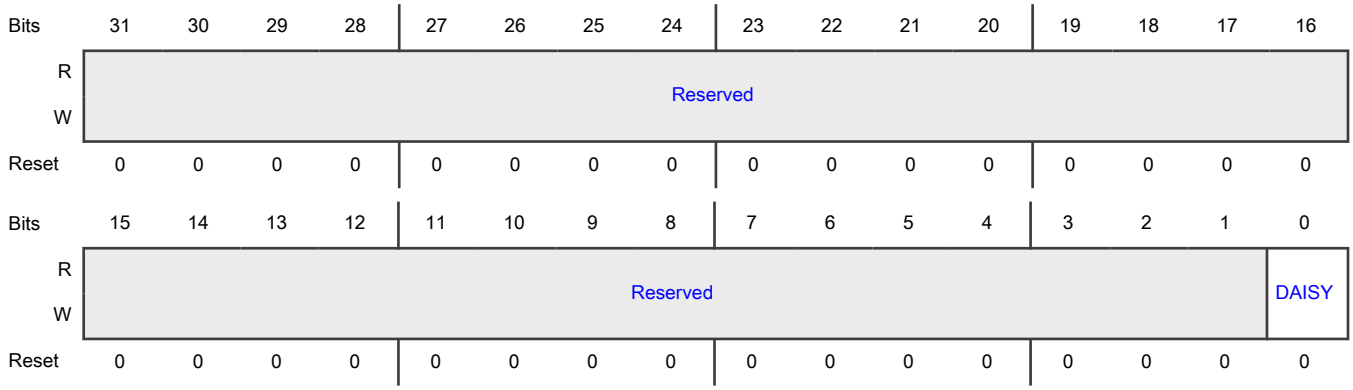
27.3.1.283 USDHC3_IPP_DAT2_IN_SELECT_INPUT DAISY Register (USDHC3_IPP_DAT2_IN_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
USDHC3_IPP_DAT2_IN_SELECT_INPUT	468h

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: ipp_dat2_in 0 - Selecting Pad: GPIO_IO26 for Mode: ALT1 1 - Selecting Pad: SD3_DATA2 for Mode: ALT0

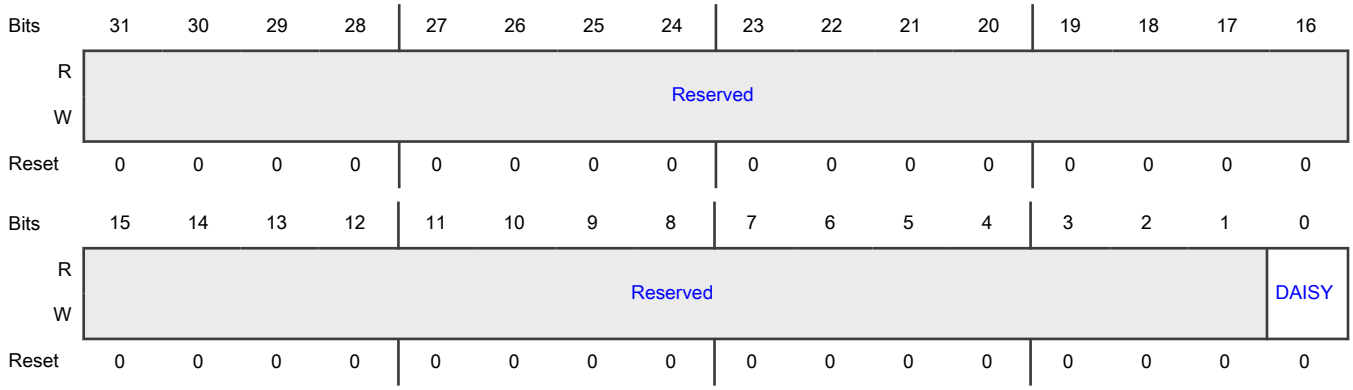
27.3.1.284 USDHC3_IPP_DAT3_IN_SELECT_INPUT DAISY Register (USDHC3_IPP_DAT3_IN_SELECT_INPUT)

DAISY Register

Offset

Register	Offset
USDHC3_IPP_DAT3_IN_SELECT_INPUT	46Ch

Diagram



Fields

Field	Description
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: ipp_dat3_in 0 - Selecting Pad: GPIO_IO27 for Mode: ALT1 1 - Selecting Pad: SD3_DATA3 for Mode: ALT0

Chapter 28
General-Purpose Input/Output (GPIO)

28.1 Chip-specific GPIO information

Table 133. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The chip supports four instances of GPIO. GPIO1 is in AONMIX and GPIO2, GPIO3, GPIO4 in WAKEUPMIX. For more details, please refer the Pin Assignments spreadsheet attached with this document.

GPIO1_IO[5], GPIO1_IO[7], GPIO1_IO[11] and GPIO1_IO[13] are output only or tristated.

For the GPIO1,GPIO2 and GPIO4, the following bits are Reserved in the PCNS, PCNP, PCNP, PDOR, PCOR, PTOR, PDIR, PDDR, PIDR, ISFRn registers.

- Bit[31:17] should be Reserved for GPIO1
- Bit[31:30] should be Reserved for GPIO2
- Bit[31:30] should be Reserved for GPIO4

28.2 Overview

GPIO communicates to the processor core via a zero wait-state interface for maximum pin performance.

28.2.1 Block diagram

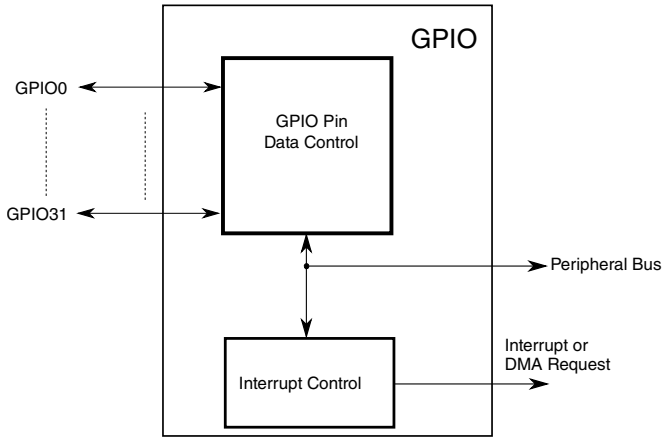


Figure 59. Block diagram

28.2.2 Features

- Port Data Input (PDIR) displays the logic value on each pin when the pin is configured for any digital function, provided the corresponding Port Control and Interrupt modules for that pin are enabled.
- Port Data Output (PDOR), with corresponding set, clear, and toggle registers, controls the output data of each pin when the pin is configured for the GPIO function.
- Port Data Direction (PDDR) controls the direction of each pin when the pin is configured for the GPIO function.
- Port Input Disable (PIDR) controls disabling of the input for each general-purpose pin.
- Pin interrupts:
  - Interrupt flags and enable registers for each pin are functional in all digital pin muxing modes.
  - Support for interrupt or DMA request is configured for each pin.
  - Support for edge-sensitive (rising, falling, or both) or level-sensitive (low, high) interrupts is configured for each pin.
  - Asynchronous wake-up in Low-Power mode.
  - GPIO generates a total of 2 interrupts, and 2 DMA requests.
  - Each pin can be used to generate a single interrupt, or DMA request.
- Protection registers:
  - Each pin is configured for secure or nonsecure and privileged or nonprivileged access.
  - Each interrupt, and DMA request domain is configured for secure or nonsecure and privileged or nonprivileged access.

28.3 Functional description

28.3.1 Low-Power mode

You can configure GPIO to exit Low-Power mode via an asynchronous wake-up signal if an enabled interrupt is detected.

28.3.2 Debug mode

GPIO remains functional in Debug mode.

28.3.3 General-purpose input

The logic state of each pin is available via Port Data Input (PDIR) if:

- The corresponding field in Port Input Disable (PIDR) is 0.
- The pin is configured for a digital function.

28.3.4 General-purpose output

The logic state of each pin is controlled via Port Data Output (PDOR) and Port Data Direction (PDDR), provided the pin is configured for the GPIO function. The following table depicts the conditions for a pin to be configured as input or output.

Table 134. General-purpose output

If	Then
A pin is configured for the GPIO function and the corresponding PDDR field is 0	The pin is configured as an input.
A pin is configured for the GPIO function and the corresponding PDDR field is 1	The pin is configured as an output and the logic state of the pin is equal to the corresponding PDOR field.

For efficient bit manipulation on the general-purpose outputs, pin data set, pin data clear, and pin data toggle registers allow one or more outputs within one port to be set, cleared, or toggled from a single register write, eliminating the need for a read-modify-write operation to prevent changing pins accidentally.

28.3.5 Clocking

GPIO receives a single clock, which is used for register access and synchronization with external pin inputs. There are no special considerations.

28.3.6 Reset

GPIO receives a single reset, which resets the peripheral. There are no special considerations.

28.3.7 External interrupts

The external interrupt capability of GPIO is available in all digital pin muxing modes.

GPIO generates a total of 2 interrupts, and 2 DMA requests. You can configure each pin individually for interrupt, or DMA requests:

- Each output implements a separate interrupt status flag (ISF) register for that domain.
- Each output generates a single interrupt.
- Each output generates a single DMA request.

You can configure each pin individually for any of the external interrupt modes shown in the following table.

Table 135. Available pin configurations for external interrupts

Signal conditions	Software polling using flags	Interrupts	DMA requests
Rising-edge	Yes	Yes	Yes
Falling-edge	Yes	Yes	Yes
Rising- and falling-edge	Yes	Yes	Yes
High-level	—	Yes	—
Low-level	—	Yes	—

The interrupt status flag is set when the configured edge or level is detected on the pin. Unless GPIO is in Low-Power mode, the input is first synchronized to the system clock to detect the configured level or edge transition.

GPIO generates a pin interrupt that asserts when the interrupt status flag is set for any enabled interrupt for that output. The interrupt negates after the interrupt status flags for all enabled interrupts are cleared by writing a logic 1 to either ISFR0[ISF*n*] or ICR0[ISF].

GPIO generates a DMA request that asserts when the interrupt status flag is set for any enabled DMA request in that output. The DMA request negates after the DMA transfer is completed because that clears the interrupt status flags for all enabled DMA requests.

In Low-Power mode, the interrupt status flag for any enabled interrupt is asynchronously set if the required level or edge is detected. This also generates an asynchronous wake-up signal to exit Low-Power mode.

28.3.8 Global interrupt control

The two global interrupt control registers (Global Interrupt Control Low (GICLR) and Global Interrupt Control High (GICHR)) allow a single register write to update the upper 16 bits of Interrupt Control a (ICR0 - ICR31) for up to 16 pins, all with the same value.

These global interrupt control registers allow you to quickly configure multiple pins within the same port with the same interrupt configuration.

The global interrupt control registers are write-only registers and always read 0.

28.3.9 DMA

GPIO generates 2 requests. For each pin, you can configure DMA requests to assert for either rising, falling, or both edge detection. See External interrupts for more information.

28.3.10 Access protection

GPIO implements access protection registers (PCNS and PCNP) for each pin, as shown in Table 136.

Table 136. PCNS and PCNP register access protection implementation

Access	Description
Secure	You can write to or read the pins configured for secure access only in the Secure state.
Nonsecure	You can write to or read the pins configured for nonsecure access only in the Nonsecure state.
Privilege	You can write to the pins configured for privilege access only in the Privilege state.
Nonprivilege	You can write to the pins configured for nonprivilege access in both the Privilege and Nonprivilege states.

GPIO implements access protection registers (ICNS and ICNP) for each interrupt, and DMA request as shown in Table 137.

Table 137. ICNS and ICNP register access protection implementation

Access	Description
Secure	You can configure the outputs programmed for secure access only in the Secure state.
Nonsecure	You can configure the outputs programmed for nonsecure access only in the Nonsecure state.
Privilege	You can configure the outputs programmed for privilege access only in the Privilege state.
Nonprivilege	You can configure the outputs programmed for nonprivilege access in both the Privilege and Nonprivilege states.

You can write to the access protection registers (PCNS, PCNP, ICNS, and ICNP) only in the Secure-Privilege state; you can optionally lock these registers until the next reset.

Configuring a pin interrupt, or DMA request requires the following access permissions:

- Access permission to configure the pin.
- Access permission to either configure to the desired interrupt, or DMA request, or writing 0 to ICR*n*[IRQC].
- Access permission to either configure with the existing interrupt,, or DMA request, or if the current value of ICR*n*[IRQC] is 0.
- The interrupt configuration is not locked.

28.4 External signals

Table 138. External signals

Signal	Description		Direction
GPIO31–GPIO0	General-purpose input/output		I/O
	State meaning	Asserted: The pin is logic 1. Deasserted: The pin is logic 0.	

Table continues on the next page...

Table 138. External signals (continued)

Signal	Description		Direction
	Timing	Assertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock. Deassertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock.	

NOTE

Not all pins within each GPIO are implemented on each chip. See the "Signal Multiplexing" chapter for the number of GPIO pins within each port available on this chip.

28.5 Initialization

To initialize GPIO:

1. Initialize the GPIO pins for the output function:
   a. Configure the output logic value for each pin by using Port Data Output (PDOR).
   b. Configure the direction for each pin by using Port Data Direction (PDDR).
2. Initialize the interrupt function by writing to Interrupt Control a (ICR0 - ICR31) for the corresponding pins and desired configuration. If the pin is previously used for a different function, first write 0100_0000h to Interrupt Control a (ICR0 - ICR31) to disable the previous function and clear the flag.

28.6 Application information

GPIO includes the following applications:

- Reading the state of a single pin by performing a byte read of Pin Data (P0DR - P31DR).
- Updating the state of a single pin by performing a byte write to Pin Data (P0DR - P31DR).
- Reading the state of multiple pins by reading Port Data Input (PDIR).
- Updating the state of multiple pins by using the following ways:
  - Writing to Port Data Output (PDOR).
  - Writing to Port Set Output (PSOR) to write 1 to Port Data Output (PDOR).
  - Writing to Port Clear Output (PCOR) to write 0 to Port Data Output (PDOR).
  - Writing to Port Toggle Output (PTOR) to toggle Port Data Output (PDOR).

28.7 Memory map and register definition

The GPIO registers support 8-bit, 16-bit, or 32-bit accesses. Any read or write access to the GPIO memory space that is outside the valid memory map results in a bus error.

NOTE

For simplicity, each GPIO port's register appears with the same width of 32 bits, corresponding to 32 pins. The actual number of pins per port (and therefore the number of usable control bits per port register) is chip-specific. See the "Signal Multiplexing" chapter for the exact control bits of each port.

28.7.1 GPIO register descriptions

28.7.1.1 GPIO memory map

gpio1 base address: 4740_0000h

gpio2 base address: 4381_0000h

gpio3 base address: 4382_0000h

gpio4 base address: 4383_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VERID)	32	R	0201_0001
4	Parameter (PARAM)	32	R	0000_0002
C	Lock (LOCK)	32	RW	0000_0000
10	Pin Control Nonsecure (PCNS)	32	RW	0000_0000
14	Interrupt Control Nonsecure (ICNS)	32	RW	0000_0000
18	Pin Control Nonprivilege (PCNP)	32	RW	0000_0000
1C	Interrupt Control Nonprivilege (ICNP)	32	RW	0000_0000
40	Port Data Output (PDOR)	32	RW	0000_0000
44	Port Set Output (PSOR)	32	RW	0000_0000
48	Port Clear Output (PCOR)	32	RW	0000_0000
4C	Port Toggle Output (PTOR)	32	RW	0000_0000
50	Port Data Input (PDIR)	32	R	0000_0000
54	Port Data Direction (PDDR)	32	RW	0000_0000
58	Port Input Disable (PIDR)	32	RW	0000_0000
60 - 7F	Pin Data (P0DR - P31DR)	8	RW	00
80 - FC	Interrupt Control a (ICR0 - ICR31)	32	RW	0000_0000
100	Global Interrupt Control Low (GICLR)	32	RW	0000_0000
104	Global Interrupt Control High (GICHR)	32	RW	0000_0000
120 - 124	Interrupt Status Flag (ISFR0 - ISFR1)	32	RW	0000_0000

28.7.1.2 Version ID (VERID)

Indicates the version ID number of each GPIO.

Offset

Register	Offset
VERID	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31-24 MAJOR	Major Version Number Indicates the major version number for the specification.
23-16 MINOR	Minor Version Number Indicates the minor version number for the specification.
15-0 FEATURE	Feature Specification Number Indicates the feature set number. 0000_0000_0000_0000 - Basic implementation 0000_0000_0000_0001 - Protection registers implemented

28.7.1.3 Parameter (PARAM)

Indicates the interrupt number of each GPIO.

Offset

Register	Offset
PARAM	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												IRQNUM			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fields

Field	Description
31-4 —	Reserved
3-0 IRQNUM	Interrupt Number Indicates the number of interrupt, or DMA request domains.

28.7.1.4 Lock (LOCK)

Locks the nonsecure and nonprivilege access protection registers. You can write to this register only in the Secure-Privilege state.

Offset

Register	Offset
LOCK	Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												ICNP	PCNP	ICNS	PCNS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved
3 ICNP	Lock ICNP Locks ICNP. If this field is 0, you can write to ICNP in the Secure-Privilege state. If this field is 1, you cannot write to ICNP until the next reset. 0 - Writable in Secure-Privilege state 1 - Not writable until the next reset
2 PCNP	Lock PCNP Locks Pin Control Nonprivilege (PCNP). If this field is 0, you can write to PCNP in the Secure-Privilege state. If this field is 1, you cannot write to PCNP until the next reset. 0 - Writable in Secure-Privilege state 1 - Not writable until the next reset
1 ICNS	Lock ICNS Locks ICNS. If this field is 0, you can write to ICNS in the Secure-Privilege state. If this field is 1, you cannot write to ICNS until the next reset. 0 - Writable in Secure-Privilege state 1 - Not writable until the next reset
0 PCNS	Lock PCNS Locks Pin Control Nonsecure (PCNS). If this field is 0, you can write to PCNS in the Secure-Privilege state. If this field is 1, you cannot write to PCNS until the next reset. 0 - Writable in Secure-Privilege state 1 - Not writable until the next reset

28.7.1.5 Pin Control Nonsecure (PCNS)

Configures secure or nonsecure access protection for each pin. You can write to this register only in the Secure-Privilege state if it is not locked (LOCK[PCNS] = 0).

Offset

Register	Offset
PCNS	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NSE31	NSE30	NSE29	NSE28	NSE27	NSE26	NSE25	NSE24	NSE23	NSE22	NSE21	NSE20	NSE19	NSE18	NSE17	NSE16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NSE15	NSE14	NSE13	NSE12	NSE11	NSE10	NSE9	NSE8	NSE7	NSE6	NSE5	NSE4	NSE3	NSE2	NSE1	NSE0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 NSEn	Nonsecure Enable Configures secure or nonsecure access protection for each pin. If this field is 0, the pin is configured for secure access. You can read or write to the corresponding pin's registers and fields only in the Secure state. When you access the corresponding pin's registers in the Nonsecure state, all fields in the registers related to that pin are read zero (with writes ignored). If this field is 1, the pin is configured for nonsecure access. You can read or write to the corresponding pin's registers and fields only in the Nonsecure state. When you access the corresponding pin's registers in the Secure state, all fields in the registers related to that pin are read zero (with writes ignored). 0 - Secure access 1 - Nonsecure access

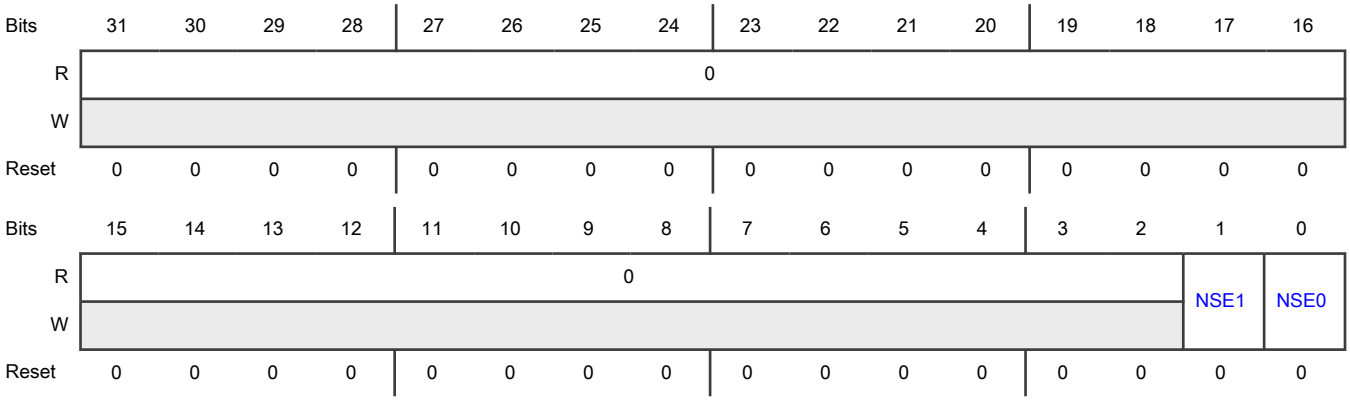
28.7.1.6 Interrupt Control Nonsecure (ICNS)

Configures secure and nonsecure access protection for each interrupt, or DMA request. You can update this register only in the Secure-Privilege state if it is not locked (LOCK[ICNS] = 0).

Offset

Register	Offset
ICNS	14h

Diagram



Fields

Field	Description
31-2 —	Reserved
1-0 NSEn	Nonsecure Enable Configures secure or nonsecure access protection for each interrupt, or DMA request. If this field is 0, the interrupt, or DMA request is configured for secure access. You can configure a pin to use the corresponding interrupt, or DMA request, or reconfigure a pin that is already configured to use the corresponding interrupt, or DMA request only in the Secure state. If this field is 1, the interrupt, or DMA request is configured for nonsecure access. You can configure a pin to use the corresponding interrupt, or DMA request, or reconfigure a pin that is already configured to use the corresponding interrupt, or DMA request only in the Nonsecure state. **NOTE** See the GPIO chip-specific information to determine which GPIO instances support interrupt, DMA request, or trigger capabilities. 0 - Secure access 1 - Nonsecure access

28.7.1.7 Pin Control Nonprivilege (PCNP)

Configures access protection on each pin either for privilege or for both privilege and nonprivilege accesses. You can update this register only in the Secure-Privilege state if it is not locked (LOCK[PCNP] = 0).

Offset

Register	Offset
PCNP	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NPE31	NPE30	NPE29	NPE28	NPE27	NPE26	NPE25	NPE24	NPE23	NPE22	NPE21	NPE20	NPE19	NPE18	NPE17	NPE16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NPE15	NPE14	NPE13	NPE12	NPE11	NPE10	NPE9	NPE8	NPE7	NPE6	NPE5	NPE4	NPE3	NPE2	NPE1	NPE0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 NPEn	Nonprivilege Enable Configures privilege or nonprivilege access protection for each pin. If this field is 0, the pin is configured for privilege access. Write access to the corresponding pin's registers and fields is allowed only in the Privilege state. When you access the corresponding pin's registers and fields in the Nonprivilege state, all fields related to that pin in this GPIO are readable (with writes ignored). If this field is 1, the pin is configured for nonprivilege access; read or write access to the corresponding pin's registers is allowed in both Privilege and Nonprivilege states. 0 - Privilege access 1 - Nonprivilege access

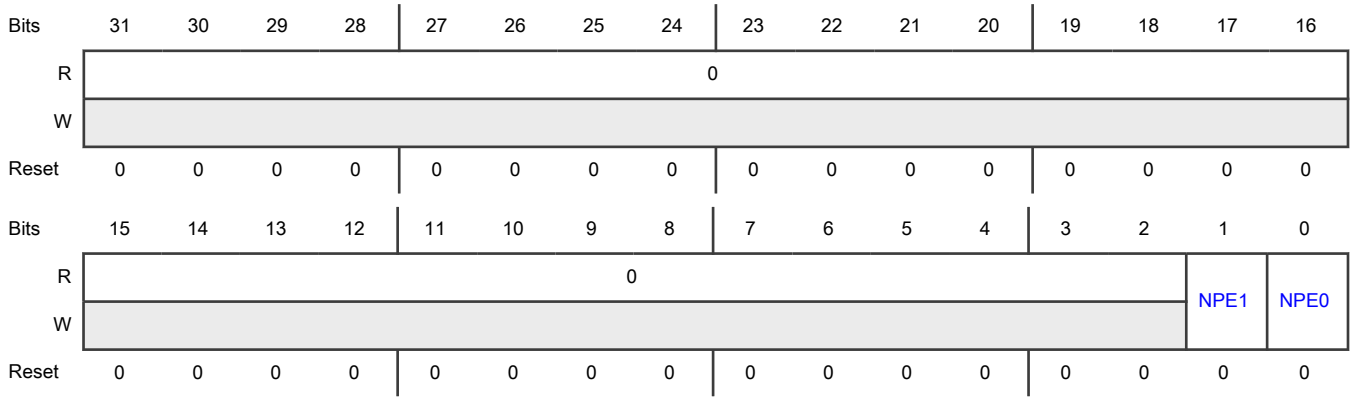
28.7.1.8 Interrupt Control Nonprivilege (ICNP)

Configures privilege and nonprivilege access protection for each interrupt, or DMA request. You can update this register only in the Secure-Privilege state if it is not locked (LOCK[ICNP] = 0).

Offset

Register	Offset
ICNP	1Ch

Diagram



Fields

Field	Description
31-2 —	Reserved
1-0 NPEn	Nonprivilege Enable Configures privilege or nonprivilege access protection for each interrupt, or DMA request. If this field is 0, the pin is configured for privilege access. You can configure a pin to use the corresponding interrupt, or DMA request, or reconfigure a pin that is already configured to use the corresponding interrupt, or DMA request only in the Privilege state. If this field is 1, the pin is configured for nonprivilege access. In either Privilege or Nonprivilege state, you can configure a pin to use the corresponding interrupt, or DMA request, or reconfigure a pin that is already configured to use the corresponding interrupt, or DMA request. **NOTE** See the GPIO chip-specific information to determine which GPIO instances support interrupt, DMA request, or trigger capabilities. 0 - Privilege access 1 - Nonprivilege access

28.7.1.9 Port Data Output (PDOR)

Configures the logic levels that are driven on each general-purpose output pin.

NOTE

Do not modify the pin configuration registers associated with pins that are not available in your selected package. By default, these unbonded pins are set to the Disable state for lowest power consumption.

Offset

Register	Offset
PDOR	40h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PDO31	PDO30	PDO29	PDO28	PDO27	PDO26	PDO25	PDO24	PDO23	PDO22	PDO21	PDO20	PDO19	PDO18	PDO17	PDO16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDO15	PDO14	PDO13	PDO12	PDO11	PDO10	PDO9	PDO8	PDO7	PDO6	PDO5	PDO4	PDO3	PDO2	PDO1	PDO0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PDOn	Port Data Output Configures the logic level on the pin if it is configured for general-purpose output. If this field is 0, logic level 0 is driven on the pin, if the pin is configured for general-purpose output. If this field is 1, logic level 1 is driven on the pin, if the pin is configured for general-purpose output. **NOTE** Reading the fields for unbonded pins returns an undefined value. 0 - Logic level 0 1 - Logic level 1

28.7.1.10 Port Set Output (PSOR)

Updates the corresponding fields of Port Data Output (PDOR) to become 1.

Offset

Register	Offset
PSOR	44h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTSO 31	PTSO 30	PTSO 29	PTSO 28	PTSO 27	PTSO 26	PTSO 25	PTSO 24	PTSO 23	PTSO 22	PTSO 21	PTSO 20	PTSO 19	PTSO 18	PTSO 17	PTSO 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTSO 15	PTSO 14	PTSO 13	PTSO 12	PTSO 11	PTSO 10	PTSO 9	PTSO 8	PTSO 7	PTSO 6	PTSO 5	PTSO 4	PTSO 3	PTSO 2	PTSO 1	PTSO 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PTSOn	Port Set Output Updates the content of the corresponding field in Port Data Output (PDOR). If this field is 0, the corresponding PDOR field does not change. If this field is 1, the corresponding PDOR field becomes 1. 0 - No change 1 - Corresponding field in PDOR becomes 1

28.7.1.11 Port Clear Output (PCOR)

Updates the corresponding fields of Port Data Output (PDOR) to become 0.

Offset

Register	Offset
PCOR	48h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTCO 31	PTCO 30	PTCO 29	PTCO 28	PTCO 27	PTCO 26	PTCO 25	PTCO 24	PTCO 23	PTCO 22	PTCO 21	PTCO 20	PTCO 19	PTCO 18	PTCO 17	PTCO 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTCO 15	PTCO 14	PTCO 13	PTCO 12	PTCO 11	PTCO 10	PTCO 9	PTCO 8	PTCO 7	PTCO 6	PTCO 5	PTCO 4	PTCO 3	PTCO 2	PTCO 1	PTCO 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PTCOn	Port Clear Output Updates the content of the corresponding field in Port Data Output (PDOR). If this field is 0, the corresponding PDOR field does not change. If this field is 1, the corresponding PDOR field becomes 0. 0 - No change 1 - Corresponding field in PDOR becomes 0

28.7.1.12 Port Toggle Output (PTOR)

Updates the corresponding fields of Port Data Output (PDOR) to set to the inverse of their current logic states.

Offset

Register	Offset
PTOR	4Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTTO 31	PTTO 30	PTTO 29	PTTO 28	PTTO 27	PTTO 26	PTTO 25	PTTO 24	PTTO 23	PTTO 22	PTTO 21	PTTO 20	PTTO 19	PTTO 18	PTTO 17	PTTO 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTTO 15	PTTO 14	PTTO 13	PTTO 12	PTTO 11	PTTO 10	PTTO 9	PTTO 8	PTTO 7	PTTO 6	PTTO 5	PTTO 4	PTTO 3	PTTO 2	PTTO 1	PTTO 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PTTOn	Port Toggle Output Updates the content of the corresponding field in Port Data Output (PDOR). If this field is 0, the corresponding PDOR field does not change. If this field is 1, the corresponding PDOR field is set to the inverse of its current logic state. 0 - No change 1 - Set to the inverse of its current logic state

28.7.1.13 Port Data Input (PDIR)

Captures the logic levels of each general-purpose input pin.

NOTE

Do not modify the pin configuration registers associated with the pins that are not available in your selected package. By default, these unbonded pins are set to the Disable state for lowest power consumption.

Offset

Register	Offset
PDIR	50h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PDI31	PDI30	PDI29	PDI28	PDI27	PDI26	PDI25	PDI24	PDI23	PDI22	PDI21	PDI20	PDI19	PDI18	PDI17	PDI16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PDI15	PDI14	PDI13	PDI12	PDI11	PDI10	PDI9	PDI8	PDI7	PDI6	PDI5	PDI4	PDI3	PDI2	PDI1	PDI0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PDIn	Port Data Input Indicates the logic level of the pin that is configured for use by a digital function. If this field is 0, the pin logic level is logic 0 or is not configured or implemented for use by a digital function. If this field is 1, the pin logic level is logic 1. 0 - Logic 0 1 - Logic 1

28.7.1.14 Port Data Direction (PDDR)

Configures the individual port pins for input or output.

Offset

Register	Offset
PDDR	54h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PDD31	PDD30	PDD29	PDD28	PDD27	PDD26	PDD25	PDD24	PDD23	PDD22	PDD21	PDD20	PDD19	PDD18	PDD17	PDD16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDD15	PDD14	PDD13	PDD12	PDD11	PDD10	PDD9	PDD8	PDD7	PDD6	PDD5	PDD4	PDD3	PDD2	PDD1	PDD0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PDDn	Port Data Direction Configures individual port pins for input or output. If this field is 0, the pin is configured as general-purpose input for the GPIO function. If this field is 1, the pin is configured as general-purpose output for the GPIO function. 0 - Input 1 - Output

28.7.1.15 Port Input Disable (PIDR)

Disables the input for each general-purpose pin, which prevents the value from being reported in Port Data Input (PDIR).

Offset

Register	Offset
PIDR	58h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24	PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PIDn	Port Input Disable Disables a pin for general-purpose input. If this field is 0, the pin is configured for general-purpose input, provided the pin is configured for a digital function. If this field is 1, the pin is disabled for general-purpose input. 0 - Configured for general-purpose input 1 - Disabled for general-purpose input

28.7.1.16 Pin Data (P0DR - P31DR)

Configures the data feature of a pin. Pins that are unimplemented or not configured for a digital function read zero.

NOTE

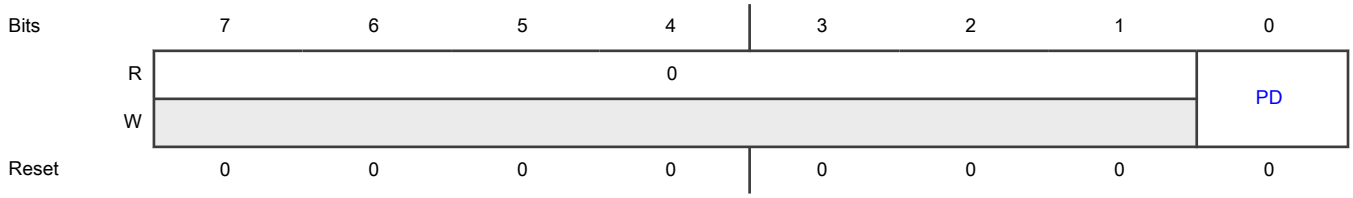
You must not modify Pin Data (P0DR - P31DR) associated with the pins that are not available in your selected package. These unbonded pins are, by default, set to the Disable state for lowest power consumption.

Offset

For a = 0 to 31:

Register	Offset
PaDR	60h + (a × 1h)

Diagram



Fields

Field	Description
7-1 —	Reserved
0 PD	Pin Data (I/O) Specifies the pin logic level. This field updates the corresponding field in Port Data Output (PDOR); reading this field returns the value in the corresponding field of Port Data Input (PDIR). If this field is 0, the pin logic level is logic zero or not configured for use by a digital function. If this field is 1, the pin logic level is logic one. 0 - Logic zero 1 - Logic one

28.7.1.17 Interrupt Control a (ICR0 - ICR31)

Configures interrupt features on each pin.

Offset

For a = 0 to 31:

Register	Offset
ICRa	80h + (a × 4h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							ISF	LK	0		IRQS	IRQC			
W								W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-25 —	Reserved
24 ISF	Interrupt Status Flag Indicates whether the configured interrupt is detected. The pin interrupt configuration is valid in all digital pin muxing modes. The fields in Interrupt Status Flag (ISFR0 - ISFR1) have the same function. ISF can be cleared with either register field. If the pin is configured to generate a DMA request, then the corresponding flag is cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic 1 is written to the flag. If the pin is configured for a level-sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not detected 1 - Detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When writing 0 - No effect 1 - Clear the flag
23 LK	Lock Locks ICR[23:0], and you cannot update the locked field until the next system reset. If this field is 0, interrupt configuration by ICR[23:0] is not locked and can be updated. If this field is 1, interrupt configuration by ICR[23:0] is locked and cannot be updated until the next system reset. 0 - Not locked 1 - Locked
22-21 —	Reserved
20 IRQS	Interrupt Select Configures the selected interrupt, or DMA request. **NOTE** See the GPIO chip-specific information to determine which GPIO instances support interrupt, DMA request, or trigger capabilities. 0 - Interrupt, or DMA request 0 1 - Interrupt, or DMA request 1
19-16 IRQC	Interrupt Configuration Specifies the ISF and DMA request configuration. The pin interrupt configuration is valid in all digital pin muxing modes. When changing the interrupt configuration, it is recommended to first disable ISF and then write the new configuration. The corresponding pin is configured to generate interrupt, or DMA request. **NOTE** See the GPIO chip-specific information to determine the GPIO instances that support interrupt, DMA request, or trigger capabilities. 0000 - ISF is disabled 0001 - ISF and DMA request on rising edge 0010 - ISF and DMA request on falling edge 0011 - ISF and DMA request on either edge 0100 - Reserved 0101 - ISF sets on rising edge 0110 - ISF sets on falling edge 0111 - ISF sets on either edge 1000 - ISF and interrupt when logic 0

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1001 - ISF and interrupt on rising edge 1010 - ISF and interrupt on falling edge 1011 - ISF and Interrupt on either edge 1100 - ISF and interrupt when logic 1 1101 - Reserved 1110 - Reserved 1111 - Reserved
15-0 —	Reserved

28.7.1.18 Global Interrupt Control Low (GICLR)

Updates any combination of the lower 16 Interrupt Control registers with the same value. This register supports only 32-bit writes and ignores any 16-bit or 8-bit writes.

Offset

Register	Offset
GICLR	100h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	GIWD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	GIWE 15	GIWE 14	GIWE 13	GIWE 12	GIWE 11	GIWE 10	GIWE 9	GIWE 8	GIWE 7	GIWE 6	GIWE 5	GIWE 4	GIWE 3	GIWE 2	GIWE 1	GIWE 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16	Global Interrupt Write Data

Table continues on the next page...

Table continued from the previous page...

Field	Description
GIWD	Indicates the write value that is written to the upper 16 bits of Interrupt Control a (ICR0 - ICR31), selected by GIWE.
15-0 GIWEn	Global Interrupt Write Enable Indicates whether the upper 16 bits of the corresponding Interrupt Control a (ICR0 - ICR31) are updated with the value in GIWD. 0 - Not updated 1 - Updated

28.7.1.19 Global Interrupt Control High (GICHR)

Updates any combination of the upper 16 Interrupt Control registers with the same value. This register supports only 32-bit writes and ignores any 16-bit or 8-bit writes.

Offset

Register	Offset
GICHR	104h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	GIWD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	GIWE 31	GIWE 30	GIWE 29	GIWE 28	GIWE 27	GIWE 26	GIWE 25	GIWE 24	GIWE 23	GIWE 22	GIWE 21	GIWE 20	GIWE 19	GIWE 18	GIWE 17	GIWE 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 GIWD	Global Interrupt Write Data Indicates the write value that is written to the upper 16 bits of Interrupt Control a (ICR0 - ICR31), selected by GIWE.
15-0	Global Interrupt Write Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
GIWEn	Indicates whether the upper 16 bits of the corresponding Interrupt Control a (ICR0 - ICR31) are updated with the value in GIWD. 0 - Not updated. 1 - Updated

28.7.1.20 Interrupt Status Flag (ISFR0 - ISFR1)

Indicates whether the related configured interrupt is detected on each pin. The pin interrupt configuration is valid in all digital pin muxing modes. The ISF for each pin is also visible in the corresponding Interrupt Control register, and each flag can be cleared in either location.

There is a separate ISF register for each interrupt, or DMA request domain. Each status flag is only visible in the register that corresponds to that flag's domain, as configured in ICR*n*[IRQS].

Offset

Register	Offset
ISFR0	120h
ISFR1	124h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ISF31	ISF30	ISF29	ISF28	ISF27	ISF26	ISF25	ISF24	ISF23	ISF22	ISF21	ISF20	ISF19	ISF18	ISF17	ISF16
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISF15	ISF14	ISF13	ISF12	ISF11	ISF10	ISF9	ISF8	ISF7	ISF6	ISF5	ISF4	ISF3	ISF2	ISF1	ISF0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ISFn	Interrupt Status Flag Indicates the detection of the configured interrupt on each pin of the same number. If this field is 0, the configured interrupt is not detected on the pin of the same number. If this field is 1, the configured interrupt is detected on the pin of the same number. If the pin is configured to generate a DMA request, then the corresponding flag is cleared automatically at the completion of the requested DMA transfer. Otherwise, the

Table continues on the next page...

Field	Description
	flag remains set until a logic 1 is written to the flag. If the pin is configured for a level-sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not detected 1 - Detected When writing 0 - No effect 1 - Clear the flag

Chapter 29
Clock and Power Overview

29.1 Overview

This chip targets many applications with power requirements that include low power consumption, long battery life, always-on capability, instant-on capability, and doesn't require active cooling. To meet these requirements, the chip design focuses on reducing current consumption, while simultaneously enabling the maximum level of peak performance, and also a balanced level of sustained performance. The chip architecture uses a wide range of power-management techniques, used in combination, for system design flexibility.

This chapter describes these architecture details as a high-level clock and power management overview of the chip.

29.2 Components of Clock and Power Management

The centralized clock generation, power generation and distribution are implemented by the following blocks:

Table 139. Clock and Power Components

Name	Description
PLL	The chip supports Arm PLL, SYSTEM PLL, AUDIO PLL, DRAM PLL and VIDEO PLL.
Crystal OSC (XTALOSC)	The XTALOSC includes a 24 MHz oscillator. The 24MHz crystal oscillator is the primary clock source for all the PLLs, and clock generation for the CPU and high-speed interfaces as well as the low-speed clock source for CCM/SRC/GPC.
PLLs and PFDs	The PLLs and their associated PFDs generate the clocks with various frequencies required to feed the CCM clock generator that supplies the different functional blocks. See the PLL and PFD section in Clock Controller Module (CCM) for information on the PLL and PFD architecture, functional description and programming model.
Clock Control Module (CCM)	The CCM module provides control for clock generation, division, distribution, synchronization, and coarse-level gating. See Clock Controller Module (CCM) for information on the CCM architecture, functional description and programming model.
Low Power Clock Gating (LPCG)	The LPCG distributes the clocks to all blocks in the SoC and handles automated clock gating and block level software-controllable clock gating. See Clock Controller Module (CCM) for information on the LPCG architecture and functional description.
General Power Controller (GPC)	This module controls the power state of the SoC. The GPC handles the power gating under low power modes and also manages the power up / power down sequences. See the General Power Controller (GPC) chapter for information on the GPC architecture, functional description and programming model.
System Reset Controller (SRC)	This module generates the reset signals to all the modules in the SoC. The SRC will appropriately assert reset signals for power transitions, entry, and exit. See the System Reset Controller (SRC) chapter for information on the SRC architecture, functional description and programming model.

Table continues on the next page...

Table 139. Clock and Power Components (continued)

Name	Description
IP Controller (IPC)	The clock roots are delivered to each module through the IPC, which contains the clock gating logic for each clock. The control signals for the clock gating control in the CCM is the source for clock enable signals. Since some of the clocks need to be gated off during reset, the SRC also sends clock gate signals to the IPC.

Together, the modules listed above provide enhanced power-management features with the centralized control for the clock, reset, and power-management signals on the SoC.

29.3 Power Management

The power management design on the chip satisfies many use cases that depend on a key balance of performance and low-power consumption. In order to meet these criteria, the system power management needs to be dynamic and flexible. This is accomplished on a multi-core system by utilizing a power scheme that considers each CPU domain, and arbitrates the appropriate power mode configuration to satisfy the needs of both CPUs, and the system as a whole. The following are additional features supported by the power management architecture:

- Programmable power mode transition rules
- Independent operation mode for individual CPU cores
- Intelligent clock and power management for multi-core architecture
- Unified power management interface (UPI) for all on-chip resources, including clock, power, reset, and peripherals

NOTE

- When switchable MIXes, such as MLMIX/MEDIAMIX/DDRMIX/WKUPMIX, are powered off, corresponding opposite Subsystem sides in NICMIX can generate Slave Error (SLVERR).
- Always-on MIXes, such as HSIOMIX/AONMIX, cannot be powered off, so cannot generate SLVERR by subsystem.
- When clock gating is enabled, subsystem cannot generate SLVERR for both switchable/always-on MIX.

29.3.1 Power Modes

The chip supports the following power modes: -

- RUN: In this mode, the Cortex®-A55 CPU is active and running, some portion can be shut off for power saving.
- Low Power RUN: This mode is defined as a very low power run mode with all external power rails are on. In this mode, all the unnecessary power domain (MIX) can be off, except AONMIX and the internal modules required, such as OSC24M/PLL. Cortex-M33 CPU in AONMIX handles all the computing and data processing. Cortex-A55 is power down and DRAM can be in self-refresh/retention mode. All the modules in the AONMIX, such as SAI/CAN/LPUART, can be used directly. To use modules in other power domain, such as WAKEUPMIX, additional peripherals and related power by Cortex-M33 can be turned on as needed.
- IDLE: The CPU can automatically enter this mode when there is no thread running and all high-speed devices are not active. CPU can be put into power gated state but with L3 data retained, DRAM and bus clock are reduced, most of the internal logic is clock gated but still remain powered. Compared to RUN mode, all the external power from PMIC remains the same, and most of the modules remain their state, so the interrupt response in this mode is very small.
- SUSPEND: It is the most power saving mode where all the clocks are off and all the unnecessary power supplies are off. Cortex®-A55 CPU is fully power gated, all internal digital logic and analog circuit that can be power down will be off, all PHYs are power gated. VDD_SOC (and related digital supply) Voltage is reduced to the “Suspend Mode” voltage. The exit time from this mode will be much longer than IDLE but the power consumption will also be much lower.

- BBSM: This mode is also called RTC mode. In this mode, only the power for the BBSM domain remain on to keep RTC and BBSM logic alive.
- OFF: This mode has all power rails OFF. BBSM, IDLE and SUSPEND are referred as Low Power Mode for the chip, SUSPEND and IDLE are the two typical low power modes defined based on the use case of Linux kernel.

29.3.1.1 Low Power Modes

The state of each module in the IDLE mode, SUSPEND mode are defined in the table below.

	IDLE	SUSPEND[1]
CCM LPM Mode	WAIT	STOP
ARM A55 CPU0	OFF	OFF
ARM A55 CPU1	OFF	OFF
Shared L3 Cache	ON	OFF
DISPLAY	OFF	OFF
DRAM Controller & PHY	ON	OFF
ARM_PLL	OFF	OFF
DRAM_PLL	OFF	OFF
SYSTEM_PLL1	ON	OFF
XTAL	ON	OFF
RTC	ON	ON
External DRAM Device	Self-Refresh [2]	Self-Refresh [3]
DDR clock	266MHz	OFF
AXI clock	133MHz	OFF
Module clocks	ON as needed	OFF
GPIO wakeup	Yes	Yes
RTC wakeup	Yes	Yes
USB remote wakeup	Yes	No [4]
Other wakeup source	Yes	No
WAKEUPMIX	ON	OFF
MLMIX	ON	OFF
NICMIX	ON as needed	OFF

1. In suspend mode, the following modules act as async wakeup sources that can wakeup the chip. These are capable of generating an interrupt without needing to read/write SRAMs.

   GPIO

   UART

   FlexCAN

   ENET QoS

   ENET FEC

LPTMR: count from osc32k

USB(DP/DM Pins)

SDIO

2. Automatic enter self-refresh when there is no DRAM access
3. Put into self-refresh mode by SW before entering low power mode
4. Remote wakeup can be supported if the USB PHY power is on in this mode

29.3.1.2 CPU Power Modes

The Cortex®-A55 and Cortex®-M33 CPU platforms support the power states that are defined as CPU modes. The CPU Modes are detailed below.

Table 140. CPU Modes

CPU Mode	Description
RUN	The Cortex®-A55 CPU is active and running.
WAIT	The Cortex®-A55 and Cortex®-M33 CPU is in the WFI state. The CPU core and L1 cache are clock gated, TCM and peripheral can still be alive. The chip can wakeup from this mode from IRQ with a very short latency.
STOP	The Cortex®-M33 CPU is in the WFI state. The CPU core, L1 cache, TCM, and it's private peripherals are all clock gated. The chip requires a longer time to wakeup from this mode.
SUSPEND	The Cortex®-A55 and Cortex®-M33 CPU core and L1 cache are power gated (except the cache). The TCM is in retention mode. SUSPEND mode consumes the least amount of power, while keeping parts of the system alive. But the chip requires a much longer time to wakeup from this mode.
OFF	The Cortex-A55 core supports a full shutdown mode where power can be removed completely and no state is retained.

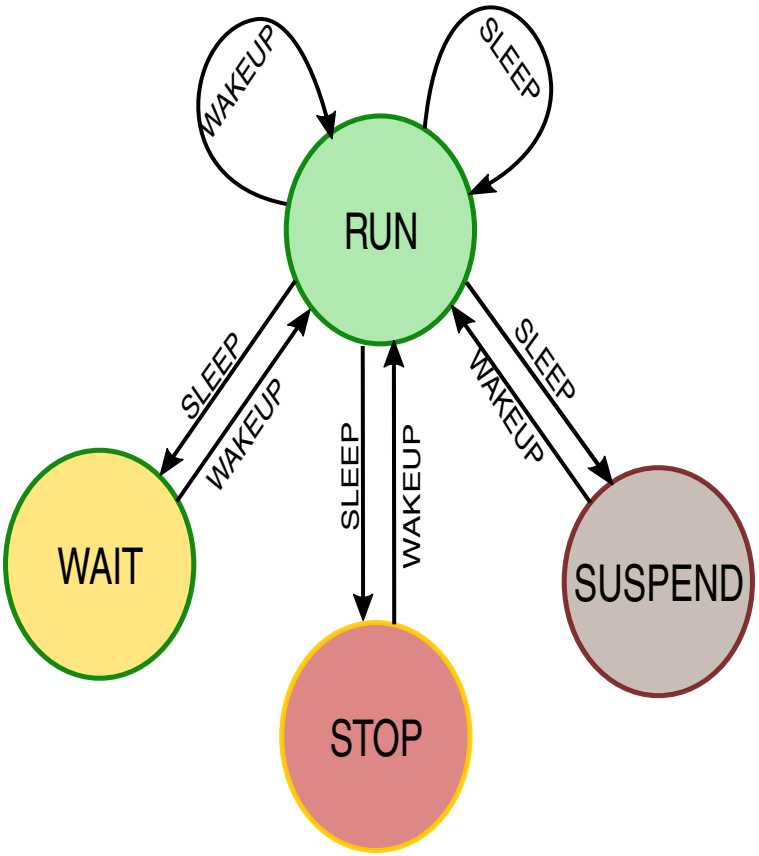


Figure 60. CPU Modes

29.3.2 Power Sequence

The power up and power down sequence in i.MX93 is optimized for use with an external PMIC.

Power Up:

1. NVCC_BBSM_1P8 supply must be turned on. The chip will assert PMIC_ON_REQ at this point in time.
2. VDD_SOC digital voltage supplies should be turned on.
3. VDD analog, PHY and PLL supplies should be turned on.
4. All remaining 1.8V supplies should be turned on including VDD_1P8 analog, PHY and PLL supplies and 1.8V NVCC IO supplies.
5. DDR IO supplies should be turned on.
6. All 3.3V supplies must be turned on, including 3.3V NVCC IO supplies and VDD_USB_3P3.

NOTE

3.3V supply step may be simultaneous with either the 1.8V or the DDR supplies if desired.

7. POR_B should be released. It should be asserted during the entire power up sequence.

Power Down:

- NVCC_BBSM_1P8 supply must be turned off after any other power supply.
- VDD_SOC must be turned off after the other (non-BBSM) power rails or at the same time as other (non-BBSM) rails. No sequence for other power rails during power down.

29.4 Clocks

29.4.1 Clock System Diagram

The overall clock system for the chip is shown in the diagram below.

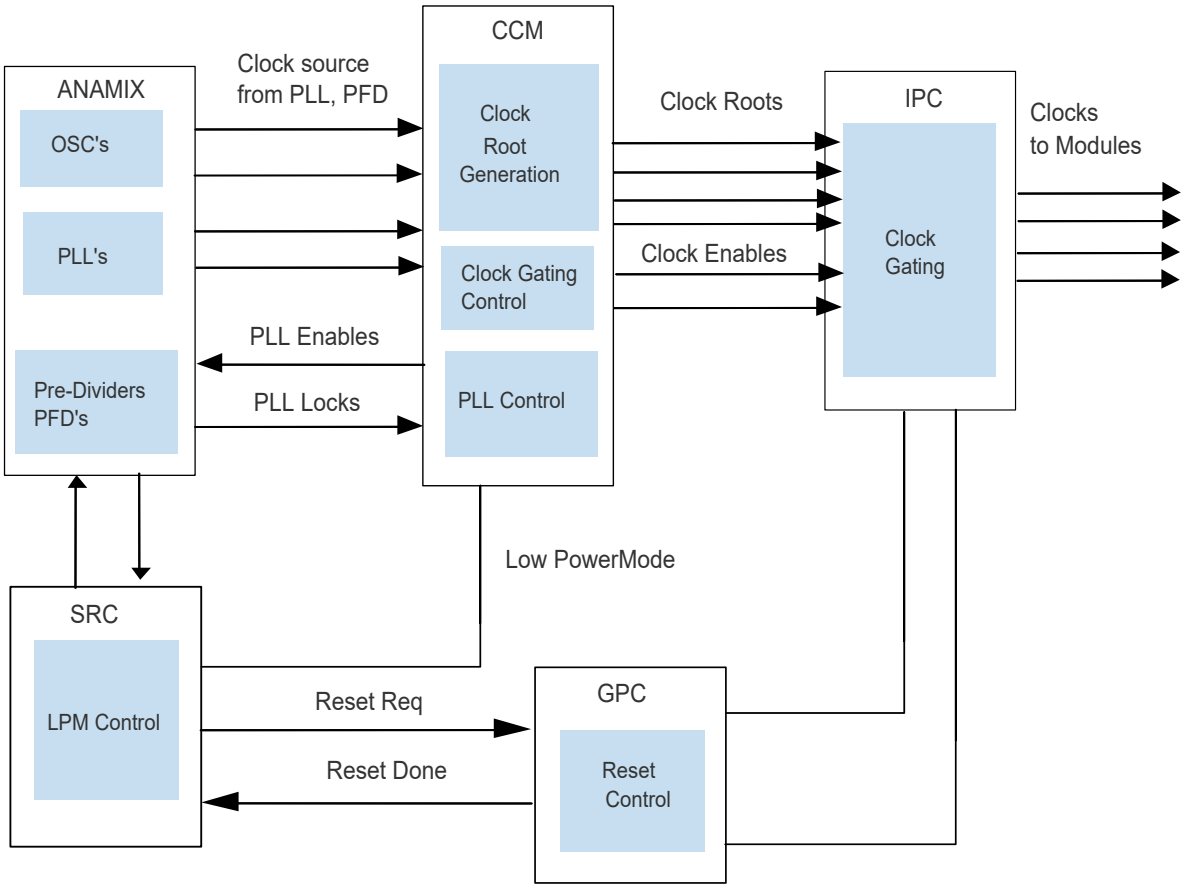


Figure 61. Clock System

The OSCs, PLLs, PFDs and Pre-Dividers generate the clock sources with fixed or variable frequency. The clock sources feed the clock root generation logic inside CCM, which generates numerous clock roots required for core, bus, and peripheral blocks. These clock roots are branched to each module through the LPCG logic, which contains the clock gating cells for each clock. Control signals for the LPCGs are in CCM (clock enable signal). Some of the clocks need to be gated off during reset, because of this, the SRC module will also send clock gate signals to the LPCG.

In low power mode, GPC will drive the low power mode state signal to CCM, then CCM will enable clock gating based on the configuration. At the same time, CCM may also deassert enable signals for PLLs and PFDs, so these modules can power off during low power modes. When the chip enters STOP mode, CCM may deassert the OSC enable signal to shut off the OSC, it may also assert the STOP mode signal to set OSC to low power mode.

The detailed functional description for each block will be described in the following sections.

29.4.2 Oscillators

This chip has two XTAL modules:

- 24MHz XTAL module for system clock
- 32KHz XTAL module for RTC

Each XTAL module includes:

- Crystal oscillator to generate reference clock;
- Digital control logics for the XTAL.

Each XTAL module supports the following mode through register configuration:

- Normal oscillator mode - XTAL generates stable square wave based on the crystal oscillator input.
- Bypass mode - external clock can be input through XTAL pad.

29.4.3 PLL and PFD

The PLLs are used in the chip to generate the high speed clocks required by the modules in SoC. A few PLLs are equipped with Phase Fractional Dividers (PFDs) in order to generate additional frequencies. The following PLLs are used in this chip:

Table 141. PLLs

PLL	Description	Spread Spectrum	Configurable
ARM_PLL	The ARM_PLL/INTEGER PLL is used to generate the clock for the Arm Cortex-A55 platform. It is a programmable integer-frequency multiplier PLL, capable of an output frequency over 1 GHz. This PLL is physically located near the high-speed Cortex-A55 processor.	-	Yes
SYS_PLL1	This is a fractional PLL fixed at a fixed VCO frequency of 4.0GHz. Attached to this integer PLL are 3 PFD (phase frequency detector) circuits, also referred to as DFS (digital frequency synthesis) circuits, which generate Fclkout = FVCO/(MFI+MFN/5)). The PLL PFDs also have a fixed /2 divisor output, so that /2 of each of the 3 clocks are also available for use by the CCM. These clock sources will be used for the majority of AXI/AHB buses in the chip.	Yes[1]	Yes
AUDIO_PLL	An AVPLL used to generate reference clocks, mainly for serial audio interfaces, and external audio codecs. This fractional multiplier PLL provides a low-jitter and high precision audio clock with standardized audio frequencies.	Yes	Yes
VIDEO_PLL	An AVPLL used to generate reference clocks, mainly for display and video interfaces. This fractional multiplier PLL provides a low-jitter and high-precision video clock at standardized video frequencies.	Yes	Yes

1. SYSTEM_PLL1 is also used to generate the 125MHz clocking for Ethernet RGMII. So spread spectrum on SYSTEM_PLL1 can only be enabled when Ethernet is not used.

NOTE

Please see the datasheet for all supported maximum PLL frequencies.

Each PLL can use one of the following clock sources as it's reference clock:

- Primary Option: 24 MHz clock from 24 MHz oscillator (XTAL).

The SYS_PLL, DRAM_PLL, AUDIO_PLL and VIDEO_PLL have the capability to spread spectrum the generated signal. They do not require exact/constant frequency, and can be changed as a part of dynamic frequency scaling procedure and/or can be spread-spectrum modulated.

The main output clocks for the PLLs (PLLx_CLK) are not configurable, but their PFDs are. Each PFD works independently by interpolating the VCO of the PLL to which it is connected. It effectively takes the PLL VCO frequency and produces 18/N x Fvco at it's output where N ranges from 12 to 35. PFD is a completely digital design with no analog components or feedback loops. The frequency switch time is much faster than a PLL because keeping the base PLL locked and changing the integer N only changes the logical combination of the interpolated outputs of the VCO. Note that the PFD not only enables faster frequency changes than a PLL, but also allows the configuration to be safely changed "on-the-fly" without going through the output clock disabling/enabling process.

The basic logical connections for the PLLs, PFDs and Dividers is shown in the diagram below:

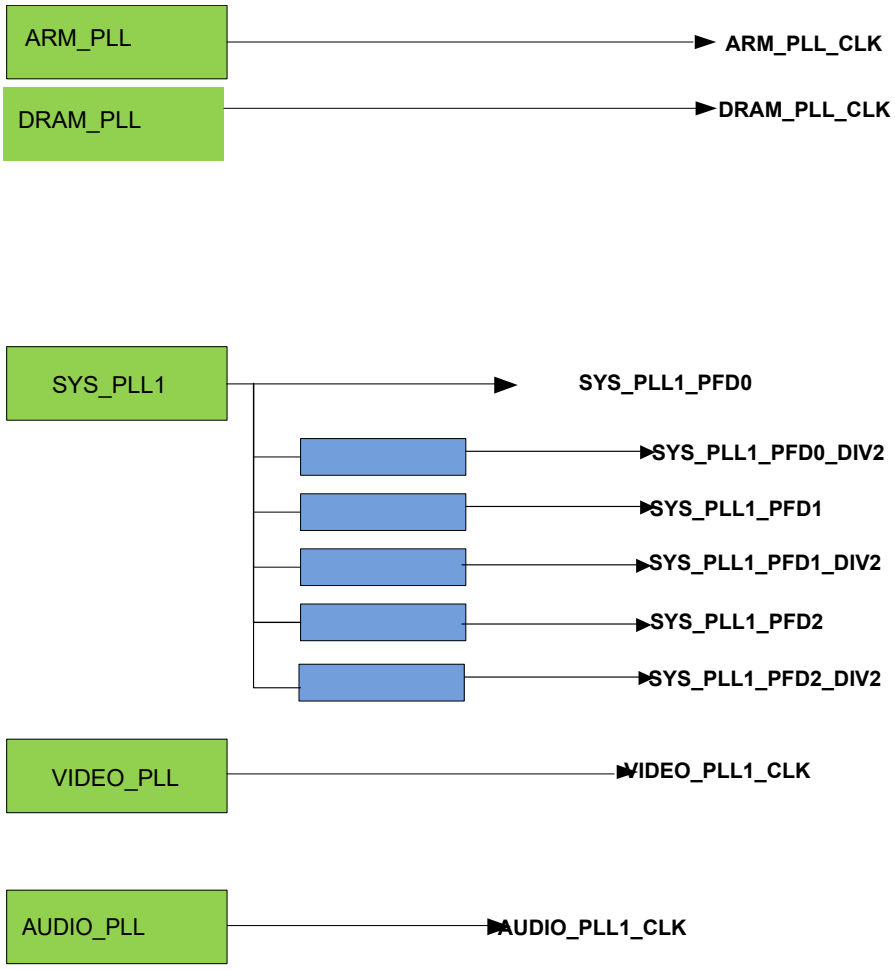


Figure 62. PLL and PFD

PLL configuration and control functions are accessible via configuration and status registers. The CCM block contains the detailed descriptions of the memory mapped registers, and control functions of the clock generation sub-module. Please see CCM for more information.

29.4.4 Clock Root Generation

Clock generation in the CCM creates clock roots for on-chip peripherals. Each clock root will have a dedicated clock slice. It takes the clock source from Oscillators, PLL/PFD or Pre-Dividers, and generate the clock root with required frequency.

The CCM can manage on-chip peripheral clocks by controlling the low power clock gating cells (LPCGs) before the clock roots enter the peripherals.

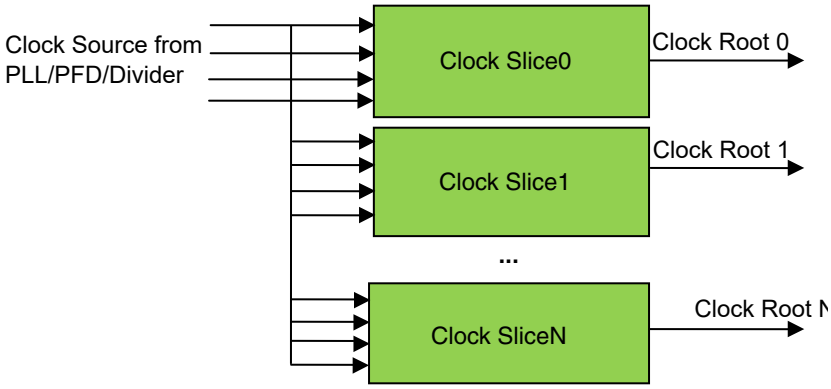


Figure 63. Clock Root Generation from Clock Slice

The LPCGs are implemented to control peripheral functions, and there could be several LPCGs for each functions in a peripheral.

For details on the programming model, please refer to CCM Memory Map/Register Definition.

29.4.5 Low Power Clock Gating (LPCG)

The clock roots generated inside CCM are distributed to the entire SoC through LPCG. The LPCG block receives the root clocks from CCM and splits them to clock branches for each functional block. The clock branches are individually gated clocks.

See the CCM chapter for more information about clock gating.

Chapter 30
Clock Controller Module (CCM)

30.1 Overview

The Clock Control Module (CCM) manages the on-chip module clocks. The oscillators, PLLs, and Phase Fractional Dividers (PFD) will generate clock sources with fixed or variable frequencies.

30.1.1 Block diagram

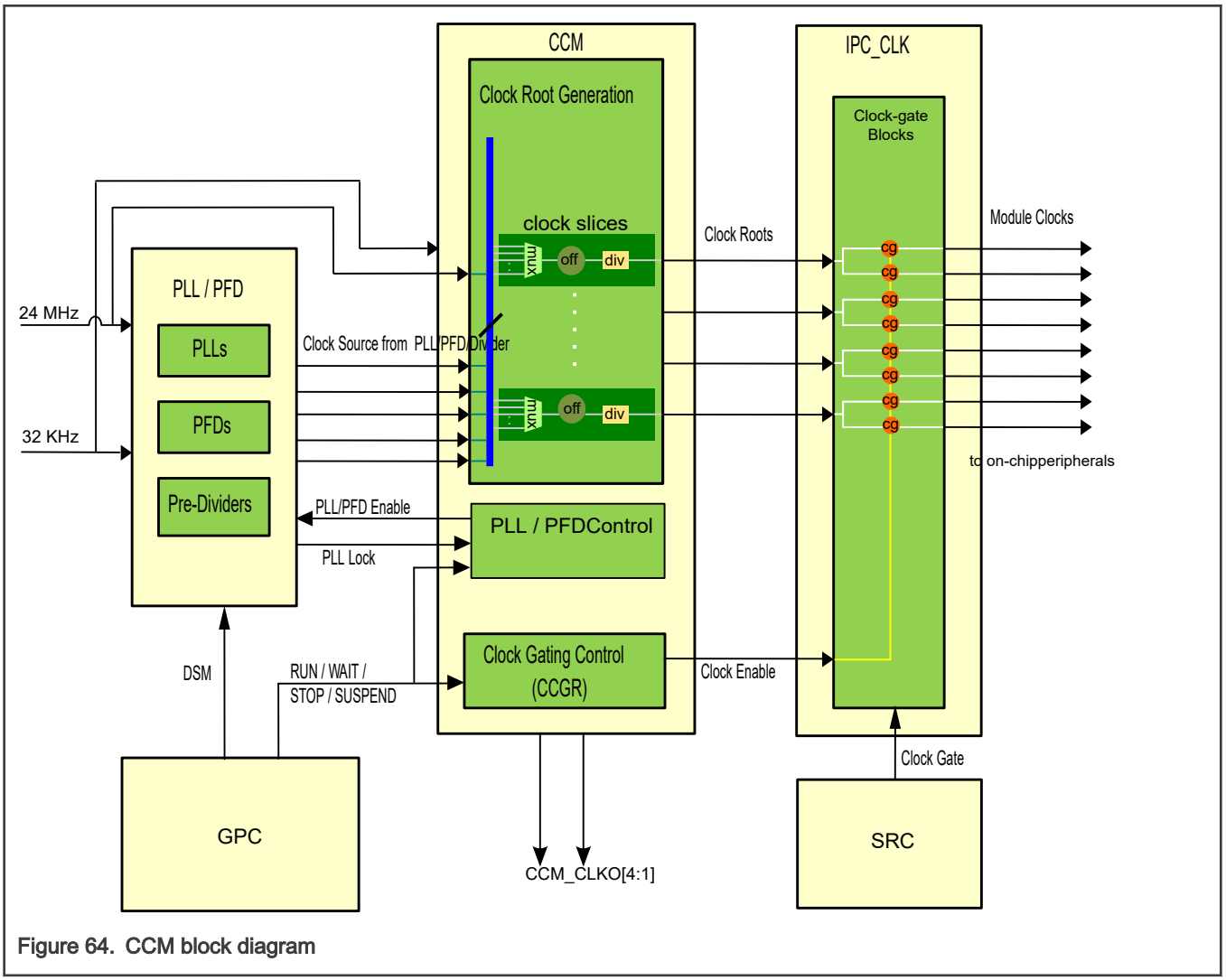


Figure 64. CCM block diagram

The clock root generation logic inside CCM will generate various clock roots required for core, bus, and peripheral blocks. These clock roots will be delivered to each module through IPC_CLK, which contains the clock gating logic for each clock.

The control signal for CCGRs in CCM will be the source for clock enable signals. Since some of the clocks need to be gated off during the reset, the System Reset Controller (SRC) will also send clock gate signals to the IPC_CLK.

In low power mode, the General Power Controller(GPC) will drive the low power mode state signal to CCM, and CCM will enable clock gating based on the configuration. At the same time, CCM may also de-assert enable signals for PLLs/PDFs, so these modules can be powered off during the low power mode.

30.1.2 Features

The CCM includes the following features:

- Clock Root
  - Up to 4 clock sources for each clock root
  - 8-bit divider support up to divide by 256 in each clock root
  - Each clock root can be shutdown by software
- Peripheral clock can be auto gated according to CPU work mode, software configurations, or system work mode
- PLLs and PFDs can be auto shutdown according to CPU work mode, software configurations, or system work mode
- Access Control
  - Two access control schemes: Trustzone, and Domain
  - Write access to each component can be protected independently
  - Independent setting for user, privileged, secure, non-secure
  - Domain mode access can be controlled by whitelist
  - Access control setting can be locked independently

30.2 System Clocks

The table found here shows the CCM output clocks' system-level connectivity.

NOTE

LPIT1 is clocked by AON bus and LPIT2 is clocked by WAKEUP bus. LPIT1 and LPIT2 have their trigger output 0 used as input by CAN1 and CAN2 respectively. Because of this, these LPIT timer instances are clocked by the bus clock - the same clock used to clock the respective CAN IP (bus_aon_clk_root and bus_wakeup_clk_root respectively) in order to ensure that these signals are timed synchronously between these modules.

Table 142. System Clocks Table

Module	Module Clock (instance.clock)	Clock Root Module	Low Power Clock Gating (LPCG)
ipsyncm_anamix	ipsyncm_anamix.ipg_master_clk	bus_aon_clk_root	LPCG7
aonmix.trdc	trdc_mdac_a3.hclk	m33_clk_root	LPCG26
	trdc_mdac_a3.ipg_clk	m33_clk_root	
	trdc_mgr_aon.ipg_clk	m33_clk_root	
	trdc_mgr_aon.hclk	m33_clk_root	
	trdc_mdac_a4.ipg_clk	m33_clk_root	
	trdc_mdac_a4.hclk	m33_clk_root	
	trdc_mdac_a2.ipg_clk	m33_clk_root	
	trdc_mdac_a2.hclk	m33_clk_root	
	trdc_mbc_a0.ipg_clk	m33_clk_root	
	trdc_mbc_a0.hclk	m33_clk_root	

Table continues on the next page...

Table 142. System Clocks Table (continued)

	trdc_mrc_a0.hclk	m33_clk_root	
	trdc_mrc_a0.ipg_clk	m33_clk_root	
	trdc_mbc_a1.hclk	m33_clk_root	
	trdc_mbc_a1.ipg_clk	m33_clk_root	
	ipsync_m33_trdc_aon.ipg_slave_clk	m33_clk_root	
	ipsync_m33_trdc_aon.ipg_master_clk	m33_clk_root	
can1	can1.ipg_clk_pe	can1_clk_root	LPCG50
	can1.ipg_clk_pe_nogate	can1_clk_root	
	can1.ipt_clk	bus_aon_clk_root	
	can1.ipg_clk	bus_aon_clk_root	
	can1.ipg_clk_s	bus_aon_clk_root	
	can1.ipg_clk_chi	bus_aon_clk_root	
	can1.osc_clk	osc_24m	
can2	can2.osc_clk	osc_24m	LPCG51
	can2.ipg_clk	bus_wakeup_clk_root	
	can2.ipg_clk_chi	bus_wakeup_clk_root	
	can2.ipt_clk	bus_wakeup_clk_root	
	can2.ipg_clk_pe_nogate	can2_clk_root	
	can2.ipg_clk_pe	can2_clk_root	
	can2.ipg_clk_s	bus_wakeup_clk_root	
eDMA3_1	eDMA3_1.ipg_clk	m33_clk_root	LPCG21
	eDMA3_1.hclk	m33_clk_root	
	ipsync_eDMA3_1.ipg_slave_clk	m33_clk_root	
	ipsync_eDMA3_1.ipg_master_clk	bus_aon_clk_root	
eDMA4_2	ipsync_eDMA4_2.ipg_slave_clk	wakeup_axi_clk_root	LPCG22
	eDMA4_2.ipg_clk	wakeup_axi_clk_root	
	eDMA4_2.axi_clk	wakeup_axi_clk_root	
	ipsync_eDMA4_2.ipg_master_clk	bus_wakeup_clk_root	
i3c1	i3c1.PCLK	bus_aon_clk_root	LPCG76
	i3c1.FCLK	i3c1_clk_root	

Table continues on the next page...

Table 142. System Clocks Table (continued)

i3c2	i3c2.PCLK	bus_aon_clk_root	LPCG77
	i3c2.FCLK	i3c2_clk_root	
lpi2c1	lpi2c1.ipg_clk_s	bus_aon_clk_root	LPCG60
	lpi2c1.ipg_clk	bus_aon_clk_root	
	lpi2c1.lpi2c_clk	lpi2c1_clk_root	
lpi2c2	lpi2c2.ipg_clk_s	bus_aon_clk_root	LPCG61
	lpi2c2.ipg_clk	bus_aon_clk_root	
	lpi2c2.lpi2c_clk	lpi2c2_clk_root	
lpi2c3	lpi2c3.ipg_clk_s	bus_wakeup_clk_root	LPCG62
	lpi2c3.ipg_clk	bus_wakeup_clk_root	
	lpi2c3.lpi2c_clk	lpi2c3_clk_root	
lpi2c4	lpi2c4.ipg_clk_s	bus_wakeup_clk_root	LPCG63
	lpi2c4.ipg_clk	bus_wakeup_clk_root	
	lpi2c4.lpi2c_clk	lpi2c4_clk_root	
lpi2c5	lpi2c5.ipg_clk_s	bus_wakeup_clk_root	LPCG64
	lpi2c5.ipg_clk	bus_wakeup_clk_root	
	lpi2c5.lpi2c_clk	lpi2c5_clk_root	
lpi2c6	lpi2c6.ipg_clk_s	bus_wakeup_clk_root	LPCG65
	lpi2c6.ipg_clk	bus_wakeup_clk_root	
	lpi2c6.lpi2c_clk	lpi2c6_clk_root	
lpi2c7	lpi2c7.ipg_clk_s	bus_wakeup_clk_root	LPCG66
	lpi2c7.ipg_clk	bus_wakeup_clk_root	
	lpi2c7.lpi2c_clk	lpi2c7_clk_root	
lpi2c8	lpi2c8.ipg_clk_s	bus_wakeup_clk_root	LPCG67
	lpi2c8.ipg_clk	bus_wakeup_clk_root	
	lpi2c8.lpi2c_clk	lpi2c8_clk_root	
lpit1	lpit1.ipg_clk_s	bus_aon_clk_root	LPCG40
	lpit1.ipg_clk	bus_aon_clk_root	
	lpit1.ipg_ungated_per_clk	bus_aon_clk_root	
lpit2	lpit2.ipg_ungated_per_clk	bus_wakeup_clk_root	LPCG41
	lpit2.ipg_clk_s	bus_wakeup_clk_root	
	lpit2.ipg_clk	bus_wakeup_clk_root	
lpspi1	lpspi1.lpspi_clk	lpspi1_clk_root	LPCG68

Table continues on the next page...

Table 142. System Clocks Table (continued)

	lpspi1.ipg_clk	bus_aon_clk_root	
	lpspi1.ipg_clk_s	bus_aon_clk_root	
lpspi2	lpspi2.lpspi_clk	lpspi2_clk_root	LPCG69
	lpspi2.ipg_clk	bus_aon_clk_root	
	lpspi2.ipg_clk_s	bus_aon_clk_root	
lpspi3	lpspi3.lpspi_clk	lpspi3_clk_root	LPCG70
	lpspi3.ipg_clk_s	bus_wakeup_clk_root	
	lpspi3.ipg_clk	bus_wakeup_clk_root	
lpspi4	lpspi4.lpspi_clk	lpspi4_clk_root	LPCG71
	lpspi4.ipg_clk_s	bus_wakeup_clk_root	
	lpspi4.ipg_clk	bus_wakeup_clk_root	
lpspi5	lpspi5.lpspi_clk	lpspi5_clk_root	LPCG72
	lpspi5.ipg_clk_s	bus_wakeup_clk_root	
	lpspi5.ipg_clk	bus_wakeup_clk_root	
lpspi6	lpspi6.lpspi_clk	lpspi6_clk_root	LPCG73
	lpspi6.ipg_clk_s	bus_wakeup_clk_root	
	lpspi6.ipg_clk	bus_wakeup_clk_root	
lpspi7	lpspi7.lpspi_clk	lpspi7_clk_root	LPCG74
	lpspi7.ipg_clk_s	bus_wakeup_clk_root	
	lpspi7.ipg_clk	bus_wakeup_clk_root	
lpspi8	lpspi8.lpspi_clk	lpspi8_clk_root	LPCG75
	lpspi8.ipg_clk_s	bus_wakeup_clk_root	
	lpspi8.ipg_clk	bus_wakeup_clk_root	
lptmr1	lptmr1.ipg_clk_s	bus_aon_clk_root	LPCG42
	lptmr1.ipg_clk_irclk	lptmr1_clk_root	
	lptmr1.ipg_clk_1khz	32k_clk_root	
	lptmr1.ipg_clk_32khz	32k_clk_root	
	lptmr1.ipg_clk_erclk	32k_clk_root	
lptmr2	lptmr2.ipg_clk_irclk	lptmr2_clk_root	LPCG43
	lptmr2.ipg_clk_s	bus_wakeup_clk_root	
	lptmr2.ipg_clk_1khz	32k_clk_root	
	lptmr2.ipg_clk_32khz	32k_clk_root	
	lptmr2.ipg_clk_erclk	32k_clk_root	

Table continues on the next page...

Table 142. System Clocks Table (continued)

lpuart1	lpuart1.pclk	bus_aon_clk_root	LPCG52
	lpuart1.lpuart_clk	lpuart1_clk_root	
	lpuart1.lpuart_clk_n	lpuart1_clk_root	
lpuart2	lpuart2.pclk	bus_aon_clk_root	LPCG53
	lpuart2.lpuart_clk	lpuart2_clk_root	
	lpuart2.lpuart_clk_n	lpuart2_clk_root	
lpuart3	lpuart3.pclk	bus_wakeup_clk_root	LPCG54
	lpuart3.lpuart_clk	lpuart3_clk_root	
	lpuart3.lpuart_clk_n	lpuart3_clk_root	
lpuart4	lpuart4.pclk	bus_wakeup_clk_root	LPCG55
	lpuart4.lpuart_clk	lpuart4_clk_root	
	lpuart4.lpuart_clk_n	lpuart4_clk_root	
lpuart5	lpuart5.pclk	bus_wakeup_clk_root	LPCG56
	lpuart5.lpuart_clk	lpuart5_clk_root	
	lpuart5.lpuart_clk_n	lpuart5_clk_root	
lpuart6	lpuart6.pclk	bus_wakeup_clk_root	LPCG57
	lpuart6.lpuart_clk	lpuart6_clk_root	
	lpuart6.lpuart_clk_n	lpuart6_clk_root	
lpuart7	lpuart7.pclk	bus_wakeup_clk_root	LPCG58
	lpuart7.lpuart_clk	lpuart7_clk_root	
	lpuart7.lpuart_clk_n	lpuart7_clk_root	
lpuart8	lpuart8.pclk	bus_wakeup_clk_root	LPCG59
	lpuart8.lpuart_clk	lpuart8_clk_root	
	lpuart8.lpuart_clk_n	lpuart8_clk_root	
romcp	ipsync_romcp1.ipg_master_clk	bus_aon_clk_root	LPCG24
	ipsync_romcp1.ipg_slave_clk	m33_clk_root	
	romcp1.hclk	m33_clk_root	
	romcp1.hclk_bist	m33_clk_root	
	romcp1_mem.hclk	m33_clk_root	
	romcp1.hclk_reg	m33_clk_root	
sai1	sai1.ipg_clk	bus_aon_clk_root	LPCG81
	sai1.ipg_clk_s	bus_aon_clk_root	
	sai1.ipg_clk_sai_mclk	sai1_clk_root	

Table continues on the next page...

Table 142. System Clocks Table (continued)

sai2	sai2.ipg_clk_sai_mclk	sai2_clk_root	LPCG82
	sai2.ipg_clk	bus_wakeup_clk_root	
	sai2.ipg_clk_s	bus_wakeup_clk_root	
sai3	sai3.ipg_clk_sai_mclk	sai3_clk_root	LPCG83
	sai3.ipg_clk	bus_wakeup_clk_root	
	sai3.ipg_clk_s	bus_wakeup_clk_root	
sema1	sema1.clk	bus_aon_clk_root	LPCG17
sema2	sema2.clk	bus_wakeup_clk_root	LPCG18
aips1	aips1.hclk	bus_aon_clk_root	LPCG5
	aips1.ipg_clk	bus_aon_clk_root	
aips2	aips2.hclk	bus_wakeup_clk_root	LPCG4
	aips2.ipg_clk	bus_wakeup_clk_root	
aips3	aips3.hclk	bus_wakeup_clk_root	LPCG4
	aips3.ipg_clk	bus_wakeup_clk_root	
aips4	ahb_async_aips4.HCLKM	nic_apb_clk_root	LPCG103
	aips4.hclk	nic_apb_clk_root	
	aips4.ipg_clk	nic_apb_clk_root	
iomuxc	iomuxc_gpr.ipg_clk_s	bus_aon_clk_root	LPCG5
	iomuxc.ipg_clk_s	bus_aon_clk_root	
aonmix.ahbasync	ahbasync_bridge_aips1.HCLKM	bus_aon_clk_root	LPCG5
	ahbasync_bridge_aips1.HCLKS	m33_clk_root	
	ahbasync_slv_aon_s6.HCLKS	m33_clk_root	
	ahbasync_bridge_ELE_soc_out.HCLKM	m33_clk_root	
	ahbasync_bridge_ELE_soc_in.HCLKS	m33_clk_root	
cm33	cm33.FCLK_LP	m33_clk_root	LPCG5
cm33	cm33.ipg_clk_trdc	m33_clk_root	LPCG1
	cm33.ipg_clk_perfmon	m33_clk_root	
	cm33.PLAT_HCLK	m33_clk_root	
	cm33.FCLK	m33_clk_root	
	cm33.ATCLK	m33_clk_root	

Table continues on the next page...

Table 142. System Clocks Table (continued)

mu1	mu1.ipg_clk_mua	bus_aon_clk_root	LPCG19
	mu1.ipg_clk_s_mua	bus_aon_clk_root	
	mu1.ipg_clk_mub	bus_aon_clk_root	LPCG20
	mu1.ipg_clk_s_mub	bus_aon_clk_root	
mu2	mu2.ipg_clk_mua	bus_wakeup_clk_root	LPCG19
	mu2.ipg_clk_s_mua	bus_wakeup_clk_root	
	mu2.ipg_clk_mub	bus_wakeup_clk_root	LPCG20
	mu2.ipg_clk_s_mub	bus_wakeup_clk_root	
timer	sys_timer_9rt.CLK__sys_ctr1__ipg_clk	bus_aon_clk_root	LPCG122
	sys_timer_9rt.CLK__sys_ctr1__ipg_clk_s	bus_aon_clk_root	
tpm1	tpm1.lptpm_clk	bus_aon_clk_root	LPCG44
	tpm1.ipg_clk_s	bus_aon_clk_root	
	tpm1.ipg_clk	bus_aon_clk_root	
tpm2	tpm2.ipg_clk	bus_aon_clk_root	LPCG45
	tpm2.ipg_clk_s	bus_aon_clk_root	
	tpm2.lptpm_clk	tpm2_clk_root	
tpm3	tpm3.lptpm_clk	bus_wakeup_clk_root	LPCG46
	tpm3.ipg_clk_s	bus_wakeup_clk_root	
	tpm3.ipg_clk	bus_wakeup_clk_root	
tpm4	tpm4.lptpm_clk	tpm4_clk_root	LPCG47
	tpm4.ipg_clk	bus_wakeup_clk_root	
	tpm4.ipg_clk_s	bus_wakeup_clk_root	
tpm5	tpm5.lptpm_clk	tpm5_clk_root	LPCG48
	tpm5.ipg_clk	bus_wakeup_clk_root	
	tpm5.ipg_clk_s	bus_wakeup_clk_root	
tpm6	tpm6.lptpm_clk	tpm6_clk_root	LPCG49
	tpm6.ipg_clk	bus_wakeup_clk_root	
	tpm6.ipg_clk_s	bus_wakeup_clk_root	
wdog1	wdog1.ipg_clk_s	bus_aon_clk_root	LPCG12
	wdog1.ipg_clk_ungated	bus_aon_clk_root	
	wdog1.lpo_clk	32k_clk_root	
	wdog1.ipg_clk_gated	bus_aon_clk_root	

Table continues on the next page...

Table 142. System Clocks Table (continued)

	wdog1.int_clk	bus_aon_clk_root	
	wdog1.ext_clk	osc_24m	
wdog2	wdog2.ext_clk	osc_24m	LPCG13
	wdog2.ipg_clk_gated	bus_aon_clk_root	
	wdog2.ipg_clk_s	bus_aon_clk_root	
	wdog2.ipg_clk_ungated	bus_aon_clk_root	
	wdog2.int_clk	bus_aon_clk_root	
	wdog2.lpo_clk	32k_clk_root	
wdog3	wdog3.ipg_clk_gated	bus_wakeup_clk_root	LPCG14
	wdog3.ipg_clk_s	bus_wakeup_clk_root	
	wdog3.ipg_clk_ungated	bus_wakeup_clk_root	
	wdog3.int_clk	bus_wakeup_clk_root	
	wdog3.lpo_clk	32k_clk_root	
	wdog3.ext_clk	osc_24m	
wdog4	wdog4.ipg_clk_gated	bus_wakeup_clk_root	LPCG15
	wdog4.ipg_clk_s	bus_wakeup_clk_root	
	wdog4.ipg_clk_ungated	bus_wakeup_clk_root	
	wdog4.int_clk	bus_wakeup_clk_root	
	wdog4.lpo_clk	32k_clk_root	
	wdog4.ext_clk	osc_24m	
wdog5	wdog5.ipg_clk_gated	bus_wakeup_clk_root	LPCG16
	wdog5.ipg_clk_s	bus_wakeup_clk_root	
	wdog5.ipg_clk_ungated	bus_wakeup_clk_root	
	wdog5.int_clk	bus_wakeup_clk_root	
	wdog5.lpo_clk	32k_clk_root	
	wdog5.ext_clk	osc_24m	
gpio1	gpio1.hclk	m33_clk_root	LPCG34
gpio2	gpio2.hclk	bus_wakeup_clk_root	LPCG35
gpio3	gpio3.hclk	bus_wakeup_clk_root	LPCG36
gpio4	gpio4.hclk	bus_wakeup_clk_root	LPCG37
tstmr1	sys_timer_9rt.tstmr1.ipg_clk	bus_aon_clk_root	LPCG123
	sys_timer_9rt.tstmr1.ipg_clk_s	bus_aon_clk_root	
tstmr2	tstmr2.ipg_clk_s	bus_wakeup_clk_root	LPCG124

Table continues on the next page...

Table 142. System Clocks Table (continued)

	tstmr2.ipg_clk	bus_wakeup_clk_root	
pdm	pdm.ipg_clk_app_nonstop	pdm_clk_root	LPCG107
	pdm.ipg_clk_app	pdm_clk_root	
dbg_trace	dbg_trace.swo_TRACECLKIN	swo_trace_clk_root	LPCG32
	ahb2apb_dbg_trace.ahb_clk	bus_wakeup_clk_root	LPCG31
	dbg_trace.dapclk	bus_wakeup_clk_root	
	dbg_trace.hclk_ispap	bus_wakeup_clk_root	
	dbg_trace.csclk	wakeup_axi_clk_root	
enet (Ethernet MAC)	enet.ipp_ind_mac0_txclk	enet_ref_clk_root	LPCG120
	enet.ipg_clk_time	enet_timer1_clk_root	
	enet.ipg_clk_mac0_s	wakeup_axi_clk_root	
	enet.ipg_clk	wakeup_axi_clk_root	
	enet.ipg_clk_s	wakeup_axi_clk_root	
	enet.ipg_clk_mac0	wakeup_axi_clk_root	
enet_qos (Ethernet Controller with TSN)	CLK__ccm_enet_qos_clock_generate__enet_qos_clk	enet_clk_root	LPCG121
	enet_qos.clk_ptp_ref_i	enet_timer2_clk_root	
	enet_qos.clk_csr_i	wakeup_axi_clk_root	
	enet_qos.aclk_i	wakeup_axi_clk_root	
flexio1	flexio1.flexio_clk	flexio1_clk_root	LPCG38
	flexio1.pclk	bus_wakeup_clk_root	
flexio2	flexio2.flexio_clk	flexio2_clk_root	LPCG39
	flexio2.pclk	bus_wakeup_clk_root	
flexspi	flexspi1.hclk	wakeup_axi_clk_root	LPCG68
	flexspi1_mem.hclk	wakeup_axi_clk_root	
	flexspi1.ipg_clk	wakeup_axi_clk_root	
	flexspi1.ipg_clk_sfck	flexspi1_clk_root	
mqs1	mqs1.hmclk	sai1_clk_root	LPCG108
	mqs1.bclk	sai1_clk_root	
mqs2	mqs2.hmclk	sai3_clk_root	LPCG109
	mqs2.bclk	sai3_clk_root	
usdhc1	usdhc1.ipp_dft8_clk	usdhc1_clk_root	LPCG78
	usdhc1.ipp_dft7_clk	usdhc1_clk_root	

Table continues on the next page...

Table 142. System Clocks Table (continued)

	usdhc1.ipp_dft6_clk	usdhc1_clk_root	
	usdhc1.ipp_dft4_clk	mipi_test_byte_clk_root	
	usdhc1.ipp_dft3_clk	mipi_test_byte_clk_root	
	usdhc1.ipp_dft2_clk	mipi_test_byte_clk_root	
	usdhc1.ipp_dft1_clk	mipi_test_byte_clk_root	
	usdhc1.ipg_clk_s	bus_wakeup_clk_root	
	usdhc1.ipg_clk	bus_wakeup_clk_root	
	usdhc1.ipg_clk_perclk	usdhc1_clk_root	
	usdhc1.hclk	wakeup_axi_clk_root	
usdhc2	usdhc2.ipp_dft8_clk	usdhc2_clk_root	LPCG79
	usdhc2.ipp_dft7_clk	usdhc2_clk_root	
	usdhc2.ipp_dft6_clk	usdhc2_clk_root	
	usdhc2.ipp_dft4_clk	mipi_test_byte_clk_root	
	usdhc2.ipp_dft3_clk	mipi_test_byte_clk_root	
	usdhc2.ipp_dft2_clk	mipi_test_byte_clk_root	
	usdhc2.ipp_dft1_clk	mipi_test_byte_clk_root	
	usdhc2.ipg_clk_s	bus_wakeup_clk_root	
	usdhc2.ipg_clk	bus_wakeup_clk_root	
	usdhc2.ipg_clk_perclk	usdhc2_clk_root	
	usdhc2.hclk	wakeup_axi_clk_root	
usdhc3	usdhc3.ipp_dft8_clk	usdhc3_clk_root	LPCG79
	usdhc3.ipp_dft7_clk	usdhc3_clk_root	
	usdhc3.ipp_dft6_clk	usdhc3_clk_root	
	usdhc3.ipp_dft4_clk	mipi_test_byte_clk_root	
	usdhc3.ipp_dft3_clk	mipi_test_byte_clk_root	
	usdhc3.ipp_dft2_clk	mipi_test_byte_clk_root	
	usdhc3.ipp_dft1_clk	mipi_test_byte_clk_root	
	usdhc3.ipg_clk_s	bus_wakeup_clk_root	
	usdhc3.ipg_clk	bus_wakeup_clk_root	
	usdhc3.ipg_clk_perclk	usdhc3_clk_root	
	usdhc3.hclk	wakeup_axi_clk_root	
ADC	adc.ipg_clk	adc_clk_root	LPCG11
ca55mix.trout	ca55mix.trout_wrapper.pclk	arm_a55_mtr_bus_clk_root	LPCG2

Table continues on the next page...

Table 142. System Clocks Table (continued)

ddr_apb	apbm.pclkm	dram_apb_clk_root	LPCG98
	ddr_bsd_rst.apbclk	dram_apb_clk_root	
	apb2ips_ddrc.clk	dram_apb_clk_root	
	ddr_top.ipg_clk_skyblue	dram_apb_clk_root	
	ddr_ss_reg_blk.pclk	dram_apb_clk_root	
	apb2ips_mtr.clk	dram_apb_clk_root	
	ddrmix_mbist_part.mtr_bus_clk	dram_apb_clk_root	
ddrmix	ddr_top.BypassPclk	dram_alt_clk_root	LPCG97
	ddr_top.DfiCtlClk	dram_alt_clk_root	LPCG94
nicmix	nicmix.ahb2apb_ddrmix.ahb_clk	nic_clk_root	LPCG101
	nicmix.apb_async_slv_ddrmix.pclks	nic_clk_root	
	nicmix.ahb2axi_hsiomix.CLK	nic_clk_root	
	nicmix.ahb2axi_mediamix.CLK	nic_clk_root	
	nicmix.ahb_async_aips4.HCLKS	nic_clk_root	
	nicmix.axi_limit_wakeupmix.clk	nic_clk_root	
	nicmix.dac_cache_r.hclk	nic_clk_root	
	nicmix.dac_cache_r.ipg_clk	nic_clk_root	
	nicmix.dac_cache_w.hclk	nic_clk_root	
	nicmix.dac_cache_w.ipg_clk	nic_clk_root	
	nicmix.dac_mlmix0.hclk	nic_clk_root	
	nicmix.dac_mlmix0.ipg_clk	nic_clk_root	
	nicmix.dac_mlmix1.hclk	nic_clk_root	
	nicmix.dac_mlmix1.ipg_clk	nic_clk_root	
	nicmix.dexsc_ocram.aclk_exsc	nic_clk_root	
	nicmix.dexsc_ocram.ipg_clk	nic_clk_root	
	nicmix.nic400_central.mainclk	nic_clk_root	
	nicmix.nic400_central.mainclk_r	nic_clk_root	
	nicmix.mecc256_ocram.clk	nic_clk_root	

Table continues on the next page...

Table 142. System Clocks Table (continued)

nicmix.mecc256	nicmix.mecc256_ocram.ipg_clk	nic_apb_clk_root	LPCG111
	nicmix.mecc256_ocram.ipg_clk_s	nic_apb_clk_root	
nicmix.ssi	nicmix.ssi_mst_ddrmix.clk	nic_clk_root	LPCG117
	nicmix.ssi_slv_hsiomix.clk	nic_clk_root	
	nicmix.ssi_mst_wakeupmix.clk	nic_clk_root	
	nicmix.ssi_mst_gic600.clk	nic_clk_root	
	nicmix.ssi_slv_mlmix1.clk	nic_clk_root	
	nicmix.ssi_slv_mlmix0.clk	nic_clk_root	
	nicmix.ssi_slv_mediamix.clk	nic_clk_root	
	nicmix.ssi_slv_gic600.clk	nic_clk_root	
	nicmix.ssi_slv_cache.clk	nic_clk_root	
	nicmix.ssi_slv_wakeupmix.clk	nic_clk_root	
pxp	mediamix.pxp.async_apb_clk	media_apb_clk_root	LPCG90
	mediamix.pxp.clk	media_axi_clk_root	
isi	d_ip_isi_syn.apb_clk	media_apb_clk_root	LPCG91
	d_ip_isi_syn.ipg_proc_ungated_clk	media_axi_clk_root	
	d_ip_isi_syn.axi_ungated_clk	media_axi_clk_root	
	d_ip_isi_syn.ipt_test_mode_clk	media_axi_clk_root	
lcdif	lcdif_1.apb_clk	media_apb_clk_root	LPCG89
	lcdif_1.b_clk	media_axi_clk_root	
	lcdif_1.pix_clk	media_disp_pix_clk_root	
ldb	ldb.ch_0_serial_clk	media_ldb_clk_root	LPCG88
	ldb.ch_1_serial_clk	media_ldb_clk_root	
mipi_csi2	mipi_csi2_host_inst.pclk	media_apb_clk_root	LPCG86
	mipi_csi2_host_inst.pixclk	cam_pix_clk_root	
dsi	mediamix.dsi_pix_mem.rclk	media_apb_clk_root	LPCG87
	mediamix.dsi_pld_mem.wclk	media_apb_clk_root	
	mediamix.mipi_dsi_host_inst.pclk	media_apb_clk_root	
	mediamix.mipi_dsi_host_inst.refclk	osc_24m	

Table continues on the next page...

Table 142. System Clocks Table (continued)

ML	mlmix.npu_core.CLK	ml_clk_root	LPCG113
	mlmix.ssi_axim1.clk	ml_clk_root	
	mlmix.ssi_axim0.clk	ml_clk_root	
	mlmix.pl301_axi2apb.mainclk	ml_clk_root	
USB_Controller	U_USB_Controller.ipg_ahb_clk	hsiomix_clk_root	LPCG104
	U_USB_Controller.ipg_clk_s	hsiomix_clk_root	LPCG105
	U_USB_Controller.ipg_clk_s_pl301	hsiomix_clk_root	
	U_USB_Controller.ipg_clk_32khz	32k_clk_root	LPCG119
CSI Parallel camera	mipi_csi2_host_inst.pixclk	cam_pix_clk_root	LPCG86

30.3 Functional description

CCM is comprised of several functional blocks:

- Clock Root
- Clock Gate
- Clock Source
- General Purpose Registers (GPR)

These functions can be configured under various CCM modes. The first section will describe the ways to configure these functions under different CCM modes. And the other sections will describe these functions independently.

30.3.1 Configuration of Functions

CCM functional blocks support the following modes:

- Direct Configuration mode
- CPU Low Power mode
- Auto mode (only for clock sources)

The default CCM mode is Direct Configuration mode.

30.3.2 Clock Sources

Clock roots are generated from various clock sources, which can be either oscillators or PLLs. Clock source is implemented to control PLLs and oscillators, to be automatically turned OFF or ON on system low power actions.

The following table shows the clock sources for the CCM:

Table 143. Clock Sources

Control Channel	Clock Source	Description
OSCPLL0	OSC_24M_CLK	OSC 24MHz CLK

Table continues on the next page...

Table 143. Clock Sources (continued)

Control Channel	Clock Source	Description
OSCPLL1	ARM PLL	ARM PLL VCO (Not connected to the clock tree)
OSCPLL2	ARM PLL CLK	ARM PLL output clock
OSCPLL3	SYSTEM PLL	SYSTEM PLL VCO (Not connected to the clock tree)
OSCPLL4	SYSTEM PLL PFD0	SYSTEM PLL PFD0 (OSCPLL channel can reset SYSTEM PLL PFD. It handles DFS[ENABLE].)
OSCPLL5	SYSTEM PLL PFD0 CLK	SYSTEM PLL PFD0 clock (OSCPLL channel can gate off SYSTEM PLL PFD0 CLK. It handles DFS[CLKOUT_EN].)
OSCPLL6	SYSTEM PLL PFD0 DIV2 CLK	SYSTEM PLL PFD0 divided by 2 clock
OSCPLL7	SYSTEM PLL PFD1	SYSTEM PLL PFD1 (OSCPLL channel can reset SYSTEM PLL PFD. It handles DFS[ENABLE].)
OSCPLL8	SYSTEM PLL PFD1 CLK	SYSTEM PLL PFD1 clock (OSCPLL channel can gate off SYSTEM PLL PFD0 CLK. It handles DFS[CLKOUT_EN].)
OSCPLL9	SYSTEM PLL PFD1 DIV2 CLK	SYSTEM PLL PFD1 divided by 2 clock
OSCPLL10	SYSTEM PLL PFD2	SYSTEM PLL PFD2 (OSCPLL channel can reset SYSTEM PLL PFD. It handles DFS[ENABLE].)
OSCPLL11	SYSTEM PLL PFD2 CLK	SYSTEM PLL PFD2 clock (OSCPLL channel can gate off SYSTEM PLL PFD0 CLK. It handles DFS[CLKOUT_EN].)
OSCPLL12	SYSTEM PLL PFD2 DIV2 CLK	SYSTEM PLL PFD2 divided by 2 clock
OSCPLL13	AUDIO PLL	AUDIO PLL VCO (Not connected to the clock tree)
OSCPLL14	AUDIO PLL CLK	AUDIO PLL output clock
OSCPLL15	DRAM PLL	DRAM PLL VCO (Not connected to the clock tree)
OSCPLL16	DRAM PLL CLK	DRAM PLL output clock
OSCPLL17	VIDEO PLL	VIDEO PLL VCO (Not connected to the clock tree)
OSCPLL18	VIDEO PLL CLK	VIDEO PLL output clock

NOTE

The clock source calculates whether the clock is required for each CPU domain. If clock source is not needed for any domain, clock will be shutdown.

Clock Source blocks have 3 work mode: Direct Control mode, CPULPM mode and Auto mode. Direct Control mode is the default mode. After reset, all Clock Source blocks work in Direct Control mode.

All registers are readable for any domain in any mode, but write access is restricted for protection. Trustzone unauthenticated access will be ignored.

The OSCPLLn_AUTHEN and OSCPLLn_DIRECT registers can be changed in any mode, while the write access should be from the domain granted by OSCPLLn_AUTHEN[WHITE_LIST] and Trustzone authenticated. The default value of OSCPLLn_AUTHEN[WHITE_LIST] is 0xFFFF, which means all 16 domains are granted.

The OSCPLLn_LPM_CUR registers can only be written in CPULPM mode, otherwise the write access will be ignored. And the write access in CPULPM mode should also comply with the Trustzone access control.

In Direct Control mode, the ON/OFF of clock source is determined by OSCPLLn_DIRECT[ON] directly. The default state of OSCPLL is ON.

In CPULPM mode, the ON/OFF of clock source is dependent on the settings in OSCPLLn_LPMs and the current mode every domain is in. There are 3-bit clock dependent level settings for each domain respectively. The bigger the 3-bit value is, the stronger the clock is dependent. Valid values are as follows:

- 0: this clock is not needed in any mode
- 1: this clock is needed in RUN mode, but not needed in WAIT, STOP, SESPEND mode
- 2: this clock is needed in RUN and WAIT mode, but not needed in STOP, SUSPEND mode
- 3: this clock is needed in RUN, WAIT and STOP mode, but not needed in SUSPEND mode
- 4: this clock is needed in RUN, WAIT, STOP and SUSPEND mode

NOTE

The mode above means the mode of domain (CPU platform) instead of CCM mode. There are 4 CPU mode: RUN, WAIT, STOP and SUSPEND. If the CPU is in SUSPEND mode, all clocks are treated as NOT needed.

For PLLs which are used as input to CCM for use as clock roots by the SoC (this in general includes all PLLs in ANAMIX but not SoC-specific PLLs elsewhere in the SoC which are only used by one MIX or PHY), an AUTO_CTRL feature is supported in the CCM. When AUTO_CTRL is enabled, if a PLL is not selected by any CCM clock root, then that PLL will be disabled automatically. Similarly, if a PLL has multiple PFD outputs and a given PFD output is not selected by any CCM clock root, then that PFD output will be disabled automatically.

NOTE

To enable OSCPLL AUTO mode , you need to

1. Enable HW control in OSCPLL by setting CTRL[HW_CTRL_SEL]
2. Enable AUTOMODE in the OSCPLL CCM by setting OSCPLLa_AUTHEN[AUTO_CTRL] to 0x1
3. Set OSCPLLa_DIRECT[ON] to 0x0

30.3.3 Clock Root

CCM clock root generation contains multiple clock root channels. All clock roots are asynchronous each other, even all the clock root setting are the same.

Each clock channel contains an 4-to-1 MUX, and a 8-bit divider. The clock MUX selects 1 clock out of 4 clock inputs. The 8-bit divider can divide selected clock by up to 256. The clock output of the clock channel can be gated off.

The clock root channel setting can be changed from any value to any value, and at any time. It is also possible to change the clock setting when the clock root channel is gating off. If more than one setting is changed at the same time, internal logic will use a

procedure to change the setting. The procedure will make sure the clock output is not faster than the current clock root setting or target clock root setting. If the application does care about the clock during transition, software will need to change the setting one field at a time. This typically happens when a clock must be faster than the given frequency. In this case, the application needs to change each field one at a time and wait for the field to take effect before changing the next field.

Although a clock root channel is designed to be able to switch clocks regardless of clock input status. When the current selected clock input is off, the application needs to avoid changing the clock MUX after the clock input is turned on. This will lead to unstable behavior for the clock root and loading peripherals. To avoid this scenario, the application can either switch the clock MUX before turning on the clock input, or wait until the clock input is stable before changing the MUX option.

The division fractor of the divider is equal to CLOCK_ROOTn_CONTROL[DIV]+1.

All registers are readable for any domain in any mode, but write access are restricted. Unauthenticated write access will be ignored. Only the access from the domains on the whitelist can change the value of CLOCK_ROOTn_CONTROL register, after passing trustzone authentication.

NOTE

LPIT1, TPM1 is clocked by AON bus and LPIT2, TPM3 is clocked by WAKEUP bus. LPIT1, TPM1 and LPIT2, TPM3 have their trigger output 0 used as input by CAN1 and CAN2 respectively. Because of this, these instances are clocked by the bus clock - the same clock used to clock the respective CAN module (bus_aon_clk_root and bus_wakeup_clk_root respectively) in order to ensure that these signals are timed synchronously between these modules.

Table 144. Clock Root Table

Slice Num	Clock Root	Max Freq UD (MHz)	Max Freq NM (MHz)	Max Freq OD (MHz)	Offset	Source Select
0	arm_a55_periph_clk_root	200	333.33	400	0x0000	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-SYS_PLL_PFD2
1	arm_a55_mtr_bus_clk_root	133.33	133.33	133.33	0x0080	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
2	arm_a55_clk_root	500	800	1000	0x0100	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-SYS_PLL_PFD2
3	m33_clk_root	133.33	200	250	0x0180	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
4	ele_clk_root	133.33	200	250	0x0200	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2

Table continues on the next page...

Table 144. Clock Root Table (continued)

Slice Num	Clock Root	Max Freq UD (MHz)	Max Freq NM (MHz)	Max Freq OD (MHz)	Offset	Source Select
						10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
5	bus_wakeup_clk_root	133.33	133.33	133.33	0x0280	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
6	bus_aon_clk_root	133.33	133.33	133.33	0x0300	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
7	wakeup_axi_clk_root	200	312.5	400	0x0380	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-SYS_PLL_PFD2
8	swo_trace_clk_root	133.33	133.33	133.33	0x0400	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
9	m33_systick_clk_root	24	24	24	0x0480	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
10	flexio1_clk_root	80	80	80	0x0500	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
11	flexio2_clk_root	80	80	80	0x0580	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
14	lptmr1_clk_root	80	80	80	0x0700	00-OSC_24M_CLK

Table continues on the next page...

Table 144. Clock Root Table (continued)

Slice Num	Clock Root	Max Freq UD (MHz)	Max Freq NM (MHz)	Max Freq OD (MHz)	Offset	Source Select
						01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
15	lptmr2_clk_root	80	80	80	0x0780	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
17	tpm2_clk_root	83.33	83.33	83.33	0x0880	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-AUDIO_PLL_CLK 11-EXT_CLK
19	tpm4_clk_root	83.33	83.33	83.33	0x0980	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-AUDIO_PLL_CLK 11-EXT_CLK
20	tpm5_clk_root	83.33	83.33	83.33	0x0a00	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-AUDIO_PLL_CLK 11-EXT_CLK
21	tpm6_clk_root	83.33	83.33	83.33	0x0a80	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-AUDIO_PLL_CLK 11-EXT_CLK
22	flexspi1_clk_root	266.66	400	400	0x0b00	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-SYS_PLL_PFD2
23	can1_clk_root	80	80	80	0x0b80	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK

Table continues on the next page...

Table 144. Clock Root Table (continued)

Slice Num	Clock Root	Max Freq UD (MHz)	Max Freq NM (MHz)	Max Freq OD (MHz)	Offset	Source Select
24	can2_clk_root	80	80	80	0x0c00	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
25	lpuart1_clk_root	133.33	133.33	133.33	0x0c80	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
26	lpuart2_clk_root	133.33	133.33	133.33	0x0d00	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
27	lpuart3_clk_root	133.33	133.33	133.33	0x0d80	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
28	lpuart4_clk_root	133.33	133.33	133.33	0x0e00	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
29	lpuart5_clk_root	133.33	133.33	133.33	0x0e80	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
30	lpuart6_clk_root	133.33	133.33	133.33	0x0f00	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
31	lpuart7_clk_root	133.33	133.33	133.33	0x0f80	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2

Table continues on the next page...

Table 144. Clock Root Table (continued)

Slice Num	Clock Root	Max Freq UD (MHz)	Max Freq NM (MHz)	Max Freq OD (MHz)	Offset	Source Select
						11-VIDEO_PLL_CLK
32	lpuart8_clk_root	133.33	133.33	133.33	0x1000	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
33	lpi2c1_clk_root	133.33	133.33	133.33	0x1080	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
34	lpi2c2_clk_root	133.33	133.33	133.33	0x1100	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
35	lpi2c3_clk_root	133.33	133.33	133.33	0x1180	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
36	lpi2c4_clk_root	133.33	133.33	133.33	0x1200	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
37	lpi2c5_clk_root	133.33	133.33	133.33	0x1280	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
38	lpi2c6_clk_root	133.33	133.33	133.33	0x1300	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
39	lpi2c7_clk_root	133.33	133.33	133.33	0x1380	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2

Table continues on the next page...

Table 144. Clock Root Table (continued)

Slice Num	Clock Root	Max Freq UD (MHz)	Max Freq NM (MHz)	Max Freq OD (MHz)	Offset	Source Select
						10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
40	lpi2c8_clk_root	133.33	133.33	133.33	0x1400	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
41	lpspi1_clk_root	133.33	133.33	133.33	0x1480	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
42	lpspi2_clk_root	133.33	133.33	133.33	0x1500	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
43	lpspi3_clk_root	133.33	133.33	133.33	0x1580	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
44	lpspi4_clk_root	133.33	133.33	133.33	0x1600	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
45	lpspi5_clk_root	133.33	133.33	133.33	0x1680	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
46	lpspi6_clk_root	133.33	133.33	133.33	0x1700	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
47	lpspi7_clk_root	133.33	133.33	133.33	0x1780	00-OSC_24M_CLK

Table continues on the next page...

Table 144. Clock Root Table (continued)

Slice Num	Clock Root	Max Freq UD (MHz)	Max Freq NM (MHz)	Max Freq OD (MHz)	Offset	Source Select
						01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
48	lpspi8_clk_root	133.33	133.33	133.33	0x1800	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
49	i3c1_clk_root	133.33.33	133.33	133.33	0x1880	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
50	i3c2_clk_root	133.33	133.33	133.33	0x1900	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
51	usdhc1_clk_root	266.67	400	400	0x1980	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-SYS_PLL_PFD2
52	usdhc2_clk_root	266.67	400	400	0x1a00	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-SYS_PLL_PFD2
53	usdhc3_clk_root	266.67	400	400	0x1a80	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-SYS_PLL_PFD2
54	sai1_clk_root	66.67	66.67	66.67	0x1b00	00-OSC_24M_CLK 01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-EXT_CLK

Table continues on the next page...

Table 144. Clock Root Table (continued)

Slice Num	Clock Root	Max Freq UD (MHz)	Max Freq NM (MHz)	Max Freq OD (MHz)	Offset	Source Select
55	sai2_clk_root	66.67	66.67	66.67	0x1b80	00-OSC_24M_CLK 01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-EXT_CLK
56	sai3_clk_root	66.67	66.67	66.67	0x1c00	00-OSC_24M_CLK 01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-EXT_CLK
57	ccm_cko1_clk_root	266.67	266.67	266.67	0x1c80	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-AUDIO_PLL_CLK
58	ccm_cko2_clk_root	266.67	266.67	266	0x1d00	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-VIDEO_PLL_CLK
59	ccm_cko3_clk_root	266.67	266.67	266	0x1d80	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-AUDIO_PLL_CLK
60	ccm_cko4_clk_root	266.67	266.67	266	0x1e00	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-VIDEO_PLL_CLK
61	hsio_clk_root	133.33	133.33	133.33	0x1e80	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
62	hsio_usb_test_60m_clk_root	62.5	62.5	62.5	0x1f00	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2

Table continues on the next page...

Table 144. Clock Root Table (continued)

Slice Num	Clock Root	Max Freq UD (MHz)	Max Freq NM (MHz)	Max Freq OD (MHz)	Offset	Source Select
						11-VIDEO_PLL_CLK
63	hsio_acscan_80m_clk_root	80	80	80	0x1f80	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
64	hsio_acscan_480m_clk_root	-	-	480	0x2000	00-OSC_24M_CLK 01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-SYS_PLL_PFD2
65	nic_clk_root	250	400	500	0x2080	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-SYS_PLL_PFD2
66	nic_apb_clk_root	133.33	133.33	133.33	0x2100	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
67	ml_apb_clk_root	133.33	133.33	133.33	0x2180	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
68	ml_clk_root	500	800	1000	0x2200	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-SYS_PLL_PFD2
69	media_axi_clk_root	200	333.33	400	0x2280	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-SYS_PLL_PFD2
70	media_apb_clk_root	125	125	125	0x2300	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2

Table continues on the next page...

Table 144. Clock Root Table (continued)

Slice Num	Clock Root	Max Freq UD (MHz)	Max Freq NM (MHz)	Max Freq OD (MHz)	Offset	Source Select
						10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
71	media_ldb_clk_root	560	560	560	0x2380	00-OSC_24M_CLK 01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-SYS_PLL_PFD0
72	media_disp_pix_clk_root	140	200	200	0x2400	00-OSC_24M_CLK 01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-SYS_PLL_PFD0
73	cam_pix_clk_root	140	200	200	0x2480	00-OSC_24M_CLK 01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-SYS_PLL_PFD0
75	mipi_phy_cfg_clk_root	24	24	24	0x2580	00-OSC_24M_CLK 01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-SYS_PLL_PFD0
76	dram_alt_clk_root	625	625	625	0x2600	00-OSC_24M_CLK 01-SYS_PLL_PFD0 10-SYS_PLL_PFD1 11-SYS_PLL_PFD2
77	dram_apb_clk_root	133.33	133.33	133.33	0x2680	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-SYS_PLL_PFD2_DIV2
78	adc_clk_root	80	80	80	0x2700	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
79	pdm_clk_root	133.33	200	200	0x2780	00-OSC_24M_CLK

Table continues on the next page...

Table 144. Clock Root Table (continued)

Slice Num	Clock Root	Max Freq UD (MHz)	Max Freq NM (MHz)	Max Freq OD (MHz)	Offset	Source Select
						01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-EXT_CLK
80	tstmr1_clk_root	24	24	24	0x2800	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
81	tstmr2_clk_root	24	24	24	0x2880	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
82	mqs1_clk_root	66.67	66.67	66.67	0x2900	00-OSC_24M_CLK 01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-EXT_CLK
83	mqs2_clk_root	66.67	66.67	66.67	0x2980	00-OSC_24M_CLK 01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-EXT_CLK
84	audio_xcvr_clk_root	200	200	200	0x2a00	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-SYS_PLL_PFD2_DIV2
85	spdif_clk_root	66.67	66.67	66.67	0x2a80	00-OSC_24M_CLK 01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-EXT_CLK
86	enet_clk_root	250	250	250	0x2b00	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-SYS_PLL_PFD2_DIV2

Table continues on the next page...

Table 144. Clock Root Table (continued)

Slice Num	Clock Root	Max Freq UD (MHz)	Max Freq NM (MHz)	Max Freq OD (MHz)	Offset	Source Select
87	enet_timer1_clk_root	100	100	100	0x2b80	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
88	enet_timer2_clk_root	100	100	100	0x2c00	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
89	enet_ref_clk_root	250	250	250	0x2c80	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-SYS_PLL_PFD2_DIV2
91	i3c1_slow_clk_root	24	24	24	0x2d80	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
92	i3c2_slow_clk_root	24	24	24	0x2e00	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
93	usb_phy_burunin_clk_root	50	50	50	0x2e80	00-OSC_24M_CLK 01-SYS_PLL_PFD0_DIV2 10-SYS_PLL_PFD1_DIV2 11-VIDEO_PLL_CLK
94	pal_cam_scan_clk_root	80	80	80	0x2f00	00-OSC_24M_CLK 01-AUDIO_PLL_CLK 10-VIDEO_PLL_CLK 11-SYS_PLL_PFD2

30.3.4 Clock Gate

Before a clock root goes to on-chip peripherals, the clock roots go through the IPC_CLK. These IPC_CLK are implemented to perform automatic clock gating when a domain enters and leaves low-power states. The IPC_CLKs are mapped to peripherals

so software can be simplified. IPC_CLKs are also implemented to control peripheral functions, and different IPC_CLKs could interface different clocks of the peripheral.

Clock gates have 2 work modes: Direct Control mode and CPULPM mode. Direct Control mode is the implicit mode. After reset, all clock gates work in Direct Control mode.

All registers are readable for any domain in any mode, but write access is restricted for protection. Trustzone unauthenticated access will be ignored.

The LPCGn_AUTHEN and LPCGn_DIRECT registers can be changed in any mode, while the write access should be from the domain granted by LPCGn_AUTHEN[WHITE_LIST] and Trustzone authenticated. The default value of LPCGn_AUTHEN[WHITE_LIST] is 0xFFFF, which means all 16 domains are granted.

The LPCGn_LPM_CUR registers can only be written in CPULPM mode, otherwise the write access will be ignored. And the write access in CPULPM mode should also comply with the Trustzone access control.

In Direct Control mode, the ON/OFF of LPCG is determined by LPCGn_DIRECT[ON] directly. The default state of LPCG is ON.

In CPULPM mode, the ON/OFF of LPCG is dependent on the settings in LPCGn_LPMs and the current mode every domain is in. There are 3-bit clock dependent level settings for each domain respectively. The bigger the 3-bit value is, the stronger the clock is dependent. Valid values are as follows:

- 0: this clock is not needed in any mode
- 1: this clock is needed in RUN mode, but not needed in WAIT, STOP, SUSPEND mode
- 2: this clock is needed in RUN and WAIT mode, but not needed in STOP, SUSPEND mode
- 3: this clock is needed in RUN, WAIT, STOP, SUSPEND mode

NOTE

The mode above means the mode of domain (CPU platform) instead of CCM mode. There are 4 CPU mode: RUN, WAIT, STOP and SUSPEND. If the CPU is in SUSPEND mode, all clocks are treated as NOT needed.

In CPULPM mode, LPCG will calculate whether the clock is required by any CPU domain. If clock is not needed for any domain, the clock will be shutdown.

NOTE

LPCG3 gates ele_clk_root for EdgeLock Secure Enclave.

NOTE

The tstmr1_clk_root, tstmr2_clk_root, mqs1_clk_root, mqs2_clk_root are not used.

Table 145. Clock Gate Table

Gating Channel	Generated Gate Signal
LPCG0	clk_enable_a55_ch
LPCG1	clk_enable_cm33_ch
LPCG2	clk_enable_arm_trout_ch
LPCG3	clk_enable_ele_ch
LPCG4	clk_enable_sim_wakeup_ch
LPCG5	clk_enable_sim_aon_ch
LPCG6	clk_enable_sim_mega_ch
LPCG7	clk_enable_anadig_ch

Table continues on the next page...

Table 145. Clock Gate Table (continued)

Gating Channel	Generated Gate Signal
LPCG8	clk_enable_src_ch
LPCG9	clk_enable_ccm_ch
LPCG10	clk_enable_gpc_ch
LPCG11	clk_enable_adc1_ch
LPCG12	clk_enable_wdog1_ch
LPCG13	clk_enable_wdog2_ch
LPCG14	clk_enable_wdog3_ch
LPCG15	clk_enable_wdog4_ch
LPCG16	clk_enable_wdog5_ch
LPCG17	clk_enable_sema1_ch
LPCG18	clk_enable_sema2_ch
LPCG19	clk_enable_mu_a_ch
LPCG20	clk_enable_mu_b_ch
LPCG21	clk_enable_edma3_1_ch
LPCG22	clk_enable_edma4_2_ch
LPCG23	clk_enable_romcp_a55_ch
LPCG24	clk_enable_romcp_m33_ch
LPCG25	clk_enable_flexspi1_ch
LPCG26	clk_enable_aon_trdc_ch
LPCG27	clk_enable_wkup_trdc_ch
LPCG28	clk_enable_ocotp_ch
LPCG29	clk_enable_bbsm_hp_ch
LPCG30	clk_enable_bbsm_ch
LPCG31	clk_enable_cstrace_ch
LPCG32	clk_enable_csswo_ch
LPCG33	clk_enable_iomuxc_ch
LPCG34	clk_enable_gpio1_ch
LPCG35	clk_enable_gpio2_ch
LPCG36	clk_enable_gpio3_ch
LPCG37	clk_enable_gpio4_ch
LPCG38	clk_enable_flexio1_ch
LPCG39	clk_enable_flexio2_ch

Table continues on the next page...

Table 145. Clock Gate Table (continued)

Gating Channel	Generated Gate Signal
LPCG40	clk_enable_lpit1_ch
LPCG41	clk_enable_lpit2_ch
LPCG42	clk_enable_lptmr1_ch
LPCG43	clk_enable_lptmr2_ch
LPCG44	clk_enable_tpm1_ch
LPCG45	clk_enable_tpm2_ch
LPCG46	clk_enable_tpm3_ch
LPCG47	clk_enable_tpm4_ch
LPCG48	clk_enable_tpm5_ch
LPCG49	clk_enable_tpm6_ch
LPCG50	clk_enable_can1_ch
LPCG51	clk_enable_can2_ch
LPCG52	clk_enable_lpuart1_ch
LPCG53	clk_enable_lpuart2_ch
LPCG54	clk_enable_lpuart3_ch
LPCG55	clk_enable_lpuart4_ch
LPCG56	clk_enable_lpuart5_ch
LPCG57	clk_enable_lpuart6_ch
LPCG58	clk_enable_lpuart7_ch
LPCG59	clk_enable_lpuart8_ch
LPCG60	clk_enable_lpi2c1_ch
LPCG61	clk_enable_lpi2c2_ch
LPCG62	clk_enable_lpi2c3_ch
LPCG63	clk_enable_lpi2c4_ch
LPCG64	clk_enable_lpi2c5_ch
LPCG65	clk_enable_lpi2c6_ch
LPCG66	clk_enable_lpi2c7_ch
LPCG67	clk_enable_lpi2c8_ch
LPCG68	clk_enable_lpspi1_ch
LPCG69	clk_enable_lpspi2_ch
LPCG70	clk_enable_lpspi3_ch
LPCG71	clk_enable_lpspi4_ch

Table continues on the next page...

Table 145. Clock Gate Table (continued)

Gating Channel	Generated Gate Signal
LPCG72	clk_enable_lpspi5_ch
LPCG73	clk_enable_lpspi6_ch
LPCG74	clk_enable_lpspi7_ch
LPCG75	clk_enable_lpspi8_ch
LPCG76	clk_enable_i3c1_ch
LPCG77	clk_enable_i3c2_ch
LPCG78	clk_enable_usdhc1_ch
LPCG79	clk_enable_usdhc2_ch
LPCG80	clk_enable_usdhc3_ch
LPCG81	clk_enable_sai1_ch
LPCG82	clk_enable_sai2_ch
LPCG83	clk_enable_sai3_ch
LPCG84	clk_enable_ssi_w2ao_ch
LPCG85	clk_enable_ssi_ao2w_ch
LPCG86	clk_enable_mipi_csi_ch
LPCG87	clk_enable_mipi_dsi_ch
LPCG88	clk_enable_lvds_ch
LPCG89	clk_enable_lcdif_ch
LPCG90	clk_enable_pxp_ch
LPCG91	clk_enable_isi_ch
LPCG92	clk_enable_nic_media_ch
LPCG93	clk_enable_ddr_dfi_ch
LPCG94	clk_enable_ddr_ctl_ch
LPCG95	clk_enable_ddr_dfi_ctl_ch
LPCG96	clk_enable_ddr_ssi_ch
LPCG97	clk_enable_ddr_bypass_ch
LPCG98	clk_enable_ddr_apb_ch
LPCG99	clk_enable_ddr_drampll_ch
LPCG100	clk_enable_ddr_clk_ctl_ch
LPCG101	clk_enable_nic_central_ch
LPCG102	clk_enable_gic600_ch
LPCG103	clk_enable_nic_apb_ch

Table continues on the next page...

Table 145. Clock Gate Table (continued)

Gating Channel	Generated Gate Signal
LPCG104	clk_enable_usb_controller_ch
LPCG105	clk_enable_usb_test_60m_ch
LPCG106	clk_enable_hsio_trout_24m_ch
LPCG107	clk_enable_pdm_ch
LPCG108	clk_enable_mqs1_ch
LPCG109	clk_enable_mqs2_ch
LPCG110	clk_enable_aud_xcvr_ch
LPCG111	clk_enable_nicmix_mecc_ch
LPCG112	clk_enable_spdif_ch
LPCG113	clk_enable_ssi_ml2nic_ch
LPCG114	clk_enable_ssi_med2nic_ch
LPCG115	clk_enable_ssi_hsio2nic_ch
LPCG116	clk_enable_ssi_w2nic_ch
LPCG117	clk_enable_ssi_nic2w_ch
LPCG118	clk_enable_ssi_nic2ddr_ch
LPCG119	clk_enable_hsio_32k_ch
LPCG120	clk_enable_enet1_ch
LPCG121	clk_enable_enet_qos_ch
LPCG122	clk_enable_sys_cnt_ch
LPCG123	clk_enable_tstmr1_ch
LPCG124	clk_enable_tstmr2_ch
LPCG125	clk_enable_tmc_ch

30.3.5 General Purpose Registers (GPR)

The general purpose registers provide miscellaneous controls and data saving functions. The GPRs are divided in two regions - shared and private region.

In shared region, all CPU domains access the same registers. If any CPU updates the value in the shared region, all the other CPU domains can read out the new value. The registers have an access control bit field to protect it from being changed unexpectedly.

In private region, each CPU domain has its own register. A CPU domain can only access registers belonging to itself. Since all the registers share the same address, a CPU domain will not have access to registers belonging to another CPU domain.

Shared registers can be read by any domain at any time, while write access to shared registers must be from domains granted by GPR_SHAREDn_AUTHEN[WHITE_LIST] and Trustzone authenticated. Every shared register has independent whitelist.

Private registers can be and only can be read or written by corresponding domain with Trustzone authentication.

GPR_SHAREDn_AUTHEN and GPR_PRIVATEn_AUTHEN registers can be read by any domain at any time, but the write access must be from domains granted by respective whitelist and Trustzone authenticated.

Type	GPR	Position	Width	Description
Shared	0	0	2	EXT_CLK_SEL. Selecting ext_clk from 4 external clocks: • 00b - Select ext1_clk. • 01b - Select ext2_clk. • 10b - Select ext3_clk. • 11b - Select ext4_clk. Changing the value of this field when ext_clk is used is not allowed, otherwise ext_clk will be gated off. **NOTE** Only EXT_CLK1 exists.
Shared	1	32	1	arm_pll_sel.
Shared	2	64	1	dram_pll_sel.

30.4 Reset

The reset of CCM will be released along with system early reset and clocks are provided during system reset. For each power domain, early reset of the power domain in which that clock gate resides, will be fed for correct power up and reset flow.

30.5 CCM Memory Map and Register definition

30.5.1 CCM register descriptions

30.5.1.1 D_IP_MX9DUAL_CCM memory map

CCM_CTRL base address: 4445_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Clock Root Control Register (CLOCK_ROOT0_CONTROL)	32	RW	0000_0000
4	Clock Root Control Register (CLOCK_ROOT0_CONTROL_SET)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8	Clock Root Control Register (CLOCK_ROOT0_CONTROL_CLR)	32	RW	0000_0000
C	Clock Root Control Register (CLOCK_ROOT0_CONTROL_TOG)	32	RW	0000_0000
20	Clock root working status (CLOCK_ROOT0_STATUS0)	32	RW	0000_0000
30	Clock root access control (CLOCK_ROOT0_AUTHEN)	32	RW	FFFF_0000
34	Clock root access control (CLOCK_ROOT0_AUTHEN_SET)	32	RW	FFFF_0000
38	Clock root access control (CLOCK_ROOT0_AUTHEN_CLR)	32	RW	FFFF_0000
3C	Clock root access control (CLOCK_ROOT0_AUTHEN_TOG)	32	RW	FFFF_0000
80	Clock Root Control Register (CLOCK_ROOT1_CONTROL)	32	RW	0000_0000
84	Clock Root Control Register (CLOCK_ROOT1_CONTROL_SET)	32	RW	0000_0000
88	Clock Root Control Register (CLOCK_ROOT1_CONTROL_CLR)	32	RW	0000_0000
8C	Clock Root Control Register (CLOCK_ROOT1_CONTROL_TOG)	32	RW	0000_0000
A0	Clock root working status (CLOCK_ROOT1_STATUS0)	32	RW	0000_0000
B0	Clock root access control (CLOCK_ROOT1_AUTHEN)	32	RW	FFFF_0000
B4	Clock root access control (CLOCK_ROOT1_AUTHEN_SET)	32	RW	FFFF_0000
B8	Clock root access control (CLOCK_ROOT1_AUTHEN_CLR)	32	RW	FFFF_0000
BC	Clock root access control (CLOCK_ROOT1_AUTHEN_TOG)	32	RW	FFFF_0000
100	Clock Root Control Register (CLOCK_ROOT2_CONTROL)	32	RW	0000_0000
104	Clock Root Control Register (CLOCK_ROOT2_CONTROL_SET)	32	RW	0000_0000
108	Clock Root Control Register (CLOCK_ROOT2_CONTROL_CLR)	32	RW	0000_0000
10C	Clock Root Control Register (CLOCK_ROOT2_CONTROL_TOG)	32	RW	0000_0000
120	Clock root working status (CLOCK_ROOT2_STATUS0)	32	RW	0000_0000
130	Clock root access control (CLOCK_ROOT2_AUTHEN)	32	RW	FFFF_0000
134	Clock root access control (CLOCK_ROOT2_AUTHEN_SET)	32	RW	FFFF_0000
138	Clock root access control (CLOCK_ROOT2_AUTHEN_CLR)	32	RW	FFFF_0000
13C	Clock root access control (CLOCK_ROOT2_AUTHEN_TOG)	32	RW	FFFF_0000
180	Clock Root Control Register (CLOCK_ROOT3_CONTROL)	32	RW	0000_0000
184	Clock Root Control Register (CLOCK_ROOT3_CONTROL_SET)	32	RW	0000_0000
188	Clock Root Control Register (CLOCK_ROOT3_CONTROL_CLR)	32	RW	0000_0000
18C	Clock Root Control Register (CLOCK_ROOT3_CONTROL_TOG)	32	RW	0000_0000
1A0	Clock root working status (CLOCK_ROOT3_STATUS0)	32	RW	0000_0000
1B0	Clock root access control (CLOCK_ROOT3_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1B4	Clock root access control (CLOCK_ROOT3_AUTHEN_SET)	32	RW	FFFF_0000
1B8	Clock root access control (CLOCK_ROOT3_AUTHEN_CLR)	32	RW	FFFF_0000
1BC	Clock root access control (CLOCK_ROOT3_AUTHEN_TOG)	32	RW	FFFF_0000
200	Clock Root Control Register (CLOCK_ROOT4_CONTROL)	32	RW	0000_0000
204	Clock Root Control Register (CLOCK_ROOT4_CONTROL_SET)	32	RW	0000_0000
208	Clock Root Control Register (CLOCK_ROOT4_CONTROL_CLR)	32	RW	0000_0000
20C	Clock Root Control Register (CLOCK_ROOT4_CONTROL_TOG)	32	RW	0000_0000
220	Clock root working status (CLOCK_ROOT4_STATUS0)	32	RW	0000_0000
230	Clock root access control (CLOCK_ROOT4_AUTHEN)	32	RW	FFFF_0000
234	Clock root access control (CLOCK_ROOT4_AUTHEN_SET)	32	RW	FFFF_0000
238	Clock root access control (CLOCK_ROOT4_AUTHEN_CLR)	32	RW	FFFF_0000
23C	Clock root access control (CLOCK_ROOT4_AUTHEN_TOG)	32	RW	FFFF_0000
280	Clock Root Control Register (CLOCK_ROOT5_CONTROL)	32	RW	0000_0000
284	Clock Root Control Register (CLOCK_ROOT5_CONTROL_SET)	32	RW	0000_0000
288	Clock Root Control Register (CLOCK_ROOT5_CONTROL_CLR)	32	RW	0000_0000
28C	Clock Root Control Register (CLOCK_ROOT5_CONTROL_TOG)	32	RW	0000_0000
2A0	Clock root working status (CLOCK_ROOT5_STATUS0)	32	RW	0000_0000
2B0	Clock root access control (CLOCK_ROOT5_AUTHEN)	32	RW	FFFF_0000
2B4	Clock root access control (CLOCK_ROOT5_AUTHEN_SET)	32	RW	FFFF_0000
2B8	Clock root access control (CLOCK_ROOT5_AUTHEN_CLR)	32	RW	FFFF_0000
2BC	Clock root access control (CLOCK_ROOT5_AUTHEN_TOG)	32	RW	FFFF_0000
300	Clock Root Control Register (CLOCK_ROOT6_CONTROL)	32	RW	0000_0000
304	Clock Root Control Register (CLOCK_ROOT6_CONTROL_SET)	32	RW	0000_0000
308	Clock Root Control Register (CLOCK_ROOT6_CONTROL_CLR)	32	RW	0000_0000
30C	Clock Root Control Register (CLOCK_ROOT6_CONTROL_TOG)	32	RW	0000_0000
320	Clock root working status (CLOCK_ROOT6_STATUS0)	32	RW	0000_0000
330	Clock root access control (CLOCK_ROOT6_AUTHEN)	32	RW	FFFF_0000
334	Clock root access control (CLOCK_ROOT6_AUTHEN_SET)	32	RW	FFFF_0000
338	Clock root access control (CLOCK_ROOT6_AUTHEN_CLR)	32	RW	FFFF_0000
33C	Clock root access control (CLOCK_ROOT6_AUTHEN_TOG)	32	RW	FFFF_0000
380	Clock Root Control Register (CLOCK_ROOT7_CONTROL)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
384	Clock Root Control Register (CLOCK_ROOT7_CONTROL_SET)	32	RW	0000_0000
388	Clock Root Control Register (CLOCK_ROOT7_CONTROL_CLR)	32	RW	0000_0000
38C	Clock Root Control Register (CLOCK_ROOT7_CONTROL_TOG)	32	RW	0000_0000
3A0	Clock root working status (CLOCK_ROOT7_STATUS0)	32	RW	0000_0000
3B0	Clock root access control (CLOCK_ROOT7_AUTHEN)	32	RW	FFFF_0000
3B4	Clock root access control (CLOCK_ROOT7_AUTHEN_SET)	32	RW	FFFF_0000
3B8	Clock root access control (CLOCK_ROOT7_AUTHEN_CLR)	32	RW	FFFF_0000
3BC	Clock root access control (CLOCK_ROOT7_AUTHEN_TOG)	32	RW	FFFF_0000
400	Clock Root Control Register (CLOCK_ROOT8_CONTROL)	32	RW	0000_0000
404	Clock Root Control Register (CLOCK_ROOT8_CONTROL_SET)	32	RW	0000_0000
408	Clock Root Control Register (CLOCK_ROOT8_CONTROL_CLR)	32	RW	0000_0000
40C	Clock Root Control Register (CLOCK_ROOT8_CONTROL_TOG)	32	RW	0000_0000
420	Clock root working status (CLOCK_ROOT8_STATUS0)	32	RW	0000_0000
430	Clock root access control (CLOCK_ROOT8_AUTHEN)	32	RW	FFFF_0000
434	Clock root access control (CLOCK_ROOT8_AUTHEN_SET)	32	RW	FFFF_0000
438	Clock root access control (CLOCK_ROOT8_AUTHEN_CLR)	32	RW	FFFF_0000
43C	Clock root access control (CLOCK_ROOT8_AUTHEN_TOG)	32	RW	FFFF_0000
480	Clock Root Control Register (CLOCK_ROOT9_CONTROL)	32	RW	0000_0000
484	Clock Root Control Register (CLOCK_ROOT9_CONTROL_SET)	32	RW	0000_0000
488	Clock Root Control Register (CLOCK_ROOT9_CONTROL_CLR)	32	RW	0000_0000
48C	Clock Root Control Register (CLOCK_ROOT9_CONTROL_TOG)	32	RW	0000_0000
4A0	Clock root working status (CLOCK_ROOT9_STATUS0)	32	RW	0000_0000
4B0	Clock root access control (CLOCK_ROOT9_AUTHEN)	32	RW	FFFF_0000
4B4	Clock root access control (CLOCK_ROOT9_AUTHEN_SET)	32	RW	FFFF_0000
4B8	Clock root access control (CLOCK_ROOT9_AUTHEN_CLR)	32	RW	FFFF_0000
4BC	Clock root access control (CLOCK_ROOT9_AUTHEN_TOG)	32	RW	FFFF_0000
500	Clock Root Control Register (CLOCK_ROOT10_CONTROL)	32	RW	0000_0000
504	Clock Root Control Register (CLOCK_ROOT10_CONTROL_SET)	32	RW	0000_0000
508	Clock Root Control Register (CLOCK_ROOT10_CONTROL_CLR)	32	RW	0000_0000
50C	Clock Root Control Register (CLOCK_ROOT10_CONTROL_TOG)	32	RW	0000_0000
520	Clock root working status (CLOCK_ROOT10_STATUS0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
530	Clock root access control (CLOCK_ROOT10_AUTHEN)	32	RW	FFFF_0000
534	Clock root access control (CLOCK_ROOT10_AUTHEN_SET)	32	RW	FFFF_0000
538	Clock root access control (CLOCK_ROOT10_AUTHEN_CLR)	32	RW	FFFF_0000
53C	Clock root access control (CLOCK_ROOT10_AUTHEN_TOG)	32	RW	FFFF_0000
580	Clock Root Control Register (CLOCK_ROOT11_CONTROL)	32	RW	0000_0000
584	Clock Root Control Register (CLOCK_ROOT11_CONTROL_SET)	32	RW	0000_0000
588	Clock Root Control Register (CLOCK_ROOT11_CONTROL_CLR)	32	RW	0000_0000
58C	Clock Root Control Register (CLOCK_ROOT11_CONTROL_TOG)	32	RW	0000_0000
5A0	Clock root working status (CLOCK_ROOT11_STATUS0)	32	RW	0000_0000
5B0	Clock root access control (CLOCK_ROOT11_AUTHEN)	32	RW	FFFF_0000
5B4	Clock root access control (CLOCK_ROOT11_AUTHEN_SET)	32	RW	FFFF_0000
5B8	Clock root access control (CLOCK_ROOT11_AUTHEN_CLR)	32	RW	FFFF_0000
5BC	Clock root access control (CLOCK_ROOT11_AUTHEN_TOG)	32	RW	FFFF_0000
600	Clock Root Control Register (CLOCK_ROOT12_CONTROL)	32	RW	0000_0000
604	Clock Root Control Register (CLOCK_ROOT12_CONTROL_SET)	32	RW	0000_0000
608	Clock Root Control Register (CLOCK_ROOT12_CONTROL_CLR)	32	RW	0000_0000
60C	Clock Root Control Register (CLOCK_ROOT12_CONTROL_TOG)	32	RW	0000_0000
620	Clock root working status (CLOCK_ROOT12_STATUS0)	32	RW	0000_0000
630	Clock root access control (CLOCK_ROOT12_AUTHEN)	32	RW	FFFF_0000
634	Clock root access control (CLOCK_ROOT12_AUTHEN_SET)	32	RW	FFFF_0000
638	Clock root access control (CLOCK_ROOT12_AUTHEN_CLR)	32	RW	FFFF_0000
63C	Clock root access control (CLOCK_ROOT12_AUTHEN_TOG)	32	RW	FFFF_0000
680	Clock Root Control Register (CLOCK_ROOT13_CONTROL)	32	RW	0000_0000
684	Clock Root Control Register (CLOCK_ROOT13_CONTROL_SET)	32	RW	0000_0000
688	Clock Root Control Register (CLOCK_ROOT13_CONTROL_CLR)	32	RW	0000_0000
68C	Clock Root Control Register (CLOCK_ROOT13_CONTROL_TOG)	32	RW	0000_0000
6A0	Clock root working status (CLOCK_ROOT13_STATUS0)	32	RW	0000_0000
6B0	Clock root access control (CLOCK_ROOT13_AUTHEN)	32	RW	FFFF_0000
6B4	Clock root access control (CLOCK_ROOT13_AUTHEN_SET)	32	RW	FFFF_0000
6B8	Clock root access control (CLOCK_ROOT13_AUTHEN_CLR)	32	RW	FFFF_0000
6BC	Clock root access control (CLOCK_ROOT13_AUTHEN_TOG)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
700	Clock Root Control Register (CLOCK_ROOT14_CONTROL)	32	RW	0000_0000
704	Clock Root Control Register (CLOCK_ROOT14_CONTROL_SET)	32	RW	0000_0000
708	Clock Root Control Register (CLOCK_ROOT14_CONTROL_CLR)	32	RW	0000_0000
70C	Clock Root Control Register (CLOCK_ROOT14_CONTROL_TOG)	32	RW	0000_0000
720	Clock root working status (CLOCK_ROOT14_STATUS0)	32	RW	0000_0000
730	Clock root access control (CLOCK_ROOT14_AUTHEN)	32	RW	FFFF_0000
734	Clock root access control (CLOCK_ROOT14_AUTHEN_SET)	32	RW	FFFF_0000
738	Clock root access control (CLOCK_ROOT14_AUTHEN_CLR)	32	RW	FFFF_0000
73C	Clock root access control (CLOCK_ROOT14_AUTHEN_TOG)	32	RW	FFFF_0000
780	Clock Root Control Register (CLOCK_ROOT15_CONTROL)	32	RW	0000_0000
784	Clock Root Control Register (CLOCK_ROOT15_CONTROL_SET)	32	RW	0000_0000
788	Clock Root Control Register (CLOCK_ROOT15_CONTROL_CLR)	32	RW	0000_0000
78C	Clock Root Control Register (CLOCK_ROOT15_CONTROL_TOG)	32	RW	0000_0000
7A0	Clock root working status (CLOCK_ROOT15_STATUS0)	32	RW	0000_0000
7B0	Clock root access control (CLOCK_ROOT15_AUTHEN)	32	RW	FFFF_0000
7B4	Clock root access control (CLOCK_ROOT15_AUTHEN_SET)	32	RW	FFFF_0000
7B8	Clock root access control (CLOCK_ROOT15_AUTHEN_CLR)	32	RW	FFFF_0000
7BC	Clock root access control (CLOCK_ROOT15_AUTHEN_TOG)	32	RW	FFFF_0000
800	Clock Root Control Register (CLOCK_ROOT16_CONTROL)	32	RW	0000_0000
804	Clock Root Control Register (CLOCK_ROOT16_CONTROL_SET)	32	RW	0000_0000
808	Clock Root Control Register (CLOCK_ROOT16_CONTROL_CLR)	32	RW	0000_0000
80C	Clock Root Control Register (CLOCK_ROOT16_CONTROL_TOG)	32	RW	0000_0000
820	Clock root working status (CLOCK_ROOT16_STATUS0)	32	RW	0000_0000
830	Clock root access control (CLOCK_ROOT16_AUTHEN)	32	RW	FFFF_0000
834	Clock root access control (CLOCK_ROOT16_AUTHEN_SET)	32	RW	FFFF_0000
838	Clock root access control (CLOCK_ROOT16_AUTHEN_CLR)	32	RW	FFFF_0000
83C	Clock root access control (CLOCK_ROOT16_AUTHEN_TOG)	32	RW	FFFF_0000
880	Clock Root Control Register (CLOCK_ROOT17_CONTROL)	32	RW	0000_0000
884	Clock Root Control Register (CLOCK_ROOT17_CONTROL_SET)	32	RW	0000_0000
888	Clock Root Control Register (CLOCK_ROOT17_CONTROL_CLR)	32	RW	0000_0000
88C	Clock Root Control Register (CLOCK_ROOT17_CONTROL_TOG)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8A0	Clock root working status (CLOCK_ROOT17_STATUS0)	32	RW	0000_0000
8B0	Clock root access control (CLOCK_ROOT17_AUTHEN)	32	RW	FFFF_0000
8B4	Clock root access control (CLOCK_ROOT17_AUTHEN_SET)	32	RW	FFFF_0000
8B8	Clock root access control (CLOCK_ROOT17_AUTHEN_CLR)	32	RW	FFFF_0000
8BC	Clock root access control (CLOCK_ROOT17_AUTHEN_TOG)	32	RW	FFFF_0000
900	Clock Root Control Register (CLOCK_ROOT18_CONTROL)	32	RW	0000_0000
904	Clock Root Control Register (CLOCK_ROOT18_CONTROL_SET)	32	RW	0000_0000
908	Clock Root Control Register (CLOCK_ROOT18_CONTROL_CLR)	32	RW	0000_0000
90C	Clock Root Control Register (CLOCK_ROOT18_CONTROL_TOG)	32	RW	0000_0000
920	Clock root working status (CLOCK_ROOT18_STATUS0)	32	RW	0000_0000
930	Clock root access control (CLOCK_ROOT18_AUTHEN)	32	RW	FFFF_0000
934	Clock root access control (CLOCK_ROOT18_AUTHEN_SET)	32	RW	FFFF_0000
938	Clock root access control (CLOCK_ROOT18_AUTHEN_CLR)	32	RW	FFFF_0000
93C	Clock root access control (CLOCK_ROOT18_AUTHEN_TOG)	32	RW	FFFF_0000
980	Clock Root Control Register (CLOCK_ROOT19_CONTROL)	32	RW	0000_0000
984	Clock Root Control Register (CLOCK_ROOT19_CONTROL_SET)	32	RW	0000_0000
988	Clock Root Control Register (CLOCK_ROOT19_CONTROL_CLR)	32	RW	0000_0000
98C	Clock Root Control Register (CLOCK_ROOT19_CONTROL_TOG)	32	RW	0000_0000
9A0	Clock root working status (CLOCK_ROOT19_STATUS0)	32	RW	0000_0000
9B0	Clock root access control (CLOCK_ROOT19_AUTHEN)	32	RW	FFFF_0000
9B4	Clock root access control (CLOCK_ROOT19_AUTHEN_SET)	32	RW	FFFF_0000
9B8	Clock root access control (CLOCK_ROOT19_AUTHEN_CLR)	32	RW	FFFF_0000
9BC	Clock root access control (CLOCK_ROOT19_AUTHEN_TOG)	32	RW	FFFF_0000
A00	Clock Root Control Register (CLOCK_ROOT20_CONTROL)	32	RW	0000_0000
A04	Clock Root Control Register (CLOCK_ROOT20_CONTROL_SET)	32	RW	0000_0000
A08	Clock Root Control Register (CLOCK_ROOT20_CONTROL_CLR)	32	RW	0000_0000
A0C	Clock Root Control Register (CLOCK_ROOT20_CONTROL_TOG)	32	RW	0000_0000
A20	Clock root working status (CLOCK_ROOT20_STATUS0)	32	RW	0000_0000
A30	Clock root access control (CLOCK_ROOT20_AUTHEN)	32	RW	FFFF_0000
A34	Clock root access control (CLOCK_ROOT20_AUTHEN_SET)	32	RW	FFFF_0000
A38	Clock root access control (CLOCK_ROOT20_AUTHEN_CLR)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
A3C	Clock root access control (CLOCK_ROOT20_AUTHEN_TOG)	32	RW	FFFF_0000
A80	Clock Root Control Register (CLOCK_ROOT21_CONTROL)	32	RW	0000_0000
A84	Clock Root Control Register (CLOCK_ROOT21_CONTROL_SET)	32	RW	0000_0000
A88	Clock Root Control Register (CLOCK_ROOT21_CONTROL_CLR)	32	RW	0000_0000
A8C	Clock Root Control Register (CLOCK_ROOT21_CONTROL_TOG)	32	RW	0000_0000
AA0	Clock root working status (CLOCK_ROOT21_STATUS0)	32	RW	0000_0000
AB0	Clock root access control (CLOCK_ROOT21_AUTHEN)	32	RW	FFFF_0000
AB4	Clock root access control (CLOCK_ROOT21_AUTHEN_SET)	32	RW	FFFF_0000
AB8	Clock root access control (CLOCK_ROOT21_AUTHEN_CLR)	32	RW	FFFF_0000
ABC	Clock root access control (CLOCK_ROOT21_AUTHEN_TOG)	32	RW	FFFF_0000
B00	Clock Root Control Register (CLOCK_ROOT22_CONTROL)	32	RW	0000_0000
B04	Clock Root Control Register (CLOCK_ROOT22_CONTROL_SET)	32	RW	0000_0000
B08	Clock Root Control Register (CLOCK_ROOT22_CONTROL_CLR)	32	RW	0000_0000
B0C	Clock Root Control Register (CLOCK_ROOT22_CONTROL_TOG)	32	RW	0000_0000
B20	Clock root working status (CLOCK_ROOT22_STATUS0)	32	RW	0000_0000
B30	Clock root access control (CLOCK_ROOT22_AUTHEN)	32	RW	FFFF_0000
B34	Clock root access control (CLOCK_ROOT22_AUTHEN_SET)	32	RW	FFFF_0000
B38	Clock root access control (CLOCK_ROOT22_AUTHEN_CLR)	32	RW	FFFF_0000
B3C	Clock root access control (CLOCK_ROOT22_AUTHEN_TOG)	32	RW	FFFF_0000
B80	Clock Root Control Register (CLOCK_ROOT23_CONTROL)	32	RW	0000_0000
B84	Clock Root Control Register (CLOCK_ROOT23_CONTROL_SET)	32	RW	0000_0000
B88	Clock Root Control Register (CLOCK_ROOT23_CONTROL_CLR)	32	RW	0000_0000
B8C	Clock Root Control Register (CLOCK_ROOT23_CONTROL_TOG)	32	RW	0000_0000
BA0	Clock root working status (CLOCK_ROOT23_STATUS0)	32	RW	0000_0000
BB0	Clock root access control (CLOCK_ROOT23_AUTHEN)	32	RW	FFFF_0000
BB4	Clock root access control (CLOCK_ROOT23_AUTHEN_SET)	32	RW	FFFF_0000
BB8	Clock root access control (CLOCK_ROOT23_AUTHEN_CLR)	32	RW	FFFF_0000
BBC	Clock root access control (CLOCK_ROOT23_AUTHEN_TOG)	32	RW	FFFF_0000
C00	Clock Root Control Register (CLOCK_ROOT24_CONTROL)	32	RW	0000_0000
C04	Clock Root Control Register (CLOCK_ROOT24_CONTROL_SET)	32	RW	0000_0000
C08	Clock Root Control Register (CLOCK_ROOT24_CONTROL_CLR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
C0C	Clock Root Control Register (CLOCK_ROOT24_CONTROL_TOG)	32	RW	0000_0000
C20	Clock root working status (CLOCK_ROOT24_STATUS0)	32	RW	0000_0000
C30	Clock root access control (CLOCK_ROOT24_AUTHEN)	32	RW	FFFF_0000
C34	Clock root access control (CLOCK_ROOT24_AUTHEN_SET)	32	RW	FFFF_0000
C38	Clock root access control (CLOCK_ROOT24_AUTHEN_CLR)	32	RW	FFFF_0000
C3C	Clock root access control (CLOCK_ROOT24_AUTHEN_TOG)	32	RW	FFFF_0000
C80	Clock Root Control Register (CLOCK_ROOT25_CONTROL)	32	RW	0000_0000
C84	Clock Root Control Register (CLOCK_ROOT25_CONTROL_SET)	32	RW	0000_0000
C88	Clock Root Control Register (CLOCK_ROOT25_CONTROL_CLR)	32	RW	0000_0000
C8C	Clock Root Control Register (CLOCK_ROOT25_CONTROL_TOG)	32	RW	0000_0000
CA0	Clock root working status (CLOCK_ROOT25_STATUS0)	32	RW	0000_0000
CB0	Clock root access control (CLOCK_ROOT25_AUTHEN)	32	RW	FFFF_0000
CB4	Clock root access control (CLOCK_ROOT25_AUTHEN_SET)	32	RW	FFFF_0000
CB8	Clock root access control (CLOCK_ROOT25_AUTHEN_CLR)	32	RW	FFFF_0000
CBC	Clock root access control (CLOCK_ROOT25_AUTHEN_TOG)	32	RW	FFFF_0000
D00	Clock Root Control Register (CLOCK_ROOT26_CONTROL)	32	RW	0000_0000
D04	Clock Root Control Register (CLOCK_ROOT26_CONTROL_SET)	32	RW	0000_0000
D08	Clock Root Control Register (CLOCK_ROOT26_CONTROL_CLR)	32	RW	0000_0000
D0C	Clock Root Control Register (CLOCK_ROOT26_CONTROL_TOG)	32	RW	0000_0000
D20	Clock root working status (CLOCK_ROOT26_STATUS0)	32	RW	0000_0000
D30	Clock root access control (CLOCK_ROOT26_AUTHEN)	32	RW	FFFF_0000
D34	Clock root access control (CLOCK_ROOT26_AUTHEN_SET)	32	RW	FFFF_0000
D38	Clock root access control (CLOCK_ROOT26_AUTHEN_CLR)	32	RW	FFFF_0000
D3C	Clock root access control (CLOCK_ROOT26_AUTHEN_TOG)	32	RW	FFFF_0000
D80	Clock Root Control Register (CLOCK_ROOT27_CONTROL)	32	RW	0000_0000
D84	Clock Root Control Register (CLOCK_ROOT27_CONTROL_SET)	32	RW	0000_0000
D88	Clock Root Control Register (CLOCK_ROOT27_CONTROL_CLR)	32	RW	0000_0000
D8C	Clock Root Control Register (CLOCK_ROOT27_CONTROL_TOG)	32	RW	0000_0000
DA0	Clock root working status (CLOCK_ROOT27_STATUS0)	32	RW	0000_0000
DB0	Clock root access control (CLOCK_ROOT27_AUTHEN)	32	RW	FFFF_0000
DB4	Clock root access control (CLOCK_ROOT27_AUTHEN_SET)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
DB8	Clock root access control (CLOCK_ROOT27_AUTHEN_CLR)	32	RW	FFFF_0000
DBC	Clock root access control (CLOCK_ROOT27_AUTHEN_TOG)	32	RW	FFFF_0000
E00	Clock Root Control Register (CLOCK_ROOT28_CONTROL)	32	RW	0000_0000
E04	Clock Root Control Register (CLOCK_ROOT28_CONTROL_SET)	32	RW	0000_0000
E08	Clock Root Control Register (CLOCK_ROOT28_CONTROL_CLR)	32	RW	0000_0000
E0C	Clock Root Control Register (CLOCK_ROOT28_CONTROL_TOG)	32	RW	0000_0000
E20	Clock root working status (CLOCK_ROOT28_STATUS0)	32	RW	0000_0000
E30	Clock root access control (CLOCK_ROOT28_AUTHEN)	32	RW	FFFF_0000
E34	Clock root access control (CLOCK_ROOT28_AUTHEN_SET)	32	RW	FFFF_0000
E38	Clock root access control (CLOCK_ROOT28_AUTHEN_CLR)	32	RW	FFFF_0000
E3C	Clock root access control (CLOCK_ROOT28_AUTHEN_TOG)	32	RW	FFFF_0000
E80	Clock Root Control Register (CLOCK_ROOT29_CONTROL)	32	RW	0000_0000
E84	Clock Root Control Register (CLOCK_ROOT29_CONTROL_SET)	32	RW	0000_0000
E88	Clock Root Control Register (CLOCK_ROOT29_CONTROL_CLR)	32	RW	0000_0000
E8C	Clock Root Control Register (CLOCK_ROOT29_CONTROL_TOG)	32	RW	0000_0000
EA0	Clock root working status (CLOCK_ROOT29_STATUS0)	32	RW	0000_0000
EB0	Clock root access control (CLOCK_ROOT29_AUTHEN)	32	RW	FFFF_0000
EB4	Clock root access control (CLOCK_ROOT29_AUTHEN_SET)	32	RW	FFFF_0000
EB8	Clock root access control (CLOCK_ROOT29_AUTHEN_CLR)	32	RW	FFFF_0000
EBC	Clock root access control (CLOCK_ROOT29_AUTHEN_TOG)	32	RW	FFFF_0000
F00	Clock Root Control Register (CLOCK_ROOT30_CONTROL)	32	RW	0000_0000
F04	Clock Root Control Register (CLOCK_ROOT30_CONTROL_SET)	32	RW	0000_0000
F08	Clock Root Control Register (CLOCK_ROOT30_CONTROL_CLR)	32	RW	0000_0000
F0C	Clock Root Control Register (CLOCK_ROOT30_CONTROL_TOG)	32	RW	0000_0000
F20	Clock root working status (CLOCK_ROOT30_STATUS0)	32	RW	0000_0000
F30	Clock root access control (CLOCK_ROOT30_AUTHEN)	32	RW	FFFF_0000
F34	Clock root access control (CLOCK_ROOT30_AUTHEN_SET)	32	RW	FFFF_0000
F38	Clock root access control (CLOCK_ROOT30_AUTHEN_CLR)	32	RW	FFFF_0000
F3C	Clock root access control (CLOCK_ROOT30_AUTHEN_TOG)	32	RW	FFFF_0000
F80	Clock Root Control Register (CLOCK_ROOT31_CONTROL)	32	RW	0000_0000
F84	Clock Root Control Register (CLOCK_ROOT31_CONTROL_SET)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
F88	Clock Root Control Register (CLOCK_ROOT31_CONTROL_CLR)	32	RW	0000_0000
F8C	Clock Root Control Register (CLOCK_ROOT31_CONTROL_TOG)	32	RW	0000_0000
FA0	Clock root working status (CLOCK_ROOT31_STATUS0)	32	RW	0000_0000
FB0	Clock root access control (CLOCK_ROOT31_AUTHEN)	32	RW	FFFF_0000
FB4	Clock root access control (CLOCK_ROOT31_AUTHEN_SET)	32	RW	FFFF_0000
FB8	Clock root access control (CLOCK_ROOT31_AUTHEN_CLR)	32	RW	FFFF_0000
FBC	Clock root access control (CLOCK_ROOT31_AUTHEN_TOG)	32	RW	FFFF_0000
1000	Clock Root Control Register (CLOCK_ROOT32_CONTROL)	32	RW	0000_0000
1004	Clock Root Control Register (CLOCK_ROOT32_CONTROL_SET)	32	RW	0000_0000
1008	Clock Root Control Register (CLOCK_ROOT32_CONTROL_CLR)	32	RW	0000_0000
100C	Clock Root Control Register (CLOCK_ROOT32_CONTROL_TOG)	32	RW	0000_0000
1020	Clock root working status (CLOCK_ROOT32_STATUS0)	32	RW	0000_0000
1030	Clock root access control (CLOCK_ROOT32_AUTHEN)	32	RW	FFFF_0000
1034	Clock root access control (CLOCK_ROOT32_AUTHEN_SET)	32	RW	FFFF_0000
1038	Clock root access control (CLOCK_ROOT32_AUTHEN_CLR)	32	RW	FFFF_0000
103C	Clock root access control (CLOCK_ROOT32_AUTHEN_TOG)	32	RW	FFFF_0000
1080	Clock Root Control Register (CLOCK_ROOT33_CONTROL)	32	RW	0000_0000
1084	Clock Root Control Register (CLOCK_ROOT33_CONTROL_SET)	32	RW	0000_0000
1088	Clock Root Control Register (CLOCK_ROOT33_CONTROL_CLR)	32	RW	0000_0000
108C	Clock Root Control Register (CLOCK_ROOT33_CONTROL_TOG)	32	RW	0000_0000
10A0	Clock root working status (CLOCK_ROOT33_STATUS0)	32	RW	0000_0000
10B0	Clock root access control (CLOCK_ROOT33_AUTHEN)	32	RW	FFFF_0000
10B4	Clock root access control (CLOCK_ROOT33_AUTHEN_SET)	32	RW	FFFF_0000
10B8	Clock root access control (CLOCK_ROOT33_AUTHEN_CLR)	32	RW	FFFF_0000
10BC	Clock root access control (CLOCK_ROOT33_AUTHEN_TOG)	32	RW	FFFF_0000
1100	Clock Root Control Register (CLOCK_ROOT34_CONTROL)	32	RW	0000_0000
1104	Clock Root Control Register (CLOCK_ROOT34_CONTROL_SET)	32	RW	0000_0000
1108	Clock Root Control Register (CLOCK_ROOT34_CONTROL_CLR)	32	RW	0000_0000
110C	Clock Root Control Register (CLOCK_ROOT34_CONTROL_TOG)	32	RW	0000_0000
1120	Clock root working status (CLOCK_ROOT34_STATUS0)	32	RW	0000_0000
1130	Clock root access control (CLOCK_ROOT34_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1134	Clock root access control (CLOCK_ROOT34_AUTHEN_SET)	32	RW	FFFF_0000
1138	Clock root access control (CLOCK_ROOT34_AUTHEN_CLR)	32	RW	FFFF_0000
113C	Clock root access control (CLOCK_ROOT34_AUTHEN_TOG)	32	RW	FFFF_0000
1180	Clock Root Control Register (CLOCK_ROOT35_CONTROL)	32	RW	0000_0000
1184	Clock Root Control Register (CLOCK_ROOT35_CONTROL_SET)	32	RW	0000_0000
1188	Clock Root Control Register (CLOCK_ROOT35_CONTROL_CLR)	32	RW	0000_0000
118C	Clock Root Control Register (CLOCK_ROOT35_CONTROL_TOG)	32	RW	0000_0000
11A0	Clock root working status (CLOCK_ROOT35_STATUS0)	32	RW	0000_0000
11B0	Clock root access control (CLOCK_ROOT35_AUTHEN)	32	RW	FFFF_0000
11B4	Clock root access control (CLOCK_ROOT35_AUTHEN_SET)	32	RW	FFFF_0000
11B8	Clock root access control (CLOCK_ROOT35_AUTHEN_CLR)	32	RW	FFFF_0000
11BC	Clock root access control (CLOCK_ROOT35_AUTHEN_TOG)	32	RW	FFFF_0000
1200	Clock Root Control Register (CLOCK_ROOT36_CONTROL)	32	RW	0000_0000
1204	Clock Root Control Register (CLOCK_ROOT36_CONTROL_SET)	32	RW	0000_0000
1208	Clock Root Control Register (CLOCK_ROOT36_CONTROL_CLR)	32	RW	0000_0000
120C	Clock Root Control Register (CLOCK_ROOT36_CONTROL_TOG)	32	RW	0000_0000
1220	Clock root working status (CLOCK_ROOT36_STATUS0)	32	RW	0000_0000
1230	Clock root access control (CLOCK_ROOT36_AUTHEN)	32	RW	FFFF_0000
1234	Clock root access control (CLOCK_ROOT36_AUTHEN_SET)	32	RW	FFFF_0000
1238	Clock root access control (CLOCK_ROOT36_AUTHEN_CLR)	32	RW	FFFF_0000
123C	Clock root access control (CLOCK_ROOT36_AUTHEN_TOG)	32	RW	FFFF_0000
1280	Clock Root Control Register (CLOCK_ROOT37_CONTROL)	32	RW	0000_0000
1284	Clock Root Control Register (CLOCK_ROOT37_CONTROL_SET)	32	RW	0000_0000
1288	Clock Root Control Register (CLOCK_ROOT37_CONTROL_CLR)	32	RW	0000_0000
128C	Clock Root Control Register (CLOCK_ROOT37_CONTROL_TOG)	32	RW	0000_0000
12A0	Clock root working status (CLOCK_ROOT37_STATUS0)	32	RW	0000_0000
12B0	Clock root access control (CLOCK_ROOT37_AUTHEN)	32	RW	FFFF_0000
12B4	Clock root access control (CLOCK_ROOT37_AUTHEN_SET)	32	RW	FFFF_0000
12B8	Clock root access control (CLOCK_ROOT37_AUTHEN_CLR)	32	RW	FFFF_0000
12BC	Clock root access control (CLOCK_ROOT37_AUTHEN_TOG)	32	RW	FFFF_0000
1300	Clock Root Control Register (CLOCK_ROOT38_CONTROL)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1304	Clock Root Control Register (CLOCK_ROOT38_CONTROL_SET)	32	RW	0000_0000
1308	Clock Root Control Register (CLOCK_ROOT38_CONTROL_CLR)	32	RW	0000_0000
130C	Clock Root Control Register (CLOCK_ROOT38_CONTROL_TOG)	32	RW	0000_0000
1320	Clock root working status (CLOCK_ROOT38_STATUS0)	32	RW	0000_0000
1330	Clock root access control (CLOCK_ROOT38_AUTHEN)	32	RW	FFFF_0000
1334	Clock root access control (CLOCK_ROOT38_AUTHEN_SET)	32	RW	FFFF_0000
1338	Clock root access control (CLOCK_ROOT38_AUTHEN_CLR)	32	RW	FFFF_0000
133C	Clock root access control (CLOCK_ROOT38_AUTHEN_TOG)	32	RW	FFFF_0000
1380	Clock Root Control Register (CLOCK_ROOT39_CONTROL)	32	RW	0000_0000
1384	Clock Root Control Register (CLOCK_ROOT39_CONTROL_SET)	32	RW	0000_0000
1388	Clock Root Control Register (CLOCK_ROOT39_CONTROL_CLR)	32	RW	0000_0000
138C	Clock Root Control Register (CLOCK_ROOT39_CONTROL_TOG)	32	RW	0000_0000
13A0	Clock root working status (CLOCK_ROOT39_STATUS0)	32	RW	0000_0000
13B0	Clock root access control (CLOCK_ROOT39_AUTHEN)	32	RW	FFFF_0000
13B4	Clock root access control (CLOCK_ROOT39_AUTHEN_SET)	32	RW	FFFF_0000
13B8	Clock root access control (CLOCK_ROOT39_AUTHEN_CLR)	32	RW	FFFF_0000
13BC	Clock root access control (CLOCK_ROOT39_AUTHEN_TOG)	32	RW	FFFF_0000
1400	Clock Root Control Register (CLOCK_ROOT40_CONTROL)	32	RW	0000_0000
1404	Clock Root Control Register (CLOCK_ROOT40_CONTROL_SET)	32	RW	0000_0000
1408	Clock Root Control Register (CLOCK_ROOT40_CONTROL_CLR)	32	RW	0000_0000
140C	Clock Root Control Register (CLOCK_ROOT40_CONTROL_TOG)	32	RW	0000_0000
1420	Clock root working status (CLOCK_ROOT40_STATUS0)	32	RW	0000_0000
1430	Clock root access control (CLOCK_ROOT40_AUTHEN)	32	RW	FFFF_0000
1434	Clock root access control (CLOCK_ROOT40_AUTHEN_SET)	32	RW	FFFF_0000
1438	Clock root access control (CLOCK_ROOT40_AUTHEN_CLR)	32	RW	FFFF_0000
143C	Clock root access control (CLOCK_ROOT40_AUTHEN_TOG)	32	RW	FFFF_0000
1480	Clock Root Control Register (CLOCK_ROOT41_CONTROL)	32	RW	0000_0000
1484	Clock Root Control Register (CLOCK_ROOT41_CONTROL_SET)	32	RW	0000_0000
1488	Clock Root Control Register (CLOCK_ROOT41_CONTROL_CLR)	32	RW	0000_0000
148C	Clock Root Control Register (CLOCK_ROOT41_CONTROL_TOG)	32	RW	0000_0000
14A0	Clock root working status (CLOCK_ROOT41_STATUS0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
14B0	Clock root access control (CLOCK_ROOT41_AUTHEN)	32	RW	FFFF_0000
14B4	Clock root access control (CLOCK_ROOT41_AUTHEN_SET)	32	RW	FFFF_0000
14B8	Clock root access control (CLOCK_ROOT41_AUTHEN_CLR)	32	RW	FFFF_0000
14BC	Clock root access control (CLOCK_ROOT41_AUTHEN_TOG)	32	RW	FFFF_0000
1500	Clock Root Control Register (CLOCK_ROOT42_CONTROL)	32	RW	0000_0000
1504	Clock Root Control Register (CLOCK_ROOT42_CONTROL_SET)	32	RW	0000_0000
1508	Clock Root Control Register (CLOCK_ROOT42_CONTROL_CLR)	32	RW	0000_0000
150C	Clock Root Control Register (CLOCK_ROOT42_CONTROL_TOG)	32	RW	0000_0000
1520	Clock root working status (CLOCK_ROOT42_STATUS0)	32	RW	0000_0000
1530	Clock root access control (CLOCK_ROOT42_AUTHEN)	32	RW	FFFF_0000
1534	Clock root access control (CLOCK_ROOT42_AUTHEN_SET)	32	RW	FFFF_0000
1538	Clock root access control (CLOCK_ROOT42_AUTHEN_CLR)	32	RW	FFFF_0000
153C	Clock root access control (CLOCK_ROOT42_AUTHEN_TOG)	32	RW	FFFF_0000
1580	Clock Root Control Register (CLOCK_ROOT43_CONTROL)	32	RW	0000_0000
1584	Clock Root Control Register (CLOCK_ROOT43_CONTROL_SET)	32	RW	0000_0000
1588	Clock Root Control Register (CLOCK_ROOT43_CONTROL_CLR)	32	RW	0000_0000
158C	Clock Root Control Register (CLOCK_ROOT43_CONTROL_TOG)	32	RW	0000_0000
15A0	Clock root working status (CLOCK_ROOT43_STATUS0)	32	RW	0000_0000
15B0	Clock root access control (CLOCK_ROOT43_AUTHEN)	32	RW	FFFF_0000
15B4	Clock root access control (CLOCK_ROOT43_AUTHEN_SET)	32	RW	FFFF_0000
15B8	Clock root access control (CLOCK_ROOT43_AUTHEN_CLR)	32	RW	FFFF_0000
15BC	Clock root access control (CLOCK_ROOT43_AUTHEN_TOG)	32	RW	FFFF_0000
1600	Clock Root Control Register (CLOCK_ROOT44_CONTROL)	32	RW	0000_0000
1604	Clock Root Control Register (CLOCK_ROOT44_CONTROL_SET)	32	RW	0000_0000
1608	Clock Root Control Register (CLOCK_ROOT44_CONTROL_CLR)	32	RW	0000_0000
160C	Clock Root Control Register (CLOCK_ROOT44_CONTROL_TOG)	32	RW	0000_0000
1620	Clock root working status (CLOCK_ROOT44_STATUS0)	32	RW	0000_0000
1630	Clock root access control (CLOCK_ROOT44_AUTHEN)	32	RW	FFFF_0000
1634	Clock root access control (CLOCK_ROOT44_AUTHEN_SET)	32	RW	FFFF_0000
1638	Clock root access control (CLOCK_ROOT44_AUTHEN_CLR)	32	RW	FFFF_0000
163C	Clock root access control (CLOCK_ROOT44_AUTHEN_TOG)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1680	Clock Root Control Register (CLOCK_ROOT45_CONTROL)	32	RW	0000_0000
1684	Clock Root Control Register (CLOCK_ROOT45_CONTROL_SET)	32	RW	0000_0000
1688	Clock Root Control Register (CLOCK_ROOT45_CONTROL_CLR)	32	RW	0000_0000
168C	Clock Root Control Register (CLOCK_ROOT45_CONTROL_TOG)	32	RW	0000_0000
16A0	Clock root working status (CLOCK_ROOT45_STATUS0)	32	RW	0000_0000
16B0	Clock root access control (CLOCK_ROOT45_AUTHEN)	32	RW	FFFF_0000
16B4	Clock root access control (CLOCK_ROOT45_AUTHEN_SET)	32	RW	FFFF_0000
16B8	Clock root access control (CLOCK_ROOT45_AUTHEN_CLR)	32	RW	FFFF_0000
16BC	Clock root access control (CLOCK_ROOT45_AUTHEN_TOG)	32	RW	FFFF_0000
1700	Clock Root Control Register (CLOCK_ROOT46_CONTROL)	32	RW	0000_0000
1704	Clock Root Control Register (CLOCK_ROOT46_CONTROL_SET)	32	RW	0000_0000
1708	Clock Root Control Register (CLOCK_ROOT46_CONTROL_CLR)	32	RW	0000_0000
170C	Clock Root Control Register (CLOCK_ROOT46_CONTROL_TOG)	32	RW	0000_0000
1720	Clock root working status (CLOCK_ROOT46_STATUS0)	32	RW	0000_0000
1730	Clock root access control (CLOCK_ROOT46_AUTHEN)	32	RW	FFFF_0000
1734	Clock root access control (CLOCK_ROOT46_AUTHEN_SET)	32	RW	FFFF_0000
1738	Clock root access control (CLOCK_ROOT46_AUTHEN_CLR)	32	RW	FFFF_0000
173C	Clock root access control (CLOCK_ROOT46_AUTHEN_TOG)	32	RW	FFFF_0000
1780	Clock Root Control Register (CLOCK_ROOT47_CONTROL)	32	RW	0000_0000
1784	Clock Root Control Register (CLOCK_ROOT47_CONTROL_SET)	32	RW	0000_0000
1788	Clock Root Control Register (CLOCK_ROOT47_CONTROL_CLR)	32	RW	0000_0000
178C	Clock Root Control Register (CLOCK_ROOT47_CONTROL_TOG)	32	RW	0000_0000
17A0	Clock root working status (CLOCK_ROOT47_STATUS0)	32	RW	0000_0000
17B0	Clock root access control (CLOCK_ROOT47_AUTHEN)	32	RW	FFFF_0000
17B4	Clock root access control (CLOCK_ROOT47_AUTHEN_SET)	32	RW	FFFF_0000
17B8	Clock root access control (CLOCK_ROOT47_AUTHEN_CLR)	32	RW	FFFF_0000
17BC	Clock root access control (CLOCK_ROOT47_AUTHEN_TOG)	32	RW	FFFF_0000
1800	Clock Root Control Register (CLOCK_ROOT48_CONTROL)	32	RW	0000_0000
1804	Clock Root Control Register (CLOCK_ROOT48_CONTROL_SET)	32	RW	0000_0000
1808	Clock Root Control Register (CLOCK_ROOT48_CONTROL_CLR)	32	RW	0000_0000
180C	Clock Root Control Register (CLOCK_ROOT48_CONTROL_TOG)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1820	Clock root working status (CLOCK_ROOT48_STATUS0)	32	RW	0000_0000
1830	Clock root access control (CLOCK_ROOT48_AUTHEN)	32	RW	FFFF_0000
1834	Clock root access control (CLOCK_ROOT48_AUTHEN_SET)	32	RW	FFFF_0000
1838	Clock root access control (CLOCK_ROOT48_AUTHEN_CLR)	32	RW	FFFF_0000
183C	Clock root access control (CLOCK_ROOT48_AUTHEN_TOG)	32	RW	FFFF_0000
1880	Clock Root Control Register (CLOCK_ROOT49_CONTROL)	32	RW	0000_0000
1884	Clock Root Control Register (CLOCK_ROOT49_CONTROL_SET)	32	RW	0000_0000
1888	Clock Root Control Register (CLOCK_ROOT49_CONTROL_CLR)	32	RW	0000_0000
188C	Clock Root Control Register (CLOCK_ROOT49_CONTROL_TOG)	32	RW	0000_0000
18A0	Clock root working status (CLOCK_ROOT49_STATUS0)	32	RW	0000_0000
18B0	Clock root access control (CLOCK_ROOT49_AUTHEN)	32	RW	FFFF_0000
18B4	Clock root access control (CLOCK_ROOT49_AUTHEN_SET)	32	RW	FFFF_0000
18B8	Clock root access control (CLOCK_ROOT49_AUTHEN_CLR)	32	RW	FFFF_0000
18BC	Clock root access control (CLOCK_ROOT49_AUTHEN_TOG)	32	RW	FFFF_0000
1900	Clock Root Control Register (CLOCK_ROOT50_CONTROL)	32	RW	0000_0000
1904	Clock Root Control Register (CLOCK_ROOT50_CONTROL_SET)	32	RW	0000_0000
1908	Clock Root Control Register (CLOCK_ROOT50_CONTROL_CLR)	32	RW	0000_0000
190C	Clock Root Control Register (CLOCK_ROOT50_CONTROL_TOG)	32	RW	0000_0000
1920	Clock root working status (CLOCK_ROOT50_STATUS0)	32	RW	0000_0000
1930	Clock root access control (CLOCK_ROOT50_AUTHEN)	32	RW	FFFF_0000
1934	Clock root access control (CLOCK_ROOT50_AUTHEN_SET)	32	RW	FFFF_0000
1938	Clock root access control (CLOCK_ROOT50_AUTHEN_CLR)	32	RW	FFFF_0000
193C	Clock root access control (CLOCK_ROOT50_AUTHEN_TOG)	32	RW	FFFF_0000
1980	Clock Root Control Register (CLOCK_ROOT51_CONTROL)	32	RW	0000_0000
1984	Clock Root Control Register (CLOCK_ROOT51_CONTROL_SET)	32	RW	0000_0000
1988	Clock Root Control Register (CLOCK_ROOT51_CONTROL_CLR)	32	RW	0000_0000
198C	Clock Root Control Register (CLOCK_ROOT51_CONTROL_TOG)	32	RW	0000_0000
19A0	Clock root working status (CLOCK_ROOT51_STATUS0)	32	RW	0000_0000
19B0	Clock root access control (CLOCK_ROOT51_AUTHEN)	32	RW	FFFF_0000
19B4	Clock root access control (CLOCK_ROOT51_AUTHEN_SET)	32	RW	FFFF_0000
19B8	Clock root access control (CLOCK_ROOT51_AUTHEN_CLR)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
19BC	Clock root access control (CLOCK_ROOT51_AUTHEN_TOG)	32	RW	FFFF_0000
1A00	Clock Root Control Register (CLOCK_ROOT52_CONTROL)	32	RW	0000_0000
1A04	Clock Root Control Register (CLOCK_ROOT52_CONTROL_SET)	32	RW	0000_0000
1A08	Clock Root Control Register (CLOCK_ROOT52_CONTROL_CLR)	32	RW	0000_0000
1A0C	Clock Root Control Register (CLOCK_ROOT52_CONTROL_TOG)	32	RW	0000_0000
1A20	Clock root working status (CLOCK_ROOT52_STATUS0)	32	RW	0000_0000
1A30	Clock root access control (CLOCK_ROOT52_AUTHEN)	32	RW	FFFF_0000
1A34	Clock root access control (CLOCK_ROOT52_AUTHEN_SET)	32	RW	FFFF_0000
1A38	Clock root access control (CLOCK_ROOT52_AUTHEN_CLR)	32	RW	FFFF_0000
1A3C	Clock root access control (CLOCK_ROOT52_AUTHEN_TOG)	32	RW	FFFF_0000
1A80	Clock Root Control Register (CLOCK_ROOT53_CONTROL)	32	RW	0000_0000
1A84	Clock Root Control Register (CLOCK_ROOT53_CONTROL_SET)	32	RW	0000_0000
1A88	Clock Root Control Register (CLOCK_ROOT53_CONTROL_CLR)	32	RW	0000_0000
1A8C	Clock Root Control Register (CLOCK_ROOT53_CONTROL_TOG)	32	RW	0000_0000
1AA0	Clock root working status (CLOCK_ROOT53_STATUS0)	32	RW	0000_0000
1AB0	Clock root access control (CLOCK_ROOT53_AUTHEN)	32	RW	FFFF_0000
1AB4	Clock root access control (CLOCK_ROOT53_AUTHEN_SET)	32	RW	FFFF_0000
1AB8	Clock root access control (CLOCK_ROOT53_AUTHEN_CLR)	32	RW	FFFF_0000
1ABC	Clock root access control (CLOCK_ROOT53_AUTHEN_TOG)	32	RW	FFFF_0000
1B00	Clock Root Control Register (CLOCK_ROOT54_CONTROL)	32	RW	0000_0000
1B04	Clock Root Control Register (CLOCK_ROOT54_CONTROL_SET)	32	RW	0000_0000
1B08	Clock Root Control Register (CLOCK_ROOT54_CONTROL_CLR)	32	RW	0000_0000
1B0C	Clock Root Control Register (CLOCK_ROOT54_CONTROL_TOG)	32	RW	0000_0000
1B20	Clock root working status (CLOCK_ROOT54_STATUS0)	32	RW	0000_0000
1B30	Clock root access control (CLOCK_ROOT54_AUTHEN)	32	RW	FFFF_0000
1B34	Clock root access control (CLOCK_ROOT54_AUTHEN_SET)	32	RW	FFFF_0000
1B38	Clock root access control (CLOCK_ROOT54_AUTHEN_CLR)	32	RW	FFFF_0000
1B3C	Clock root access control (CLOCK_ROOT54_AUTHEN_TOG)	32	RW	FFFF_0000
1B80	Clock Root Control Register (CLOCK_ROOT55_CONTROL)	32	RW	0000_0000
1B84	Clock Root Control Register (CLOCK_ROOT55_CONTROL_SET)	32	RW	0000_0000
1B88	Clock Root Control Register (CLOCK_ROOT55_CONTROL_CLR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1B8C	Clock Root Control Register (CLOCK_ROOT55_CONTROL_TOG)	32	RW	0000_0000
1BA0	Clock root working status (CLOCK_ROOT55_STATUS0)	32	RW	0000_0000
1BB0	Clock root access control (CLOCK_ROOT55_AUTHEN)	32	RW	FFFF_0000
1BB4	Clock root access control (CLOCK_ROOT55_AUTHEN_SET)	32	RW	FFFF_0000
1BB8	Clock root access control (CLOCK_ROOT55_AUTHEN_CLR)	32	RW	FFFF_0000
1BBC	Clock root access control (CLOCK_ROOT55_AUTHEN_TOG)	32	RW	FFFF_0000
1C00	Clock Root Control Register (CLOCK_ROOT56_CONTROL)	32	RW	0000_0000
1C04	Clock Root Control Register (CLOCK_ROOT56_CONTROL_SET)	32	RW	0000_0000
1C08	Clock Root Control Register (CLOCK_ROOT56_CONTROL_CLR)	32	RW	0000_0000
1C0C	Clock Root Control Register (CLOCK_ROOT56_CONTROL_TOG)	32	RW	0000_0000
1C20	Clock root working status (CLOCK_ROOT56_STATUS0)	32	RW	0000_0000
1C30	Clock root access control (CLOCK_ROOT56_AUTHEN)	32	RW	FFFF_0000
1C34	Clock root access control (CLOCK_ROOT56_AUTHEN_SET)	32	RW	FFFF_0000
1C38	Clock root access control (CLOCK_ROOT56_AUTHEN_CLR)	32	RW	FFFF_0000
1C3C	Clock root access control (CLOCK_ROOT56_AUTHEN_TOG)	32	RW	FFFF_0000
1C80	Clock Root Control Register (CLOCK_ROOT57_CONTROL)	32	RW	0000_0000
1C84	Clock Root Control Register (CLOCK_ROOT57_CONTROL_SET)	32	RW	0000_0000
1C88	Clock Root Control Register (CLOCK_ROOT57_CONTROL_CLR)	32	RW	0000_0000
1C8C	Clock Root Control Register (CLOCK_ROOT57_CONTROL_TOG)	32	RW	0000_0000
1CA0	Clock root working status (CLOCK_ROOT57_STATUS0)	32	RW	0000_0000
1CB0	Clock root access control (CLOCK_ROOT57_AUTHEN)	32	RW	FFFF_0000
1CB4	Clock root access control (CLOCK_ROOT57_AUTHEN_SET)	32	RW	FFFF_0000
1CB8	Clock root access control (CLOCK_ROOT57_AUTHEN_CLR)	32	RW	FFFF_0000
1CBC	Clock root access control (CLOCK_ROOT57_AUTHEN_TOG)	32	RW	FFFF_0000
1D00	Clock Root Control Register (CLOCK_ROOT58_CONTROL)	32	RW	0000_0000
1D04	Clock Root Control Register (CLOCK_ROOT58_CONTROL_SET)	32	RW	0000_0000
1D08	Clock Root Control Register (CLOCK_ROOT58_CONTROL_CLR)	32	RW	0000_0000
1D0C	Clock Root Control Register (CLOCK_ROOT58_CONTROL_TOG)	32	RW	0000_0000
1D20	Clock root working status (CLOCK_ROOT58_STATUS0)	32	RW	0000_0000
1D30	Clock root access control (CLOCK_ROOT58_AUTHEN)	32	RW	FFFF_0000
1D34	Clock root access control (CLOCK_ROOT58_AUTHEN_SET)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1D38	Clock root access control (CLOCK_ROOT58_AUTHEN_CLR)	32	RW	FFFF_0000
1D3C	Clock root access control (CLOCK_ROOT58_AUTHEN_TOG)	32	RW	FFFF_0000
1D80	Clock Root Control Register (CLOCK_ROOT59_CONTROL)	32	RW	0000_0000
1D84	Clock Root Control Register (CLOCK_ROOT59_CONTROL_SET)	32	RW	0000_0000
1D88	Clock Root Control Register (CLOCK_ROOT59_CONTROL_CLR)	32	RW	0000_0000
1D8C	Clock Root Control Register (CLOCK_ROOT59_CONTROL_TOG)	32	RW	0000_0000
1DA0	Clock root working status (CLOCK_ROOT59_STATUS0)	32	RW	0000_0000
1DB0	Clock root access control (CLOCK_ROOT59_AUTHEN)	32	RW	FFFF_0000
1DB4	Clock root access control (CLOCK_ROOT59_AUTHEN_SET)	32	RW	FFFF_0000
1DB8	Clock root access control (CLOCK_ROOT59_AUTHEN_CLR)	32	RW	FFFF_0000
1DBC	Clock root access control (CLOCK_ROOT59_AUTHEN_TOG)	32	RW	FFFF_0000
1E00	Clock Root Control Register (CLOCK_ROOT60_CONTROL)	32	RW	0000_0000
1E04	Clock Root Control Register (CLOCK_ROOT60_CONTROL_SET)	32	RW	0000_0000
1E08	Clock Root Control Register (CLOCK_ROOT60_CONTROL_CLR)	32	RW	0000_0000
1E0C	Clock Root Control Register (CLOCK_ROOT60_CONTROL_TOG)	32	RW	0000_0000
1E20	Clock root working status (CLOCK_ROOT60_STATUS0)	32	RW	0000_0000
1E30	Clock root access control (CLOCK_ROOT60_AUTHEN)	32	RW	FFFF_0000
1E34	Clock root access control (CLOCK_ROOT60_AUTHEN_SET)	32	RW	FFFF_0000
1E38	Clock root access control (CLOCK_ROOT60_AUTHEN_CLR)	32	RW	FFFF_0000
1E3C	Clock root access control (CLOCK_ROOT60_AUTHEN_TOG)	32	RW	FFFF_0000
1E80	Clock Root Control Register (CLOCK_ROOT61_CONTROL)	32	RW	0000_0000
1E84	Clock Root Control Register (CLOCK_ROOT61_CONTROL_SET)	32	RW	0000_0000
1E88	Clock Root Control Register (CLOCK_ROOT61_CONTROL_CLR)	32	RW	0000_0000
1E8C	Clock Root Control Register (CLOCK_ROOT61_CONTROL_TOG)	32	RW	0000_0000
1EA0	Clock root working status (CLOCK_ROOT61_STATUS0)	32	RW	0000_0000
1EB0	Clock root access control (CLOCK_ROOT61_AUTHEN)	32	RW	FFFF_0000
1EB4	Clock root access control (CLOCK_ROOT61_AUTHEN_SET)	32	RW	FFFF_0000
1EB8	Clock root access control (CLOCK_ROOT61_AUTHEN_CLR)	32	RW	FFFF_0000
1EBC	Clock root access control (CLOCK_ROOT61_AUTHEN_TOG)	32	RW	FFFF_0000
1F00	Clock Root Control Register (CLOCK_ROOT62_CONTROL)	32	RW	0000_0000
1F04	Clock Root Control Register (CLOCK_ROOT62_CONTROL_SET)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1F08	Clock Root Control Register (CLOCK_ROOT62_CONTROL_CLR)	32	RW	0000_0000
1F0C	Clock Root Control Register (CLOCK_ROOT62_CONTROL_TOG)	32	RW	0000_0000
1F20	Clock root working status (CLOCK_ROOT62_STATUS0)	32	RW	0000_0000
1F30	Clock root access control (CLOCK_ROOT62_AUTHEN)	32	RW	FFFF_0000
1F34	Clock root access control (CLOCK_ROOT62_AUTHEN_SET)	32	RW	FFFF_0000
1F38	Clock root access control (CLOCK_ROOT62_AUTHEN_CLR)	32	RW	FFFF_0000
1F3C	Clock root access control (CLOCK_ROOT62_AUTHEN_TOG)	32	RW	FFFF_0000
1F80	Clock Root Control Register (CLOCK_ROOT63_CONTROL)	32	RW	0000_0000
1F84	Clock Root Control Register (CLOCK_ROOT63_CONTROL_SET)	32	RW	0000_0000
1F88	Clock Root Control Register (CLOCK_ROOT63_CONTROL_CLR)	32	RW	0000_0000
1F8C	Clock Root Control Register (CLOCK_ROOT63_CONTROL_TOG)	32	RW	0000_0000
1FA0	Clock root working status (CLOCK_ROOT63_STATUS0)	32	RW	0000_0000
1FB0	Clock root access control (CLOCK_ROOT63_AUTHEN)	32	RW	FFFF_0000
1FB4	Clock root access control (CLOCK_ROOT63_AUTHEN_SET)	32	RW	FFFF_0000
1FB8	Clock root access control (CLOCK_ROOT63_AUTHEN_CLR)	32	RW	FFFF_0000
1FBC	Clock root access control (CLOCK_ROOT63_AUTHEN_TOG)	32	RW	FFFF_0000
2000	Clock Root Control Register (CLOCK_ROOT64_CONTROL)	32	RW	0000_0000
2004	Clock Root Control Register (CLOCK_ROOT64_CONTROL_SET)	32	RW	0000_0000
2008	Clock Root Control Register (CLOCK_ROOT64_CONTROL_CLR)	32	RW	0000_0000
200C	Clock Root Control Register (CLOCK_ROOT64_CONTROL_TOG)	32	RW	0000_0000
2020	Clock root working status (CLOCK_ROOT64_STATUS0)	32	RW	0000_0000
2030	Clock root access control (CLOCK_ROOT64_AUTHEN)	32	RW	FFFF_0000
2034	Clock root access control (CLOCK_ROOT64_AUTHEN_SET)	32	RW	FFFF_0000
2038	Clock root access control (CLOCK_ROOT64_AUTHEN_CLR)	32	RW	FFFF_0000
203C	Clock root access control (CLOCK_ROOT64_AUTHEN_TOG)	32	RW	FFFF_0000
2080	Clock Root Control Register (CLOCK_ROOT65_CONTROL)	32	RW	0000_0000
2084	Clock Root Control Register (CLOCK_ROOT65_CONTROL_SET)	32	RW	0000_0000
2088	Clock Root Control Register (CLOCK_ROOT65_CONTROL_CLR)	32	RW	0000_0000
208C	Clock Root Control Register (CLOCK_ROOT65_CONTROL_TOG)	32	RW	0000_0000
20A0	Clock root working status (CLOCK_ROOT65_STATUS0)	32	RW	0000_0000
20B0	Clock root access control (CLOCK_ROOT65_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
20B4	Clock root access control (CLOCK_ROOT65_AUTHEN_SET)	32	RW	FFFF_0000
20B8	Clock root access control (CLOCK_ROOT65_AUTHEN_CLR)	32	RW	FFFF_0000
20BC	Clock root access control (CLOCK_ROOT65_AUTHEN_TOG)	32	RW	FFFF_0000
2100	Clock Root Control Register (CLOCK_ROOT66_CONTROL)	32	RW	0000_0000
2104	Clock Root Control Register (CLOCK_ROOT66_CONTROL_SET)	32	RW	0000_0000
2108	Clock Root Control Register (CLOCK_ROOT66_CONTROL_CLR)	32	RW	0000_0000
210C	Clock Root Control Register (CLOCK_ROOT66_CONTROL_TOG)	32	RW	0000_0000
2120	Clock root working status (CLOCK_ROOT66_STATUS0)	32	RW	0000_0000
2130	Clock root access control (CLOCK_ROOT66_AUTHEN)	32	RW	FFFF_0000
2134	Clock root access control (CLOCK_ROOT66_AUTHEN_SET)	32	RW	FFFF_0000
2138	Clock root access control (CLOCK_ROOT66_AUTHEN_CLR)	32	RW	FFFF_0000
213C	Clock root access control (CLOCK_ROOT66_AUTHEN_TOG)	32	RW	FFFF_0000
2180	Clock Root Control Register (CLOCK_ROOT67_CONTROL)	32	RW	0000_0000
2184	Clock Root Control Register (CLOCK_ROOT67_CONTROL_SET)	32	RW	0000_0000
2188	Clock Root Control Register (CLOCK_ROOT67_CONTROL_CLR)	32	RW	0000_0000
218C	Clock Root Control Register (CLOCK_ROOT67_CONTROL_TOG)	32	RW	0000_0000
21A0	Clock root working status (CLOCK_ROOT67_STATUS0)	32	RW	0000_0000
21B0	Clock root access control (CLOCK_ROOT67_AUTHEN)	32	RW	FFFF_0000
21B4	Clock root access control (CLOCK_ROOT67_AUTHEN_SET)	32	RW	FFFF_0000
21B8	Clock root access control (CLOCK_ROOT67_AUTHEN_CLR)	32	RW	FFFF_0000
21BC	Clock root access control (CLOCK_ROOT67_AUTHEN_TOG)	32	RW	FFFF_0000
2200	Clock Root Control Register (CLOCK_ROOT68_CONTROL)	32	RW	0000_0000
2204	Clock Root Control Register (CLOCK_ROOT68_CONTROL_SET)	32	RW	0000_0000
2208	Clock Root Control Register (CLOCK_ROOT68_CONTROL_CLR)	32	RW	0000_0000
220C	Clock Root Control Register (CLOCK_ROOT68_CONTROL_TOG)	32	RW	0000_0000
2220	Clock root working status (CLOCK_ROOT68_STATUS0)	32	RW	0000_0000
2230	Clock root access control (CLOCK_ROOT68_AUTHEN)	32	RW	FFFF_0000
2234	Clock root access control (CLOCK_ROOT68_AUTHEN_SET)	32	RW	FFFF_0000
2238	Clock root access control (CLOCK_ROOT68_AUTHEN_CLR)	32	RW	FFFF_0000
223C	Clock root access control (CLOCK_ROOT68_AUTHEN_TOG)	32	RW	FFFF_0000
2280	Clock Root Control Register (CLOCK_ROOT69_CONTROL)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2284	Clock Root Control Register (CLOCK_ROOT69_CONTROL_SET)	32	RW	0000_0000
2288	Clock Root Control Register (CLOCK_ROOT69_CONTROL_CLR)	32	RW	0000_0000
228C	Clock Root Control Register (CLOCK_ROOT69_CONTROL_TOG)	32	RW	0000_0000
22A0	Clock root working status (CLOCK_ROOT69_STATUS0)	32	RW	0000_0000
22B0	Clock root access control (CLOCK_ROOT69_AUTHEN)	32	RW	FFFF_0000
22B4	Clock root access control (CLOCK_ROOT69_AUTHEN_SET)	32	RW	FFFF_0000
22B8	Clock root access control (CLOCK_ROOT69_AUTHEN_CLR)	32	RW	FFFF_0000
22BC	Clock root access control (CLOCK_ROOT69_AUTHEN_TOG)	32	RW	FFFF_0000
2300	Clock Root Control Register (CLOCK_ROOT70_CONTROL)	32	RW	0000_0000
2304	Clock Root Control Register (CLOCK_ROOT70_CONTROL_SET)	32	RW	0000_0000
2308	Clock Root Control Register (CLOCK_ROOT70_CONTROL_CLR)	32	RW	0000_0000
230C	Clock Root Control Register (CLOCK_ROOT70_CONTROL_TOG)	32	RW	0000_0000
2320	Clock root working status (CLOCK_ROOT70_STATUS0)	32	RW	0000_0000
2330	Clock root access control (CLOCK_ROOT70_AUTHEN)	32	RW	FFFF_0000
2334	Clock root access control (CLOCK_ROOT70_AUTHEN_SET)	32	RW	FFFF_0000
2338	Clock root access control (CLOCK_ROOT70_AUTHEN_CLR)	32	RW	FFFF_0000
233C	Clock root access control (CLOCK_ROOT70_AUTHEN_TOG)	32	RW	FFFF_0000
2380	Clock Root Control Register (CLOCK_ROOT71_CONTROL)	32	RW	0000_0000
2384	Clock Root Control Register (CLOCK_ROOT71_CONTROL_SET)	32	RW	0000_0000
2388	Clock Root Control Register (CLOCK_ROOT71_CONTROL_CLR)	32	RW	0000_0000
238C	Clock Root Control Register (CLOCK_ROOT71_CONTROL_TOG)	32	RW	0000_0000
23A0	Clock root working status (CLOCK_ROOT71_STATUS0)	32	RW	0000_0000
23B0	Clock root access control (CLOCK_ROOT71_AUTHEN)	32	RW	FFFF_0000
23B4	Clock root access control (CLOCK_ROOT71_AUTHEN_SET)	32	RW	FFFF_0000
23B8	Clock root access control (CLOCK_ROOT71_AUTHEN_CLR)	32	RW	FFFF_0000
23BC	Clock root access control (CLOCK_ROOT71_AUTHEN_TOG)	32	RW	FFFF_0000
2400	Clock Root Control Register (CLOCK_ROOT72_CONTROL)	32	RW	0000_0000
2404	Clock Root Control Register (CLOCK_ROOT72_CONTROL_SET)	32	RW	0000_0000
2408	Clock Root Control Register (CLOCK_ROOT72_CONTROL_CLR)	32	RW	0000_0000
240C	Clock Root Control Register (CLOCK_ROOT72_CONTROL_TOG)	32	RW	0000_0000
2420	Clock root working status (CLOCK_ROOT72_STATUS0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2430	Clock root access control (CLOCK_ROOT72_AUTHEN)	32	RW	FFFF_0000
2434	Clock root access control (CLOCK_ROOT72_AUTHEN_SET)	32	RW	FFFF_0000
2438	Clock root access control (CLOCK_ROOT72_AUTHEN_CLR)	32	RW	FFFF_0000
243C	Clock root access control (CLOCK_ROOT72_AUTHEN_TOG)	32	RW	FFFF_0000
2480	Clock Root Control Register (CLOCK_ROOT73_CONTROL)	32	RW	0000_0000
2484	Clock Root Control Register (CLOCK_ROOT73_CONTROL_SET)	32	RW	0000_0000
2488	Clock Root Control Register (CLOCK_ROOT73_CONTROL_CLR)	32	RW	0000_0000
248C	Clock Root Control Register (CLOCK_ROOT73_CONTROL_TOG)	32	RW	0000_0000
24A0	Clock root working status (CLOCK_ROOT73_STATUS0)	32	RW	0000_0000
24B0	Clock root access control (CLOCK_ROOT73_AUTHEN)	32	RW	FFFF_0000
24B4	Clock root access control (CLOCK_ROOT73_AUTHEN_SET)	32	RW	FFFF_0000
24B8	Clock root access control (CLOCK_ROOT73_AUTHEN_CLR)	32	RW	FFFF_0000
24BC	Clock root access control (CLOCK_ROOT73_AUTHEN_TOG)	32	RW	FFFF_0000
2500	Clock Root Control Register (CLOCK_ROOT74_CONTROL)	32	RW	0000_0000
2504	Clock Root Control Register (CLOCK_ROOT74_CONTROL_SET)	32	RW	0000_0000
2508	Clock Root Control Register (CLOCK_ROOT74_CONTROL_CLR)	32	RW	0000_0000
250C	Clock Root Control Register (CLOCK_ROOT74_CONTROL_TOG)	32	RW	0000_0000
2520	Clock root working status (CLOCK_ROOT74_STATUS0)	32	RW	0000_0000
2530	Clock root access control (CLOCK_ROOT74_AUTHEN)	32	RW	FFFF_0000
2534	Clock root access control (CLOCK_ROOT74_AUTHEN_SET)	32	RW	FFFF_0000
2538	Clock root access control (CLOCK_ROOT74_AUTHEN_CLR)	32	RW	FFFF_0000
253C	Clock root access control (CLOCK_ROOT74_AUTHEN_TOG)	32	RW	FFFF_0000
2580	Clock Root Control Register (CLOCK_ROOT75_CONTROL)	32	RW	0000_0000
2584	Clock Root Control Register (CLOCK_ROOT75_CONTROL_SET)	32	RW	0000_0000
2588	Clock Root Control Register (CLOCK_ROOT75_CONTROL_CLR)	32	RW	0000_0000
258C	Clock Root Control Register (CLOCK_ROOT75_CONTROL_TOG)	32	RW	0000_0000
25A0	Clock root working status (CLOCK_ROOT75_STATUS0)	32	RW	0000_0000
25B0	Clock root access control (CLOCK_ROOT75_AUTHEN)	32	RW	FFFF_0000
25B4	Clock root access control (CLOCK_ROOT75_AUTHEN_SET)	32	RW	FFFF_0000
25B8	Clock root access control (CLOCK_ROOT75_AUTHEN_CLR)	32	RW	FFFF_0000
25BC	Clock root access control (CLOCK_ROOT75_AUTHEN_TOG)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2600	Clock Root Control Register (CLOCK_ROOT76_CONTROL)	32	RW	0000_0000
2604	Clock Root Control Register (CLOCK_ROOT76_CONTROL_SET)	32	RW	0000_0000
2608	Clock Root Control Register (CLOCK_ROOT76_CONTROL_CLR)	32	RW	0000_0000
260C	Clock Root Control Register (CLOCK_ROOT76_CONTROL_TOG)	32	RW	0000_0000
2620	Clock root working status (CLOCK_ROOT76_STATUS0)	32	RW	0000_0000
2630	Clock root access control (CLOCK_ROOT76_AUTHEN)	32	RW	FFFF_0000
2634	Clock root access control (CLOCK_ROOT76_AUTHEN_SET)	32	RW	FFFF_0000
2638	Clock root access control (CLOCK_ROOT76_AUTHEN_CLR)	32	RW	FFFF_0000
263C	Clock root access control (CLOCK_ROOT76_AUTHEN_TOG)	32	RW	FFFF_0000
2680	Clock Root Control Register (CLOCK_ROOT77_CONTROL)	32	RW	0000_0000
2684	Clock Root Control Register (CLOCK_ROOT77_CONTROL_SET)	32	RW	0000_0000
2688	Clock Root Control Register (CLOCK_ROOT77_CONTROL_CLR)	32	RW	0000_0000
268C	Clock Root Control Register (CLOCK_ROOT77_CONTROL_TOG)	32	RW	0000_0000
26A0	Clock root working status (CLOCK_ROOT77_STATUS0)	32	RW	0000_0000
26B0	Clock root access control (CLOCK_ROOT77_AUTHEN)	32	RW	FFFF_0000
26B4	Clock root access control (CLOCK_ROOT77_AUTHEN_SET)	32	RW	FFFF_0000
26B8	Clock root access control (CLOCK_ROOT77_AUTHEN_CLR)	32	RW	FFFF_0000
26BC	Clock root access control (CLOCK_ROOT77_AUTHEN_TOG)	32	RW	FFFF_0000
2700	Clock Root Control Register (CLOCK_ROOT78_CONTROL)	32	RW	0000_0000
2704	Clock Root Control Register (CLOCK_ROOT78_CONTROL_SET)	32	RW	0000_0000
2708	Clock Root Control Register (CLOCK_ROOT78_CONTROL_CLR)	32	RW	0000_0000
270C	Clock Root Control Register (CLOCK_ROOT78_CONTROL_TOG)	32	RW	0000_0000
2720	Clock root working status (CLOCK_ROOT78_STATUS0)	32	RW	0000_0000
2730	Clock root access control (CLOCK_ROOT78_AUTHEN)	32	RW	FFFF_0000
2734	Clock root access control (CLOCK_ROOT78_AUTHEN_SET)	32	RW	FFFF_0000
2738	Clock root access control (CLOCK_ROOT78_AUTHEN_CLR)	32	RW	FFFF_0000
273C	Clock root access control (CLOCK_ROOT78_AUTHEN_TOG)	32	RW	FFFF_0000
2780	Clock Root Control Register (CLOCK_ROOT79_CONTROL)	32	RW	0000_0000
2784	Clock Root Control Register (CLOCK_ROOT79_CONTROL_SET)	32	RW	0000_0000
2788	Clock Root Control Register (CLOCK_ROOT79_CONTROL_CLR)	32	RW	0000_0000
278C	Clock Root Control Register (CLOCK_ROOT79_CONTROL_TOG)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
27A0	Clock root working status (CLOCK_ROOT79_STATUS0)	32	RW	0000_0000
27B0	Clock root access control (CLOCK_ROOT79_AUTHEN)	32	RW	FFFF_0000
27B4	Clock root access control (CLOCK_ROOT79_AUTHEN_SET)	32	RW	FFFF_0000
27B8	Clock root access control (CLOCK_ROOT79_AUTHEN_CLR)	32	RW	FFFF_0000
27BC	Clock root access control (CLOCK_ROOT79_AUTHEN_TOG)	32	RW	FFFF_0000
2800	Clock Root Control Register (CLOCK_ROOT80_CONTROL)	32	RW	0000_0000
2804	Clock Root Control Register (CLOCK_ROOT80_CONTROL_SET)	32	RW	0000_0000
2808	Clock Root Control Register (CLOCK_ROOT80_CONTROL_CLR)	32	RW	0000_0000
280C	Clock Root Control Register (CLOCK_ROOT80_CONTROL_TOG)	32	RW	0000_0000
2820	Clock root working status (CLOCK_ROOT80_STATUS0)	32	RW	0000_0000
2830	Clock root access control (CLOCK_ROOT80_AUTHEN)	32	RW	FFFF_0000
2834	Clock root access control (CLOCK_ROOT80_AUTHEN_SET)	32	RW	FFFF_0000
2838	Clock root access control (CLOCK_ROOT80_AUTHEN_CLR)	32	RW	FFFF_0000
283C	Clock root access control (CLOCK_ROOT80_AUTHEN_TOG)	32	RW	FFFF_0000
2880	Clock Root Control Register (CLOCK_ROOT81_CONTROL)	32	RW	0000_0000
2884	Clock Root Control Register (CLOCK_ROOT81_CONTROL_SET)	32	RW	0000_0000
2888	Clock Root Control Register (CLOCK_ROOT81_CONTROL_CLR)	32	RW	0000_0000
288C	Clock Root Control Register (CLOCK_ROOT81_CONTROL_TOG)	32	RW	0000_0000
28A0	Clock root working status (CLOCK_ROOT81_STATUS0)	32	RW	0000_0000
28B0	Clock root access control (CLOCK_ROOT81_AUTHEN)	32	RW	FFFF_0000
28B4	Clock root access control (CLOCK_ROOT81_AUTHEN_SET)	32	RW	FFFF_0000
28B8	Clock root access control (CLOCK_ROOT81_AUTHEN_CLR)	32	RW	FFFF_0000
28BC	Clock root access control (CLOCK_ROOT81_AUTHEN_TOG)	32	RW	FFFF_0000
2900	Clock Root Control Register (CLOCK_ROOT82_CONTROL)	32	RW	0000_0000
2904	Clock Root Control Register (CLOCK_ROOT82_CONTROL_SET)	32	RW	0000_0000
2908	Clock Root Control Register (CLOCK_ROOT82_CONTROL_CLR)	32	RW	0000_0000
290C	Clock Root Control Register (CLOCK_ROOT82_CONTROL_TOG)	32	RW	0000_0000
2920	Clock root working status (CLOCK_ROOT82_STATUS0)	32	RW	0000_0000
2930	Clock root access control (CLOCK_ROOT82_AUTHEN)	32	RW	FFFF_0000
2934	Clock root access control (CLOCK_ROOT82_AUTHEN_SET)	32	RW	FFFF_0000
2938	Clock root access control (CLOCK_ROOT82_AUTHEN_CLR)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
293C	Clock root access control (CLOCK_ROOT82_AUTHEN_TOG)	32	RW	FFFF_0000
2980	Clock Root Control Register (CLOCK_ROOT83_CONTROL)	32	RW	0000_0000
2984	Clock Root Control Register (CLOCK_ROOT83_CONTROL_SET)	32	RW	0000_0000
2988	Clock Root Control Register (CLOCK_ROOT83_CONTROL_CLR)	32	RW	0000_0000
298C	Clock Root Control Register (CLOCK_ROOT83_CONTROL_TOG)	32	RW	0000_0000
29A0	Clock root working status (CLOCK_ROOT83_STATUS0)	32	RW	0000_0000
29B0	Clock root access control (CLOCK_ROOT83_AUTHEN)	32	RW	FFFF_0000
29B4	Clock root access control (CLOCK_ROOT83_AUTHEN_SET)	32	RW	FFFF_0000
29B8	Clock root access control (CLOCK_ROOT83_AUTHEN_CLR)	32	RW	FFFF_0000
29BC	Clock root access control (CLOCK_ROOT83_AUTHEN_TOG)	32	RW	FFFF_0000
2A00	Clock Root Control Register (CLOCK_ROOT84_CONTROL)	32	RW	0000_0000
2A04	Clock Root Control Register (CLOCK_ROOT84_CONTROL_SET)	32	RW	0000_0000
2A08	Clock Root Control Register (CLOCK_ROOT84_CONTROL_CLR)	32	RW	0000_0000
2A0C	Clock Root Control Register (CLOCK_ROOT84_CONTROL_TOG)	32	RW	0000_0000
2A20	Clock root working status (CLOCK_ROOT84_STATUS0)	32	RW	0000_0000
2A30	Clock root access control (CLOCK_ROOT84_AUTHEN)	32	RW	FFFF_0000
2A34	Clock root access control (CLOCK_ROOT84_AUTHEN_SET)	32	RW	FFFF_0000
2A38	Clock root access control (CLOCK_ROOT84_AUTHEN_CLR)	32	RW	FFFF_0000
2A3C	Clock root access control (CLOCK_ROOT84_AUTHEN_TOG)	32	RW	FFFF_0000
2A80	Clock Root Control Register (CLOCK_ROOT85_CONTROL)	32	RW	0000_0000
2A84	Clock Root Control Register (CLOCK_ROOT85_CONTROL_SET)	32	RW	0000_0000
2A88	Clock Root Control Register (CLOCK_ROOT85_CONTROL_CLR)	32	RW	0000_0000
2A8C	Clock Root Control Register (CLOCK_ROOT85_CONTROL_TOG)	32	RW	0000_0000
2AA0	Clock root working status (CLOCK_ROOT85_STATUS0)	32	RW	0000_0000
2AB0	Clock root access control (CLOCK_ROOT85_AUTHEN)	32	RW	FFFF_0000
2AB4	Clock root access control (CLOCK_ROOT85_AUTHEN_SET)	32	RW	FFFF_0000
2AB8	Clock root access control (CLOCK_ROOT85_AUTHEN_CLR)	32	RW	FFFF_0000
2ABC	Clock root access control (CLOCK_ROOT85_AUTHEN_TOG)	32	RW	FFFF_0000
2B00	Clock Root Control Register (CLOCK_ROOT86_CONTROL)	32	RW	0000_0000
2B04	Clock Root Control Register (CLOCK_ROOT86_CONTROL_SET)	32	RW	0000_0000
2B08	Clock Root Control Register (CLOCK_ROOT86_CONTROL_CLR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2B0C	Clock Root Control Register (CLOCK_ROOT86_CONTROL_TOG)	32	RW	0000_0000
2B20	Clock root working status (CLOCK_ROOT86_STATUS0)	32	RW	0000_0000
2B30	Clock root access control (CLOCK_ROOT86_AUTHEN)	32	RW	FFFF_0000
2B34	Clock root access control (CLOCK_ROOT86_AUTHEN_SET)	32	RW	FFFF_0000
2B38	Clock root access control (CLOCK_ROOT86_AUTHEN_CLR)	32	RW	FFFF_0000
2B3C	Clock root access control (CLOCK_ROOT86_AUTHEN_TOG)	32	RW	FFFF_0000
2B80	Clock Root Control Register (CLOCK_ROOT87_CONTROL)	32	RW	0000_0000
2B84	Clock Root Control Register (CLOCK_ROOT87_CONTROL_SET)	32	RW	0000_0000
2B88	Clock Root Control Register (CLOCK_ROOT87_CONTROL_CLR)	32	RW	0000_0000
2B8C	Clock Root Control Register (CLOCK_ROOT87_CONTROL_TOG)	32	RW	0000_0000
2BA0	Clock root working status (CLOCK_ROOT87_STATUS0)	32	RW	0000_0000
2BB0	Clock root access control (CLOCK_ROOT87_AUTHEN)	32	RW	FFFF_0000
2BB4	Clock root access control (CLOCK_ROOT87_AUTHEN_SET)	32	RW	FFFF_0000
2BB8	Clock root access control (CLOCK_ROOT87_AUTHEN_CLR)	32	RW	FFFF_0000
2BBC	Clock root access control (CLOCK_ROOT87_AUTHEN_TOG)	32	RW	FFFF_0000
2C00	Clock Root Control Register (CLOCK_ROOT88_CONTROL)	32	RW	0000_0000
2C04	Clock Root Control Register (CLOCK_ROOT88_CONTROL_SET)	32	RW	0000_0000
2C08	Clock Root Control Register (CLOCK_ROOT88_CONTROL_CLR)	32	RW	0000_0000
2C0C	Clock Root Control Register (CLOCK_ROOT88_CONTROL_TOG)	32	RW	0000_0000
2C20	Clock root working status (CLOCK_ROOT88_STATUS0)	32	RW	0000_0000
2C30	Clock root access control (CLOCK_ROOT88_AUTHEN)	32	RW	FFFF_0000
2C34	Clock root access control (CLOCK_ROOT88_AUTHEN_SET)	32	RW	FFFF_0000
2C38	Clock root access control (CLOCK_ROOT88_AUTHEN_CLR)	32	RW	FFFF_0000
2C3C	Clock root access control (CLOCK_ROOT88_AUTHEN_TOG)	32	RW	FFFF_0000
2C80	Clock Root Control Register (CLOCK_ROOT89_CONTROL)	32	RW	0000_0000
2C84	Clock Root Control Register (CLOCK_ROOT89_CONTROL_SET)	32	RW	0000_0000
2C88	Clock Root Control Register (CLOCK_ROOT89_CONTROL_CLR)	32	RW	0000_0000
2C8C	Clock Root Control Register (CLOCK_ROOT89_CONTROL_TOG)	32	RW	0000_0000
2CA0	Clock root working status (CLOCK_ROOT89_STATUS0)	32	RW	0000_0000
2CB0	Clock root access control (CLOCK_ROOT89_AUTHEN)	32	RW	FFFF_0000
2CB4	Clock root access control (CLOCK_ROOT89_AUTHEN_SET)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2CB8	Clock root access control (CLOCK_ROOT89_AUTHEN_CLR)	32	RW	FFFF_0000
2CBC	Clock root access control (CLOCK_ROOT89_AUTHEN_TOG)	32	RW	FFFF_0000
2D00	Clock Root Control Register (CLOCK_ROOT90_CONTROL)	32	RW	0000_0000
2D04	Clock Root Control Register (CLOCK_ROOT90_CONTROL_SET)	32	RW	0000_0000
2D08	Clock Root Control Register (CLOCK_ROOT90_CONTROL_CLR)	32	RW	0000_0000
2D0C	Clock Root Control Register (CLOCK_ROOT90_CONTROL_TOG)	32	RW	0000_0000
2D20	Clock root working status (CLOCK_ROOT90_STATUS0)	32	RW	0000_0000
2D30	Clock root access control (CLOCK_ROOT90_AUTHEN)	32	RW	FFFF_0000
2D34	Clock root access control (CLOCK_ROOT90_AUTHEN_SET)	32	RW	FFFF_0000
2D38	Clock root access control (CLOCK_ROOT90_AUTHEN_CLR)	32	RW	FFFF_0000
2D3C	Clock root access control (CLOCK_ROOT90_AUTHEN_TOG)	32	RW	FFFF_0000
2D80	Clock Root Control Register (CLOCK_ROOT91_CONTROL)	32	RW	0000_0000
2D84	Clock Root Control Register (CLOCK_ROOT91_CONTROL_SET)	32	RW	0000_0000
2D88	Clock Root Control Register (CLOCK_ROOT91_CONTROL_CLR)	32	RW	0000_0000
2D8C	Clock Root Control Register (CLOCK_ROOT91_CONTROL_TOG)	32	RW	0000_0000
2DA0	Clock root working status (CLOCK_ROOT91_STATUS0)	32	RW	0000_0000
2DB0	Clock root access control (CLOCK_ROOT91_AUTHEN)	32	RW	FFFF_0000
2DB4	Clock root access control (CLOCK_ROOT91_AUTHEN_SET)	32	RW	FFFF_0000
2DB8	Clock root access control (CLOCK_ROOT91_AUTHEN_CLR)	32	RW	FFFF_0000
2DBC	Clock root access control (CLOCK_ROOT91_AUTHEN_TOG)	32	RW	FFFF_0000
2E00	Clock Root Control Register (CLOCK_ROOT92_CONTROL)	32	RW	0000_0000
2E04	Clock Root Control Register (CLOCK_ROOT92_CONTROL_SET)	32	RW	0000_0000
2E08	Clock Root Control Register (CLOCK_ROOT92_CONTROL_CLR)	32	RW	0000_0000
2E0C	Clock Root Control Register (CLOCK_ROOT92_CONTROL_TOG)	32	RW	0000_0000
2E20	Clock root working status (CLOCK_ROOT92_STATUS0)	32	RW	0000_0000
2E30	Clock root access control (CLOCK_ROOT92_AUTHEN)	32	RW	FFFF_0000
2E34	Clock root access control (CLOCK_ROOT92_AUTHEN_SET)	32	RW	FFFF_0000
2E38	Clock root access control (CLOCK_ROOT92_AUTHEN_CLR)	32	RW	FFFF_0000
2E3C	Clock root access control (CLOCK_ROOT92_AUTHEN_TOG)	32	RW	FFFF_0000
2E80	Clock Root Control Register (CLOCK_ROOT93_CONTROL)	32	RW	0000_0000
2E84	Clock Root Control Register (CLOCK_ROOT93_CONTROL_SET)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2E88	Clock Root Control Register (CLOCK_ROOT93_CONTROL_CLR)	32	RW	0000_0000
2E8C	Clock Root Control Register (CLOCK_ROOT93_CONTROL_TOG)	32	RW	0000_0000
2EA0	Clock root working status (CLOCK_ROOT93_STATUS0)	32	RW	0000_0000
2EB0	Clock root access control (CLOCK_ROOT93_AUTHEN)	32	RW	FFFF_0000
2EB4	Clock root access control (CLOCK_ROOT93_AUTHEN_SET)	32	RW	FFFF_0000
2EB8	Clock root access control (CLOCK_ROOT93_AUTHEN_CLR)	32	RW	FFFF_0000
2EBC	Clock root access control (CLOCK_ROOT93_AUTHEN_TOG)	32	RW	FFFF_0000
2F00	Clock Root Control Register (CLOCK_ROOT94_CONTROL)	32	RW	0000_0000
2F04	Clock Root Control Register (CLOCK_ROOT94_CONTROL_SET)	32	RW	0000_0000
2F08	Clock Root Control Register (CLOCK_ROOT94_CONTROL_CLR)	32	RW	0000_0000
2F0C	Clock Root Control Register (CLOCK_ROOT94_CONTROL_TOG)	32	RW	0000_0000
2F20	Clock root working status (CLOCK_ROOT94_STATUS0)	32	RW	0000_0000
2F30	Clock root access control (CLOCK_ROOT94_AUTHEN)	32	RW	FFFF_0000
2F34	Clock root access control (CLOCK_ROOT94_AUTHEN_SET)	32	RW	FFFF_0000
2F38	Clock root access control (CLOCK_ROOT94_AUTHEN_CLR)	32	RW	FFFF_0000
2F3C	Clock root access control (CLOCK_ROOT94_AUTHEN_TOG)	32	RW	FFFF_0000
4800	General Purpose Register (GPR_SHARED0)	32	R	0000_0000
4804	General Purpose Register (GPR_SHARED0_SET)	32	R	0000_0000
4808	General Purpose Register (GPR_SHARED0_CLR)	32	R	0000_0000
480C	General Purpose Register (GPR_SHARED0_TOG)	32	R	0000_0000
4810	GPR access control (GPR_SHARED0_AUTHEN)	32	RW	FFFF_0000
4814	GPR access control (GPR_SHARED0_AUTHEN_SET)	32	RW	FFFF_0000
4818	GPR access control (GPR_SHARED0_AUTHEN_CLR)	32	RW	FFFF_0000
481C	GPR access control (GPR_SHARED0_AUTHEN_TOG)	32	RW	FFFF_0000
4820	General Purpose Register (GPR_SHARED1)	32	RW	0000_0000
4824	General Purpose Register (GPR_SHARED1_SET)	32	RW	0000_0000
4828	General Purpose Register (GPR_SHARED1_CLR)	32	RW	0000_0000
482C	General Purpose Register (GPR_SHARED1_TOG)	32	RW	0000_0000
4830	GPR access control (GPR_SHARED1_AUTHEN)	32	RW	FFFF_0000
4834	GPR access control (GPR_SHARED1_AUTHEN_SET)	32	RW	FFFF_0000
4838	GPR access control (GPR_SHARED1_AUTHEN_CLR)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
483C	GPR access control (GPR_SHARED1_AUTHEN_TOG)	32	RW	FFFF_0000
4840	General Purpose Register (GPR_SHARED2)	32	RW	0000_0000
4844	General Purpose Register (GPR_SHARED2_SET)	32	RW	0000_0000
4848	General Purpose Register (GPR_SHARED2_CLR)	32	RW	0000_0000
484C	General Purpose Register (GPR_SHARED2_TOG)	32	RW	0000_0000
4850	GPR access control (GPR_SHARED2_AUTHEN)	32	RW	FFFF_0000
4854	GPR access control (GPR_SHARED2_AUTHEN_SET)	32	RW	FFFF_0000
4858	GPR access control (GPR_SHARED2_AUTHEN_CLR)	32	RW	FFFF_0000
485C	GPR access control (GPR_SHARED2_AUTHEN_TOG)	32	RW	FFFF_0000
4860	General Purpose Register (GPR_SHARED3)	32	R	0000_0000
4864	General Purpose Register (GPR_SHARED3_SET)	32	R	0000_0000
4868	General Purpose Register (GPR_SHARED3_CLR)	32	R	0000_0000
486C	General Purpose Register (GPR_SHARED3_TOG)	32	R	0000_0000
4870	GPR access control (GPR_SHARED3_AUTHEN)	32	RW	FFFF_0000
4874	GPR access control (GPR_SHARED3_AUTHEN_SET)	32	RW	FFFF_0000
4878	GPR access control (GPR_SHARED3_AUTHEN_CLR)	32	RW	FFFF_0000
487C	GPR access control (GPR_SHARED3_AUTHEN_TOG)	32	RW	FFFF_0000
4880	General Purpose Register (GPR_SHARED4)	32	R	0000_0000
4884	General Purpose Register (GPR_SHARED4_SET)	32	R	0000_0000
4888	General Purpose Register (GPR_SHARED4_CLR)	32	R	0000_0000
488C	General Purpose Register (GPR_SHARED4_TOG)	32	R	0000_0000
4890	GPR access control (GPR_SHARED4_AUTHEN)	32	RW	FFFF_0000
4894	GPR access control (GPR_SHARED4_AUTHEN_SET)	32	RW	FFFF_0000
4898	GPR access control (GPR_SHARED4_AUTHEN_CLR)	32	RW	FFFF_0000
489C	GPR access control (GPR_SHARED4_AUTHEN_TOG)	32	RW	FFFF_0000
48A0	General Purpose Register (GPR_SHARED5)	32	R	0000_0000
48A4	General Purpose Register (GPR_SHARED5_SET)	32	R	0000_0000
48A8	General Purpose Register (GPR_SHARED5_CLR)	32	R	0000_0000
48AC	General Purpose Register (GPR_SHARED5_TOG)	32	R	0000_0000
48B0	GPR access control (GPR_SHARED5_AUTHEN)	32	RW	FFFF_0000
48B4	GPR access control (GPR_SHARED5_AUTHEN_SET)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
48B8	GPR access control (GPR_SHARED5_AUTHEN_CLR)	32	RW	FFFF_0000
48BC	GPR access control (GPR_SHARED5_AUTHEN_TOG)	32	RW	FFFF_0000
48C0	General Purpose Register (GPR_SHARED6)	32	R	0000_0000
48C4	General Purpose Register (GPR_SHARED6_SET)	32	R	0000_0000
48C8	General Purpose Register (GPR_SHARED6_CLR)	32	R	0000_0000
48CC	General Purpose Register (GPR_SHARED6_TOG)	32	R	0000_0000
48D0	GPR access control (GPR_SHARED6_AUTHEN)	32	RW	FFFF_0000
48D4	GPR access control (GPR_SHARED6_AUTHEN_SET)	32	RW	FFFF_0000
48D8	GPR access control (GPR_SHARED6_AUTHEN_CLR)	32	RW	FFFF_0000
48DC	GPR access control (GPR_SHARED6_AUTHEN_TOG)	32	RW	FFFF_0000
48E0	General Purpose Register (GPR_SHARED7)	32	R	0000_0000
48E4	General Purpose Register (GPR_SHARED7_SET)	32	R	0000_0000
48E8	General Purpose Register (GPR_SHARED7_CLR)	32	R	0000_0000
48EC	General Purpose Register (GPR_SHARED7_TOG)	32	R	0000_0000
48F0	GPR access control (GPR_SHARED7_AUTHEN)	32	RW	FFFF_0000
48F4	GPR access control (GPR_SHARED7_AUTHEN_SET)	32	RW	FFFF_0000
48F8	GPR access control (GPR_SHARED7_AUTHEN_CLR)	32	RW	FFFF_0000
48FC	GPR access control (GPR_SHARED7_AUTHEN_TOG)	32	RW	FFFF_0000
4C00	General puspose register (GPR_PRIVATE0)	32	RW	0000_0000
4C04	General puspose register (GPR_PRIVATE0_SET)	32	RW	0000_0000
4C08	General puspose register (GPR_PRIVATE0_CLR)	32	RW	0000_0000
4C0C	General puspose register (GPR_PRIVATE0_TOG)	32	RW	0000_0000
4C10	GPR access control (GPR_PRIVATE0_AUTHEN)	32	RW	FFFF_0000
4C14	GPR access control (GPR_PRIVATE0_AUTHEN_SET)	32	RW	FFFF_0000
4C18	GPR access control (GPR_PRIVATE0_AUTHEN_CLR)	32	RW	FFFF_0000
4C1C	GPR access control (GPR_PRIVATE0_AUTHEN_TOG)	32	RW	FFFF_0000
4C20	General puspose register (GPR_PRIVATE1)	32	RW	0000_0000
4C24	General puspose register (GPR_PRIVATE1_SET)	32	RW	0000_0000
4C28	General puspose register (GPR_PRIVATE1_CLR)	32	RW	0000_0000
4C2C	General puspose register (GPR_PRIVATE1_TOG)	32	RW	0000_0000
4C30	GPR access control (GPR_PRIVATE1_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4C34	GPR access control (GPR_PRIVATE1_AUTHEN_SET)	32	RW	FFFF_0000
4C38	GPR access control (GPR_PRIVATE1_AUTHEN_CLR)	32	RW	FFFF_0000
4C3C	GPR access control (GPR_PRIVATE1_AUTHEN_TOG)	32	RW	FFFF_0000
4C40	General puspose register (GPR_PRIVATE2)	32	RW	0000_0000
4C44	General puspose register (GPR_PRIVATE2_SET)	32	RW	0000_0000
4C48	General puspose register (GPR_PRIVATE2_CLR)	32	RW	0000_0000
4C4C	General puspose register (GPR_PRIVATE2_TOG)	32	RW	0000_0000
4C50	GPR access control (GPR_PRIVATE2_AUTHEN)	32	RW	FFFF_0000
4C54	GPR access control (GPR_PRIVATE2_AUTHEN_SET)	32	RW	FFFF_0000
4C58	GPR access control (GPR_PRIVATE2_AUTHEN_CLR)	32	RW	FFFF_0000
4C5C	GPR access control (GPR_PRIVATE2_AUTHEN_TOG)	32	RW	FFFF_0000
4C60	General puspose register (GPR_PRIVATE3)	32	RW	0000_0000
4C64	General puspose register (GPR_PRIVATE3_SET)	32	RW	0000_0000
4C68	General puspose register (GPR_PRIVATE3_CLR)	32	RW	0000_0000
4C6C	General puspose register (GPR_PRIVATE3_TOG)	32	RW	0000_0000
4C70	GPR access control (GPR_PRIVATE3_AUTHEN)	32	RW	FFFF_0000
4C74	GPR access control (GPR_PRIVATE3_AUTHEN_SET)	32	RW	FFFF_0000
4C78	GPR access control (GPR_PRIVATE3_AUTHEN_CLR)	32	RW	FFFF_0000
4C7C	GPR access control (GPR_PRIVATE3_AUTHEN_TOG)	32	RW	FFFF_0000
4C80	General puspose register (GPR_PRIVATE4)	32	RW	0000_0000
4C84	General puspose register (GPR_PRIVATE4_SET)	32	RW	0000_0000
4C88	General puspose register (GPR_PRIVATE4_CLR)	32	RW	0000_0000
4C8C	General puspose register (GPR_PRIVATE4_TOG)	32	RW	0000_0000
4C90	GPR access control (GPR_PRIVATE4_AUTHEN)	32	RW	FFFF_0000
4C94	GPR access control (GPR_PRIVATE4_AUTHEN_SET)	32	RW	FFFF_0000
4C98	GPR access control (GPR_PRIVATE4_AUTHEN_CLR)	32	RW	FFFF_0000
4C9C	GPR access control (GPR_PRIVATE4_AUTHEN_TOG)	32	RW	FFFF_0000
4CA0	General puspose register (GPR_PRIVATE5)	32	RW	0000_0000
4CA4	General puspose register (GPR_PRIVATE5_SET)	32	RW	0000_0000
4CA8	General puspose register (GPR_PRIVATE5_CLR)	32	RW	0000_0000
4CAC	General puspose register (GPR_PRIVATE5_TOG)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4CB0	GPR access control (GPR_PRIVATE5_AUTHEN)	32	RW	FFFF_0000
4CB4	GPR access control (GPR_PRIVATE5_AUTHEN_SET)	32	RW	FFFF_0000
4CB8	GPR access control (GPR_PRIVATE5_AUTHEN_CLR)	32	RW	FFFF_0000
4CBC	GPR access control (GPR_PRIVATE5_AUTHEN_TOG)	32	RW	FFFF_0000
4CC0	General puspose register (GPR_PRIVATE6)	32	RW	0000_0000
4CC4	General puspose register (GPR_PRIVATE6_SET)	32	RW	0000_0000
4CC8	General puspose register (GPR_PRIVATE6_CLR)	32	RW	0000_0000
4CCC	General puspose register (GPR_PRIVATE6_TOG)	32	RW	0000_0000
4CD0	GPR access control (GPR_PRIVATE6_AUTHEN)	32	RW	FFFF_0000
4CD4	GPR access control (GPR_PRIVATE6_AUTHEN_SET)	32	RW	FFFF_0000
4CD8	GPR access control (GPR_PRIVATE6_AUTHEN_CLR)	32	RW	FFFF_0000
4CDC	GPR access control (GPR_PRIVATE6_AUTHEN_TOG)	32	RW	FFFF_0000
4CE0	General puspose register (GPR_PRIVATE7)	32	RW	0000_0000
4CE4	General puspose register (GPR_PRIVATE7_SET)	32	RW	0000_0000
4CE8	General puspose register (GPR_PRIVATE7_CLR)	32	RW	0000_0000
4CEC	General puspose register (GPR_PRIVATE7_TOG)	32	RW	0000_0000
4CF0	GPR access control (GPR_PRIVATE7_AUTHEN)	32	RW	FFFF_0000
4CF4	GPR access control (GPR_PRIVATE7_AUTHEN_SET)	32	RW	FFFF_0000
4CF8	GPR access control (GPR_PRIVATE7_AUTHEN_CLR)	32	RW	FFFF_0000
4CFC	GPR access control (GPR_PRIVATE7_AUTHEN_TOG)	32	RW	FFFF_0000
5000	Clock source direct control (OSCPLL0_DIRECT)	32	RW	0000_0001
5004	Low power mode information transfer status (OSCPLL0_LPM_STATUS0)	32	R	0000_0000
5008	Low power mode information transfer status (OSCPLL0_LPM_STATUS1)	32	R	0000_0000
5010	Clock source low power mode setting (OSCPLL0_LPM0)	32	RW	0000_0000
5014	clock source low power mode setting (OSCPLL0_LPM1)	32	RW	0000_0000
501C	LPM setting of current CPU domain (OSCPLL0_LPM_CUR)	32	R	0000_0000
5020	Clock source working status (OSCPLL0_STATUS0)	32	R	0000_0001
5024	Clock source domain status (OSCPLL0_STATUS1)	32	R	0000_FFFF
5030	Clock Source access control (OSCPLL0_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
5040	Clock source direct control (OSCPLL1_DIRECT)	32	RW	0000_0001
5044	Low power mode information transfer status (OSCPLL1_LPM_STATUS0)	32	R	0000_0000
5048	Low power mode information transfer status (OSCPLL1_LPM_STATUS1)	32	R	0000_0000
5050	Clock source low power mode setting (OSCPLL1_LPM0)	32	RW	0000_0000
5054	clock source low power mode setting (OSCPLL1_LPM1)	32	RW	0000_0000
505C	LPM setting of current CPU domain (OSCPLL1_LPM_CUR)	32	R	0000_0000
5060	Clock source working status (OSCPLL1_STATUS0)	32	R	0000_0001
5064	Clock source domain status (OSCPLL1_STATUS1)	32	R	0000_FFFF
5070	Clock Source access control (OSCPLL1_AUTHEN)	32	RW	FFFF_0000
5080	Clock source direct control (OSCPLL2_DIRECT)	32	RW	0000_0001
5084	Low power mode information transfer status (OSCPLL2_LPM_STATUS0)	32	R	0000_0000
5088	Low power mode information transfer status (OSCPLL2_LPM_STATUS1)	32	R	0000_0000
5090	Clock source low power mode setting (OSCPLL2_LPM0)	32	RW	0000_0000
5094	clock source low power mode setting (OSCPLL2_LPM1)	32	RW	0000_0000
509C	LPM setting of current CPU domain (OSCPLL2_LPM_CUR)	32	R	0000_0000
50A0	Clock source working status (OSCPLL2_STATUS0)	32	R	0000_0001
50A4	Clock source domain status (OSCPLL2_STATUS1)	32	R	0000_FFFF
50B0	Clock Source access control (OSCPLL2_AUTHEN)	32	RW	FFFF_0000
50C0	Clock source direct control (OSCPLL3_DIRECT)	32	RW	0000_0001
50C4	Low power mode information transfer status (OSCPLL3_LPM_STATUS0)	32	R	0000_0000
50C8	Low power mode information transfer status (OSCPLL3_LPM_STATUS1)	32	R	0000_0000
50D0	Clock source low power mode setting (OSCPLL3_LPM0)	32	RW	0000_0000
50D4	clock source low power mode setting (OSCPLL3_LPM1)	32	RW	0000_0000
50DC	LPM setting of current CPU domain (OSCPLL3_LPM_CUR)	32	R	0000_0000
50E0	Clock source working status (OSCPLL3_STATUS0)	32	R	0000_0001
50E4	Clock source domain status (OSCPLL3_STATUS1)	32	R	0000_FFFF
50F0	Clock Source access control (OSCPLL3_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
5100	Clock source direct control (OSCPLL4_DIRECT)	32	RW	0000_0001
5104	Low power mode information transfer status (OSCPLL4_LPM_STATUS0)	32	R	0000_0000
5108	Low power mode information transfer status (OSCPLL4_LPM_STATUS1)	32	R	0000_0000
5110	Clock source low power mode setting (OSCPLL4_LPM0)	32	RW	0000_0000
5114	clock source low power mode setting (OSCPLL4_LPM1)	32	RW	0000_0000
511C	LPM setting of current CPU domain (OSCPLL4_LPM_CUR)	32	R	0000_0000
5120	Clock source working status (OSCPLL4_STATUS0)	32	R	0000_0001
5124	Clock source domain status (OSCPLL4_STATUS1)	32	R	0000_FFFF
5130	Clock Source access control (OSCPLL4_AUTHEN)	32	RW	FFFF_0000
5140	Clock source direct control (OSCPLL5_DIRECT)	32	RW	0000_0001
5144	Low power mode information transfer status (OSCPLL5_LPM_STATUS0)	32	R	0000_0000
5148	Low power mode information transfer status (OSCPLL5_LPM_STATUS1)	32	R	0000_0000
5150	Clock source low power mode setting (OSCPLL5_LPM0)	32	RW	0000_0000
5154	clock source low power mode setting (OSCPLL5_LPM1)	32	RW	0000_0000
515C	LPM setting of current CPU domain (OSCPLL5_LPM_CUR)	32	R	0000_0000
5160	Clock source working status (OSCPLL5_STATUS0)	32	R	0000_0001
5164	Clock source domain status (OSCPLL5_STATUS1)	32	R	0000_FFFF
5170	Clock Source access control (OSCPLL5_AUTHEN)	32	RW	FFFF_0000
5180	Clock source direct control (OSCPLL6_DIRECT)	32	RW	0000_0001
5184	Low power mode information transfer status (OSCPLL6_LPM_STATUS0)	32	R	0000_0000
5188	Low power mode information transfer status (OSCPLL6_LPM_STATUS1)	32	R	0000_0000
5190	Clock source low power mode setting (OSCPLL6_LPM0)	32	RW	0000_0000
5194	clock source low power mode setting (OSCPLL6_LPM1)	32	RW	0000_0000
519C	LPM setting of current CPU domain (OSCPLL6_LPM_CUR)	32	R	0000_0000
51A0	Clock source working status (OSCPLL6_STATUS0)	32	R	0000_0001
51A4	Clock source domain status (OSCPLL6_STATUS1)	32	R	0000_FFFF
51B0	Clock Source access control (OSCPLL6_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
51C0	Clock source direct control (OSCPLL7_DIRECT)	32	RW	0000_0001
51C4	Low power mode information transfer status (OSCPLL7_LPM_STATUS0)	32	R	0000_0000
51C8	Low power mode information transfer status (OSCPLL7_LPM_STATUS1)	32	R	0000_0000
51D0	Clock source low power mode setting (OSCPLL7_LPM0)	32	RW	0000_0000
51D4	clock source low power mode setting (OSCPLL7_LPM1)	32	RW	0000_0000
51DC	LPM setting of current CPU domain (OSCPLL7_LPM_CUR)	32	R	0000_0000
51E0	Clock source working status (OSCPLL7_STATUS0)	32	R	0000_0001
51E4	Clock source domain status (OSCPLL7_STATUS1)	32	R	0000_FFFF
51F0	Clock Source access control (OSCPLL7_AUTHEN)	32	RW	FFFF_0000
5200	Clock source direct control (OSCPLL8_DIRECT)	32	RW	0000_0001
5204	Low power mode information transfer status (OSCPLL8_LPM_STATUS0)	32	R	0000_0000
5208	Low power mode information transfer status (OSCPLL8_LPM_STATUS1)	32	R	0000_0000
5210	Clock source low power mode setting (OSCPLL8_LPM0)	32	RW	0000_0000
5214	clock source low power mode setting (OSCPLL8_LPM1)	32	RW	0000_0000
521C	LPM setting of current CPU domain (OSCPLL8_LPM_CUR)	32	R	0000_0000
5220	Clock source working status (OSCPLL8_STATUS0)	32	R	0000_0001
5224	Clock source domain status (OSCPLL8_STATUS1)	32	R	0000_FFFF
5230	Clock Source access control (OSCPLL8_AUTHEN)	32	RW	FFFF_0000
5240	Clock source direct control (OSCPLL9_DIRECT)	32	RW	0000_0001
5244	Low power mode information transfer status (OSCPLL9_LPM_STATUS0)	32	R	0000_0000
5248	Low power mode information transfer status (OSCPLL9_LPM_STATUS1)	32	R	0000_0000
5250	Clock source low power mode setting (OSCPLL9_LPM0)	32	RW	0000_0000
5254	clock source low power mode setting (OSCPLL9_LPM1)	32	RW	0000_0000
525C	LPM setting of current CPU domain (OSCPLL9_LPM_CUR)	32	R	0000_0000
5260	Clock source working status (OSCPLL9_STATUS0)	32	R	0000_0001
5264	Clock source domain status (OSCPLL9_STATUS1)	32	R	0000_FFFF
5270	Clock Source access control (OSCPLL9_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
5280	Clock source direct control (OSCPLL10_DIRECT)	32	RW	0000_0001
5284	Low power mode information transfer status (OSCPLL10_LPM_STATUS0)	32	R	0000_0000
5288	Low power mode information transfer status (OSCPLL10_LPM_STATUS1)	32	R	0000_0000
5290	Clock source low power mode setting (OSCPLL10_LPM0)	32	RW	0000_0000
5294	clock source low power mode setting (OSCPLL10_LPM1)	32	RW	0000_0000
529C	LPM setting of current CPU domain (OSCPLL10_LPM_CUR)	32	R	0000_0000
52A0	Clock source working status (OSCPLL10_STATUS0)	32	R	0000_0001
52A4	Clock source domain status (OSCPLL10_STATUS1)	32	R	0000_FFFF
52B0	Clock Source access control (OSCPLL10_AUTHEN)	32	RW	FFFF_0000
52C0	Clock source direct control (OSCPLL11_DIRECT)	32	RW	0000_0001
52C4	Low power mode information transfer status (OSCPLL11_LPM_STATUS0)	32	R	0000_0000
52C8	Low power mode information transfer status (OSCPLL11_LPM_STATUS1)	32	R	0000_0000
52D0	Clock source low power mode setting (OSCPLL11_LPM0)	32	RW	0000_0000
52D4	clock source low power mode setting (OSCPLL11_LPM1)	32	RW	0000_0000
52DC	LPM setting of current CPU domain (OSCPLL11_LPM_CUR)	32	R	0000_0000
52E0	Clock source working status (OSCPLL11_STATUS0)	32	R	0000_0001
52E4	Clock source domain status (OSCPLL11_STATUS1)	32	R	0000_FFFF
52F0	Clock Source access control (OSCPLL11_AUTHEN)	32	RW	FFFF_0000
5300	Clock source direct control (OSCPLL12_DIRECT)	32	RW	0000_0001
5304	Low power mode information transfer status (OSCPLL12_LPM_STATUS0)	32	R	0000_0000
5308	Low power mode information transfer status (OSCPLL12_LPM_STATUS1)	32	R	0000_0000
5310	Clock source low power mode setting (OSCPLL12_LPM0)	32	RW	0000_0000
5314	clock source low power mode setting (OSCPLL12_LPM1)	32	RW	0000_0000
531C	LPM setting of current CPU domain (OSCPLL12_LPM_CUR)	32	R	0000_0000
5320	Clock source working status (OSCPLL12_STATUS0)	32	R	0000_0001
5324	Clock source domain status (OSCPLL12_STATUS1)	32	R	0000_FFFF
5330	Clock Source access control (OSCPLL12_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
5340	Clock source direct control (OSCPLL13_DIRECT)	32	RW	0000_0001
5344	Low power mode information transfer status (OSCPLL13_LPM_STATUS0)	32	R	0000_0000
5348	Low power mode information transfer status (OSCPLL13_LPM_STATUS1)	32	R	0000_0000
5350	Clock source low power mode setting (OSCPLL13_LPM0)	32	RW	0000_0000
5354	clock source low power mode setting (OSCPLL13_LPM1)	32	RW	0000_0000
535C	LPM setting of current CPU domain (OSCPLL13_LPM_CUR)	32	R	0000_0000
5360	Clock source working status (OSCPLL13_STATUS0)	32	R	0000_0001
5364	Clock source domain status (OSCPLL13_STATUS1)	32	R	0000_FFFF
5370	Clock Source access control (OSCPLL13_AUTHEN)	32	RW	FFFF_0000
5380	Clock source direct control (OSCPLL14_DIRECT)	32	RW	0000_0001
5384	Low power mode information transfer status (OSCPLL14_LPM_STATUS0)	32	R	0000_0000
5388	Low power mode information transfer status (OSCPLL14_LPM_STATUS1)	32	R	0000_0000
5390	Clock source low power mode setting (OSCPLL14_LPM0)	32	RW	0000_0000
5394	clock source low power mode setting (OSCPLL14_LPM1)	32	RW	0000_0000
539C	LPM setting of current CPU domain (OSCPLL14_LPM_CUR)	32	R	0000_0000
53A0	Clock source working status (OSCPLL14_STATUS0)	32	R	0000_0001
53A4	Clock source domain status (OSCPLL14_STATUS1)	32	R	0000_FFFF
53B0	Clock Source access control (OSCPLL14_AUTHEN)	32	RW	FFFF_0000
53C0	Clock source direct control (OSCPLL15_DIRECT)	32	RW	0000_0001
53C4	Low power mode information transfer status (OSCPLL15_LPM_STATUS0)	32	R	0000_0000
53C8	Low power mode information transfer status (OSCPLL15_LPM_STATUS1)	32	R	0000_0000
53D0	Clock source low power mode setting (OSCPLL15_LPM0)	32	RW	0000_0000
53D4	clock source low power mode setting (OSCPLL15_LPM1)	32	RW	0000_0000
53DC	LPM setting of current CPU domain (OSCPLL15_LPM_CUR)	32	R	0000_0000
53E0	Clock source working status (OSCPLL15_STATUS0)	32	R	0000_0001
53E4	Clock source domain status (OSCPLL15_STATUS1)	32	R	0000_FFFF
53F0	Clock Source access control (OSCPLL15_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
5400	Clock source direct control (OSCPLL16_DIRECT)	32	RW	0000_0001
5404	Low power mode information transfer status (OSCPLL16_LPM_STATUS0)	32	R	0000_0000
5408	Low power mode information transfer status (OSCPLL16_LPM_STATUS1)	32	R	0000_0000
5410	Clock source low power mode setting (OSCPLL16_LPM0)	32	RW	0000_0000
5414	clock source low power mode setting (OSCPLL16_LPM1)	32	RW	0000_0000
541C	LPM setting of current CPU domain (OSCPLL16_LPM_CUR)	32	R	0000_0000
5420	Clock source working status (OSCPLL16_STATUS0)	32	R	0000_0001
5424	Clock source domain status (OSCPLL16_STATUS1)	32	R	0000_FFFF
5430	Clock Source access control (OSCPLL16_AUTHEN)	32	RW	FFFF_0000
5440	Clock source direct control (OSCPLL17_DIRECT)	32	RW	0000_0001
5444	Low power mode information transfer status (OSCPLL17_LPM_STATUS0)	32	R	0000_0000
5448	Low power mode information transfer status (OSCPLL17_LPM_STATUS1)	32	R	0000_0000
5450	Clock source low power mode setting (OSCPLL17_LPM0)	32	RW	0000_0000
5454	clock source low power mode setting (OSCPLL17_LPM1)	32	RW	0000_0000
545C	LPM setting of current CPU domain (OSCPLL17_LPM_CUR)	32	R	0000_0000
5460	Clock source working status (OSCPLL17_STATUS0)	32	R	0000_0001
5464	Clock source domain status (OSCPLL17_STATUS1)	32	R	0000_FFFF
5470	Clock Source access control (OSCPLL17_AUTHEN)	32	RW	FFFF_0000
5480	Clock source direct control (OSCPLL18_DIRECT)	32	RW	0000_0001
5484	Low power mode information transfer status (OSCPLL18_LPM_STATUS0)	32	R	0000_0000
5488	Low power mode information transfer status (OSCPLL18_LPM_STATUS1)	32	R	0000_0000
5490	Clock source low power mode setting (OSCPLL18_LPM0)	32	RW	0000_0000
5494	clock source low power mode setting (OSCPLL18_LPM1)	32	RW	0000_0000
549C	LPM setting of current CPU domain (OSCPLL18_LPM_CUR)	32	R	0000_0000
54A0	Clock source working status (OSCPLL18_STATUS0)	32	R	0000_0001
54A4	Clock source domain status (OSCPLL18_STATUS1)	32	R	0000_FFFF
54B0	Clock Source access control (OSCPLL18_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8000	LPCG direct control (LPCG0_DIRECT)	32	RW	0000_0001
8004	Low power mode information transfer status (LPCG0_LPM_STATUS0)	32	R	0000_0000
8008	Low power mode information transfer status (LPCG0_LPM_STATUS1)	32	R	0000_0000
8010	LPCG low power mode setting (LPCG0_LPM0)	32	RW	0000_0000
8014	LPCG low power mode setting (LPCG0_LPM1)	32	RW	0000_0000
801C	LPM setting of current CPU domain (LPCG0_LPM_CUR)	32	RW	0000_0000
8020	LPCG working status (LPCG0_STATUS0)	32	R	0000_0001
8024	LPCG domain status (LPCG0_STATUS1)	32	R	0000_FFFF
8030	LPCG access control (LPCG0_AUTHEN)	32	RW	FFFF_0000
8040	LPCG direct control (LPCG1_DIRECT)	32	RW	0000_0001
8044	Low power mode information transfer status (LPCG1_LPM_STATUS0)	32	R	0000_0000
8048	Low power mode information transfer status (LPCG1_LPM_STATUS1)	32	R	0000_0000
8050	LPCG low power mode setting (LPCG1_LPM0)	32	RW	0000_0000
8054	LPCG low power mode setting (LPCG1_LPM1)	32	RW	0000_0000
805C	LPM setting of current CPU domain (LPCG1_LPM_CUR)	32	RW	0000_0000
8060	LPCG working status (LPCG1_STATUS0)	32	R	0000_0001
8064	LPCG domain status (LPCG1_STATUS1)	32	R	0000_FFFF
8070	LPCG access control (LPCG1_AUTHEN)	32	RW	FFFF_0000
8080	LPCG direct control (LPCG2_DIRECT)	32	RW	0000_0001
8084	Low power mode information transfer status (LPCG2_LPM_STATUS0)	32	R	0000_0000
8088	Low power mode information transfer status (LPCG2_LPM_STATUS1)	32	R	0000_0000
8090	LPCG low power mode setting (LPCG2_LPM0)	32	RW	0000_0000
8094	LPCG low power mode setting (LPCG2_LPM1)	32	RW	0000_0000
809C	LPM setting of current CPU domain (LPCG2_LPM_CUR)	32	RW	0000_0000
80A0	LPCG working status (LPCG2_STATUS0)	32	R	0000_0001
80A4	LPCG domain status (LPCG2_STATUS1)	32	R	0000_FFFF
80B0	LPCG access control (LPCG2_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
80C0	LPCG direct control (LPCG3_DIRECT)	32	RW	0000_0001
80C4	Low power mode information transfer status (LPCG3_LPM_STATUS0)	32	R	0000_0000
80C8	Low power mode information transfer status (LPCG3_LPM_STATUS1)	32	R	0000_0000
80D0	LPCG low power mode setting (LPCG3_LPM0)	32	RW	0000_0000
80D4	LPCG low power mode setting (LPCG3_LPM1)	32	RW	0000_0000
80DC	LPM setting of current CPU domain (LPCG3_LPM_CUR)	32	RW	0000_0000
80E0	LPCG working status (LPCG3_STATUS0)	32	R	0000_0001
80E4	LPCG domain status (LPCG3_STATUS1)	32	R	0000_FFFF
80F0	LPCG access control (LPCG3_AUTHEN)	32	RW	FFFF_0000
8100	LPCG direct control (LPCG4_DIRECT)	32	RW	0000_0001
8104	Low power mode information transfer status (LPCG4_LPM_STATUS0)	32	R	0000_0000
8108	Low power mode information transfer status (LPCG4_LPM_STATUS1)	32	R	0000_0000
8110	LPCG low power mode setting (LPCG4_LPM0)	32	RW	0000_0000
8114	LPCG low power mode setting (LPCG4_LPM1)	32	RW	0000_0000
811C	LPM setting of current CPU domain (LPCG4_LPM_CUR)	32	RW	0000_0000
8120	LPCG working status (LPCG4_STATUS0)	32	R	0000_0001
8124	LPCG domain status (LPCG4_STATUS1)	32	R	0000_FFFF
8130	LPCG access control (LPCG4_AUTHEN)	32	RW	FFFF_0000
8140	LPCG direct control (LPCG5_DIRECT)	32	RW	0000_0001
8144	Low power mode information transfer status (LPCG5_LPM_STATUS0)	32	R	0000_0000
8148	Low power mode information transfer status (LPCG5_LPM_STATUS1)	32	R	0000_0000
8150	LPCG low power mode setting (LPCG5_LPM0)	32	RW	0000_0000
8154	LPCG low power mode setting (LPCG5_LPM1)	32	RW	0000_0000
815C	LPM setting of current CPU domain (LPCG5_LPM_CUR)	32	RW	0000_0000
8160	LPCG working status (LPCG5_STATUS0)	32	R	0000_0001
8164	LPCG domain status (LPCG5_STATUS1)	32	R	0000_FFFF
8170	LPCG access control (LPCG5_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8180	LPCG direct control (LPCG6_DIRECT)	32	RW	0000_0001
8184	Low power mode information transfer status (LPCG6_LPM_STATUS0)	32	R	0000_0000
8188	Low power mode information transfer status (LPCG6_LPM_STATUS1)	32	R	0000_0000
8190	LPCG low power mode setting (LPCG6_LPM0)	32	RW	0000_0000
8194	LPCG low power mode setting (LPCG6_LPM1)	32	RW	0000_0000
819C	LPM setting of current CPU domain (LPCG6_LPM_CUR)	32	RW	0000_0000
81A0	LPCG working status (LPCG6_STATUS0)	32	R	0000_0001
81A4	LPCG domain status (LPCG6_STATUS1)	32	R	0000_FFFF
81B0	LPCG access control (LPCG6_AUTHEN)	32	RW	FFFF_0000
81C0	LPCG direct control (LPCG7_DIRECT)	32	RW	0000_0001
81C4	Low power mode information transfer status (LPCG7_LPM_STATUS0)	32	R	0000_0000
81C8	Low power mode information transfer status (LPCG7_LPM_STATUS1)	32	R	0000_0000
81D0	LPCG low power mode setting (LPCG7_LPM0)	32	RW	0000_0000
81D4	LPCG low power mode setting (LPCG7_LPM1)	32	RW	0000_0000
81DC	LPM setting of current CPU domain (LPCG7_LPM_CUR)	32	RW	0000_0000
81E0	LPCG working status (LPCG7_STATUS0)	32	R	0000_0001
81E4	LPCG domain status (LPCG7_STATUS1)	32	R	0000_FFFF
81F0	LPCG access control (LPCG7_AUTHEN)	32	RW	FFFF_0000
8200	LPCG direct control (LPCG8_DIRECT)	32	RW	0000_0001
8204	Low power mode information transfer status (LPCG8_LPM_STATUS0)	32	R	0000_0000
8208	Low power mode information transfer status (LPCG8_LPM_STATUS1)	32	R	0000_0000
8210	LPCG low power mode setting (LPCG8_LPM0)	32	RW	0000_0000
8214	LPCG low power mode setting (LPCG8_LPM1)	32	RW	0000_0000
821C	LPM setting of current CPU domain (LPCG8_LPM_CUR)	32	RW	0000_0000
8220	LPCG working status (LPCG8_STATUS0)	32	R	0000_0001
8224	LPCG domain status (LPCG8_STATUS1)	32	R	0000_FFFF
8230	LPCG access control (LPCG8_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8240	LPCG direct control (LPCG9_DIRECT)	32	RW	0000_0001
8244	Low power mode information transfer status (LPCG9_LPM_STATUS0)	32	R	0000_0000
8248	Low power mode information transfer status (LPCG9_LPM_STATUS1)	32	R	0000_0000
8250	LPCG low power mode setting (LPCG9_LPM0)	32	RW	0000_0000
8254	LPCG low power mode setting (LPCG9_LPM1)	32	RW	0000_0000
825C	LPM setting of current CPU domain (LPCG9_LPM_CUR)	32	RW	0000_0000
8260	LPCG working status (LPCG9_STATUS0)	32	R	0000_0001
8264	LPCG domain status (LPCG9_STATUS1)	32	R	0000_FFFF
8270	LPCG access control (LPCG9_AUTHEN)	32	RW	FFFF_0000
8280	LPCG direct control (LPCG10_DIRECT)	32	RW	0000_0001
8284	Low power mode information transfer status (LPCG10_LPM_STATUS0)	32	R	0000_0000
8288	Low power mode information transfer status (LPCG10_LPM_STATUS1)	32	R	0000_0000
8290	LPCG low power mode setting (LPCG10_LPM0)	32	RW	0000_0000
8294	LPCG low power mode setting (LPCG10_LPM1)	32	RW	0000_0000
829C	LPM setting of current CPU domain (LPCG10_LPM_CUR)	32	RW	0000_0000
82A0	LPCG working status (LPCG10_STATUS0)	32	R	0000_0001
82A4	LPCG domain status (LPCG10_STATUS1)	32	R	0000_FFFF
82B0	LPCG access control (LPCG10_AUTHEN)	32	RW	FFFF_0000
82C0	LPCG direct control (LPCG11_DIRECT)	32	RW	0000_0001
82C4	Low power mode information transfer status (LPCG11_LPM_STATUS0)	32	R	0000_0000
82C8	Low power mode information transfer status (LPCG11_LPM_STATUS1)	32	R	0000_0000
82D0	LPCG low power mode setting (LPCG11_LPM0)	32	RW	0000_0000
82D4	LPCG low power mode setting (LPCG11_LPM1)	32	RW	0000_0000
82DC	LPM setting of current CPU domain (LPCG11_LPM_CUR)	32	RW	0000_0000
82E0	LPCG working status (LPCG11_STATUS0)	32	R	0000_0001
82E4	LPCG domain status (LPCG11_STATUS1)	32	R	0000_FFFF
82F0	LPCG access control (LPCG11_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8300	LPCG direct control (LPCG12_DIRECT)	32	RW	0000_0001
8304	Low power mode information transfer status (LPCG12_LPM_STATUS0)	32	R	0000_0000
8308	Low power mode information transfer status (LPCG12_LPM_STATUS1)	32	R	0000_0000
8310	LPCG low power mode setting (LPCG12_LPM0)	32	RW	0000_0000
8314	LPCG low power mode setting (LPCG12_LPM1)	32	RW	0000_0000
831C	LPM setting of current CPU domain (LPCG12_LPM_CUR)	32	RW	0000_0000
8320	LPCG working status (LPCG12_STATUS0)	32	R	0000_0001
8324	LPCG domain status (LPCG12_STATUS1)	32	R	0000_FFFF
8330	LPCG access control (LPCG12_AUTHEN)	32	RW	FFFF_0000
8340	LPCG direct control (LPCG13_DIRECT)	32	RW	0000_0001
8344	Low power mode information transfer status (LPCG13_LPM_STATUS0)	32	R	0000_0000
8348	Low power mode information transfer status (LPCG13_LPM_STATUS1)	32	R	0000_0000
8350	LPCG low power mode setting (LPCG13_LPM0)	32	RW	0000_0000
8354	LPCG low power mode setting (LPCG13_LPM1)	32	RW	0000_0000
835C	LPM setting of current CPU domain (LPCG13_LPM_CUR)	32	RW	0000_0000
8360	LPCG working status (LPCG13_STATUS0)	32	R	0000_0001
8364	LPCG domain status (LPCG13_STATUS1)	32	R	0000_FFFF
8370	LPCG access control (LPCG13_AUTHEN)	32	RW	FFFF_0000
8380	LPCG direct control (LPCG14_DIRECT)	32	RW	0000_0001
8384	Low power mode information transfer status (LPCG14_LPM_STATUS0)	32	R	0000_0000
8388	Low power mode information transfer status (LPCG14_LPM_STATUS1)	32	R	0000_0000
8390	LPCG low power mode setting (LPCG14_LPM0)	32	RW	0000_0000
8394	LPCG low power mode setting (LPCG14_LPM1)	32	RW	0000_0000
839C	LPM setting of current CPU domain (LPCG14_LPM_CUR)	32	RW	0000_0000
83A0	LPCG working status (LPCG14_STATUS0)	32	R	0000_0001
83A4	LPCG domain status (LPCG14_STATUS1)	32	R	0000_FFFF
83B0	LPCG access control (LPCG14_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
83C0	LPCG direct control (LPCG15_DIRECT)	32	RW	0000_0001
83C4	Low power mode information transfer status (LPCG15_LPM_STATUS0)	32	R	0000_0000
83C8	Low power mode information transfer status (LPCG15_LPM_STATUS1)	32	R	0000_0000
83D0	LPCG low power mode setting (LPCG15_LPM0)	32	RW	0000_0000
83D4	LPCG low power mode setting (LPCG15_LPM1)	32	RW	0000_0000
83DC	LPM setting of current CPU domain (LPCG15_LPM_CUR)	32	RW	0000_0000
83E0	LPCG working status (LPCG15_STATUS0)	32	R	0000_0001
83E4	LPCG domain status (LPCG15_STATUS1)	32	R	0000_FFFF
83F0	LPCG access control (LPCG15_AUTHEN)	32	RW	FFFF_0000
8400	LPCG direct control (LPCG16_DIRECT)	32	RW	0000_0001
8404	Low power mode information transfer status (LPCG16_LPM_STATUS0)	32	R	0000_0000
8408	Low power mode information transfer status (LPCG16_LPM_STATUS1)	32	R	0000_0000
8410	LPCG low power mode setting (LPCG16_LPM0)	32	RW	0000_0000
8414	LPCG low power mode setting (LPCG16_LPM1)	32	RW	0000_0000
841C	LPM setting of current CPU domain (LPCG16_LPM_CUR)	32	RW	0000_0000
8420	LPCG working status (LPCG16_STATUS0)	32	R	0000_0001
8424	LPCG domain status (LPCG16_STATUS1)	32	R	0000_FFFF
8430	LPCG access control (LPCG16_AUTHEN)	32	RW	FFFF_0000
8440	LPCG direct control (LPCG17_DIRECT)	32	RW	0000_0001
8444	Low power mode information transfer status (LPCG17_LPM_STATUS0)	32	R	0000_0000
8448	Low power mode information transfer status (LPCG17_LPM_STATUS1)	32	R	0000_0000
8450	LPCG low power mode setting (LPCG17_LPM0)	32	RW	0000_0000
8454	LPCG low power mode setting (LPCG17_LPM1)	32	RW	0000_0000
845C	LPM setting of current CPU domain (LPCG17_LPM_CUR)	32	RW	0000_0000
8460	LPCG working status (LPCG17_STATUS0)	32	R	0000_0001
8464	LPCG domain status (LPCG17_STATUS1)	32	R	0000_FFFF
8470	LPCG access control (LPCG17_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8480	LPCG direct control (LPCG18_DIRECT)	32	RW	0000_0001
8484	Low power mode information transfer status (LPCG18_LPM_STATUS0)	32	R	0000_0000
8488	Low power mode information transfer status (LPCG18_LPM_STATUS1)	32	R	0000_0000
8490	LPCG low power mode setting (LPCG18_LPM0)	32	RW	0000_0000
8494	LPCG low power mode setting (LPCG18_LPM1)	32	RW	0000_0000
849C	LPM setting of current CPU domain (LPCG18_LPM_CUR)	32	RW	0000_0000
84A0	LPCG working status (LPCG18_STATUS0)	32	R	0000_0001
84A4	LPCG domain status (LPCG18_STATUS1)	32	R	0000_FFFF
84B0	LPCG access control (LPCG18_AUTHEN)	32	RW	FFFF_0000
84C0	LPCG direct control (LPCG19_DIRECT)	32	RW	0000_0001
84C4	Low power mode information transfer status (LPCG19_LPM_STATUS0)	32	R	0000_0000
84C8	Low power mode information transfer status (LPCG19_LPM_STATUS1)	32	R	0000_0000
84D0	LPCG low power mode setting (LPCG19_LPM0)	32	RW	0000_0000
84D4	LPCG low power mode setting (LPCG19_LPM1)	32	RW	0000_0000
84DC	LPM setting of current CPU domain (LPCG19_LPM_CUR)	32	RW	0000_0000
84E0	LPCG working status (LPCG19_STATUS0)	32	R	0000_0001
84E4	LPCG domain status (LPCG19_STATUS1)	32	R	0000_FFFF
84F0	LPCG access control (LPCG19_AUTHEN)	32	RW	FFFF_0000
8500	LPCG direct control (LPCG20_DIRECT)	32	RW	0000_0001
8504	Low power mode information transfer status (LPCG20_LPM_STATUS0)	32	R	0000_0000
8508	Low power mode information transfer status (LPCG20_LPM_STATUS1)	32	R	0000_0000
8510	LPCG low power mode setting (LPCG20_LPM0)	32	RW	0000_0000
8514	LPCG low power mode setting (LPCG20_LPM1)	32	RW	0000_0000
851C	LPM setting of current CPU domain (LPCG20_LPM_CUR)	32	RW	0000_0000
8520	LPCG working status (LPCG20_STATUS0)	32	R	0000_0001
8524	LPCG domain status (LPCG20_STATUS1)	32	R	0000_FFFF
8530	LPCG access control (LPCG20_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8540	LPCG direct control (LPCG21_DIRECT)	32	RW	0000_0001
8544	Low power mode information transfer status (LPCG21_LPM_STATUS0)	32	R	0000_0000
8548	Low power mode information transfer status (LPCG21_LPM_STATUS1)	32	R	0000_0000
8550	LPCG low power mode setting (LPCG21_LPM0)	32	RW	0000_0000
8554	LPCG low power mode setting (LPCG21_LPM1)	32	RW	0000_0000
855C	LPM setting of current CPU domain (LPCG21_LPM_CUR)	32	RW	0000_0000
8560	LPCG working status (LPCG21_STATUS0)	32	R	0000_0001
8564	LPCG domain status (LPCG21_STATUS1)	32	R	0000_FFFF
8570	LPCG access control (LPCG21_AUTHEN)	32	RW	FFFF_0000
8580	LPCG direct control (LPCG22_DIRECT)	32	RW	0000_0001
8584	Low power mode information transfer status (LPCG22_LPM_STATUS0)	32	R	0000_0000
8588	Low power mode information transfer status (LPCG22_LPM_STATUS1)	32	R	0000_0000
8590	LPCG low power mode setting (LPCG22_LPM0)	32	RW	0000_0000
8594	LPCG low power mode setting (LPCG22_LPM1)	32	RW	0000_0000
859C	LPM setting of current CPU domain (LPCG22_LPM_CUR)	32	RW	0000_0000
85A0	LPCG working status (LPCG22_STATUS0)	32	R	0000_0001
85A4	LPCG domain status (LPCG22_STATUS1)	32	R	0000_FFFF
85B0	LPCG access control (LPCG22_AUTHEN)	32	RW	FFFF_0000
85C0	LPCG direct control (LPCG23_DIRECT)	32	RW	0000_0001
85C4	Low power mode information transfer status (LPCG23_LPM_STATUS0)	32	R	0000_0000
85C8	Low power mode information transfer status (LPCG23_LPM_STATUS1)	32	R	0000_0000
85D0	LPCG low power mode setting (LPCG23_LPM0)	32	RW	0000_0000
85D4	LPCG low power mode setting (LPCG23_LPM1)	32	RW	0000_0000
85DC	LPM setting of current CPU domain (LPCG23_LPM_CUR)	32	RW	0000_0000
85E0	LPCG working status (LPCG23_STATUS0)	32	R	0000_0001
85E4	LPCG domain status (LPCG23_STATUS1)	32	R	0000_FFFF
85F0	LPCG access control (LPCG23_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8600	LPCG direct control (LPCG24_DIRECT)	32	RW	0000_0001
8604	Low power mode information transfer status (LPCG24_LPM_STATUS0)	32	R	0000_0000
8608	Low power mode information transfer status (LPCG24_LPM_STATUS1)	32	R	0000_0000
8610	LPCG low power mode setting (LPCG24_LPM0)	32	RW	0000_0000
8614	LPCG low power mode setting (LPCG24_LPM1)	32	RW	0000_0000
861C	LPM setting of current CPU domain (LPCG24_LPM_CUR)	32	RW	0000_0000
8620	LPCG working status (LPCG24_STATUS0)	32	R	0000_0001
8624	LPCG domain status (LPCG24_STATUS1)	32	R	0000_FFFF
8630	LPCG access control (LPCG24_AUTHEN)	32	RW	FFFF_0000
8640	LPCG direct control (LPCG25_DIRECT)	32	RW	0000_0001
8644	Low power mode information transfer status (LPCG25_LPM_STATUS0)	32	R	0000_0000
8648	Low power mode information transfer status (LPCG25_LPM_STATUS1)	32	R	0000_0000
8650	LPCG low power mode setting (LPCG25_LPM0)	32	RW	0000_0000
8654	LPCG low power mode setting (LPCG25_LPM1)	32	RW	0000_0000
865C	LPM setting of current CPU domain (LPCG25_LPM_CUR)	32	RW	0000_0000
8660	LPCG working status (LPCG25_STATUS0)	32	R	0000_0001
8664	LPCG domain status (LPCG25_STATUS1)	32	R	0000_FFFF
8670	LPCG access control (LPCG25_AUTHEN)	32	RW	FFFF_0000
8680	LPCG direct control (LPCG26_DIRECT)	32	RW	0000_0001
8684	Low power mode information transfer status (LPCG26_LPM_STATUS0)	32	R	0000_0000
8688	Low power mode information transfer status (LPCG26_LPM_STATUS1)	32	R	0000_0000
8690	LPCG low power mode setting (LPCG26_LPM0)	32	RW	0000_0000
8694	LPCG low power mode setting (LPCG26_LPM1)	32	RW	0000_0000
869C	LPM setting of current CPU domain (LPCG26_LPM_CUR)	32	RW	0000_0000
86A0	LPCG working status (LPCG26_STATUS0)	32	R	0000_0001
86A4	LPCG domain status (LPCG26_STATUS1)	32	R	0000_FFFF
86B0	LPCG access control (LPCG26_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
86C0	LPCG direct control (LPCG27_DIRECT)	32	RW	0000_0001
86C4	Low power mode information transfer status (LPCG27_LPM_STATUS0)	32	R	0000_0000
86C8	Low power mode information transfer status (LPCG27_LPM_STATUS1)	32	R	0000_0000
86D0	LPCG low power mode setting (LPCG27_LPM0)	32	RW	0000_0000
86D4	LPCG low power mode setting (LPCG27_LPM1)	32	RW	0000_0000
86DC	LPM setting of current CPU domain (LPCG27_LPM_CUR)	32	RW	0000_0000
86E0	LPCG working status (LPCG27_STATUS0)	32	R	0000_0001
86E4	LPCG domain status (LPCG27_STATUS1)	32	R	0000_FFFF
86F0	LPCG access control (LPCG27_AUTHEN)	32	RW	FFFF_0000
8700	LPCG direct control (LPCG28_DIRECT)	32	RW	0000_0001
8704	Low power mode information transfer status (LPCG28_LPM_STATUS0)	32	R	0000_0000
8708	Low power mode information transfer status (LPCG28_LPM_STATUS1)	32	R	0000_0000
8710	LPCG low power mode setting (LPCG28_LPM0)	32	RW	0000_0000
8714	LPCG low power mode setting (LPCG28_LPM1)	32	RW	0000_0000
871C	LPM setting of current CPU domain (LPCG28_LPM_CUR)	32	RW	0000_0000
8720	LPCG working status (LPCG28_STATUS0)	32	R	0000_0001
8724	LPCG domain status (LPCG28_STATUS1)	32	R	0000_FFFF
8730	LPCG access control (LPCG28_AUTHEN)	32	RW	FFFF_0000
8740	LPCG direct control (LPCG29_DIRECT)	32	RW	0000_0001
8744	Low power mode information transfer status (LPCG29_LPM_STATUS0)	32	R	0000_0000
8748	Low power mode information transfer status (LPCG29_LPM_STATUS1)	32	R	0000_0000
8750	LPCG low power mode setting (LPCG29_LPM0)	32	RW	0000_0000
8754	LPCG low power mode setting (LPCG29_LPM1)	32	RW	0000_0000
875C	LPM setting of current CPU domain (LPCG29_LPM_CUR)	32	RW	0000_0000
8760	LPCG working status (LPCG29_STATUS0)	32	R	0000_0001
8764	LPCG domain status (LPCG29_STATUS1)	32	R	0000_FFFF
8770	LPCG access control (LPCG29_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8780	LPCG direct control (LPCG30_DIRECT)	32	RW	0000_0001
8784	Low power mode information transfer status (LPCG30_LPM_STATUS0)	32	R	0000_0000
8788	Low power mode information transfer status (LPCG30_LPM_STATUS1)	32	R	0000_0000
8790	LPCG low power mode setting (LPCG30_LPM0)	32	RW	0000_0000
8794	LPCG low power mode setting (LPCG30_LPM1)	32	RW	0000_0000
879C	LPM setting of current CPU domain (LPCG30_LPM_CUR)	32	RW	0000_0000
87A0	LPCG working status (LPCG30_STATUS0)	32	R	0000_0001
87A4	LPCG domain status (LPCG30_STATUS1)	32	R	0000_FFFF
87B0	LPCG access control (LPCG30_AUTHEN)	32	RW	FFFF_0000
87C0	LPCG direct control (LPCG31_DIRECT)	32	RW	0000_0001
87C4	Low power mode information transfer status (LPCG31_LPM_STATUS0)	32	R	0000_0000
87C8	Low power mode information transfer status (LPCG31_LPM_STATUS1)	32	R	0000_0000
87D0	LPCG low power mode setting (LPCG31_LPM0)	32	RW	0000_0000
87D4	LPCG low power mode setting (LPCG31_LPM1)	32	RW	0000_0000
87DC	LPM setting of current CPU domain (LPCG31_LPM_CUR)	32	RW	0000_0000
87E0	LPCG working status (LPCG31_STATUS0)	32	R	0000_0001
87E4	LPCG domain status (LPCG31_STATUS1)	32	R	0000_FFFF
87F0	LPCG access control (LPCG31_AUTHEN)	32	RW	FFFF_0000
8800	LPCG direct control (LPCG32_DIRECT)	32	RW	0000_0001
8804	Low power mode information transfer status (LPCG32_LPM_STATUS0)	32	R	0000_0000
8808	Low power mode information transfer status (LPCG32_LPM_STATUS1)	32	R	0000_0000
8810	LPCG low power mode setting (LPCG32_LPM0)	32	RW	0000_0000
8814	LPCG low power mode setting (LPCG32_LPM1)	32	RW	0000_0000
881C	LPM setting of current CPU domain (LPCG32_LPM_CUR)	32	RW	0000_0000
8820	LPCG working status (LPCG32_STATUS0)	32	R	0000_0001
8824	LPCG domain status (LPCG32_STATUS1)	32	R	0000_FFFF
8830	LPCG access control (LPCG32_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8840	LPCG direct control (LPCG33_DIRECT)	32	RW	0000_0001
8844	Low power mode information transfer status (LPCG33_LPM_STATUS0)	32	R	0000_0000
8848	Low power mode information transfer status (LPCG33_LPM_STATUS1)	32	R	0000_0000
8850	LPCG low power mode setting (LPCG33_LPM0)	32	RW	0000_0000
8854	LPCG low power mode setting (LPCG33_LPM1)	32	RW	0000_0000
885C	LPM setting of current CPU domain (LPCG33_LPM_CUR)	32	RW	0000_0000
8860	LPCG working status (LPCG33_STATUS0)	32	R	0000_0001
8864	LPCG domain status (LPCG33_STATUS1)	32	R	0000_FFFF
8870	LPCG access control (LPCG33_AUTHEN)	32	RW	FFFF_0000
8880	LPCG direct control (LPCG34_DIRECT)	32	RW	0000_0001
8884	Low power mode information transfer status (LPCG34_LPM_STATUS0)	32	R	0000_0000
8888	Low power mode information transfer status (LPCG34_LPM_STATUS1)	32	R	0000_0000
8890	LPCG low power mode setting (LPCG34_LPM0)	32	RW	0000_0000
8894	LPCG low power mode setting (LPCG34_LPM1)	32	RW	0000_0000
889C	LPM setting of current CPU domain (LPCG34_LPM_CUR)	32	RW	0000_0000
88A0	LPCG working status (LPCG34_STATUS0)	32	R	0000_0001
88A4	LPCG domain status (LPCG34_STATUS1)	32	R	0000_FFFF
88B0	LPCG access control (LPCG34_AUTHEN)	32	RW	FFFF_0000
88C0	LPCG direct control (LPCG35_DIRECT)	32	RW	0000_0001
88C4	Low power mode information transfer status (LPCG35_LPM_STATUS0)	32	R	0000_0000
88C8	Low power mode information transfer status (LPCG35_LPM_STATUS1)	32	R	0000_0000
88D0	LPCG low power mode setting (LPCG35_LPM0)	32	RW	0000_0000
88D4	LPCG low power mode setting (LPCG35_LPM1)	32	RW	0000_0000
88DC	LPM setting of current CPU domain (LPCG35_LPM_CUR)	32	RW	0000_0000
88E0	LPCG working status (LPCG35_STATUS0)	32	R	0000_0001
88E4	LPCG domain status (LPCG35_STATUS1)	32	R	0000_FFFF
88F0	LPCG access control (LPCG35_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8900	LPCG direct control (LPCG36_DIRECT)	32	RW	0000_0001
8904	Low power mode information transfer status (LPCG36_LPM_STATUS0)	32	R	0000_0000
8908	Low power mode information transfer status (LPCG36_LPM_STATUS1)	32	R	0000_0000
8910	LPCG low power mode setting (LPCG36_LPM0)	32	RW	0000_0000
8914	LPCG low power mode setting (LPCG36_LPM1)	32	RW	0000_0000
891C	LPM setting of current CPU domain (LPCG36_LPM_CUR)	32	RW	0000_0000
8920	LPCG working status (LPCG36_STATUS0)	32	R	0000_0001
8924	LPCG domain status (LPCG36_STATUS1)	32	R	0000_FFFF
8930	LPCG access control (LPCG36_AUTHEN)	32	RW	FFFF_0000
8940	LPCG direct control (LPCG37_DIRECT)	32	RW	0000_0001
8944	Low power mode information transfer status (LPCG37_LPM_STATUS0)	32	R	0000_0000
8948	Low power mode information transfer status (LPCG37_LPM_STATUS1)	32	R	0000_0000
8950	LPCG low power mode setting (LPCG37_LPM0)	32	RW	0000_0000
8954	LPCG low power mode setting (LPCG37_LPM1)	32	RW	0000_0000
895C	LPM setting of current CPU domain (LPCG37_LPM_CUR)	32	RW	0000_0000
8960	LPCG working status (LPCG37_STATUS0)	32	R	0000_0001
8964	LPCG domain status (LPCG37_STATUS1)	32	R	0000_FFFF
8970	LPCG access control (LPCG37_AUTHEN)	32	RW	FFFF_0000
8980	LPCG direct control (LPCG38_DIRECT)	32	RW	0000_0001
8984	Low power mode information transfer status (LPCG38_LPM_STATUS0)	32	R	0000_0000
8988	Low power mode information transfer status (LPCG38_LPM_STATUS1)	32	R	0000_0000
8990	LPCG low power mode setting (LPCG38_LPM0)	32	RW	0000_0000
8994	LPCG low power mode setting (LPCG38_LPM1)	32	RW	0000_0000
899C	LPM setting of current CPU domain (LPCG38_LPM_CUR)	32	RW	0000_0000
89A0	LPCG working status (LPCG38_STATUS0)	32	R	0000_0001
89A4	LPCG domain status (LPCG38_STATUS1)	32	R	0000_FFFF
89B0	LPCG access control (LPCG38_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
89C0	LPCG direct control (LPCG39_DIRECT)	32	RW	0000_0001
89C4	Low power mode information transfer status (LPCG39_LPM_STATUS0)	32	R	0000_0000
89C8	Low power mode information transfer status (LPCG39_LPM_STATUS1)	32	R	0000_0000
89D0	LPCG low power mode setting (LPCG39_LPM0)	32	RW	0000_0000
89D4	LPCG low power mode setting (LPCG39_LPM1)	32	RW	0000_0000
89DC	LPM setting of current CPU domain (LPCG39_LPM_CUR)	32	RW	0000_0000
89E0	LPCG working status (LPCG39_STATUS0)	32	R	0000_0001
89E4	LPCG domain status (LPCG39_STATUS1)	32	R	0000_FFFF
89F0	LPCG access control (LPCG39_AUTHEN)	32	RW	FFFF_0000
8A00	LPCG direct control (LPCG40_DIRECT)	32	RW	0000_0001
8A04	Low power mode information transfer status (LPCG40_LPM_STATUS0)	32	R	0000_0000
8A08	Low power mode information transfer status (LPCG40_LPM_STATUS1)	32	R	0000_0000
8A10	LPCG low power mode setting (LPCG40_LPM0)	32	RW	0000_0000
8A14	LPCG low power mode setting (LPCG40_LPM1)	32	RW	0000_0000
8A1C	LPM setting of current CPU domain (LPCG40_LPM_CUR)	32	RW	0000_0000
8A20	LPCG working status (LPCG40_STATUS0)	32	R	0000_0001
8A24	LPCG domain status (LPCG40_STATUS1)	32	R	0000_FFFF
8A30	LPCG access control (LPCG40_AUTHEN)	32	RW	FFFF_0000
8A40	LPCG direct control (LPCG41_DIRECT)	32	RW	0000_0001
8A44	Low power mode information transfer status (LPCG41_LPM_STATUS0)	32	R	0000_0000
8A48	Low power mode information transfer status (LPCG41_LPM_STATUS1)	32	R	0000_0000
8A50	LPCG low power mode setting (LPCG41_LPM0)	32	RW	0000_0000
8A54	LPCG low power mode setting (LPCG41_LPM1)	32	RW	0000_0000
8A5C	LPM setting of current CPU domain (LPCG41_LPM_CUR)	32	RW	0000_0000
8A60	LPCG working status (LPCG41_STATUS0)	32	R	0000_0001
8A64	LPCG domain status (LPCG41_STATUS1)	32	R	0000_FFFF
8A70	LPCG access control (LPCG41_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8A80	LPCG direct control (LPCG42_DIRECT)	32	RW	0000_0001
8A84	Low power mode information transfer status (LPCG42_LPM_STATUS0)	32	R	0000_0000
8A88	Low power mode information transfer status (LPCG42_LPM_STATUS1)	32	R	0000_0000
8A90	LPCG low power mode setting (LPCG42_LPM0)	32	RW	0000_0000
8A94	LPCG low power mode setting (LPCG42_LPM1)	32	RW	0000_0000
8A9C	LPM setting of current CPU domain (LPCG42_LPM_CUR)	32	RW	0000_0000
8AA0	LPCG working status (LPCG42_STATUS0)	32	R	0000_0001
8AA4	LPCG domain status (LPCG42_STATUS1)	32	R	0000_FFFF
8AB0	LPCG access control (LPCG42_AUTHEN)	32	RW	FFFF_0000
8AC0	LPCG direct control (LPCG43_DIRECT)	32	RW	0000_0001
8AC4	Low power mode information transfer status (LPCG43_LPM_STATUS0)	32	R	0000_0000
8AC8	Low power mode information transfer status (LPCG43_LPM_STATUS1)	32	R	0000_0000
8AD0	LPCG low power mode setting (LPCG43_LPM0)	32	RW	0000_0000
8AD4	LPCG low power mode setting (LPCG43_LPM1)	32	RW	0000_0000
8ADC	LPM setting of current CPU domain (LPCG43_LPM_CUR)	32	RW	0000_0000
8AE0	LPCG working status (LPCG43_STATUS0)	32	R	0000_0001
8AE4	LPCG domain status (LPCG43_STATUS1)	32	R	0000_FFFF
8AF0	LPCG access control (LPCG43_AUTHEN)	32	RW	FFFF_0000
8B00	LPCG direct control (LPCG44_DIRECT)	32	RW	0000_0001
8B04	Low power mode information transfer status (LPCG44_LPM_STATUS0)	32	R	0000_0000
8B08	Low power mode information transfer status (LPCG44_LPM_STATUS1)	32	R	0000_0000
8B10	LPCG low power mode setting (LPCG44_LPM0)	32	RW	0000_0000
8B14	LPCG low power mode setting (LPCG44_LPM1)	32	RW	0000_0000
8B1C	LPM setting of current CPU domain (LPCG44_LPM_CUR)	32	RW	0000_0000
8B20	LPCG working status (LPCG44_STATUS0)	32	R	0000_0001
8B24	LPCG domain status (LPCG44_STATUS1)	32	R	0000_FFFF
8B30	LPCG access control (LPCG44_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8B40	LPCG direct control (LPCG45_DIRECT)	32	RW	0000_0001
8B44	Low power mode information transfer status (LPCG45_LPM_STATUS0)	32	R	0000_0000
8B48	Low power mode information transfer status (LPCG45_LPM_STATUS1)	32	R	0000_0000
8B50	LPCG low power mode setting (LPCG45_LPM0)	32	RW	0000_0000
8B54	LPCG low power mode setting (LPCG45_LPM1)	32	RW	0000_0000
8B5C	LPM setting of current CPU domain (LPCG45_LPM_CUR)	32	RW	0000_0000
8B60	LPCG working status (LPCG45_STATUS0)	32	R	0000_0001
8B64	LPCG domain status (LPCG45_STATUS1)	32	R	0000_FFFF
8B70	LPCG access control (LPCG45_AUTHEN)	32	RW	FFFF_0000
8B80	LPCG direct control (LPCG46_DIRECT)	32	RW	0000_0001
8B84	Low power mode information transfer status (LPCG46_LPM_STATUS0)	32	R	0000_0000
8B88	Low power mode information transfer status (LPCG46_LPM_STATUS1)	32	R	0000_0000
8B90	LPCG low power mode setting (LPCG46_LPM0)	32	RW	0000_0000
8B94	LPCG low power mode setting (LPCG46_LPM1)	32	RW	0000_0000
8B9C	LPM setting of current CPU domain (LPCG46_LPM_CUR)	32	RW	0000_0000
8BA0	LPCG working status (LPCG46_STATUS0)	32	R	0000_0001
8BA4	LPCG domain status (LPCG46_STATUS1)	32	R	0000_FFFF
8BB0	LPCG access control (LPCG46_AUTHEN)	32	RW	FFFF_0000
8BC0	LPCG direct control (LPCG47_DIRECT)	32	RW	0000_0001
8BC4	Low power mode information transfer status (LPCG47_LPM_STATUS0)	32	R	0000_0000
8BC8	Low power mode information transfer status (LPCG47_LPM_STATUS1)	32	R	0000_0000
8BD0	LPCG low power mode setting (LPCG47_LPM0)	32	RW	0000_0000
8BD4	LPCG low power mode setting (LPCG47_LPM1)	32	RW	0000_0000
8BDC	LPM setting of current CPU domain (LPCG47_LPM_CUR)	32	RW	0000_0000
8BE0	LPCG working status (LPCG47_STATUS0)	32	R	0000_0001
8BE4	LPCG domain status (LPCG47_STATUS1)	32	R	0000_FFFF
8BF0	LPCG access control (LPCG47_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8C00	LPCG direct control (LPCG48_DIRECT)	32	RW	0000_0001
8C04	Low power mode information transfer status (LPCG48_LPM_STATUS0)	32	R	0000_0000
8C08	Low power mode information transfer status (LPCG48_LPM_STATUS1)	32	R	0000_0000
8C10	LPCG low power mode setting (LPCG48_LPM0)	32	RW	0000_0000
8C14	LPCG low power mode setting (LPCG48_LPM1)	32	RW	0000_0000
8C1C	LPM setting of current CPU domain (LPCG48_LPM_CUR)	32	RW	0000_0000
8C20	LPCG working status (LPCG48_STATUS0)	32	R	0000_0001
8C24	LPCG domain status (LPCG48_STATUS1)	32	R	0000_FFFF
8C30	LPCG access control (LPCG48_AUTHEN)	32	RW	FFFF_0000
8C40	LPCG direct control (LPCG49_DIRECT)	32	RW	0000_0001
8C44	Low power mode information transfer status (LPCG49_LPM_STATUS0)	32	R	0000_0000
8C48	Low power mode information transfer status (LPCG49_LPM_STATUS1)	32	R	0000_0000
8C50	LPCG low power mode setting (LPCG49_LPM0)	32	RW	0000_0000
8C54	LPCG low power mode setting (LPCG49_LPM1)	32	RW	0000_0000
8C5C	LPM setting of current CPU domain (LPCG49_LPM_CUR)	32	RW	0000_0000
8C60	LPCG working status (LPCG49_STATUS0)	32	R	0000_0001
8C64	LPCG domain status (LPCG49_STATUS1)	32	R	0000_FFFF
8C70	LPCG access control (LPCG49_AUTHEN)	32	RW	FFFF_0000
8C80	LPCG direct control (LPCG50_DIRECT)	32	RW	0000_0001
8C84	Low power mode information transfer status (LPCG50_LPM_STATUS0)	32	R	0000_0000
8C88	Low power mode information transfer status (LPCG50_LPM_STATUS1)	32	R	0000_0000
8C90	LPCG low power mode setting (LPCG50_LPM0)	32	RW	0000_0000
8C94	LPCG low power mode setting (LPCG50_LPM1)	32	RW	0000_0000
8C9C	LPM setting of current CPU domain (LPCG50_LPM_CUR)	32	RW	0000_0000
8CA0	LPCG working status (LPCG50_STATUS0)	32	R	0000_0001
8CA4	LPCG domain status (LPCG50_STATUS1)	32	R	0000_FFFF
8CB0	LPCG access control (LPCG50_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8CC0	LPCG direct control (LPCG51_DIRECT)	32	RW	0000_0001
8CC4	Low power mode information transfer status (LPCG51_LPM_STATUS0)	32	R	0000_0000
8CC8	Low power mode information transfer status (LPCG51_LPM_STATUS1)	32	R	0000_0000
8CD0	LPCG low power mode setting (LPCG51_LPM0)	32	RW	0000_0000
8CD4	LPCG low power mode setting (LPCG51_LPM1)	32	RW	0000_0000
8CDC	LPM setting of current CPU domain (LPCG51_LPM_CUR)	32	RW	0000_0000
8CE0	LPCG working status (LPCG51_STATUS0)	32	R	0000_0001
8CE4	LPCG domain status (LPCG51_STATUS1)	32	R	0000_FFFF
8CF0	LPCG access control (LPCG51_AUTHEN)	32	RW	FFFF_0000
8D00	LPCG direct control (LPCG52_DIRECT)	32	RW	0000_0001
8D04	Low power mode information transfer status (LPCG52_LPM_STATUS0)	32	R	0000_0000
8D08	Low power mode information transfer status (LPCG52_LPM_STATUS1)	32	R	0000_0000
8D10	LPCG low power mode setting (LPCG52_LPM0)	32	RW	0000_0000
8D14	LPCG low power mode setting (LPCG52_LPM1)	32	RW	0000_0000
8D1C	LPM setting of current CPU domain (LPCG52_LPM_CUR)	32	RW	0000_0000
8D20	LPCG working status (LPCG52_STATUS0)	32	R	0000_0001
8D24	LPCG domain status (LPCG52_STATUS1)	32	R	0000_FFFF
8D30	LPCG access control (LPCG52_AUTHEN)	32	RW	FFFF_0000
8D40	LPCG direct control (LPCG53_DIRECT)	32	RW	0000_0001
8D44	Low power mode information transfer status (LPCG53_LPM_STATUS0)	32	R	0000_0000
8D48	Low power mode information transfer status (LPCG53_LPM_STATUS1)	32	R	0000_0000
8D50	LPCG low power mode setting (LPCG53_LPM0)	32	RW	0000_0000
8D54	LPCG low power mode setting (LPCG53_LPM1)	32	RW	0000_0000
8D5C	LPM setting of current CPU domain (LPCG53_LPM_CUR)	32	RW	0000_0000
8D60	LPCG working status (LPCG53_STATUS0)	32	R	0000_0001
8D64	LPCG domain status (LPCG53_STATUS1)	32	R	0000_FFFF
8D70	LPCG access control (LPCG53_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8D80	LPCG direct control (LPCG54_DIRECT)	32	RW	0000_0001
8D84	Low power mode information transfer status (LPCG54_LPM_STATUS0)	32	R	0000_0000
8D88	Low power mode information transfer status (LPCG54_LPM_STATUS1)	32	R	0000_0000
8D90	LPCG low power mode setting (LPCG54_LPM0)	32	RW	0000_0000
8D94	LPCG low power mode setting (LPCG54_LPM1)	32	RW	0000_0000
8D9C	LPM setting of current CPU domain (LPCG54_LPM_CUR)	32	RW	0000_0000
8DA0	LPCG working status (LPCG54_STATUS0)	32	R	0000_0001
8DA4	LPCG domain status (LPCG54_STATUS1)	32	R	0000_FFFF
8DB0	LPCG access control (LPCG54_AUTHEN)	32	RW	FFFF_0000
8DC0	LPCG direct control (LPCG55_DIRECT)	32	RW	0000_0001
8DC4	Low power mode information transfer status (LPCG55_LPM_STATUS0)	32	R	0000_0000
8DC8	Low power mode information transfer status (LPCG55_LPM_STATUS1)	32	R	0000_0000
8DD0	LPCG low power mode setting (LPCG55_LPM0)	32	RW	0000_0000
8DD4	LPCG low power mode setting (LPCG55_LPM1)	32	RW	0000_0000
8DDC	LPM setting of current CPU domain (LPCG55_LPM_CUR)	32	RW	0000_0000
8DE0	LPCG working status (LPCG55_STATUS0)	32	R	0000_0001
8DE4	LPCG domain status (LPCG55_STATUS1)	32	R	0000_FFFF
8DF0	LPCG access control (LPCG55_AUTHEN)	32	RW	FFFF_0000
8E00	LPCG direct control (LPCG56_DIRECT)	32	RW	0000_0001
8E04	Low power mode information transfer status (LPCG56_LPM_STATUS0)	32	R	0000_0000
8E08	Low power mode information transfer status (LPCG56_LPM_STATUS1)	32	R	0000_0000
8E10	LPCG low power mode setting (LPCG56_LPM0)	32	RW	0000_0000
8E14	LPCG low power mode setting (LPCG56_LPM1)	32	RW	0000_0000
8E1C	LPM setting of current CPU domain (LPCG56_LPM_CUR)	32	RW	0000_0000
8E20	LPCG working status (LPCG56_STATUS0)	32	R	0000_0001
8E24	LPCG domain status (LPCG56_STATUS1)	32	R	0000_FFFF
8E30	LPCG access control (LPCG56_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8E40	LPCG direct control (LPCG57_DIRECT)	32	RW	0000_0001
8E44	Low power mode information transfer status (LPCG57_LPM_STATUS0)	32	R	0000_0000
8E48	Low power mode information transfer status (LPCG57_LPM_STATUS1)	32	R	0000_0000
8E50	LPCG low power mode setting (LPCG57_LPM0)	32	RW	0000_0000
8E54	LPCG low power mode setting (LPCG57_LPM1)	32	RW	0000_0000
8E5C	LPM setting of current CPU domain (LPCG57_LPM_CUR)	32	RW	0000_0000
8E60	LPCG working status (LPCG57_STATUS0)	32	R	0000_0001
8E64	LPCG domain status (LPCG57_STATUS1)	32	R	0000_FFFF
8E70	LPCG access control (LPCG57_AUTHEN)	32	RW	FFFF_0000
8E80	LPCG direct control (LPCG58_DIRECT)	32	RW	0000_0001
8E84	Low power mode information transfer status (LPCG58_LPM_STATUS0)	32	R	0000_0000
8E88	Low power mode information transfer status (LPCG58_LPM_STATUS1)	32	R	0000_0000
8E90	LPCG low power mode setting (LPCG58_LPM0)	32	RW	0000_0000
8E94	LPCG low power mode setting (LPCG58_LPM1)	32	RW	0000_0000
8E9C	LPM setting of current CPU domain (LPCG58_LPM_CUR)	32	RW	0000_0000
8EA0	LPCG working status (LPCG58_STATUS0)	32	R	0000_0001
8EA4	LPCG domain status (LPCG58_STATUS1)	32	R	0000_FFFF
8EB0	LPCG access control (LPCG58_AUTHEN)	32	RW	FFFF_0000
8EC0	LPCG direct control (LPCG59_DIRECT)	32	RW	0000_0001
8EC4	Low power mode information transfer status (LPCG59_LPM_STATUS0)	32	R	0000_0000
8EC8	Low power mode information transfer status (LPCG59_LPM_STATUS1)	32	R	0000_0000
8ED0	LPCG low power mode setting (LPCG59_LPM0)	32	RW	0000_0000
8ED4	LPCG low power mode setting (LPCG59_LPM1)	32	RW	0000_0000
8EDC	LPM setting of current CPU domain (LPCG59_LPM_CUR)	32	RW	0000_0000
8EE0	LPCG working status (LPCG59_STATUS0)	32	R	0000_0001
8EE4	LPCG domain status (LPCG59_STATUS1)	32	R	0000_FFFF
8EF0	LPCG access control (LPCG59_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8F00	LPCG direct control (LPCG60_DIRECT)	32	RW	0000_0001
8F04	Low power mode information transfer status (LPCG60_LPM_STATUS0)	32	R	0000_0000
8F08	Low power mode information transfer status (LPCG60_LPM_STATUS1)	32	R	0000_0000
8F10	LPCG low power mode setting (LPCG60_LPM0)	32	RW	0000_0000
8F14	LPCG low power mode setting (LPCG60_LPM1)	32	RW	0000_0000
8F1C	LPM setting of current CPU domain (LPCG60_LPM_CUR)	32	RW	0000_0000
8F20	LPCG working status (LPCG60_STATUS0)	32	R	0000_0001
8F24	LPCG domain status (LPCG60_STATUS1)	32	R	0000_FFFF
8F30	LPCG access control (LPCG60_AUTHEN)	32	RW	FFFF_0000
8F40	LPCG direct control (LPCG61_DIRECT)	32	RW	0000_0001
8F44	Low power mode information transfer status (LPCG61_LPM_STATUS0)	32	R	0000_0000
8F48	Low power mode information transfer status (LPCG61_LPM_STATUS1)	32	R	0000_0000
8F50	LPCG low power mode setting (LPCG61_LPM0)	32	RW	0000_0000
8F54	LPCG low power mode setting (LPCG61_LPM1)	32	RW	0000_0000
8F5C	LPM setting of current CPU domain (LPCG61_LPM_CUR)	32	RW	0000_0000
8F60	LPCG working status (LPCG61_STATUS0)	32	R	0000_0001
8F64	LPCG domain status (LPCG61_STATUS1)	32	R	0000_FFFF
8F70	LPCG access control (LPCG61_AUTHEN)	32	RW	FFFF_0000
8F80	LPCG direct control (LPCG62_DIRECT)	32	RW	0000_0001
8F84	Low power mode information transfer status (LPCG62_LPM_STATUS0)	32	R	0000_0000
8F88	Low power mode information transfer status (LPCG62_LPM_STATUS1)	32	R	0000_0000
8F90	LPCG low power mode setting (LPCG62_LPM0)	32	RW	0000_0000
8F94	LPCG low power mode setting (LPCG62_LPM1)	32	RW	0000_0000
8F9C	LPM setting of current CPU domain (LPCG62_LPM_CUR)	32	RW	0000_0000
8FA0	LPCG working status (LPCG62_STATUS0)	32	R	0000_0001
8FA4	LPCG domain status (LPCG62_STATUS1)	32	R	0000_FFFF
8FB0	LPCG access control (LPCG62_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8FC0	LPCG direct control (LPCG63_DIRECT)	32	RW	0000_0001
8FC4	Low power mode information transfer status (LPCG63_LPM_STATUS0)	32	R	0000_0000
8FC8	Low power mode information transfer status (LPCG63_LPM_STATUS1)	32	R	0000_0000
8FD0	LPCG low power mode setting (LPCG63_LPM0)	32	RW	0000_0000
8FD4	LPCG low power mode setting (LPCG63_LPM1)	32	RW	0000_0000
8FDC	LPM setting of current CPU domain (LPCG63_LPM_CUR)	32	RW	0000_0000
8FE0	LPCG working status (LPCG63_STATUS0)	32	R	0000_0001
8FE4	LPCG domain status (LPCG63_STATUS1)	32	R	0000_FFFF
8FF0	LPCG access control (LPCG63_AUTHEN)	32	RW	FFFF_0000
9000	LPCG direct control (LPCG64_DIRECT)	32	RW	0000_0001
9004	Low power mode information transfer status (LPCG64_LPM_STATUS0)	32	R	0000_0000
9008	Low power mode information transfer status (LPCG64_LPM_STATUS1)	32	R	0000_0000
9010	LPCG low power mode setting (LPCG64_LPM0)	32	RW	0000_0000
9014	LPCG low power mode setting (LPCG64_LPM1)	32	RW	0000_0000
901C	LPM setting of current CPU domain (LPCG64_LPM_CUR)	32	RW	0000_0000
9020	LPCG working status (LPCG64_STATUS0)	32	R	0000_0001
9024	LPCG domain status (LPCG64_STATUS1)	32	R	0000_FFFF
9030	LPCG access control (LPCG64_AUTHEN)	32	RW	FFFF_0000
9040	LPCG direct control (LPCG65_DIRECT)	32	RW	0000_0001
9044	Low power mode information transfer status (LPCG65_LPM_STATUS0)	32	R	0000_0000
9048	Low power mode information transfer status (LPCG65_LPM_STATUS1)	32	R	0000_0000
9050	LPCG low power mode setting (LPCG65_LPM0)	32	RW	0000_0000
9054	LPCG low power mode setting (LPCG65_LPM1)	32	RW	0000_0000
905C	LPM setting of current CPU domain (LPCG65_LPM_CUR)	32	RW	0000_0000
9060	LPCG working status (LPCG65_STATUS0)	32	R	0000_0001
9064	LPCG domain status (LPCG65_STATUS1)	32	R	0000_FFFF
9070	LPCG access control (LPCG65_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9080	LPCG direct control (LPCG66_DIRECT)	32	RW	0000_0001
9084	Low power mode information transfer status (LPCG66_LPM_STATUS0)	32	R	0000_0000
9088	Low power mode information transfer status (LPCG66_LPM_STATUS1)	32	R	0000_0000
9090	LPCG low power mode setting (LPCG66_LPM0)	32	RW	0000_0000
9094	LPCG low power mode setting (LPCG66_LPM1)	32	RW	0000_0000
909C	LPM setting of current CPU domain (LPCG66_LPM_CUR)	32	RW	0000_0000
90A0	LPCG working status (LPCG66_STATUS0)	32	R	0000_0001
90A4	LPCG domain status (LPCG66_STATUS1)	32	R	0000_FFFF
90B0	LPCG access control (LPCG66_AUTHEN)	32	RW	FFFF_0000
90C0	LPCG direct control (LPCG67_DIRECT)	32	RW	0000_0001
90C4	Low power mode information transfer status (LPCG67_LPM_STATUS0)	32	R	0000_0000
90C8	Low power mode information transfer status (LPCG67_LPM_STATUS1)	32	R	0000_0000
90D0	LPCG low power mode setting (LPCG67_LPM0)	32	RW	0000_0000
90D4	LPCG low power mode setting (LPCG67_LPM1)	32	RW	0000_0000
90DC	LPM setting of current CPU domain (LPCG67_LPM_CUR)	32	RW	0000_0000
90E0	LPCG working status (LPCG67_STATUS0)	32	R	0000_0001
90E4	LPCG domain status (LPCG67_STATUS1)	32	R	0000_FFFF
90F0	LPCG access control (LPCG67_AUTHEN)	32	RW	FFFF_0000
9100	LPCG direct control (LPCG68_DIRECT)	32	RW	0000_0001
9104	Low power mode information transfer status (LPCG68_LPM_STATUS0)	32	R	0000_0000
9108	Low power mode information transfer status (LPCG68_LPM_STATUS1)	32	R	0000_0000
9110	LPCG low power mode setting (LPCG68_LPM0)	32	RW	0000_0000
9114	LPCG low power mode setting (LPCG68_LPM1)	32	RW	0000_0000
911C	LPM setting of current CPU domain (LPCG68_LPM_CUR)	32	RW	0000_0000
9120	LPCG working status (LPCG68_STATUS0)	32	R	0000_0001
9124	LPCG domain status (LPCG68_STATUS1)	32	R	0000_FFFF
9130	LPCG access control (LPCG68_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9140	LPCG direct control (LPCG69_DIRECT)	32	RW	0000_0001
9144	Low power mode information transfer status (LPCG69_LPM_STATUS0)	32	R	0000_0000
9148	Low power mode information transfer status (LPCG69_LPM_STATUS1)	32	R	0000_0000
9150	LPCG low power mode setting (LPCG69_LPM0)	32	RW	0000_0000
9154	LPCG low power mode setting (LPCG69_LPM1)	32	RW	0000_0000
915C	LPM setting of current CPU domain (LPCG69_LPM_CUR)	32	RW	0000_0000
9160	LPCG working status (LPCG69_STATUS0)	32	R	0000_0001
9164	LPCG domain status (LPCG69_STATUS1)	32	R	0000_FFFF
9170	LPCG access control (LPCG69_AUTHEN)	32	RW	FFFF_0000
9180	LPCG direct control (LPCG70_DIRECT)	32	RW	0000_0001
9184	Low power mode information transfer status (LPCG70_LPM_STATUS0)	32	R	0000_0000
9188	Low power mode information transfer status (LPCG70_LPM_STATUS1)	32	R	0000_0000
9190	LPCG low power mode setting (LPCG70_LPM0)	32	RW	0000_0000
9194	LPCG low power mode setting (LPCG70_LPM1)	32	RW	0000_0000
919C	LPM setting of current CPU domain (LPCG70_LPM_CUR)	32	RW	0000_0000
91A0	LPCG working status (LPCG70_STATUS0)	32	R	0000_0001
91A4	LPCG domain status (LPCG70_STATUS1)	32	R	0000_FFFF
91B0	LPCG access control (LPCG70_AUTHEN)	32	RW	FFFF_0000
91C0	LPCG direct control (LPCG71_DIRECT)	32	RW	0000_0001
91C4	Low power mode information transfer status (LPCG71_LPM_STATUS0)	32	R	0000_0000
91C8	Low power mode information transfer status (LPCG71_LPM_STATUS1)	32	R	0000_0000
91D0	LPCG low power mode setting (LPCG71_LPM0)	32	RW	0000_0000
91D4	LPCG low power mode setting (LPCG71_LPM1)	32	RW	0000_0000
91DC	LPM setting of current CPU domain (LPCG71_LPM_CUR)	32	RW	0000_0000
91E0	LPCG working status (LPCG71_STATUS0)	32	R	0000_0001
91E4	LPCG domain status (LPCG71_STATUS1)	32	R	0000_FFFF
91F0	LPCG access control (LPCG71_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9200	LPCG direct control (LPCG72_DIRECT)	32	RW	0000_0001
9204	Low power mode information transfer status (LPCG72_LPM_STATUS0)	32	R	0000_0000
9208	Low power mode information transfer status (LPCG72_LPM_STATUS1)	32	R	0000_0000
9210	LPCG low power mode setting (LPCG72_LPM0)	32	RW	0000_0000
9214	LPCG low power mode setting (LPCG72_LPM1)	32	RW	0000_0000
921C	LPM setting of current CPU domain (LPCG72_LPM_CUR)	32	RW	0000_0000
9220	LPCG working status (LPCG72_STATUS0)	32	R	0000_0001
9224	LPCG domain status (LPCG72_STATUS1)	32	R	0000_FFFF
9230	LPCG access control (LPCG72_AUTHEN)	32	RW	FFFF_0000
9240	LPCG direct control (LPCG73_DIRECT)	32	RW	0000_0001
9244	Low power mode information transfer status (LPCG73_LPM_STATUS0)	32	R	0000_0000
9248	Low power mode information transfer status (LPCG73_LPM_STATUS1)	32	R	0000_0000
9250	LPCG low power mode setting (LPCG73_LPM0)	32	RW	0000_0000
9254	LPCG low power mode setting (LPCG73_LPM1)	32	RW	0000_0000
925C	LPM setting of current CPU domain (LPCG73_LPM_CUR)	32	RW	0000_0000
9260	LPCG working status (LPCG73_STATUS0)	32	R	0000_0001
9264	LPCG domain status (LPCG73_STATUS1)	32	R	0000_FFFF
9270	LPCG access control (LPCG73_AUTHEN)	32	RW	FFFF_0000
9280	LPCG direct control (LPCG74_DIRECT)	32	RW	0000_0001
9284	Low power mode information transfer status (LPCG74_LPM_STATUS0)	32	R	0000_0000
9288	Low power mode information transfer status (LPCG74_LPM_STATUS1)	32	R	0000_0000
9290	LPCG low power mode setting (LPCG74_LPM0)	32	RW	0000_0000
9294	LPCG low power mode setting (LPCG74_LPM1)	32	RW	0000_0000
929C	LPM setting of current CPU domain (LPCG74_LPM_CUR)	32	RW	0000_0000
92A0	LPCG working status (LPCG74_STATUS0)	32	R	0000_0001
92A4	LPCG domain status (LPCG74_STATUS1)	32	R	0000_FFFF
92B0	LPCG access control (LPCG74_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
92C0	LPCG direct control (LPCG75_DIRECT)	32	RW	0000_0001
92C4	Low power mode information transfer status (LPCG75_LPM_STATUS0)	32	R	0000_0000
92C8	Low power mode information transfer status (LPCG75_LPM_STATUS1)	32	R	0000_0000
92D0	LPCG low power mode setting (LPCG75_LPM0)	32	RW	0000_0000
92D4	LPCG low power mode setting (LPCG75_LPM1)	32	RW	0000_0000
92DC	LPM setting of current CPU domain (LPCG75_LPM_CUR)	32	RW	0000_0000
92E0	LPCG working status (LPCG75_STATUS0)	32	R	0000_0001
92E4	LPCG domain status (LPCG75_STATUS1)	32	R	0000_FFFF
92F0	LPCG access control (LPCG75_AUTHEN)	32	RW	FFFF_0000
9300	LPCG direct control (LPCG76_DIRECT)	32	RW	0000_0001
9304	Low power mode information transfer status (LPCG76_LPM_STATUS0)	32	R	0000_0000
9308	Low power mode information transfer status (LPCG76_LPM_STATUS1)	32	R	0000_0000
9310	LPCG low power mode setting (LPCG76_LPM0)	32	RW	0000_0000
9314	LPCG low power mode setting (LPCG76_LPM1)	32	RW	0000_0000
931C	LPM setting of current CPU domain (LPCG76_LPM_CUR)	32	RW	0000_0000
9320	LPCG working status (LPCG76_STATUS0)	32	R	0000_0001
9324	LPCG domain status (LPCG76_STATUS1)	32	R	0000_FFFF
9330	LPCG access control (LPCG76_AUTHEN)	32	RW	FFFF_0000
9340	LPCG direct control (LPCG77_DIRECT)	32	RW	0000_0001
9344	Low power mode information transfer status (LPCG77_LPM_STATUS0)	32	R	0000_0000
9348	Low power mode information transfer status (LPCG77_LPM_STATUS1)	32	R	0000_0000
9350	LPCG low power mode setting (LPCG77_LPM0)	32	RW	0000_0000
9354	LPCG low power mode setting (LPCG77_LPM1)	32	RW	0000_0000
935C	LPM setting of current CPU domain (LPCG77_LPM_CUR)	32	RW	0000_0000
9360	LPCG working status (LPCG77_STATUS0)	32	R	0000_0001
9364	LPCG domain status (LPCG77_STATUS1)	32	R	0000_FFFF
9370	LPCG access control (LPCG77_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9380	LPCG direct control (LPCG78_DIRECT)	32	RW	0000_0001
9384	Low power mode information transfer status (LPCG78_LPM_STATUS0)	32	R	0000_0000
9388	Low power mode information transfer status (LPCG78_LPM_STATUS1)	32	R	0000_0000
9390	LPCG low power mode setting (LPCG78_LPM0)	32	RW	0000_0000
9394	LPCG low power mode setting (LPCG78_LPM1)	32	RW	0000_0000
939C	LPM setting of current CPU domain (LPCG78_LPM_CUR)	32	RW	0000_0000
93A0	LPCG working status (LPCG78_STATUS0)	32	R	0000_0001
93A4	LPCG domain status (LPCG78_STATUS1)	32	R	0000_FFFF
93B0	LPCG access control (LPCG78_AUTHEN)	32	RW	FFFF_0000
93C0	LPCG direct control (LPCG79_DIRECT)	32	RW	0000_0001
93C4	Low power mode information transfer status (LPCG79_LPM_STATUS0)	32	R	0000_0000
93C8	Low power mode information transfer status (LPCG79_LPM_STATUS1)	32	R	0000_0000
93D0	LPCG low power mode setting (LPCG79_LPM0)	32	RW	0000_0000
93D4	LPCG low power mode setting (LPCG79_LPM1)	32	RW	0000_0000
93DC	LPM setting of current CPU domain (LPCG79_LPM_CUR)	32	RW	0000_0000
93E0	LPCG working status (LPCG79_STATUS0)	32	R	0000_0001
93E4	LPCG domain status (LPCG79_STATUS1)	32	R	0000_FFFF
93F0	LPCG access control (LPCG79_AUTHEN)	32	RW	FFFF_0000
9400	LPCG direct control (LPCG80_DIRECT)	32	RW	0000_0001
9404	Low power mode information transfer status (LPCG80_LPM_STATUS0)	32	R	0000_0000
9408	Low power mode information transfer status (LPCG80_LPM_STATUS1)	32	R	0000_0000
9410	LPCG low power mode setting (LPCG80_LPM0)	32	RW	0000_0000
9414	LPCG low power mode setting (LPCG80_LPM1)	32	RW	0000_0000
941C	LPM setting of current CPU domain (LPCG80_LPM_CUR)	32	RW	0000_0000
9420	LPCG working status (LPCG80_STATUS0)	32	R	0000_0001
9424	LPCG domain status (LPCG80_STATUS1)	32	R	0000_FFFF
9430	LPCG access control (LPCG80_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9440	LPCG direct control (LPCG81_DIRECT)	32	RW	0000_0001
9444	Low power mode information transfer status (LPCG81_LPM_STATUS0)	32	R	0000_0000
9448	Low power mode information transfer status (LPCG81_LPM_STATUS1)	32	R	0000_0000
9450	LPCG low power mode setting (LPCG81_LPM0)	32	RW	0000_0000
9454	LPCG low power mode setting (LPCG81_LPM1)	32	RW	0000_0000
945C	LPM setting of current CPU domain (LPCG81_LPM_CUR)	32	RW	0000_0000
9460	LPCG working status (LPCG81_STATUS0)	32	R	0000_0001
9464	LPCG domain status (LPCG81_STATUS1)	32	R	0000_FFFF
9470	LPCG access control (LPCG81_AUTHEN)	32	RW	FFFF_0000
9480	LPCG direct control (LPCG82_DIRECT)	32	RW	0000_0001
9484	Low power mode information transfer status (LPCG82_LPM_STATUS0)	32	R	0000_0000
9488	Low power mode information transfer status (LPCG82_LPM_STATUS1)	32	R	0000_0000
9490	LPCG low power mode setting (LPCG82_LPM0)	32	RW	0000_0000
9494	LPCG low power mode setting (LPCG82_LPM1)	32	RW	0000_0000
949C	LPM setting of current CPU domain (LPCG82_LPM_CUR)	32	RW	0000_0000
94A0	LPCG working status (LPCG82_STATUS0)	32	R	0000_0001
94A4	LPCG domain status (LPCG82_STATUS1)	32	R	0000_FFFF
94B0	LPCG access control (LPCG82_AUTHEN)	32	RW	FFFF_0000
94C0	LPCG direct control (LPCG83_DIRECT)	32	RW	0000_0001
94C4	Low power mode information transfer status (LPCG83_LPM_STATUS0)	32	R	0000_0000
94C8	Low power mode information transfer status (LPCG83_LPM_STATUS1)	32	R	0000_0000
94D0	LPCG low power mode setting (LPCG83_LPM0)	32	RW	0000_0000
94D4	LPCG low power mode setting (LPCG83_LPM1)	32	RW	0000_0000
94DC	LPM setting of current CPU domain (LPCG83_LPM_CUR)	32	RW	0000_0000
94E0	LPCG working status (LPCG83_STATUS0)	32	R	0000_0001
94E4	LPCG domain status (LPCG83_STATUS1)	32	R	0000_FFFF
94F0	LPCG access control (LPCG83_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9500	LPCG direct control (LPCG84_DIRECT)	32	RW	0000_0001
9504	Low power mode information transfer status (LPCG84_LPM_STATUS0)	32	R	0000_0000
9508	Low power mode information transfer status (LPCG84_LPM_STATUS1)	32	R	0000_0000
9510	LPCG low power mode setting (LPCG84_LPM0)	32	RW	0000_0000
9514	LPCG low power mode setting (LPCG84_LPM1)	32	RW	0000_0000
951C	LPM setting of current CPU domain (LPCG84_LPM_CUR)	32	RW	0000_0000
9520	LPCG working status (LPCG84_STATUS0)	32	R	0000_0001
9524	LPCG domain status (LPCG84_STATUS1)	32	R	0000_FFFF
9530	LPCG access control (LPCG84_AUTHEN)	32	RW	FFFF_0000
9540	LPCG direct control (LPCG85_DIRECT)	32	RW	0000_0001
9544	Low power mode information transfer status (LPCG85_LPM_STATUS0)	32	R	0000_0000
9548	Low power mode information transfer status (LPCG85_LPM_STATUS1)	32	R	0000_0000
9550	LPCG low power mode setting (LPCG85_LPM0)	32	RW	0000_0000
9554	LPCG low power mode setting (LPCG85_LPM1)	32	RW	0000_0000
955C	LPM setting of current CPU domain (LPCG85_LPM_CUR)	32	RW	0000_0000
9560	LPCG working status (LPCG85_STATUS0)	32	R	0000_0001
9564	LPCG domain status (LPCG85_STATUS1)	32	R	0000_FFFF
9570	LPCG access control (LPCG85_AUTHEN)	32	RW	FFFF_0000
9580	LPCG direct control (LPCG86_DIRECT)	32	RW	0000_0001
9584	Low power mode information transfer status (LPCG86_LPM_STATUS0)	32	R	0000_0000
9588	Low power mode information transfer status (LPCG86_LPM_STATUS1)	32	R	0000_0000
9590	LPCG low power mode setting (LPCG86_LPM0)	32	RW	0000_0000
9594	LPCG low power mode setting (LPCG86_LPM1)	32	RW	0000_0000
959C	LPM setting of current CPU domain (LPCG86_LPM_CUR)	32	RW	0000_0000
95A0	LPCG working status (LPCG86_STATUS0)	32	R	0000_0001
95A4	LPCG domain status (LPCG86_STATUS1)	32	R	0000_FFFF
95B0	LPCG access control (LPCG86_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
95C0	LPCG direct control (LPCG87_DIRECT)	32	RW	0000_0001
95C4	Low power mode information transfer status (LPCG87_LPM_STATUS0)	32	R	0000_0000
95C8	Low power mode information transfer status (LPCG87_LPM_STATUS1)	32	R	0000_0000
95D0	LPCG low power mode setting (LPCG87_LPM0)	32	RW	0000_0000
95D4	LPCG low power mode setting (LPCG87_LPM1)	32	RW	0000_0000
95DC	LPM setting of current CPU domain (LPCG87_LPM_CUR)	32	RW	0000_0000
95E0	LPCG working status (LPCG87_STATUS0)	32	R	0000_0001
95E4	LPCG domain status (LPCG87_STATUS1)	32	R	0000_FFFF
95F0	LPCG access control (LPCG87_AUTHEN)	32	RW	FFFF_0000
9600	LPCG direct control (LPCG88_DIRECT)	32	RW	0000_0001
9604	Low power mode information transfer status (LPCG88_LPM_STATUS0)	32	R	0000_0000
9608	Low power mode information transfer status (LPCG88_LPM_STATUS1)	32	R	0000_0000
9610	LPCG low power mode setting (LPCG88_LPM0)	32	RW	0000_0000
9614	LPCG low power mode setting (LPCG88_LPM1)	32	RW	0000_0000
961C	LPM setting of current CPU domain (LPCG88_LPM_CUR)	32	RW	0000_0000
9620	LPCG working status (LPCG88_STATUS0)	32	R	0000_0001
9624	LPCG domain status (LPCG88_STATUS1)	32	R	0000_FFFF
9630	LPCG access control (LPCG88_AUTHEN)	32	RW	FFFF_0000
9640	LPCG direct control (LPCG89_DIRECT)	32	RW	0000_0001
9644	Low power mode information transfer status (LPCG89_LPM_STATUS0)	32	R	0000_0000
9648	Low power mode information transfer status (LPCG89_LPM_STATUS1)	32	R	0000_0000
9650	LPCG low power mode setting (LPCG89_LPM0)	32	RW	0000_0000
9654	LPCG low power mode setting (LPCG89_LPM1)	32	RW	0000_0000
965C	LPM setting of current CPU domain (LPCG89_LPM_CUR)	32	RW	0000_0000
9660	LPCG working status (LPCG89_STATUS0)	32	R	0000_0001
9664	LPCG domain status (LPCG89_STATUS1)	32	R	0000_FFFF
9670	LPCG access control (LPCG89_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9680	LPCG direct control (LPCG90_DIRECT)	32	RW	0000_0001
9684	Low power mode information transfer status (LPCG90_LPM_STATUS0)	32	R	0000_0000
9688	Low power mode information transfer status (LPCG90_LPM_STATUS1)	32	R	0000_0000
9690	LPCG low power mode setting (LPCG90_LPM0)	32	RW	0000_0000
9694	LPCG low power mode setting (LPCG90_LPM1)	32	RW	0000_0000
969C	LPM setting of current CPU domain (LPCG90_LPM_CUR)	32	RW	0000_0000
96A0	LPCG working status (LPCG90_STATUS0)	32	R	0000_0001
96A4	LPCG domain status (LPCG90_STATUS1)	32	R	0000_FFFF
96B0	LPCG access control (LPCG90_AUTHEN)	32	RW	FFFF_0000
96C0	LPCG direct control (LPCG91_DIRECT)	32	RW	0000_0001
96C4	Low power mode information transfer status (LPCG91_LPM_STATUS0)	32	R	0000_0000
96C8	Low power mode information transfer status (LPCG91_LPM_STATUS1)	32	R	0000_0000
96D0	LPCG low power mode setting (LPCG91_LPM0)	32	RW	0000_0000
96D4	LPCG low power mode setting (LPCG91_LPM1)	32	RW	0000_0000
96DC	LPM setting of current CPU domain (LPCG91_LPM_CUR)	32	RW	0000_0000
96E0	LPCG working status (LPCG91_STATUS0)	32	R	0000_0001
96E4	LPCG domain status (LPCG91_STATUS1)	32	R	0000_FFFF
96F0	LPCG access control (LPCG91_AUTHEN)	32	RW	FFFF_0000
9700	LPCG direct control (LPCG92_DIRECT)	32	RW	0000_0001
9704	Low power mode information transfer status (LPCG92_LPM_STATUS0)	32	R	0000_0000
9708	Low power mode information transfer status (LPCG92_LPM_STATUS1)	32	R	0000_0000
9710	LPCG low power mode setting (LPCG92_LPM0)	32	RW	0000_0000
9714	LPCG low power mode setting (LPCG92_LPM1)	32	RW	0000_0000
971C	LPM setting of current CPU domain (LPCG92_LPM_CUR)	32	RW	0000_0000
9720	LPCG working status (LPCG92_STATUS0)	32	R	0000_0001
9724	LPCG domain status (LPCG92_STATUS1)	32	R	0000_FFFF
9730	LPCG access control (LPCG92_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9740	LPCG direct control (LPCG93_DIRECT)	32	RW	0000_0001
9744	Low power mode information transfer status (LPCG93_LPM_STATUS0)	32	R	0000_0000
9748	Low power mode information transfer status (LPCG93_LPM_STATUS1)	32	R	0000_0000
9750	LPCG low power mode setting (LPCG93_LPM0)	32	RW	0000_0000
9754	LPCG low power mode setting (LPCG93_LPM1)	32	RW	0000_0000
975C	LPM setting of current CPU domain (LPCG93_LPM_CUR)	32	RW	0000_0000
9760	LPCG working status (LPCG93_STATUS0)	32	R	0000_0001
9764	LPCG domain status (LPCG93_STATUS1)	32	R	0000_FFFF
9770	LPCG access control (LPCG93_AUTHEN)	32	RW	FFFF_0000
9780	LPCG direct control (LPCG94_DIRECT)	32	RW	0000_0001
9784	Low power mode information transfer status (LPCG94_LPM_STATUS0)	32	R	0000_0000
9788	Low power mode information transfer status (LPCG94_LPM_STATUS1)	32	R	0000_0000
9790	LPCG low power mode setting (LPCG94_LPM0)	32	RW	0000_0000
9794	LPCG low power mode setting (LPCG94_LPM1)	32	RW	0000_0000
979C	LPM setting of current CPU domain (LPCG94_LPM_CUR)	32	RW	0000_0000
97A0	LPCG working status (LPCG94_STATUS0)	32	R	0000_0001
97A4	LPCG domain status (LPCG94_STATUS1)	32	R	0000_FFFF
97B0	LPCG access control (LPCG94_AUTHEN)	32	RW	FFFF_0000
97C0	LPCG direct control (LPCG95_DIRECT)	32	RW	0000_0001
97C4	Low power mode information transfer status (LPCG95_LPM_STATUS0)	32	R	0000_0000
97C8	Low power mode information transfer status (LPCG95_LPM_STATUS1)	32	R	0000_0000
97D0	LPCG low power mode setting (LPCG95_LPM0)	32	RW	0000_0000
97D4	LPCG low power mode setting (LPCG95_LPM1)	32	RW	0000_0000
97DC	LPM setting of current CPU domain (LPCG95_LPM_CUR)	32	RW	0000_0000
97E0	LPCG working status (LPCG95_STATUS0)	32	R	0000_0001
97E4	LPCG domain status (LPCG95_STATUS1)	32	R	0000_FFFF
97F0	LPCG access control (LPCG95_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9800	LPCG direct control (LPCG96_DIRECT)	32	RW	0000_0001
9804	Low power mode information transfer status (LPCG96_LPM_STATUS0)	32	R	0000_0000
9808	Low power mode information transfer status (LPCG96_LPM_STATUS1)	32	R	0000_0000
9810	LPCG low power mode setting (LPCG96_LPM0)	32	RW	0000_0000
9814	LPCG low power mode setting (LPCG96_LPM1)	32	RW	0000_0000
981C	LPM setting of current CPU domain (LPCG96_LPM_CUR)	32	RW	0000_0000
9820	LPCG working status (LPCG96_STATUS0)	32	R	0000_0001
9824	LPCG domain status (LPCG96_STATUS1)	32	R	0000_FFFF
9830	LPCG access control (LPCG96_AUTHEN)	32	RW	FFFF_0000
9840	LPCG direct control (LPCG97_DIRECT)	32	RW	0000_0001
9844	Low power mode information transfer status (LPCG97_LPM_STATUS0)	32	R	0000_0000
9848	Low power mode information transfer status (LPCG97_LPM_STATUS1)	32	R	0000_0000
9850	LPCG low power mode setting (LPCG97_LPM0)	32	RW	0000_0000
9854	LPCG low power mode setting (LPCG97_LPM1)	32	RW	0000_0000
985C	LPM setting of current CPU domain (LPCG97_LPM_CUR)	32	RW	0000_0000
9860	LPCG working status (LPCG97_STATUS0)	32	R	0000_0001
9864	LPCG domain status (LPCG97_STATUS1)	32	R	0000_FFFF
9870	LPCG access control (LPCG97_AUTHEN)	32	RW	FFFF_0000
9880	LPCG direct control (LPCG98_DIRECT)	32	RW	0000_0001
9884	Low power mode information transfer status (LPCG98_LPM_STATUS0)	32	R	0000_0000
9888	Low power mode information transfer status (LPCG98_LPM_STATUS1)	32	R	0000_0000
9890	LPCG low power mode setting (LPCG98_LPM0)	32	RW	0000_0000
9894	LPCG low power mode setting (LPCG98_LPM1)	32	RW	0000_0000
989C	LPM setting of current CPU domain (LPCG98_LPM_CUR)	32	RW	0000_0000
98A0	LPCG working status (LPCG98_STATUS0)	32	R	0000_0001
98A4	LPCG domain status (LPCG98_STATUS1)	32	R	0000_FFFF
98B0	LPCG access control (LPCG98_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
98C0	LPCG direct control (LPCG99_DIRECT)	32	RW	0000_0001
98C4	Low power mode information transfer status (LPCG99_LPM_STATUS0)	32	R	0000_0000
98C8	Low power mode information transfer status (LPCG99_LPM_STATUS1)	32	R	0000_0000
98D0	LPCG low power mode setting (LPCG99_LPM0)	32	RW	0000_0000
98D4	LPCG low power mode setting (LPCG99_LPM1)	32	RW	0000_0000
98DC	LPM setting of current CPU domain (LPCG99_LPM_CUR)	32	RW	0000_0000
98E0	LPCG working status (LPCG99_STATUS0)	32	R	0000_0001
98E4	LPCG domain status (LPCG99_STATUS1)	32	R	0000_FFFF
98F0	LPCG access control (LPCG99_AUTHEN)	32	RW	FFFF_0000
9900	LPCG direct control (LPCG100_DIRECT)	32	RW	0000_0001
9904	Low power mode information transfer status (LPCG100_LPM_STATUS0)	32	R	0000_0000
9908	Low power mode information transfer status (LPCG100_LPM_STATUS1)	32	R	0000_0000
9910	LPCG low power mode setting (LPCG100_LPM0)	32	RW	0000_0000
9914	LPCG low power mode setting (LPCG100_LPM1)	32	RW	0000_0000
991C	LPM setting of current CPU domain (LPCG100_LPM_CUR)	32	RW	0000_0000
9920	LPCG working status (LPCG100_STATUS0)	32	R	0000_0001
9924	LPCG domain status (LPCG100_STATUS1)	32	R	0000_FFFF
9930	LPCG access control (LPCG100_AUTHEN)	32	RW	FFFF_0000
9940	LPCG direct control (LPCG101_DIRECT)	32	RW	0000_0001
9944	Low power mode information transfer status (LPCG101_LPM_STATUS0)	32	R	0000_0000
9948	Low power mode information transfer status (LPCG101_LPM_STATUS1)	32	R	0000_0000
9950	LPCG low power mode setting (LPCG101_LPM0)	32	RW	0000_0000
9954	LPCG low power mode setting (LPCG101_LPM1)	32	RW	0000_0000
995C	LPM setting of current CPU domain (LPCG101_LPM_CUR)	32	RW	0000_0000
9960	LPCG working status (LPCG101_STATUS0)	32	R	0000_0001
9964	LPCG domain status (LPCG101_STATUS1)	32	R	0000_FFFF
9970	LPCG access control (LPCG101_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9980	LPCG direct control (LPCG102_DIRECT)	32	RW	0000_0001
9984	Low power mode information transfer status (LPCG102_LPM_STATUS0)	32	R	0000_0000
9988	Low power mode information transfer status (LPCG102_LPM_STATUS1)	32	R	0000_0000
9990	LPCG low power mode setting (LPCG102_LPM0)	32	RW	0000_0000
9994	LPCG low power mode setting (LPCG102_LPM1)	32	RW	0000_0000
999C	LPM setting of current CPU domain (LPCG102_LPM_CUR)	32	RW	0000_0000
99A0	LPCG working status (LPCG102_STATUS0)	32	R	0000_0001
99A4	LPCG domain status (LPCG102_STATUS1)	32	R	0000_FFFF
99B0	LPCG access control (LPCG102_AUTHEN)	32	RW	FFFF_0000
99C0	LPCG direct control (LPCG103_DIRECT)	32	RW	0000_0001
99C4	Low power mode information transfer status (LPCG103_LPM_STATUS0)	32	R	0000_0000
99C8	Low power mode information transfer status (LPCG103_LPM_STATUS1)	32	R	0000_0000
99D0	LPCG low power mode setting (LPCG103_LPM0)	32	RW	0000_0000
99D4	LPCG low power mode setting (LPCG103_LPM1)	32	RW	0000_0000
99DC	LPM setting of current CPU domain (LPCG103_LPM_CUR)	32	RW	0000_0000
99E0	LPCG working status (LPCG103_STATUS0)	32	R	0000_0001
99E4	LPCG domain status (LPCG103_STATUS1)	32	R	0000_FFFF
99F0	LPCG access control (LPCG103_AUTHEN)	32	RW	FFFF_0000
9A00	LPCG direct control (LPCG104_DIRECT)	32	RW	0000_0001
9A04	Low power mode information transfer status (LPCG104_LPM_STATUS0)	32	R	0000_0000
9A08	Low power mode information transfer status (LPCG104_LPM_STATUS1)	32	R	0000_0000
9A10	LPCG low power mode setting (LPCG104_LPM0)	32	RW	0000_0000
9A14	LPCG low power mode setting (LPCG104_LPM1)	32	RW	0000_0000
9A1C	LPM setting of current CPU domain (LPCG104_LPM_CUR)	32	RW	0000_0000
9A20	LPCG working status (LPCG104_STATUS0)	32	R	0000_0001
9A24	LPCG domain status (LPCG104_STATUS1)	32	R	0000_FFFF
9A30	LPCG access control (LPCG104_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9A40	LPCG direct control (LPCG105_DIRECT)	32	RW	0000_0001
9A44	Low power mode information transfer status (LPCG105_LPM_STATUS0)	32	R	0000_0000
9A48	Low power mode information transfer status (LPCG105_LPM_STATUS1)	32	R	0000_0000
9A50	LPCG low power mode setting (LPCG105_LPM0)	32	RW	0000_0000
9A54	LPCG low power mode setting (LPCG105_LPM1)	32	RW	0000_0000
9A5C	LPM setting of current CPU domain (LPCG105_LPM_CUR)	32	RW	0000_0000
9A60	LPCG working status (LPCG105_STATUS0)	32	R	0000_0001
9A64	LPCG domain status (LPCG105_STATUS1)	32	R	0000_FFFF
9A70	LPCG access control (LPCG105_AUTHEN)	32	RW	FFFF_0000
9A80	LPCG direct control (LPCG106_DIRECT)	32	RW	0000_0001
9A84	Low power mode information transfer status (LPCG106_LPM_STATUS0)	32	R	0000_0000
9A88	Low power mode information transfer status (LPCG106_LPM_STATUS1)	32	R	0000_0000
9A90	LPCG low power mode setting (LPCG106_LPM0)	32	RW	0000_0000
9A94	LPCG low power mode setting (LPCG106_LPM1)	32	RW	0000_0000
9A9C	LPM setting of current CPU domain (LPCG106_LPM_CUR)	32	RW	0000_0000
9AA0	LPCG working status (LPCG106_STATUS0)	32	R	0000_0001
9AA4	LPCG domain status (LPCG106_STATUS1)	32	R	0000_FFFF
9AB0	LPCG access control (LPCG106_AUTHEN)	32	RW	FFFF_0000
9AC0	LPCG direct control (LPCG107_DIRECT)	32	RW	0000_0001
9AC4	Low power mode information transfer status (LPCG107_LPM_STATUS0)	32	R	0000_0000
9AC8	Low power mode information transfer status (LPCG107_LPM_STATUS1)	32	R	0000_0000
9AD0	LPCG low power mode setting (LPCG107_LPM0)	32	RW	0000_0000
9AD4	LPCG low power mode setting (LPCG107_LPM1)	32	RW	0000_0000
9ADC	LPM setting of current CPU domain (LPCG107_LPM_CUR)	32	RW	0000_0000
9AE0	LPCG working status (LPCG107_STATUS0)	32	R	0000_0001
9AE4	LPCG domain status (LPCG107_STATUS1)	32	R	0000_FFFF
9AF0	LPCG access control (LPCG107_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9B00	LPCG direct control (LPCG108_DIRECT)	32	RW	0000_0001
9B04	Low power mode information transfer status (LPCG108_LPM_STATUS0)	32	R	0000_0000
9B08	Low power mode information transfer status (LPCG108_LPM_STATUS1)	32	R	0000_0000
9B10	LPCG low power mode setting (LPCG108_LPM0)	32	RW	0000_0000
9B14	LPCG low power mode setting (LPCG108_LPM1)	32	RW	0000_0000
9B1C	LPM setting of current CPU domain (LPCG108_LPM_CUR)	32	RW	0000_0000
9B20	LPCG working status (LPCG108_STATUS0)	32	R	0000_0001
9B24	LPCG domain status (LPCG108_STATUS1)	32	R	0000_FFFF
9B30	LPCG access control (LPCG108_AUTHEN)	32	RW	FFFF_0000
9B40	LPCG direct control (LPCG109_DIRECT)	32	RW	0000_0001
9B44	Low power mode information transfer status (LPCG109_LPM_STATUS0)	32	R	0000_0000
9B48	Low power mode information transfer status (LPCG109_LPM_STATUS1)	32	R	0000_0000
9B50	LPCG low power mode setting (LPCG109_LPM0)	32	RW	0000_0000
9B54	LPCG low power mode setting (LPCG109_LPM1)	32	RW	0000_0000
9B5C	LPM setting of current CPU domain (LPCG109_LPM_CUR)	32	RW	0000_0000
9B60	LPCG working status (LPCG109_STATUS0)	32	R	0000_0001
9B64	LPCG domain status (LPCG109_STATUS1)	32	R	0000_FFFF
9B70	LPCG access control (LPCG109_AUTHEN)	32	RW	FFFF_0000
9B80	LPCG direct control (LPCG110_DIRECT)	32	RW	0000_0001
9B84	Low power mode information transfer status (LPCG110_LPM_STATUS0)	32	R	0000_0000
9B88	Low power mode information transfer status (LPCG110_LPM_STATUS1)	32	R	0000_0000
9B90	LPCG low power mode setting (LPCG110_LPM0)	32	RW	0000_0000
9B94	LPCG low power mode setting (LPCG110_LPM1)	32	RW	0000_0000
9B9C	LPM setting of current CPU domain (LPCG110_LPM_CUR)	32	RW	0000_0000
9BA0	LPCG working status (LPCG110_STATUS0)	32	R	0000_0001
9BA4	LPCG domain status (LPCG110_STATUS1)	32	R	0000_FFFF
9BB0	LPCG access control (LPCG110_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9BC0	LPCG direct control (LPCG111_DIRECT)	32	RW	0000_0001
9BC4	Low power mode information transfer status (LPCG111_LPM_STATUS0)	32	R	0000_0000
9BC8	Low power mode information transfer status (LPCG111_LPM_STATUS1)	32	R	0000_0000
9BD0	LPCG low power mode setting (LPCG111_LPM0)	32	RW	0000_0000
9BD4	LPCG low power mode setting (LPCG111_LPM1)	32	RW	0000_0000
9BDC	LPM setting of current CPU domain (LPCG111_LPM_CUR)	32	RW	0000_0000
9BE0	LPCG working status (LPCG111_STATUS0)	32	R	0000_0001
9BE4	LPCG domain status (LPCG111_STATUS1)	32	R	0000_FFFF
9BF0	LPCG access control (LPCG111_AUTHEN)	32	RW	FFFF_0000
9C00	LPCG direct control (LPCG112_DIRECT)	32	RW	0000_0001
9C04	Low power mode information transfer status (LPCG112_LPM_STATUS0)	32	R	0000_0000
9C08	Low power mode information transfer status (LPCG112_LPM_STATUS1)	32	R	0000_0000
9C10	LPCG low power mode setting (LPCG112_LPM0)	32	RW	0000_0000
9C14	LPCG low power mode setting (LPCG112_LPM1)	32	RW	0000_0000
9C1C	LPM setting of current CPU domain (LPCG112_LPM_CUR)	32	RW	0000_0000
9C20	LPCG working status (LPCG112_STATUS0)	32	R	0000_0001
9C24	LPCG domain status (LPCG112_STATUS1)	32	R	0000_FFFF
9C30	LPCG access control (LPCG112_AUTHEN)	32	RW	FFFF_0000
9C40	LPCG direct control (LPCG113_DIRECT)	32	RW	0000_0001
9C44	Low power mode information transfer status (LPCG113_LPM_STATUS0)	32	R	0000_0000
9C48	Low power mode information transfer status (LPCG113_LPM_STATUS1)	32	R	0000_0000
9C50	LPCG low power mode setting (LPCG113_LPM0)	32	RW	0000_0000
9C54	LPCG low power mode setting (LPCG113_LPM1)	32	RW	0000_0000
9C5C	LPM setting of current CPU domain (LPCG113_LPM_CUR)	32	RW	0000_0000
9C60	LPCG working status (LPCG113_STATUS0)	32	R	0000_0001
9C64	LPCG domain status (LPCG113_STATUS1)	32	R	0000_FFFF
9C70	LPCG access control (LPCG113_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9C80	LPCG direct control (LPCG114_DIRECT)	32	RW	0000_0001
9C84	Low power mode information transfer status (LPCG114_LPM_STATUS0)	32	R	0000_0000
9C88	Low power mode information transfer status (LPCG114_LPM_STATUS1)	32	R	0000_0000
9C90	LPCG low power mode setting (LPCG114_LPM0)	32	RW	0000_0000
9C94	LPCG low power mode setting (LPCG114_LPM1)	32	RW	0000_0000
9C9C	LPM setting of current CPU domain (LPCG114_LPM_CUR)	32	RW	0000_0000
9CA0	LPCG working status (LPCG114_STATUS0)	32	R	0000_0001
9CA4	LPCG domain status (LPCG114_STATUS1)	32	R	0000_FFFF
9CB0	LPCG access control (LPCG114_AUTHEN)	32	RW	FFFF_0000
9CC0	LPCG direct control (LPCG115_DIRECT)	32	RW	0000_0001
9CC4	Low power mode information transfer status (LPCG115_LPM_STATUS0)	32	R	0000_0000
9CC8	Low power mode information transfer status (LPCG115_LPM_STATUS1)	32	R	0000_0000
9CD0	LPCG low power mode setting (LPCG115_LPM0)	32	RW	0000_0000
9CD4	LPCG low power mode setting (LPCG115_LPM1)	32	RW	0000_0000
9CDC	LPM setting of current CPU domain (LPCG115_LPM_CUR)	32	RW	0000_0000
9CE0	LPCG working status (LPCG115_STATUS0)	32	R	0000_0001
9CE4	LPCG domain status (LPCG115_STATUS1)	32	R	0000_FFFF
9CF0	LPCG access control (LPCG115_AUTHEN)	32	RW	FFFF_0000
9D00	LPCG direct control (LPCG116_DIRECT)	32	RW	0000_0001
9D04	Low power mode information transfer status (LPCG116_LPM_STATUS0)	32	R	0000_0000
9D08	Low power mode information transfer status (LPCG116_LPM_STATUS1)	32	R	0000_0000
9D10	LPCG low power mode setting (LPCG116_LPM0)	32	RW	0000_0000
9D14	LPCG low power mode setting (LPCG116_LPM1)	32	RW	0000_0000
9D1C	LPM setting of current CPU domain (LPCG116_LPM_CUR)	32	RW	0000_0000
9D20	LPCG working status (LPCG116_STATUS0)	32	R	0000_0001
9D24	LPCG domain status (LPCG116_STATUS1)	32	R	0000_FFFF
9D30	LPCG access control (LPCG116_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9D40	LPCG direct control (LPCG117_DIRECT)	32	RW	0000_0001
9D44	Low power mode information transfer status (LPCG117_LPM_STATUS0)	32	R	0000_0000
9D48	Low power mode information transfer status (LPCG117_LPM_STATUS1)	32	R	0000_0000
9D50	LPCG low power mode setting (LPCG117_LPM0)	32	RW	0000_0000
9D54	LPCG low power mode setting (LPCG117_LPM1)	32	RW	0000_0000
9D5C	LPM setting of current CPU domain (LPCG117_LPM_CUR)	32	RW	0000_0000
9D60	LPCG working status (LPCG117_STATUS0)	32	R	0000_0001
9D64	LPCG domain status (LPCG117_STATUS1)	32	R	0000_FFFF
9D70	LPCG access control (LPCG117_AUTHEN)	32	RW	FFFF_0000
9D80	LPCG direct control (LPCG118_DIRECT)	32	RW	0000_0001
9D84	Low power mode information transfer status (LPCG118_LPM_STATUS0)	32	R	0000_0000
9D88	Low power mode information transfer status (LPCG118_LPM_STATUS1)	32	R	0000_0000
9D90	LPCG low power mode setting (LPCG118_LPM0)	32	RW	0000_0000
9D94	LPCG low power mode setting (LPCG118_LPM1)	32	RW	0000_0000
9D9C	LPM setting of current CPU domain (LPCG118_LPM_CUR)	32	RW	0000_0000
9DA0	LPCG working status (LPCG118_STATUS0)	32	R	0000_0001
9DA4	LPCG domain status (LPCG118_STATUS1)	32	R	0000_FFFF
9DB0	LPCG access control (LPCG118_AUTHEN)	32	RW	FFFF_0000
9DC0	LPCG direct control (LPCG119_DIRECT)	32	RW	0000_0001
9DC4	Low power mode information transfer status (LPCG119_LPM_STATUS0)	32	R	0000_0000
9DC8	Low power mode information transfer status (LPCG119_LPM_STATUS1)	32	R	0000_0000
9DD0	LPCG low power mode setting (LPCG119_LPM0)	32	RW	0000_0000
9DD4	LPCG low power mode setting (LPCG119_LPM1)	32	RW	0000_0000
9DDC	LPM setting of current CPU domain (LPCG119_LPM_CUR)	32	RW	0000_0000
9DE0	LPCG working status (LPCG119_STATUS0)	32	R	0000_0001
9DE4	LPCG domain status (LPCG119_STATUS1)	32	R	0000_FFFF
9DF0	LPCG access control (LPCG119_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9E00	LPCG direct control (LPCG120_DIRECT)	32	RW	0000_0001
9E04	Low power mode information transfer status (LPCG120_LPM_STATUS0)	32	R	0000_0000
9E08	Low power mode information transfer status (LPCG120_LPM_STATUS1)	32	R	0000_0000
9E10	LPCG low power mode setting (LPCG120_LPM0)	32	RW	0000_0000
9E14	LPCG low power mode setting (LPCG120_LPM1)	32	RW	0000_0000
9E1C	LPM setting of current CPU domain (LPCG120_LPM_CUR)	32	RW	0000_0000
9E20	LPCG working status (LPCG120_STATUS0)	32	R	0000_0001
9E24	LPCG domain status (LPCG120_STATUS1)	32	R	0000_FFFF
9E30	LPCG access control (LPCG120_AUTHEN)	32	RW	FFFF_0000
9E40	LPCG direct control (LPCG121_DIRECT)	32	RW	0000_0001
9E44	Low power mode information transfer status (LPCG121_LPM_STATUS0)	32	R	0000_0000
9E48	Low power mode information transfer status (LPCG121_LPM_STATUS1)	32	R	0000_0000
9E50	LPCG low power mode setting (LPCG121_LPM0)	32	RW	0000_0000
9E54	LPCG low power mode setting (LPCG121_LPM1)	32	RW	0000_0000
9E5C	LPM setting of current CPU domain (LPCG121_LPM_CUR)	32	RW	0000_0000
9E60	LPCG working status (LPCG121_STATUS0)	32	R	0000_0001
9E64	LPCG domain status (LPCG121_STATUS1)	32	R	0000_FFFF
9E70	LPCG access control (LPCG121_AUTHEN)	32	RW	FFFF_0000
9E80	LPCG direct control (LPCG122_DIRECT)	32	RW	0000_0001
9E84	Low power mode information transfer status (LPCG122_LPM_STATUS0)	32	R	0000_0000
9E88	Low power mode information transfer status (LPCG122_LPM_STATUS1)	32	R	0000_0000
9E90	LPCG low power mode setting (LPCG122_LPM0)	32	RW	0000_0000
9E94	LPCG low power mode setting (LPCG122_LPM1)	32	RW	0000_0000
9E9C	LPM setting of current CPU domain (LPCG122_LPM_CUR)	32	RW	0000_0000
9EA0	LPCG working status (LPCG122_STATUS0)	32	R	0000_0001
9EA4	LPCG domain status (LPCG122_STATUS1)	32	R	0000_FFFF
9EB0	LPCG access control (LPCG122_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9EC0	LPCG direct control (LPCG123_DIRECT)	32	RW	0000_0001
9EC4	Low power mode information transfer status (LPCG123_LPM_STATUS0)	32	R	0000_0000
9EC8	Low power mode information transfer status (LPCG123_LPM_STATUS1)	32	R	0000_0000
9ED0	LPCG low power mode setting (LPCG123_LPM0)	32	RW	0000_0000
9ED4	LPCG low power mode setting (LPCG123_LPM1)	32	RW	0000_0000
9EDC	LPM setting of current CPU domain (LPCG123_LPM_CUR)	32	RW	0000_0000
9EE0	LPCG working status (LPCG123_STATUS0)	32	R	0000_0001
9EE4	LPCG domain status (LPCG123_STATUS1)	32	R	0000_FFFF
9EF0	LPCG access control (LPCG123_AUTHEN)	32	RW	FFFF_0000
9F00	LPCG direct control (LPCG124_DIRECT)	32	RW	0000_0001
9F04	Low power mode information transfer status (LPCG124_LPM_STATUS0)	32	R	0000_0000
9F08	Low power mode information transfer status (LPCG124_LPM_STATUS1)	32	R	0000_0000
9F10	LPCG low power mode setting (LPCG124_LPM0)	32	RW	0000_0000
9F14	LPCG low power mode setting (LPCG124_LPM1)	32	RW	0000_0000
9F1C	LPM setting of current CPU domain (LPCG124_LPM_CUR)	32	RW	0000_0000
9F20	LPCG working status (LPCG124_STATUS0)	32	R	0000_0001
9F24	LPCG domain status (LPCG124_STATUS1)	32	R	0000_FFFF
9F30	LPCG access control (LPCG124_AUTHEN)	32	RW	FFFF_0000
9F40	LPCG direct control (LPCG125_DIRECT)	32	RW	0000_0001
9F44	Low power mode information transfer status (LPCG125_LPM_STATUS0)	32	R	0000_0000
9F48	Low power mode information transfer status (LPCG125_LPM_STATUS1)	32	R	0000_0000
9F50	LPCG low power mode setting (LPCG125_LPM0)	32	RW	0000_0000
9F54	LPCG low power mode setting (LPCG125_LPM1)	32	RW	0000_0000
9F5C	LPM setting of current CPU domain (LPCG125_LPM_CUR)	32	RW	0000_0000
9F60	LPCG working status (LPCG125_STATUS0)	32	R	0000_0001
9F64	LPCG domain status (LPCG125_STATUS1)	32	R	0000_FFFF
9F70	LPCG access control (LPCG125_AUTHEN)	32	RW	FFFF_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
9F80	LPCG direct control (LPCG126_DIRECT)	32	RW	0000_0001
9F84	Low power mode information transfer status (LPCG126_LPM_STATUS0)	32	R	0000_0000
9F88	Low power mode information transfer status (LPCG126_LPM_STATUS1)	32	R	0000_0000
9F90	LPCG low power mode setting (LPCG126_LPM0)	32	RW	0000_0000
9F94	LPCG low power mode setting (LPCG126_LPM1)	32	RW	0000_0000
9F9C	LPM setting of current CPU domain (LPCG126_LPM_CUR)	32	RW	0000_0000
9FA0	LPCG working status (LPCG126_STATUS0)	32	R	0000_0001
9FA4	LPCG domain status (LPCG126_STATUS1)	32	R	0000_FFFF
9FB0	LPCG access control (LPCG126_AUTHEN)	32	RW	FFFF_0000

30.5.1.2 Clock Root Control Register (CLOCK_ROOT0_CONTROL - CLOCK_ROOT94_CONTROL)

This register controls clock root generation, including enable, MUX selection and division factor.

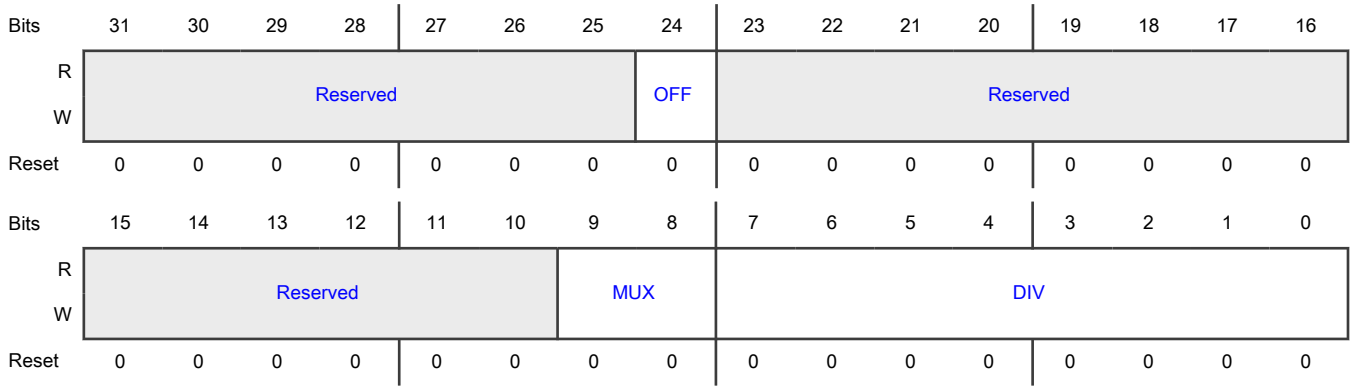
Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

For a = 0 to 94:

Register	Offset	Description
CLOCK_ROOTa_CONTROL	0h + (a × 80h)	Clock Root Control Register
CLOCK_ROOTa_CONTROL_SET	4h + (a × 80h)	Writing 1 to a bit in this register ensures that the corresponding bit in CLOCK_ROOTa_CONTROL is 1
CLOCK_ROOTa_CONTROL_CLR	8h + (a × 80h)	Writing 1 to a bit in this register ensures that the corresponding bit in CLOCK_ROOTa_CONTROL is 0
CLOCK_ROOTa_CONTROL_TOG	Ch + (a × 80h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in CLOCK_ROOTa_CONTROL

Diagram



Fields

Field	Description
31-25 —	Reserved
24 OFF	Shutdown clock root. Turn off the generated clock root or not. 0 - Clock is running. 1 - Turn off clock.
23-10 —	Reserved
9-8 MUX	Clock multiplexer. Select clock from 4 clock sources to generate the clock root. 00 - Select clock source 0 01 - Select clock source 1 10 - Select clock source 2 11 - Select clock source 3
7-0 DIV	Clock division fraction. The period of generated clock root is DIV + 1 times longer than selected clock source.

30.5.1.3 Clock root working status (CLOCK_ROOT0_STATUS0 - CLOCK_ROOT94_STATUS0)

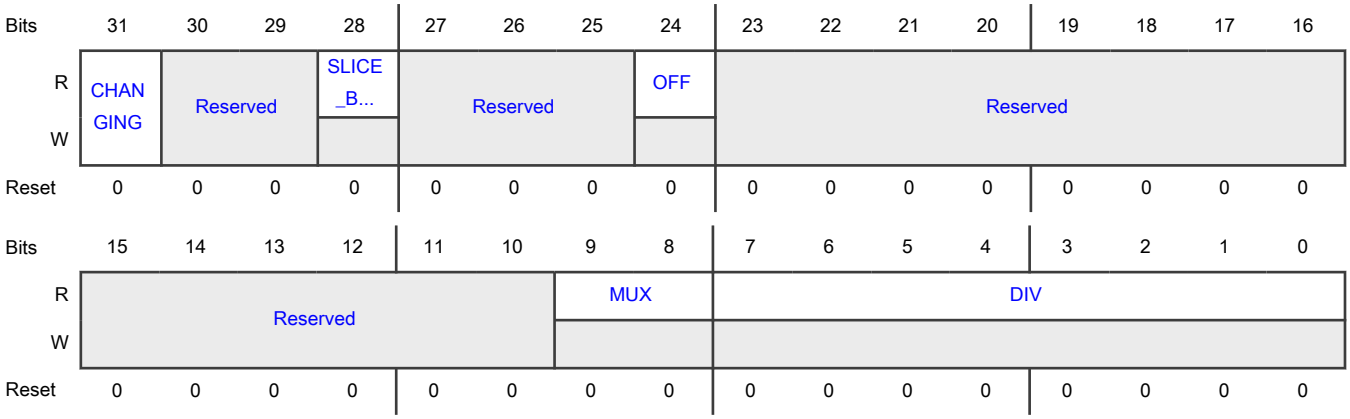
This register shows current clock root running status.

Offset

For a = 0 to 94:

Register	Offset
CLOCK_ROOTa_STATUS0	20h + (a × 80h)

Diagram



Fields

Field	Description
31 CHANGING	Internal updating in clock root This bit indicates the update status of internal clock root logic. 0 - Clock Status is not updating currently 1 - Clock generation logic is updating currently
30-29 —	Reserved
28 SLICE_BUSY	Indication for clock generation logic is applying new setting. 0 - Clock generation logic is not busy. 1 - Clock generation logic is applying new setting.
27-25 —	Reserved
24 OFF	Current clock root OFF setting Current running state of OFF field for clock root. 0 - Clock is running. 1 - Turn off clock.
23-10 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
9-8 MUX	Current clock root MUX setting Current running state of MUX field for clock root.
7-0 DIV	Current clock root DIV setting Current running state of DIV field for clock root. The period of generated clock root is DIV + 1 times longer than selected clock source.

30.5.1.4 Clock root access control (CLOCK_ROOT0_AUTHEN - CLOCK_ROOT94_AUTHEN)

This register controls the access to this Clock Root slice. The access control includes 2 parts: TrustZone control and domain based granting.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

For a = 0 to 94:

Register	Offset	Description
CLOCK_ROOTa_AUTHEN	30h + (a × 80h)	Clock root access control
CLOCK_ROOTa_AUTHEN_SET	34h + (a × 80h)	Writing 1 to a bit in this register ensures that the corresponding bit in CLOCK_ROOTa_AUTHEN is 1
CLOCK_ROOTa_AUTHEN_CLR	38h + (a × 80h)	Writing 1 to a bit in this register ensures that the corresponding bit in CLOCK_ROOTa_AUTHEN is 0
CLOCK_ROOTa_AUTHEN_TOG	3Ch + (a × 80h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in CLOCK_ROOTa_AUTHEN

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WHITE_LIST															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LOCK_LI...	Reserved			LOCK_TZ	Reserved	TZ_NS	TZ_USER	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 WHITE_LIST	Whitelist settings Each master(Core or non-Core) can be assigned to any domains among 0-15 through TRDC and through GPC domain assignment register. Once its domains are assigned in TRDC, then its accesses will have its unique domain ID. Each bit in this field represent for one domain. Bit 16-31 represent domains 0-15 respectively. Only the TRDC domains whose corresponding bits are 1 can write to this register. Write violation will fail without any exception or information. Read violation can still get read data. In LPM mode, only the GPC domains whose corresponding bits are 1 can trigger the powering down of the given clock root.
15 LOCK_LIST	Lock white list This bit locks whitelist of this Clock Root. When this bit is set, CLOCK_ROOTn_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - Whitelist is not locked. 1 - Whitelist is locked.
14-12 —	Reserved
11 LOCK_TZ	Lock TrustZone settings This bit locks TrustZone settings. When this bit is set, CLOCK_ROOTn_AUTHEN[TZ_USER] and CLOCK_ROOTn_AUTHEN[TZ_NS] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - TrustZone setting is not locked. 1 - TrustZone setting is locked.
10 —	Reserved
9 TZ_NS	Non-secure access permission Whether the registers of this Clock Root can be changed when CPU is in non-secure mode. The default value is 0, that means only the access from CPU in secure mode can change the registers of this Clock Root. 0 - Cannot be changed in Non-secure mode. 1 - Can be changed in Non-secure mode.
8 TZ_USER	User access permission Whether the registers of this Clock Root can be changed when CPU is in user mode. The default value is 0, that means only the access from CPU in supervisor mode can change the registers of this Clock Root. 0 - Clock Root settings cannot be changed in user mode. 1 - Clock Root settings can be changed in user mode.
7-0	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	

30.5.1.5 General Purpose Register (GPR_SHARED0)

These registers are shared general purpose registers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
GPR_SHARED0	4800h	General Purpose Register
GPR_SHARED0_SET	4804h	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHARED0 is 1
GPR_SHARED0_CLR	4808h	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHARED0 is 0
GPR_SHARED0_TOG	480Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_SHARED0

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 —	Reserved

30.5.1.6 GPR access control (GPR_SHARED0_AUTHEN)

This register manages GPR access control. The access control includes 2 parts: TrustZone control and domain based granting.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
GPR_SHARED0_AUTHEN	4810h	GPR access control
GPR_SHARED0_AUTHEN_SET	4814h	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHARED0_AUTHEN is 1
GPR_SHARED0_AUTHEN_CLR	4818h	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHARED0_AUTHEN is 0
GPR_SHARED0_AUTHEN_TOG	481Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_SHARED0_AUTHEN

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WHITE_LIST															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LOCK_LI...	Reserved			LOCK_TZ	Reserved	TZ_NS	TZ_USER	Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 WHITE_LIST	Whitelist settings Each master(Core or non-Core) can be assigned to any domains among 0-15 through TRDC and through GPC domain assignment register. Once its domains are assigned in TRDC, then its accesses will have its unique domain ID. Each bit in this field represent for one domain. Bit 16-31 represent domains 0-15 respectively. Only the TRDC domains whose corresponding bits are 1 can write to this register. Write violation will fail without any exception or information. Read violation can still get read data. In LPM mode, only the GPC domains whose corresponding bits are 1 can trigger the powering down of the shared GPR.
15 LOCK_LIST	Lock white list This bit lock whitelist. When this bit is set, GPR_SHAREDn_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - Whitelist is not locked. 1 - Whitelist is locked.
14-12	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
11 LOCK_TZ	Lock TrustZone settings This bit lock TrustZone settings. When this bit is set, GPR_SHAREDn_AUTHEN[TZ_USER] and GPR_SHAREDn_AUTHEN[TZ_NS] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - TrustZone settings is not locked. 1 - TrustZone settings is locked.
10 —	Reserved
9 TZ_NS	Non-secure access permission Whether the registers of this shared GPR can be changed when CPU is in non-secure mode. The default value is 0, that means only the access from CPU in secure mode can change the registers of this shared GPR. 0 - Cannot be changed in Non-secure mode. 1 - Can be changed in Non-secure mode.
8 TZ_USER	User access permission Whether the registers of this shared GPR can be changed when CPU is in user mode. The default value is 0, that means only the access from CPU in supervisor mode can change the registers of this shared GPR. 0 - Registers of shared GPR slice cannot be changed in user mode. 1 - Registers of shared GPR slice can be changed in user mode.
7-0 —	Reserved

30.5.1.7 General Purpose Register (GPR_SHARED1)

These registers are shared general purpose registers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

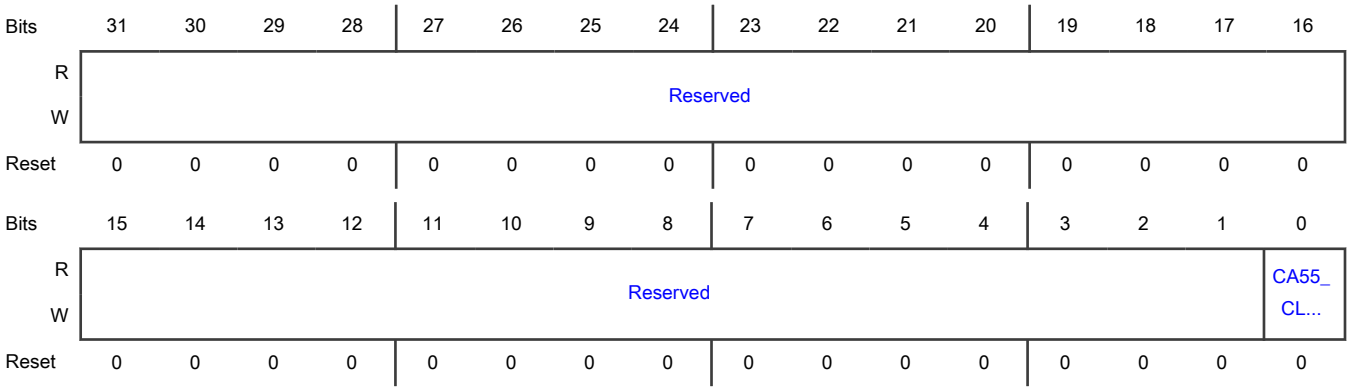
Register	Offset	Description
GPR_SHARED1	4820h	General Purpose Register
GPR_SHARED1_SET	4824h	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHARED1 is 1

Table continues on the next page...

Table continued from the previous page...

Register	Offset	Description
GPR_SHARED1_CLR	4828h	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHARED1 is 0
GPR_SHARED1_TOG	482Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_SHARED1

Diagram



Fields

Field	Description
31-1 —	Reserved
0 CA55_CLOCK_SELECT	Clock select signal between ccm clock root and ARM PLL clock This signal goes to ca55mix, used as clock select signal for ca55 0 - Clkroot_arm_a55 is used. 1 - The clock output of ARM PLL is selected.

30.5.1.8 GPR access control (GPR_SHARED1_AUTHEN)

This register manages GPR access control. The access control includes 2 parts: TrustZone control and domain based granting.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

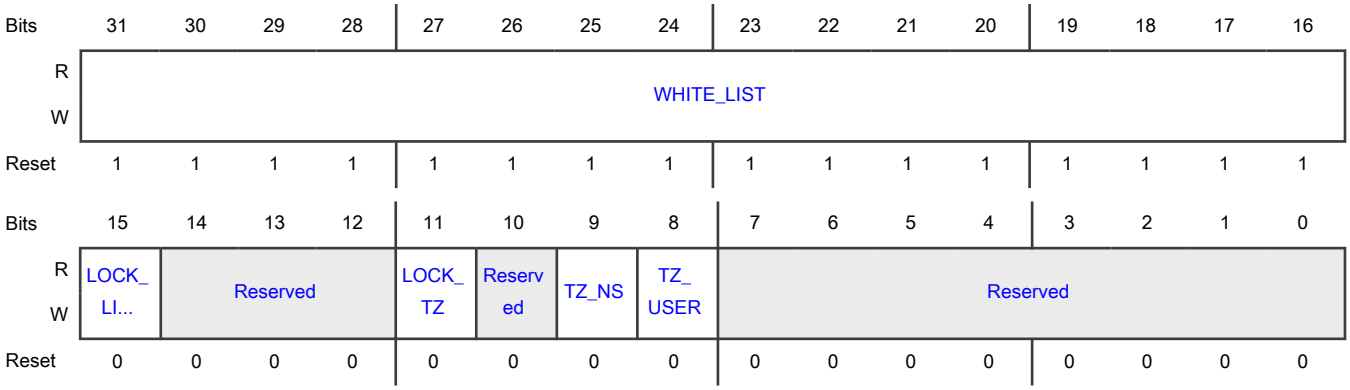
Register	Offset	Description
GPR_SHARED1_AUTHEN	4830h	GPR access control

Table continues on the next page...

Table continued from the previous page...

Register	Offset	Description
GPR_SHARED1_AUTHEN_SET	4834h	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHARED1_AUTHEN is 1
GPR_SHARED1_AUTHEN_CLR	4838h	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHARED1_AUTHEN is 0
GPR_SHARED1_AUTHEN_TOG	483Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_SHARED1_AUTHEN

Diagram



Fields

Field	Description
31-16 WHITE_LIST	Whitelist settings Each master(Core or non-Core) can be assigned to any domains among 0-15 through TRDC and through GPC domain assignment register. Once its domains are assigned in TRDC, then its accesses will have its unique domain ID. Each bit in this field represent for one domain. Bit 16-31 represent domains 0-15 respectively. Only the TRDC domains whose corresponding bits are 1 can write to this register. Write violation will fail without any exception or information. Read violation can still get read data. In LPM mode, only the GPC domains whose corresponding bits are 1 can trigger the powering down of the shared GPR.
15 LOCK_LIST	Lock white list This bit lock whitelist. When this bit is set, GPR_SHAREDn_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - Whitelist is not locked. 1 - Whitelist is locked.
14-12 —	Reserved
11	Lock TrustZone settings

Table continues on the next page...

Table continued from the previous page...

Field	Description
LOCK_TZ	This bit lock TrustZone settings. When this bit is set, GPR_SHAREDn_AUTHEN[TZ_USER] and GPR_SHAREDn_AUTHEN[TZ_NS] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - TrustZone settings is not locked. 1 - TrustZone settings is locked.
10 —	Reserved
9 TZ_NS	Non-secure access permission Whether the registers of this shared GPR can be changed when CPU is in non-secure mode. The default value is 0, that means only the access from CPU in secure mode can change the registers of this shared GPR. 0 - Cannot be changed in Non-secure mode. 1 - Can be changed in Non-secure mode.
8 TZ_USER	User access permission Whether the registers of this shared GPR can be changed when CPU is in user mode. The default value is 0, that means only the access from CPU in supervisor mode can change the registers of this shared GPR. 0 - Registers of shared GPR slice cannot be changed in user mode. 1 - Registers of shared GPR slice can be changed in user mode.
7-0 —	Reserved

30.5.1.9 General Purpose Register (GPR_SHARED2)

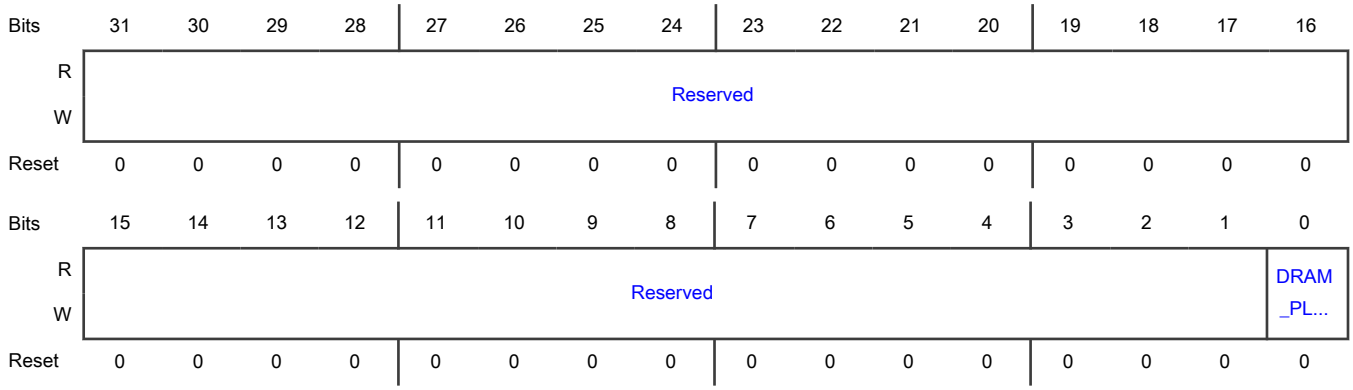
These registers are shared general purpose registers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
GPR_SHARED2	4840h	General Purpose Register
GPR_SHARED2_SET	4844h	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHARED2 is 1
GPR_SHARED2_CLR	4848h	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHARED2 is 0
GPR_SHARED2_TOG	484Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_SHARED2

Diagram



Fields

Field	Description
31-1 —	Reserved
0 DRAM_PLL_BYPASS	Clock select signal between ccm clock root and DRAM PLL clock This signal goes to ddrmix, used as clock select signal for DDR PHY. 0 - The clock output of DRAM PLL is selected. 1 - Dram_alt_clk_root is selected.

30.5.1.10 GPR access control (GPR_SHARED2_AUTHEN)

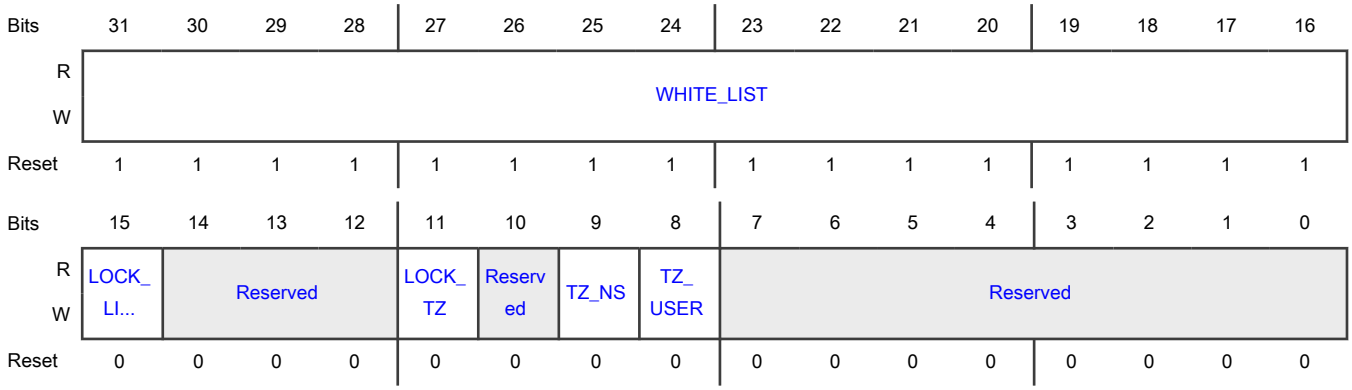
This register manages GPR access control. The access control includes 2 parts: TrustZone control and domain based granting.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
GPR_SHARED2_AUTHEN	4850h	GPR access control
GPR_SHARED2_AUTHEN_SET	4854h	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHARED2_AUTHEN is 1
GPR_SHARED2_AUTHEN_CLR	4858h	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHARED2_AUTHEN is 0
GPR_SHARED2_AUTHEN_TOG	485Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_SHARED2_AUTHEN

Diagram



Fields

Field	Description
31-16 WHITE_LIST	Whitelist settings Each master(Core or non-Core) can be assigned to any domains among 0-15 through TRDC and through GPC domain assignment register. Once its domains are assigned in TRDC, then its accesses will have its unique domain ID. Each bit in this field represent for one domain. Bit 16-31 represent domains 0-15 respectively. Only the TRDC domains whose corresponding bits are 1 can write to this register. Write violation will fail without any exception or information. Read violation can still get read data. In LPM mode, only the GPC domains whose corresponding bits are 1 can trigger the powering down of the shared GPR.
15 LOCK_LIST	Lock white list This bit lock whitelist. When this bit is set, GPR_SHAREDn_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - Whitelist is not locked. 1 - Whitelist is locked.
14-12 —	Reserved
11 LOCK_TZ	Lock TrustZone settings This bit lock TrustZone settings. When this bit is set, GPR_SHAREDn_AUTHEN[TZ_USER] and GPR_SHAREDn_AUTHEN[TZ_NS] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - TrustZone settings is not locked. 1 - TrustZone settings is locked.
10 —	Reserved
9 TZ_NS	Non-secure access permission

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Whether the registers of this shared GPR can be changed when CPU is in non-secure mode. The default value is 0, that means only the access from CPU in secure mode can change the registers of this shared GPR. 0 - Cannot be changed in Non-secure mode. 1 - Can be changed in Non-secure mode.
8 TZ_USER	User access permission Whether the registers of this shared GPR can be changed when CPU is in user mode. The default value is 0, that means only the access from CPU in supervisor mode can change the registers of this shared GPR. 0 - Registers of shared GPR slice cannot be changed in user mode. 1 - Registers of shared GPR slice can be changed in user mode.
7-0 —	Reserved

30.5.1.11 General Purpose Register (GPR_SHARED3 - GPR_SHARED7)

These registers are shared general purpose registers.

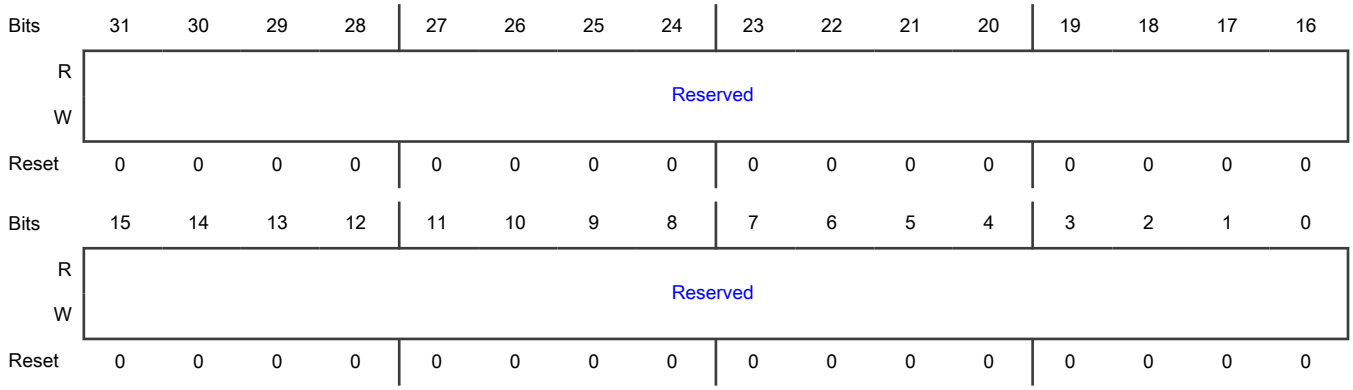
Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

For a = 3 to 7:

Register	Offset	Description
GPR_SHAREDa	4800h + (a × 20h)	General Purpose Register
GPR_SHAREDa_SET	4804h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHAREDa is 1
GPR_SHAREDa_CLR	4808h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHAREDa is 0
GPR_SHAREDa_TOG	480Ch + (a × 20h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_SHAREDa

Diagram



Fields

Field	Description
31-0 —	Reserved

30.5.1.12 GPR access control (GPR_SHARED3_AUTHEN - GPR_SHARED7_AUTHEN)

This register manages GPR access control. The access control includes 2 parts: TrustZone control and domain based granting.

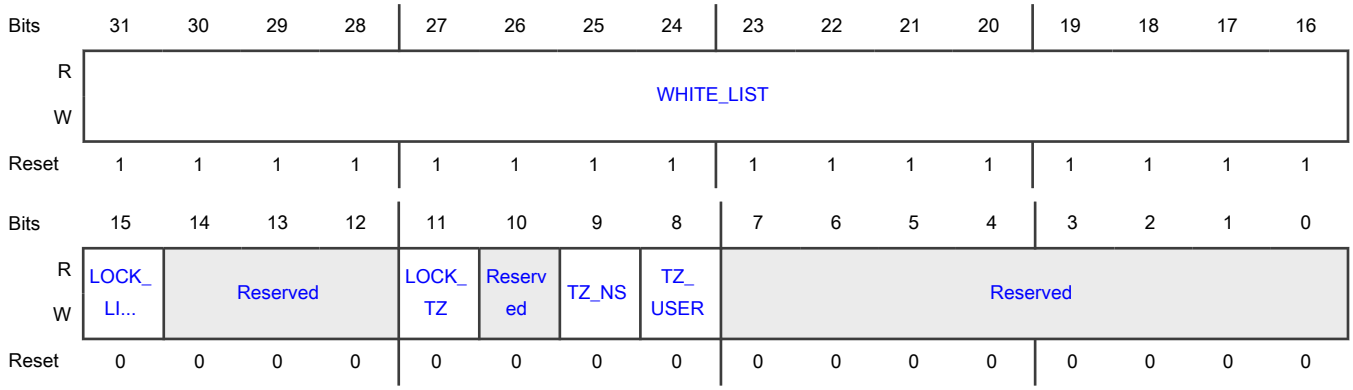
Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

For a = 3 to 7:

Register	Offset	Description
GPR_SHAREDa_AUTHEN	4810h + (a × 20h)	GPR access control
GPR_SHAREDa_AUTHEN_SET	4814h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHAREDa_AUTHEN is 1
GPR_SHAREDa_AUTHEN_CLR	4818h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHAREDa_AUTHEN is 0
GPR_SHAREDa_AUTHEN_TOG	481Ch + (a × 20h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_SHAREDa_AUTHEN

Diagram



Fields

Field	Description
31-16 WHITE_LIST	Whitelist settings Each master(Core or non-Core) can be assigned to any domains among 0-15 through TRDC and through GPC domain assignment register. Once its domains are assigned in TRDC, then its accesses will have its unique domain ID. Each bit in this field represent for one domain. Bit 16-31 represent domains 0-15 respectively. Only the TRDC domains whose corresponding bits are 1 can write to this register. Write violation will fail without any exception or information. Read violation can still get read data. In LPM mode, only the GPC domains whose corresponding bits are 1 can trigger the powering down of the shared GPR.
15 LOCK_LIST	Lock white list This bit lock whitelist. When this bit is set, GPR_SHAREDn_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - Whitelist is not locked. 1 - Whitelist is locked.
14-12 —	Reserved
11 LOCK_TZ	Lock TrustZone settings This bit lock TrustZone settings. When this bit is set, GPR_SHAREDn_AUTHEN[TZ_USER] and GPR_SHAREDn_AUTHEN[TZ_NS] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - TrustZone settings is not locked. 1 - TrustZone settings is locked.
10 —	Reserved
9 TZ_NS	Non-secure access permission

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Whether the registers of this shared GPR can be changed when CPU is in non-secure mode. The default value is 0, that means only the access from CPU in secure mode can change the registers of this shared GPR. 0 - Cannot be changed in Non-secure mode. 1 - Can be changed in Non-secure mode.
8 TZ_USER	User access permission Whether the registers of this shared GPR can be changed when CPU is in user mode. The default value is 0, that means only the access from CPU in supervisor mode can change the registers of this shared GPR. 0 - Registers of shared GPR slice cannot be changed in user mode. 1 - Registers of shared GPR slice can be changed in user mode.
7-0 —	Reserved

30.5.1.13 General puspose register (GPR_PRIVATE0 - GPR_PRIVATE7)

These registers is general purpose register.

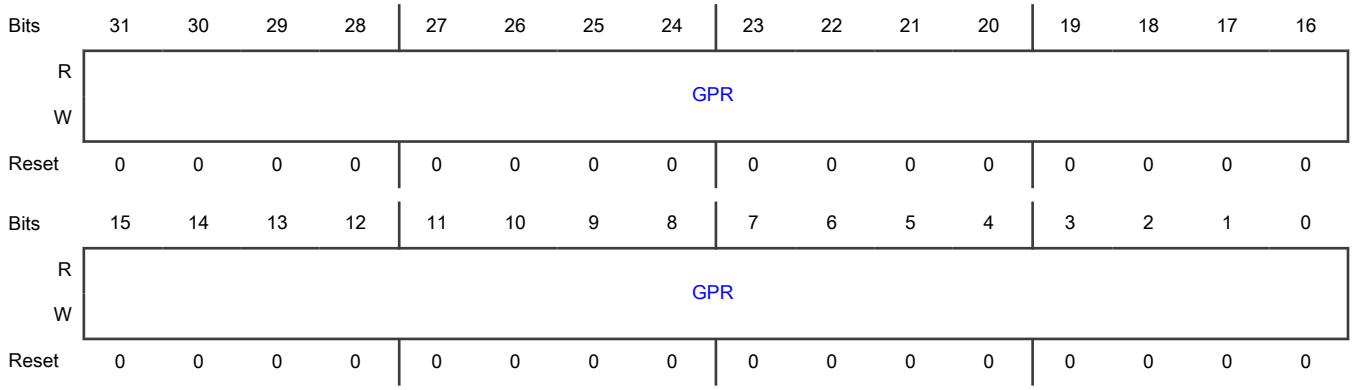
Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

For a = 0 to 7:

Register	Offset	Description
GPR_PRIVATEa	4C00h + (a × 20h)	General puspose register
GPR_PRIVATEa_SET	4C04h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_PRIVATEa is 1
GPR_PRIVATEa_CLR	4C08h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_PRIVATEa is 0
GPR_PRIVATEa_TOG	4C0Ch + (a × 20h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_PRIVATEa

Diagram



Fields

Field	Description
31-0 GPR	GP register General purpose register. This register has dedicated bits for each domain.

30.5.1.14 GPR access control (GPR_PRIVATE0_AUTHEN - GPR_PRIVATE7_AUTHEN)

This register controls the access to this private GPR slice. The access control includes 2 parts: TrustZone control and domain based granting.

NOTE

The domain based granting only influence the access to GPR_PRIVATEn_AUTHEN.

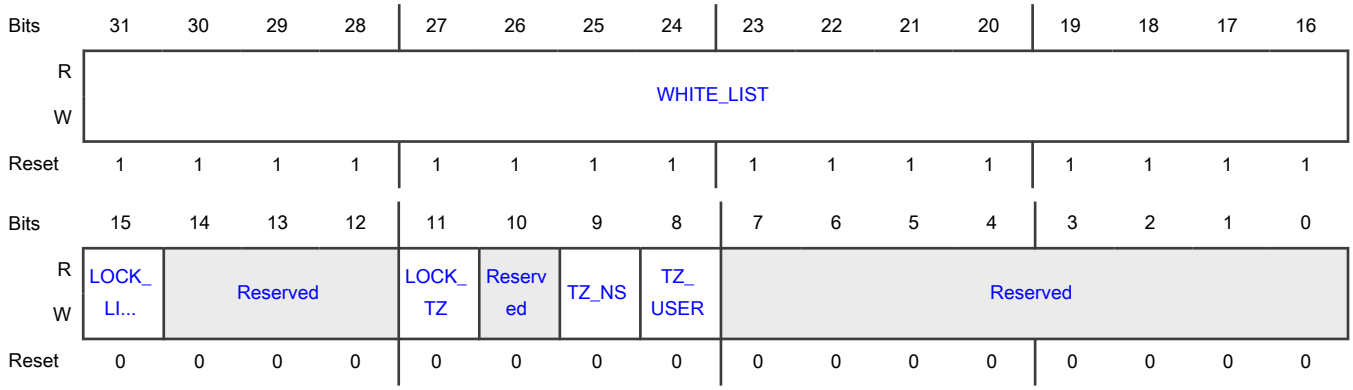
Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

For a = 0 to 7:

Register	Offset	Description
GPR_PRIVATEa_AUTHEN	4C10h + (a × 20h)	GPR access control
GPR_PRIVATEa_AUTHEN_SET	4C14h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_PRIVATEa_AUTHEN is 1
GPR_PRIVATEa_AUTHEN_CLR	4C18h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_PRIVATEa_AUTHEN is 0
GPR_PRIVATEa_AUTHEN_TOG	4C1Ch + (a × 20h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_PRIVATEa_AUTHEN

Diagram



Fields

Field	Description
31-16 WHITE_LIST	Whitelist settings Each master(Core or non-Core) can be assigned to any domains among 0-15 through TRDC and through GPC domain assignment register. Once its domains are assigned in TRDC, then its accesses will have its unique domain ID. Each bit in this field represent for one domain. Bit 16-31 represent domains 0-15 respectively. Only the TRDC domains whose corresponding bits are 1 can write to this register. Write violation will fail without any exception or information. Read violation can still get read data. In LPM mode, only the GPC domains whose corresponding bits are 1 can trigger the powering down of the given power slice.
15 LOCK_LIST	Lock white list This bit lock whitelist. When this bit is set, GPR_PRIVATEn_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - Whitelist is not locked. 1 - Whitelist is locked.
14-12 —	Reserved
11 LOCK_TZ	Lock TrustZone settings This bit lock TrustZone settings. When this bit is set, GPR_PRIVATEn_AUTHEN[TZ_USER] and GPR_PRIVATEn_AUTHEN[TZ_NS] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - TrustZone settings is not locked. 1 - TrustZone settings is locked.
10 —	Reserved
9 TZ_NS	Non-secure access permission

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Whether the registers of this private GPR can be changed when CPU is in non-secure mode. The default value is 0, that means only the access from CPU in secure mode can change the registers of this private GPR. 0 - Cannot be changed in Non-secure mode. 1 - Can be changed in Non-secure mode.
8 TZ_USER	User access permission Whether the registers of this private GPR can be changed when CPU is in user mode. The default value is 0, that means only the access from CPU in supervisor mode can change the registers of this private GPR. 0 - Registers of privat GPR cannot be changed in user mode. 1 - Registers of private GPR can be changed in user mode.
7-0 —	Reserved

30.5.1.15 Clock source direct control (OSCPLL0_DIRECT - OSCPLL18_DIRECT)

This register controls the ON/OFF of clock source when clock source is working in Direct Control mode.

Offset

For a = 0 to 18:

Register	Offset
OSCPLLa_DIRECT	5000h + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															ON
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31-1 —	Reserved
0 ON	Turn on clock source This bit controls ON/OFF of clock source. 0 - Clock source is OFF. 1 - Clock source is ON.

30.5.1.16 Low power mode information transfer status (OSCPLL0_LPM_STATUS0 - OSCPLL18_LPM_STATUS0)

This register denotes the Clock source low power mode information transfer status for DOMAIN0 ~ DOMAIM7. The information comprises the current low power mode and transfer status.

Offset

For a = 0 to 18:

Register	Offset
OSCPLLa_LPM_STATUS0	5004h + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	TRANS_R...	CPU_MODE_DOMAIN7		Reserved	TRANS_R...	CPU_MODE_DOMAIN6		Reserved	TRANS_R...	CPU_MODE_DOMAIN5		Reserved	TRANS_R...	CPU_MODE_DOMAIN4	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	TRANS_R...	CPU_MODE_DOMAIN3		Reserved	TRANS_R...	CPU_MODE_DOMAIN2		Reserved	TRANS_R...	CPU_MODE_DOMAIN1		Reserved	TRANS_R...	CPU_MODE_DOMAIN0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
30 TRANS_REQ_DOMAIN7	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
29-28 CPU_MODE_DOMAIN7	Current mode of CPU domain 7 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
27 —	Reserved
26 TRANS_REQ_DOMAIN6	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
25-24 CPU_MODE_DOMAIN6	Current mode of CPU domain 6 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
23 —	Reserved
22 TRANS_REQ_DOMAIN5	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
21-20 CPU_MODE_DOMAIN5	Current mode of CPU domain 5 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
19 —	Reserved
18	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.

Table continues on the next page...

Table continued from the previous page...

Field	Description
TRANS_REQ_DOMAIN4	
17-16 CPU_MODE_DOMAIN4	Current mode of CPU domain 4 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
15 —	Reserved
14 TRANS_REQ_DOMAIN3	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
13-12 CPU_MODE_DOMAIN3	Current mode of CPU domain 3 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
11 —	Reserved
10 TRANS_REQ_DOMAIN2	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
9-8 CPU_MODE_DOMAIN2	Current mode of CPU domain 2 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
7 —	Reserved
6 TRANS_REQ_DOMAIN1	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.

Table continues on the next page...

Table continued from the previous page...

Field	Description
5-4 CPU_MODE_DOMAIN1	Current mode of CPU domain 1 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
3 —	Reserved
2 TRANS_REQ_DOMAIN0	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
1-0 CPU_MODE_DOMAIN0	Current mode of CPU domain 0 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode

30.5.1.17 Low power mode information transfer status (OSCPLL0_LPM_STATUS1 - OSCPLL18_LPM_STATUS1)

This register denotes the Clock source low power mode information transfer status for DOMAIN8 ~ DOMAIM15. The information comprises the current low power mode and transfer status.

Offset

For a = 0 to 18:

Register	Offset
OSCPLLa_LPM_STATUS1	5008h + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	TRANS_R...	CPU_MODE_DOMAIN...		Reserved	TRANS_R...	CPU_MODE_DOMAIN...		Reserved	TRANS_R...	CPU_MODE_DOMAIN...		Reserved	TRANS_R...	CPU_MODE_DOMAIN...	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	TRANS_R...	CPU_MODE_DOMAIN...		Reserved	TRANS_R...	CPU_MODE_DOMAIN...		Reserved	TRANS_R...	CPU_MODE_DOMAIN9		Reserved	TRANS_R...	CPU_MODE_DOMAIN8	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30 TRANS_REQ_DOMAIN15	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
29-28 CPU_MODE_DOMAIN15	Current mode of CPU domain 15 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
27 —	Reserved
26 TRANS_REQ_DOMAIN14	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
25-24 CPU_MODE_DOMAIN14	Current mode of CPU domain 14 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
23	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
22 TRANS_REQ_DOMAIN13	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
21-20 CPU_MODE_DOMAIN13	Current mode of CPU domain 13 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
19 —	Reserved
18 TRANS_REQ_DOMAIN12	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
17-16 CPU_MODE_DOMAIN12	Current mode of CPU domain 12 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
15 —	Reserved
14 TRANS_REQ_DOMAIN11	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
13-12 CPU_MODE_DOMAIN11	Current mode of CPU domain 11 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
11 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
10 TRANS_REQ_DOMAIN10	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
9-8 CPU_MODE_DOMAIN10	Current mode of CPU domain 10 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
7 —	Reserved
6 TRANS_REQ_DOMAIN9	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
5-4 CPU_MODE_DOMAIN9	Current mode of CPU domain 9 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
3 —	Reserved
2 TRANS_REQ_DOMAIN8	GPC request CCM to enable or disable OSCPLL. This signal will turn to low once CCM complete response to GPC.
1-0 CPU_MODE_DOMAIN8	Current mode of CPU domain 8 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode

30.5.1.18 Clock source low power mode setting (OSCPLL0_LPM0 - OSCPLL18_LPM0)

Clock source low power mode setting for DOMAIN0 ~ DOMAIM7.

NOTE

The settings in this register are valid only when this Clock Source slice is working in CPULPM mode.

Offset

For a = 0 to 18:

Register	Offset
OSCPLLa_LPM0	5010h + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	LPM_SETTING_D7			Reserved	LPM_SETTING_D6			Reserved	LPM_SETTING_D5			Reserved	LPM_SETTING_D4		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	LPM_SETTING_D3			Reserved	LPM_SETTING_D2			Reserved	LPM_SETTING_D1			Reserved	LPM_SETTING_D0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30-28 LPM_SETTING_D7	Clock Source LPM in DOMAIN7 Clock Source low power mode setting in DOMAIN7. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
27 —	Reserved
26-24 LPM_SETTING_D6	Clock Source LPM in DOMAIN6 Clock Source low power mode setting in DOMAIN6. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
23 —	Reserved
22-20 LPM_SETTING_D5	Clock Source LPM in DOMAIN5 Clock Source low power mode setting in DOMAIN5. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
19 —	Reserved
18-16 LPM_SETTING_D4	Clock Source LPM in DOMAIN4 Clock Source low power mode setting in DOMAIN4. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
15 —	Reserved
14-12 LPM_SETTING_D3	Clock Source LPM in DOMAIN3 Clock Source low power mode setting in DOMAIN3. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
11 —	Reserved
10-8	Clock Source LPM in DOMAIN2

Table continues on the next page...

Table continued from the previous page...

Field	Description
LPM_SETTING_D2	Clock Source low power mode setting in DOMAIN2. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
7 —	Reserved
6-4 LPM_SETTING_D1	Clock Source LPM in DOMAIN1 Clock Source low power mode setting in DOMAIN1. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
3 —	Reserved
2-0 LPM_SETTING_D0	Clock Source LPM in DOMAIN0 Clock Source low power mode setting in DOMAIN0. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND Mode.

30.5.1.19 clock source low power mode setting (OSCPLL0_LPM1 - OSCPLL18_LPM1)

Clock source low power mode setting for DOMAIN8 ~ DOMAIM15.

NOTE

The settings in this register are valid only when this Clock Source slice is working in CPULPM mode.

Offset

For a = 0 to 18:

Register	Offset
OSCPLLa_LPM1	5014h + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	LPM_SETTING_D15			Reserved	LPM_SETTING_D14			Reserved	LPM_SETTING_D13			Reserved	LPM_SETTING_D12		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	LPM_SETTING_D11			Reserved	LPM_SETTING_D10			Reserved	LPM_SETTING_D9			Reserved	LPM_SETTING_D8		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30-28 LPM_SETTING_D15	Clock Source LPM in DOMAIN15 Clock Source low power mode setting in DOMAIN15. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
27 —	Reserved
26-24 LPM_SETTING_D14	Clock Source LPM in DOMAIN14 Clock Source low power mode setting in DOMAIN14. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
23	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
22-20 LPM_SETTING_D13	Clock Source LPM in DOMAIN13 Clock Source low power mode setting in DOMAIN13. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
19 —	Reserved
18-16 LPM_SETTING_D12	Clock Source LPM in DOMAIN12 Clock Source low power mode setting in DOMAIN12. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
15 —	Reserved
14-12 LPM_SETTING_D11	Clock Source LPM in DOMAIN11 Clock Source low power mode setting in DOMAIN11. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
11 —	Reserved
10-8 LPM_SETTING_D10	Clock Source LPM in DOMAIN10 Clock Source low power mode setting in DOMAIN10. 000 - Clock Source will be OFF in any CPU mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
7 —	Reserved
6-4 LPM_SETTING_D9	Clock Source LPM in DOMAIN9 Clock Source low power mode setting in DOMAIN9. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.
3 —	Reserved
2-0 LPM_SETTING_D8	Clock Source LPM in DOMAIN8 Clock Source low power mode setting in DOMAIN8. 000 - Clock Source will be OFF in any CPU mode. 001 - Clock Source will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - Clock Source will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - Clock Source will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - Clock Source will be ON in RUN/WAIT/STOP/SUSPEND mode.

30.5.1.20 LPM setting of current CPU domain (OSCPLL0_LPM_CUR - OSCPLL18_LPM_CUR)

Reading this field will returen OSCPLLn_LPM[LPM_SETTING_Dx], and x represent the CPU domain which launched the read access.

Offset

For a = 0 to 18:

Register	Offset
OSCPLLa_LPM_CUR	501Ch + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													LPM_SETTING_CUR		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-3 —	Reserved
2-0 LPM_SETTING_CUR	LPM SETTING of current CPU DOMAIN **NOTE** This field only can be changed by the cpu domain which launched the write access.

30.5.1.21 Clock source working status (OSCPLL0_STATUS0 - OSCPLL18_STATUS0)

This register shows current clock source running status.

Offset

For a = 0 to 18:

Register	Offset
OSCPLLa_STATUS0	5020h + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			IN_USE	Reserved						STATUS_...	STATUS_...	Reserved			ON
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31-13 —	Reserved
12 IN_USE	This Clock Source is being used or not. This field will be 1, if any Clock Root derived from this Clock Source is enabled. 0 - Clock Source is not being used. 1 - Clock Source is being used.
11-6 —	Reserved
5 STATUS_LATE	Late ready signal sent from PLL. This signal means clock source is ready to be used.
4 STATUS_EARLY	Early ready signal sent from PLL control module. This signal means clock source is going to be used.
3-1 —	Reserved
0 ON	Clock source current state Clock source running status. 0 - Clock source is OFF. 1 - Clock source is ON.

30.5.1.22 Clock source domain status (OSCPLL0_STATUS1 - OSCPLL18_STATUS1)

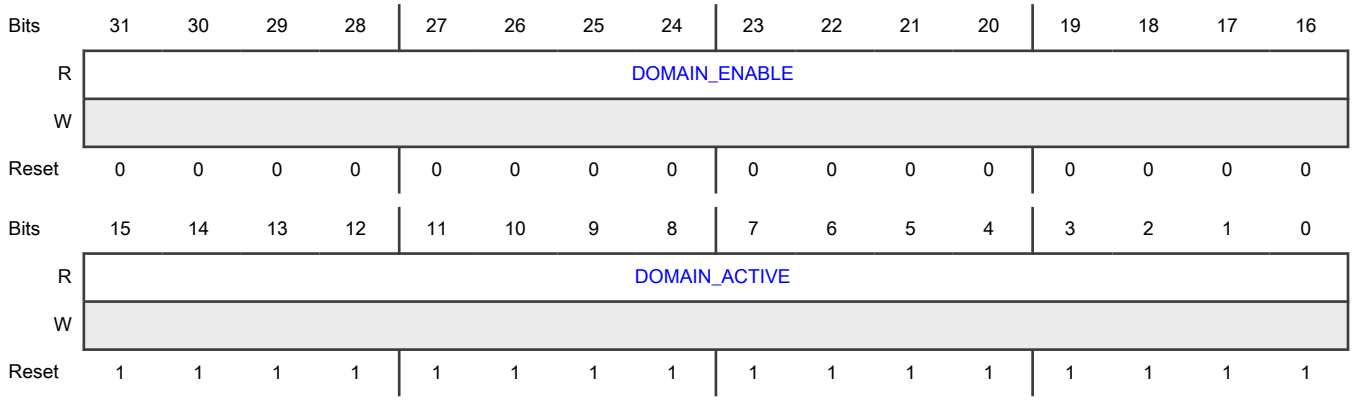
This register shows domain status of clock source.

Offset

For a = 0 to 18:

Register	Offset
OSCPLLa_STATUS1	5024h + (a × 40h)

Diagram



Fields

Field	Description
31-16 DOMAIN_ENABLE	Domain enable Domain x is enabled means that OSCPLLn_LPM_STATUS[CPU_MODE_DOMAINx] is larger than OSCPLLn_LPM[LPM_SETTING_Dx], and this clock source should be ON for DOMAINx. BIT16~BIT31 are corresponding with DOMAIN0~DOMAIN15.
15-0 DOMAIN_ACTIVE	Domain active This field is equal to OSCPLLn_AUTHEN[WHITE_LIST].

30.5.1.23 Clock Source access control (OSCPLL0_AUTHEN - OSCPLL18_AUTHEN)

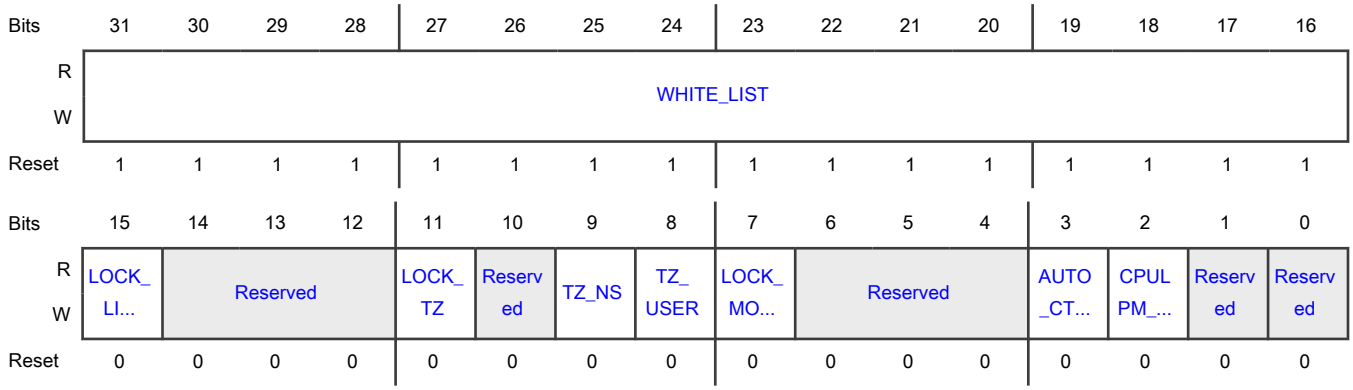
This register controls the access to this Clock Source slice. The access control includes 2 parts: TrustZone control and domain based granting.

Offset

For a = 0 to 18:

Register	Offset
OSCPLLa_AUTHEN	5030h + (a × 40h)

Diagram



Fields

Field	Description
31-16 WHITE_LIST	Whitelist Each master(Core or non-Core) can be assigned to any domains among 0-15 through TRDC and through GPC domain assignment register. Once its domains are assigned in TRDC, then its accesses will have its unique domain ID. Each bit in this field represent for one domain. Bit 16-31 represent domains 0-15 respectively. Only the TRDC domains whose corresponding bits are 1 can write to this register. Write violation will fail without any exception or information. Read violation can still get read data. In LPM mode, only the GPC domains whose corresponding bits are 1 can trigger the enabling/ disabling down of the clock Source.
15 LOCK_LIST	Lock white list This bit lock white list. When this bit is set, OSCPLLn_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - Whitelist is not locked. 1 - Whitelist is locked.
14-12 —	Reserved
11 LOCK_TZ	Lock TrustZone settings This bit locks TrustZone settings. When this bit is set, OSCPLLn_AUTHEN[TZ_USER] and OSCPLLn_AUTHEN[TZ_NS] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - TrustZone settings is not locked. 1 - TrustZone settings is locked.
10 —	Reserved
9 TZ_NS	Non-secure access permission

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Whether the registers of this Clock Source can be changed when CPU is in non-secure mode. The default value is 0, that means only the access from CPU in secure mode can change the registers of this Clock Source. 0 - Cannot be changed in Non-secure mode. 1 - Can be changed in Non-secure mode.
8 TZ_USER	User access permission Whether the registers of this Clock Source can be changed when CPU is in user mode. The default value is 0, that means only the access from CPU in supervisor mode can change the registers of this Clock Source. 0 - Clock Source settings cannot be changed in user mode. 1 - Clock Source settings can be changed in user mode.
7 LOCK_MODE	When this bit is set, CPULPM_MODE and AUTO_CREL cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - CPULPM_MODE and AUTO_CTRL is not locked. 1 - CPULPM_MODE and AUTO_CTRL is locked.
6-4 —	Reserved
3 AUTO_CTRL	Auto mode enable This filed enalbes the Auto mode. **NOTE** The actual working mode of this Clock Source is determined by OSCPLLn_AUTHEN[CPULPM_MODE] and OSCPLLn_AUTHEN[AUTO_CTRL]. 0 - Disable Auto mode 1 - Enable Auto mode
2 CPULPM_MODE	CPULPM mode enable This field enables the CPULPM mode. **NOTE** The actual working mode of this Clock Source is determined by OSCPLLn_AUTHEN[CPULPM_MODE] and OSCPLLn_AUTHEN[AUTO_CTRL]. 0 - Disable CPULPM mode. 1 - Enable CPULPM mode.
1 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 —	Reserved

30.5.1.24 LPCG direct control (LPCG0_DIRECT - LPCG126_DIRECT)

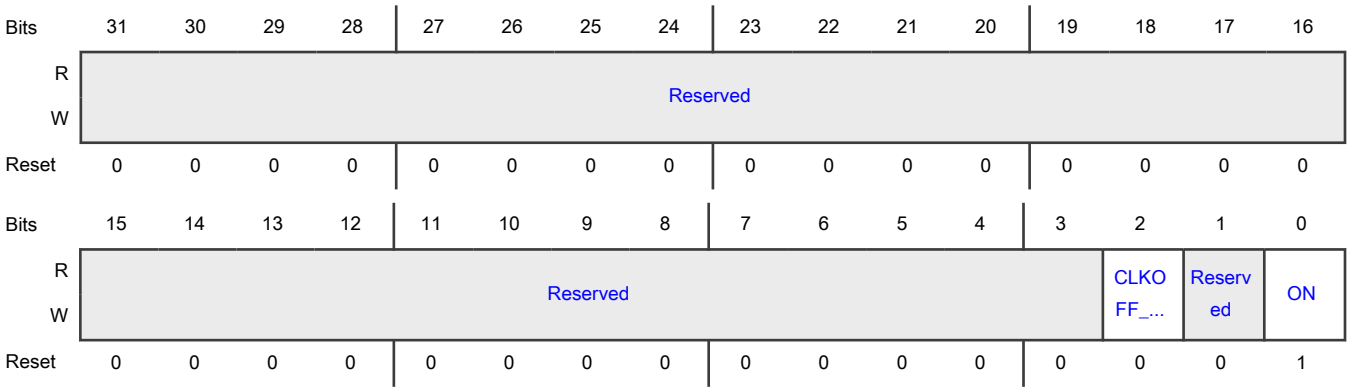
This register controls the ON/OFF of LPCG when LPCG slice is working in Direct Control mode.

Offset

For a = 0 to 126:

Register	Offset
LPCGa_DIRECT	8000h + (a × 40h)

Diagram



Fields

Field	Description
31-3 —	Reserved
2 CLKOFF_ACK_TIMEOUT_EN	Clock off handshake timeout enable This bit enables time out mechanism for clock off handshake, when enabled, clock off acknowlege will be ignored and wait for 16-cycle IPG clock timeout, when disabled, wait for clock off acknowlege. 0 - disable 1 - enable
1 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 ON	Turn on LPCG This bit controls ON/OFF of LPCG. 0 - LPCG is OFF. 1 - LPCG is ON.

30.5.1.25 Low power mode information transfer status (LPCG0_LPM_STATUS0 - LPCG126_LPM_STATUS0)

This register denotes the LPCG low power mode information transfer status for DOMAIN0 ~ DOMAIM7. The information comprises the current low power mode and transfer status.

Offset

For a = 0 to 126:

Register	Offset
LPCGa_LPM_STATUS0	8004h + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	TRANS_R...	CPU_MODE_DOMAIN7		Reserved	TRANS_R...	CPU_MODE_DOMAIN6		Reserved	TRANS_R...	CPU_MODE_DOMAIN5		Reserved	TRANS_R...	CPU_MODE_DOMAIN4	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	TRANS_R...	CPU_MODE_DOMAIN3		Reserved	TRANS_R...	CPU_MODE_DOMAIN2		Reserved	TRANS_R...	CPU_MODE_DOMAIN1		Reserved	TRANS_R...	CPU_MODE_DOMAIN0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30 TRANS_REQ_DOMAIN7	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.

Table continues on the next page...

Table continued from the previous page...

Field	Description
29-28 CPU_MODE_DOMAIN7	Current mode of CPU domain 7 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
27 —	Reserved
26 TRANS_REQ_DOMAIN6	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.
25-24 CPU_MODE_DOMAIN6	Current mode of CPU domain 6 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
23 —	Reserved
22 TRANS_REQ_DOMAIN5	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.
21-20 CPU_MODE_DOMAIN5	Current mode of CPU domain 5 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
19 —	Reserved
18 TRANS_REQ_DOMAIN4	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.
17-16	Current mode of CPU domain 4 00 - CPU is in RUN mode

Table continues on the next page...

Table continued from the previous page...

Field	Description
CPU_MODE_DOMAIN4	01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
15 —	Reserved
14 TRANS_REQ_DOMAIN3	
13-12 CPU_MODE_DOMAIN3	Current mode of CPU domain 3 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
11 —	Reserved
10 TRANS_REQ_DOMAIN2	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.
9-8 CPU_MODE_DOMAIN2	Current mode of CPU domain 2 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
7 —	Reserved
6 TRANS_REQ_DOMAIN1	
5-4 CPU_MODE_DOMAIN1	Current mode of CPU domain 1 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode

Table continues on the next page...

Table continued from the previous page...

Field	Description
	11 - CPU is in SUSPEND mode
3 —	Reserved
2 TRANS_REQ_DOMAIN0	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.
1-0 CPU_MODE_DOMAIN0	Current mode of CPU domain 0 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode

30.5.1.26 Low power mode information transfer status (LPCG0_LPM_STATUS1 - LPCG126_LPM_STATUS1)

This register denotes the LPCG low power mode information transfer status for DOMAIN8 ~ DOMAIM15. The information comprises the current low power mode and transfer status.

Offset

For a = 0 to 126:

Register	Offset
LPCGa_LPM_STATUS1	8008h + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	TRANS_R...	CPU_MODE_DOMAIN...		Reserved	TRANS_R...	CPU_MODE_DOMAIN...		Reserved	TRANS_R...	CPU_MODE_DOMAIN...		Reserved	TRANS_R...	CPU_MODE_DOMAIN...	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	TRANS_R...	CPU_MODE_DOMAIN...		Reserved	TRANS_R...	CPU_MODE_DOMAIN...		Reserved	TRANS_R...	CPU_MODE_DOMAIN9		Reserved	TRANS_R...	CPU_MODE_DOMAIN8	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30 TRANS_REQ_DOMAIN15	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.
29-28 CPU_MODE_DOMAIN15	Current mode of CPU domain 15 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
27 —	Reserved
26 TRANS_REQ_DOMAIN14	
25-24 CPU_MODE_DOMAIN14	Current mode of CPU domain 14 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
23 —	Reserved
22 TRANS_REQ_DOMAIN13	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.
21-20 CPU_MODE_DOMAIN13	Current mode of CPU domain 13 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
19 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
18 TRANS_REQ_DOMAIN12	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.
17-16 CPU_MODE_DOMAIN12	Current mode of CPU domain 12 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
15 —	Reserved
14 TRANS_REQ_DOMAIN11	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.
13-12 CPU_MODE_DOMAIN11	Current mode of CPU domain 11 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
11 —	Reserved
10 TRANS_REQ_DOMAIN10	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.
9-8 CPU_MODE_DOMAIN10	Current mode of CPU domain 10 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
7 —	Reserved
6	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.

Table continues on the next page...

Table continued from the previous page...

Field	Description
TRANS_REQ_DOMAIN9	
5-4 CPU_MODE_DOMAIN9	Current mode of CPU domain 9 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode
3 —	Reserved
2 TRANS_REQ_DOMAIN8	GPC request CCM to enable or disable LPCG. This signal will turn to low once CCM complete response to GPC.
1-0 CPU_MODE_DOMAIN8	Current mode of CPU domain 8 00 - CPU is in RUN mode 01 - CPU is in WAIT mode 10 - CPU is in STOP mode 11 - CPU is in SUSPEND mode

30.5.1.27 LPCG low power mode setting (LPCG0_LPM0 - LPCG126_LPM0)

LPCG low power mode setting for DOMAIN0 ~ DOMAIM7

Offset

For a = 0 to 126:

Register	Offset
LPCGa_LPM0	8010h + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	LPM_SETTING_D7			Reserved	LPM_SETTING_D6			Reserved	LPM_SETTING_D5			Reserved	LPM_SETTING_D4		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	LPM_SETTING_D3			Reserved	LPM_SETTING_D2			Reserved	LPM_SETTING_D1			Reserved	LPM_SETTING_D0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30-28 LPM_SETTING_D7	LPCG low power mode setting in DOMAIN7. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
27 —	Reserved
26-24 LPM_SETTING_D6	LPCG low power mode setting in DOMAIN6. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
23 —	Reserved
22-20 LPM_SETTING_D5	LPCG low power mode setting in DOMAIN5. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
19 —	Reserved
18-16 LPM_SETTING_D4	LPCG low power mode setting in DOMAIN4. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
15 —	Reserved
14-12 LPM_SETTING_D3	LPCG low power mode setting in DOMAIN3. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
11 —	Reserved
10-8 LPM_SETTING_D2	LPCG LPM in DOMAIN2 LPCG low power mode setting in DOMAIN2. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
7 —	Reserved
6-4	LPCG LPM in DOMAIN1 LPCG low power mode setting in DOMAIN1.

Table continues on the next page...

Table continued from the previous page...

Field	Description
LPM_SETTING_D1	000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
3 —	Reserved
2-0 LPM_SETTING_D0	LPCG LPM in DOMAIN0 LPCG low power mode setting in DOMAIN0. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.

30.5.1.28 LPCG low power mode setting (LPCG0_LPM1 - LPCG126_LPM1)

LPCG low power mode setting for DOMAIN8 ~ DOMAIM15.

Offset

For a = 0 to 126:

Register	Offset
LPCGa_LPM1	8014h + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	LPM_SETTING_D15			Reserved	LPM_SETTING_D14			Reserved	LPM_SETTING_D13			Reserved	LPM_SETTING_D12		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	LPM_SETTING_D11			Reserved	LPM_SETTING_D10			Reserved	LPM_SETTING_D9			Reserved	LPM_SETTING_D8		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30-28 LPM_SETTING_D15	LPCG low power mode setting in DOMAIN15. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
27 —	Reserved
26-24 LPM_SETTING_D14	LPCG low power mode setting in DOMAIN14. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
23 —	Reserved
22-20 LPM_SETTING_D13	LPCG low power mode setting in DOMAIN13. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
19 —	Reserved
18-16 LPM_SETTING_D12	LPCG low power mode setting in DOMAIN12. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15 —	Reserved
14-12 LPM_SETTING_D11	LPCG low power mode setting in DOMAIN11. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
11 —	Reserved
10-8 LPM_SETTING_D10	LPCG low power mode setting in DOMAIN10. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
7 —	Reserved
6-4 LPM_SETTING_D9	LPCG low power mode setting in DOMAIN9. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.
3 —	Reserved
2-0 LPM_SETTING_D8	LPCG low power mode setting in DOMAIN8. 000 - LPCG will be OFF in any CPU mode. 001 - LPCG will be ON in RUN mode, OFF in WAIT/STOP/SUSPEND mode. 010 - LPCG will be ON in RUN/WAIT mode, OFF in STOP/SUSPEND mode. 011 - LPCG will be ON in RUN/WAIT/STOP mode, OFF in SUSPEND mode. 100 - LPCG will be ON in RUN/WAIT/STOP/SUSPEND mode.

30.5.1.29 LPM setting of current CPU domain (LPCG0_LPM_CUR - LPCG126_LPM_CUR)

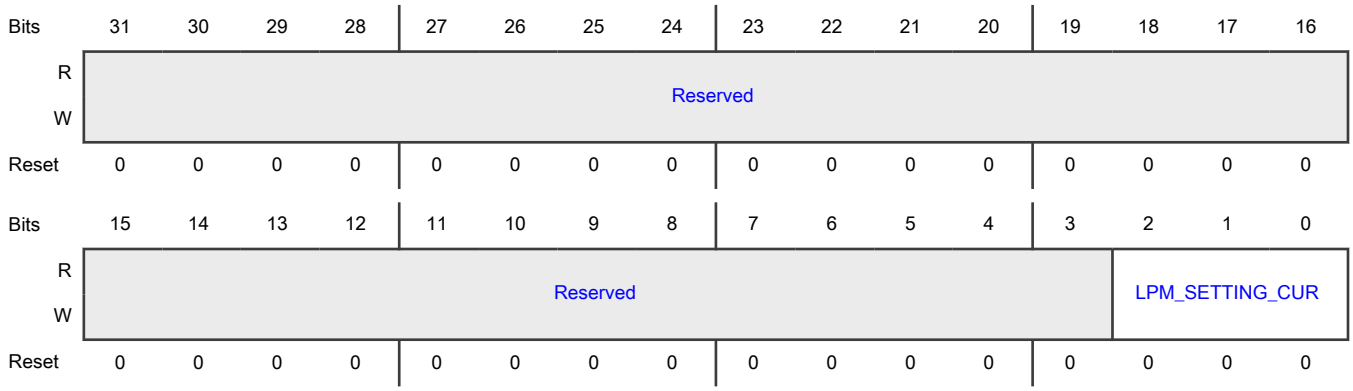
Reading this field will returen LPCGn_LPM[LPM_SETTING_Dx], and x represent the CPU domain which launched the read access.

Offset

For a = 0 to 126:

Register	Offset
LPCGa_LPM_CUR	801Ch + (a × 40h)

Diagram



Fields

Field	Description
31-3 —	Reserved
2-0 LPM_SETTING_CUR	LPM SETTING of current CPU DOMAIN **NOTE** This field only can be changed by the cpu domain which launched the write access.

30.5.1.30 LPCG working status (LPCG0_STATUS0 - LPCG126_STATUS0)

This register shows current LPCG running status.

Offset

For a = 0 to 126:

Register	Offset
LPCGa_STATUS0	8020h + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															ON
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31-1 —	Reserved
0 ON	LPCG work status 0 - LPCG is OFF. 1 - LPCG is ON.

30.5.1.31 LPCG domain status (LPCG0_STATUS1 - LPCG126_STATUS1)

This register shows domains status of LPCG.

Offset

For a = 0 to 126:

Register	Offset
LPCGa_STATUS1	8024h + (a × 40h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DOMAIN_ENABLE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DOMAIN_ACTIVE															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-16 DOMAIN_ENABLE	Domain enable Domain x is enabled means that LPCGn_LPM_STATUS[CPU_MODE_DOMAINx] is larger than LPCGn_LPM[LPM_SETTING_Dx], this LPCG should be ON for DOMAINx. BIT16~BIT31 are corresponding with DOMAIN0~DOMAIN15.
15-0 DOMAIN_ACTIVE	Domain active This field is equal to LPCGn_AUTHEN[WHITE_LIST].

30.5.1.32 LPCG access control (LPCG0_AUTHEN - LPCG126_AUTHEN)

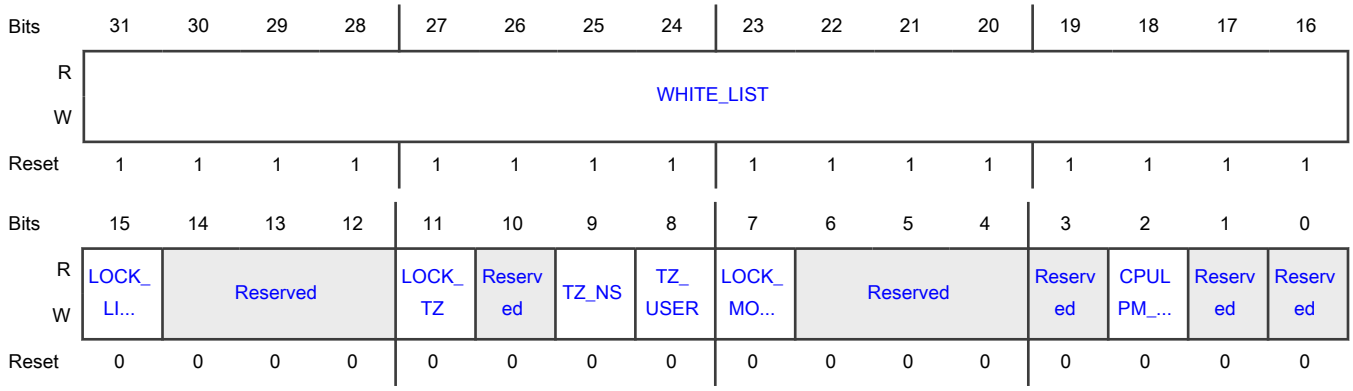
This register controls the access to this LPCG slice. The access control includes 2 parts: TrustZone control and domain based granting.

Offset

For a = 0 to 126:

Register	Offset
LPCGa_AUTHEN	8030h + (a × 40h)

Diagram



Fields

Field	Description
31-16 WHITE_LIST	Whitelist Each master(Core or non-Core) can be assigned to any domains among 0-15 through TRDC and through GPC domain assignment register. Once its domains are assigned in TRDC, then its accesses will have its unique domain ID. Each bit in this field represent for one domain. Bit 16-31 represent domains 0-15 respectively. Only the TRDC domains whose corresponding bits are 1 can write to this register. Write violation will fail without any exception or information. Read violation can still get read

Table continues on the next page...

Table continued from the previous page...

Field	Description
	data. In LPM mode, only the GPC domains whose corresponding bits are 1 can trigger the enabling/ disabling down of the LPCG slice.
15 LOCK_LIST	Lock white list This bit lock white list. When this bit is set, LPCGn_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - Whitelist is not locked. 1 - Whitelist is locked.
14-12 —	Reserved
11 LOCK_TZ	Lock TrustZone settings This bit locks TrustZone settings. When this bit is set, LPCGn_AUTHEN[TZ_USER] and LPCGn_AUTHEN[TZ_NS] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - TrustZone settings is not locked. 1 - TrustZone settings is locked.
10 —	Reserved
9 TZ_NS	Non-secure access permission Whether the registers of this LPCG can be changed when CPU is in non-secure mode. The default value is 0, that means only the access from CPU in secure mode can change the registers of this LPCG. 0 - Cannot be changed in Non-secure mode. 1 - Can be changed in Non-secure mode.
8 TZ_USER	User access permission Whether the registers of this LPCG can be changed when CPU is in user mode. The default value is 0, that means only the access from CPU in supervisor mode can change the registers of this LPCG. 0 - LPCG settings cannot be changed in user mode. 1 - LPCG settings can be changed in user mode.
7 LOCK_MODE	When this bit is set, CPULPM_MODE cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - CPULPM_MODE is not locked. 1 - CPULPM_MODE is locked.
6-4 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
3 —	Reserved
2 CPULPM_MODE	CPULPM mode enable This field enables the CPULPM mode. 0 - Disable CPULPM mode, this LPCG is in Direct Control mode. 1 - Enable CPULPM mode, this LPCG is in CPULPM mode.
1 —	Reserved
0 —	Reserved

Chapter 31
General Power Controller (GPC)

31.1 Overview

The General Power Controller(GPC) is the centralized power controller, which controls the power mode of the processor(s). The GPC takes the Wait For Interrupt (WFI) signal from CPU platforms, and wakeup events from peripherals, to determine the power mode based on the GPC power management policy. The GPC can also control the power mode transition.

31.1.1 Block diagram

The GPC block consists of the following sub-modules:

- CPU Mode Control (CMC): It contains CPU mode controller for the CPU platform. It controls the CPU mode of the CPU platforms and their private resources.
  - CMC for the Cortex-M33 core platform.
  - CMC for the Cortex-A55 core0, referred to as the core0 platform.
  - CMC for the Cortex-A55 core1, referred to as the core1 platform.
  - CMC for the Cortex-A55 cluster (SCU/L3), referred to as the CA55 platform.
- UPI CM Mapping (UPI): The connection between CMC and CPU platform is directly hard-wired, but CPU platform can be assigned to any domain in the system. This block is used to map CPU mode control signals into the correct domain.
- GLOBAL CTRL: It is used to control special resources.

The figure below shows the block diagram of the General Power Controller.

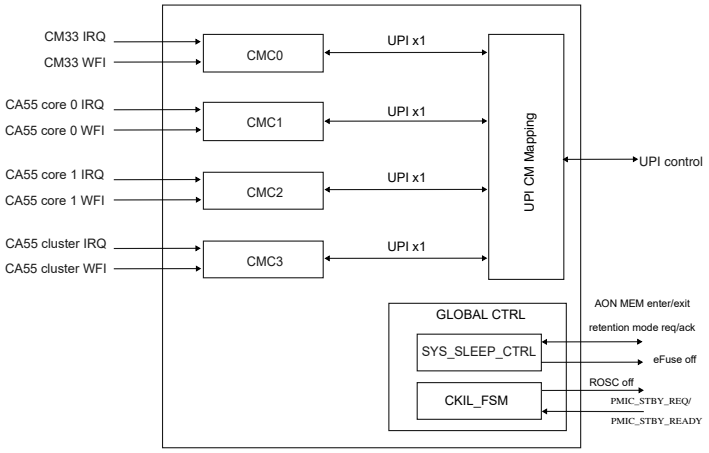


Figure 65. GPC block diagram

31.1.2 Features

The GPC includes the following features:

- Independent CPU modes for each CPU platform
- System sleep management
- Domain access control
- Controllable steps of sleep and wakeup sequence

31.2 Functional description

The following sections describe functional details of the GPC module.

31.2.1 CPU Mode Control (CMC)

The CPU mode is a power mode of the CPU platform. The following are the four CPU modes:

Table 146. CPU Modes

CPU Mode	Description
RUN	The CPU core is active and running under normal operation. All the blocks inside the CPU platform can be accessed when needed. The state of private resources are fully controlled by software configuration.
WAIT	The CPU is in WFI state, but can to get back to RUN mode with very short latency. In typical applications, the CPU will enter WAIT mode whenever there is no active thread running. In WAIT mode, the clock to the CPU core is gated off, the cache is clock gated, and the TCM is still active since there are other modules, such as DMA, that still needs access to it.
STOP	The CPU is in WFI state, and does not require an extremely short exit time. In STOP mode, the clocks to the CPU core, Cache, and TCM are all gated off. The clocks to the bus and peripherals are also gated off. This is the lowest power consumption mode without losing the state of the peripheral. When exiting from STOP mode, there is no need to re-initialize the peripherals.
SUSPEND	Entering SUSPEND mode is for lowest power consumption, and exit time is not critical. In SUSPEND mode, CPU, Cache, and peripherals are all power gated. The biggest difference between SUSPEND and STOP mode is the peripherals will be power gated. Because the peripherals are power gated, entering SUSPEND mode requires the CPU to save the state of the peripherals. When exiting from SUSPEND mode, the CPU needs to restore these states.

CPU Mode transitions happen when the following events occur:

- Sleep Event: CPU enters sleep state with WFI instruction
- Wakeup Event: Any unmasked IRQ wakeup

On a Sleep Event, the CPU has the option to enter WAIT/STOP/SUSPEND mode, or stay in RUN mode. The transition to WAIT/STOP/SUSPEND is called the sleep sequence, and the next state is configured by CM_MODE_CTRL[CPU_MODE_TARGET].

When CPU is in WAIT/STOP/SUSPEND mode, a wakeup request can wake the CPU, and bring it back to RUN mode. This is transition is called the wakeup sequence. The CM_IRQ_WAKEUP_MASK_n register selects which IRQs can wakeup the CPU platform. There are also non-IRQ requests, which can wakeup the CPU. These non-IRQ requests are configured by the CM_NON_IRQ_WAKEUP_MASK register. There is another wakeup event called non-masked interrupt (nmi_in), which can wake the CPU platforms unconditionally.

For the Cortex-A55, there is a special wakeup event called gic_wakeup_req. CM_MISC[IRQ_MUX] can be configured to select if the wakeup is from the IRQs (default selection) or the gic_wakeup_req event.

During a CPU mode transition, the GPC communicates with the system resource controllers by sending the target CPU mode and transition step. Please see the UPI interface for more detials.

If a resource is set as private to a single CPU platform, the resource's power state will change with the CPU platform's mode. If a resource is set as shared by multiple CPU platforms, the resource controller needs to look at all the current modes of the CPU platforms to determine the power state of the resource. For a resource set as shared by multiple CPU platforms, the following rules apply:

- The resource can be power gated, only when all the CPU platforms want to power down the resource.

- The resource can be powered on, when any CPU platform wants to power up the resource.

31.2.2 System Sleep Mode

When a CMC has entered SUSPEND mode, it requests the system to enter system sleep mode.
The mode can be changed by configuring CM_SYS_SLEEP_CTRL[SS_SUSPEND, SS_STOP, SS_WAIT].
The default values are CM_SYS_SLEEP_CTRL[SS_SUSPEND]=1, CM_SYS_SLEEP_CTRL[SS_STOP]=0, and CM_SYS_SLEEP_CTRL[SS_WAIT]=0. If CM_SYS_SLEEP_CTRL[SS_STOP] value is set to 1, when a CMC has entered into STOP mode, the CMC requests the system to enter system sleep mode.

The system is put into system sleep mode only when all the CMCs request the system to enter system sleep mode. If a CPU is disabled by FUSE, the GPC ignores it's CMC system sleep mode requirement. A CPU can also be forced into a system sleep status by configuring GPC_SYS_SLEEP[FORECE_CPU3_DISABLE, FORECE_CPU2_DISABLE, FORECE_CPU1_DISABLE, FORECE_CPU0_DISABLE]. When configured, the GPC will ignore the CMC system sleep mode requirement for the disabled CPU(s).

For details of each system sleep transition step, refer to Table 148.

System sleep mode has the following four functions.

- AON memories (AONMIX memory, HSIO memory, DDR PHY memory) can enter/exit retention mode with configuration of the GPC_AON_MEM_CTRL register.
- eFUSE can be power gated with configuration of the GPC_EFUSE_CTRL register.
- ROSC 24M clock can be turned off with configuration of the GPC_ROSC_CTRL register. Then the GPC works under the 32KHz clock.
- PMIC can be put into standby status with configuration of the PMIC_CTRL, PMIC_PRE_DLY_CTRL, and PMIC_STBY_ACK_CTRL registers.

31.2.3 Unified Power Management Interface (UPI)

GPC does not control the state of the resource directly. During power mode transition, it communicates with the resource controllers through the UPI (Unified Power management Interface) to accomplish the power mode transition.

GPC and resource controllers implement a handshaking mechanism for mode transitions.

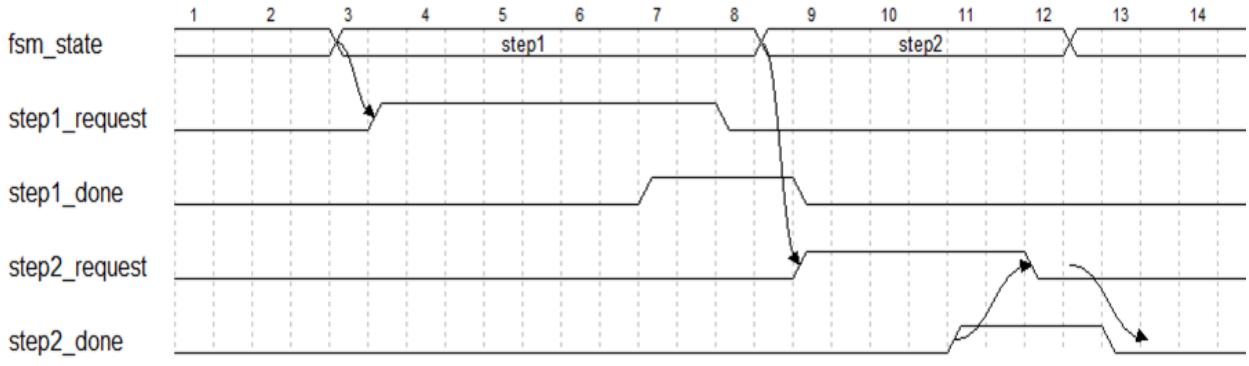


Figure 66. Handshake method

31.2.4 Low Power Sequence

31.2.4.1 CPU mode transition flow

There are four CPU mode control instances:CMC0 for CM33,CMC1 for CA55 Core 0,CMC2 for CA55 Core 1, and CMC3 for CA55 cluster. Each CMC has a state machine to control the sequence of CPU mode transition.

When CMC receives a sleep request from CPU platform, it starts the sleep sequence. The flow cannot be interrupted by a wakeup request until the SLEEP_RSV2 request completes. After the sleep sequence finishes, the CMC stays at IDLE_SLEEP and waits for a wakeup event to start wakeup sequence, which is shown in the left part of the diagram.

Figure 67. CPU mode transition flow

NOTE

SLEEP_RSV2 and WAKEUP_RSV2 are reserved status and don't have a real action.

Table 147. CPU mode transition steps

State	Description
IDLE_RUN	CPU is in RUN mode
SLEEP_A55_HDSK	GPC asserts command to SRC to handshake with CA55
SLEEP_SSAR	GPC asserts command to SRC to send SSAR system save
SLEEP_LPCG	GPC asserts command to CCM to disable LPCGs
SLEEP_PLL	GPC asserts command to CCM to turn PLLs off
SLEEP_ISO	GPC asserts command to SRC to enable isolation signals
SLEEP_RESET	GPC asserts command to SRC to assert reset signals

Table continues on the next page...

Table 147. CPU mode transition steps (continued)

State	Description
SLEEP_POWER	GPC asserts command to SRC to turn off power switch signals
SLEEP_SYS	SSC runs the system sleep transition flow
IDLE_SLEEP	CMC waits for wakeup event to start wakeup sequence
WAKEUP_SYS	SSC runs the wakeup flow
WAKEUP_POWER	GPC asserts command to SRC to de-assert reset signals
WAKEUP_RESET	GPC asserts command to SRC to turn on power switch signals
WAKEUP_ISO	GPC asserts command to SRC to disable isolation signals
WAKEUP_PLL	GPC asserts command to CCM to turn PLLs on
WAKEUP_LPCG	GPC asserts command to CCM to enable LPCGs
WAKEUP_SSAR	GPC asserts command to SRC to send SSAR restore
WAKEUP_A55_HDSK	GPC asserts command to SRC to handshake with CA55

For each step of transition above, there is a control register with the following fields:

- STEP_CNT: This field configures the delay count number.
- CNT_MODE: This field configures the count mode. There are 4 selectable modes available for the delay counter.
- DISABLE: This field disables the step, so that no request is sent out in this transition step.

There is also a read only counter to record the time when the step response is received.

31.2.4.2 System sleep transition flow

There are four transition steps for the system sleep mode. Each step has its own enable register. The system sleep in sequence can be aborted after any step completes because of the wakeup requirement.

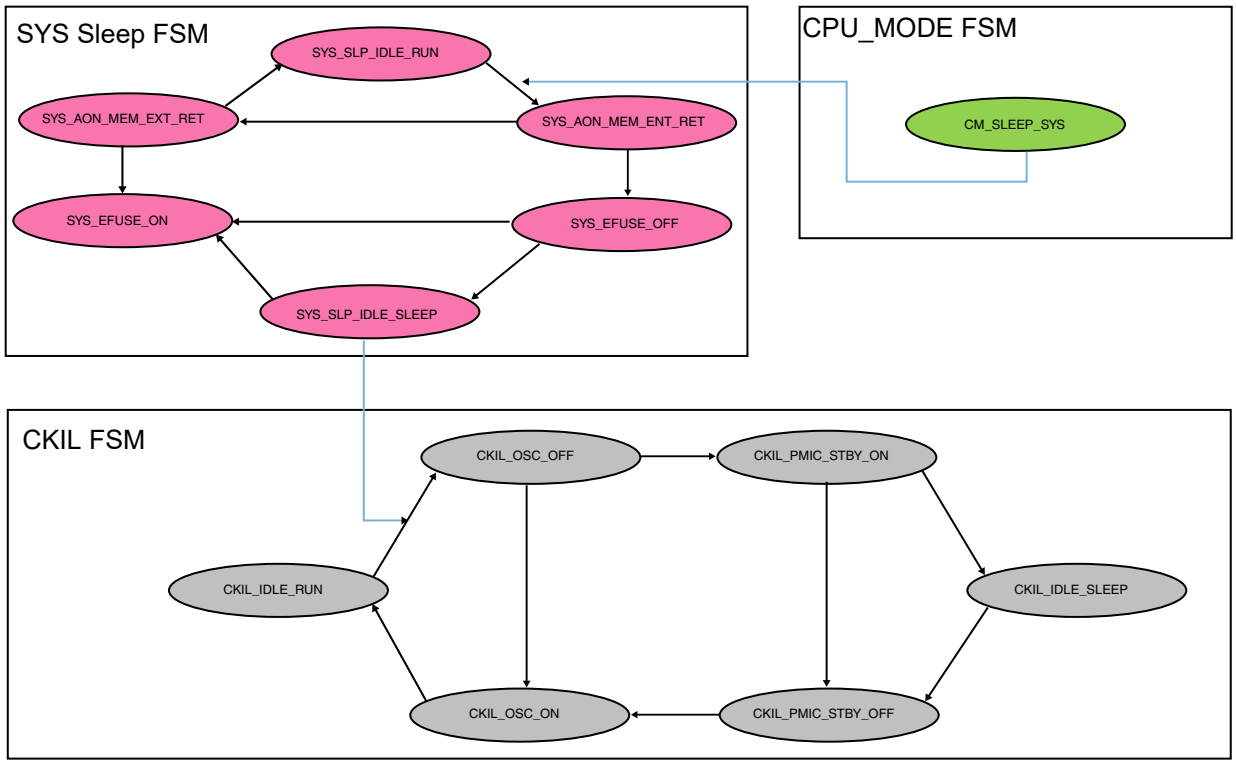


Figure 68. System sleep transition flow

Table 148. System sleep transition steps

State	Description
SYS_SLP_IDLE_RUN	Idle run
SYS_AON_MEM_ENT_RET	AON memories enter retention mode
SYS_EFUSE_OFF	Power off EFUSE
SYS_SLP_IDLE_SLEEP	Idle run
CKIL_OSC_OFF	ROSC 24M clock off
CKIL_PMIC_STBY_ON	Standby on PMIC
CKIL_IDLE_SLEEP	Idle run
CKIL_PMIC_STBY_OFF	Standby off PMIC
CKIL_OSC_ON	ROSC 24M clock on
CKIL_IDLE_RUN	Idle run
SYS_EFUSE_ON	Power on EFUSE
SYS_AON_MEM_EXT_RET	AON memories exit retention mode

31.2.5 Priority of Sleep and Wakeup

IRQs are sent to the GPC, CM33 and GIC.

When CM33 WFI and IRQs occur at the same time, the GPC will not enter the sleep sequence. IRQs can wake up the CM33 from WFI directly.

IRQs are not sent to CA55 directly, but through the GIC. Before the CA55 enters WFI, the GIC distributor register CORE_PWRDN_EN bit is set. Then executing a WFI instruction automatically masks all interrupts and wake_up events in the core. As a result, applying reset is the only way to wake up the core from this WFI. Therefore, when the CA55 CORE0/1 WFI and IRQs happen at the same time, the GPC must enter the sleep sequence.

Wakeup has a higher priority than sleep in CMC0 and CMC3. Sleep has a higher priority than wakeup in CMC1 and CMC2.

31.2.6 System Domain

31.2.6.1 Domain assignment

Each CPU platform and the CMC connected to it can be assigned to the same domain. CPU_DOMAIN[CPUn_DOMAIN] is used to select the domain assignments for CMCn.

Based on the domain assignment, a mapping mechanism between CPU platform and system domain is implemented inside GPC.

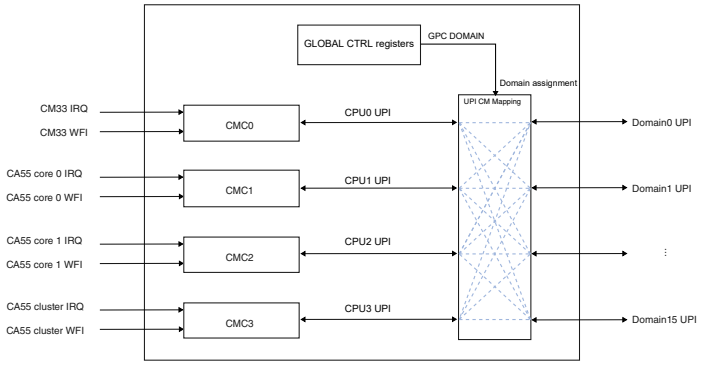


Figure 69. Domain Assignment

When more than one CMCs is assigned to same domain, a two stage priority selects which CMC can control the domain's UPI output. The first stage is CPUn_MASTER flag from GLOBAL which is enabled by register GPC_MASTER[CPUn_MASTER]. Only one CMC with CPUn_MASTER flag owns the domain UPI output. The second stage is for the case when more than one CMC has CPUn_MASTER flag or no CMC has this flag. In this case, the lower indexed CMC has higher priority.

31.2.6.2 Register Access Permissions

The GPC register access is controlled by the following registers and bitfields:

- CM_AUTHEN_CTRL register for CMC access permission
- AUTHEN_CTRL register for GLOBAL registers access permission
- x_AUTHEN_CTRL[WHITE_LIST] field configures which domain IDs can write to the registers
- x_AUTHEN_CTRL[USER] and x_AUTHEN_CTRL[NONSECURE] fields are used to set the privilege and security permission
- x_AUTHEN_CTRL[LOCK_CFG] locks configuration fields and should only be set at initialization
- x_AUTHEN_CTRL[LOCK_LIST] locks WHITE_LIST fields and should only be set at initialization

31.2.7 Resource Low Power Control

31.2.7.1 Hardware Control

The resource behavior under a CPU power mode is configurable. The following is an example of configuring a LPCGn resource under CPU power mode.

1. Configure the CCM register LPCGn_AUTHEN[CPULPM_MODE] to 1.
2. Configure the CCM register LPCGn_LPMn.
3. Configure the CCM register LPCGn_AUTHEN[WHITE_LIST].
4. Configure the GPC CMC register CM_MODE_CTRL[CPU_MODE_TRAGET]. Configure the GPC global register GPC_DOMAIN to assign the CPU control domain, and then execute the "WFI" instruction.
5. The GPC runs the Low Power Sequence up to the WAKEUP_LPCG/SLEEP_LPCG step, then sends the lpcg_request together with cpu_power_mode_n via the UPI.
6. The CCM will turn on or turn off the LPCGn clock based on the LPCGn_LPMn setting in step 2.

31.2.7.2 Resource Domain Assignment

Each resource can be assigned to any (one or multiple) of the 16 domains. This domain assignment is configured in the resource controller's register AUTHEN[WHITE_LIST]. When the resource is controlled by hardware, the WHITE_LIST field indicates which domain UPI request needs to be cared. Take LPCG-n clock gating as an example, which is illustrated below.

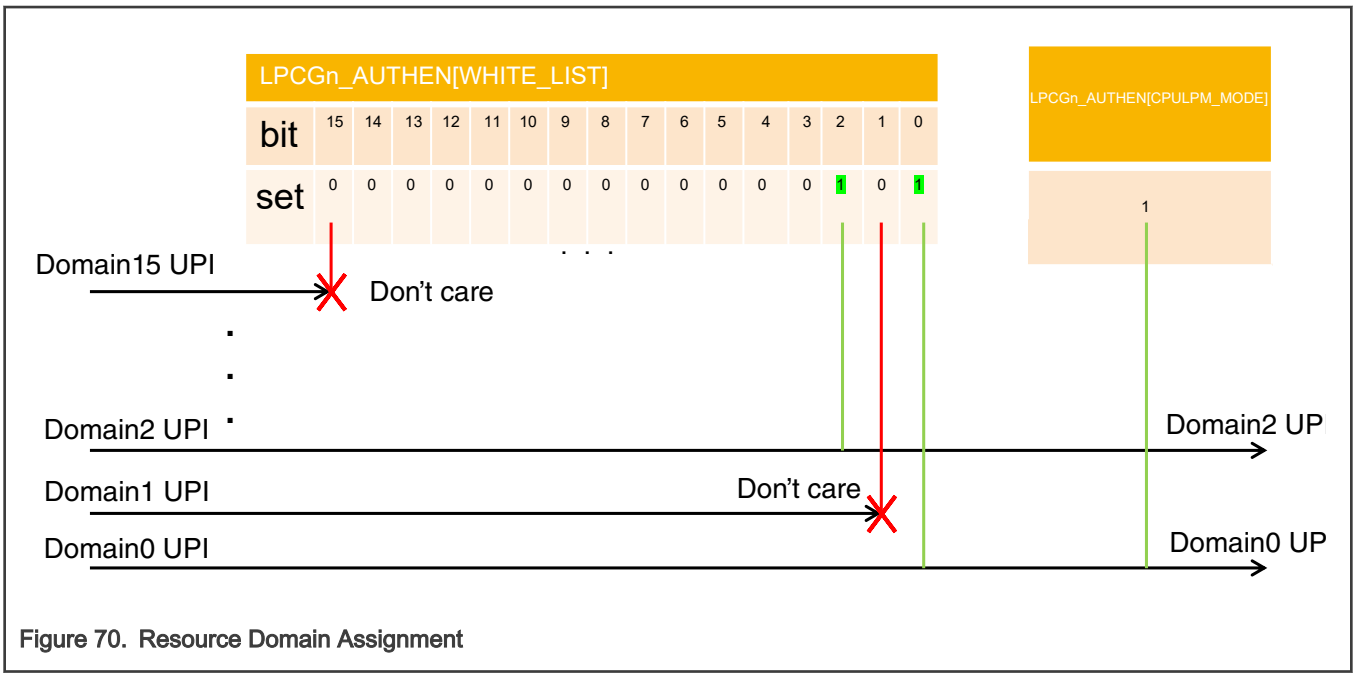


Figure 70. Resource Domain Assignment

The LPCGn_LPMn register provides 16 domain settings (LPM_SETTING_D0 - LPM_SETTING_D15), one for each of the 16 domains. The final resource shut off request is logic AND of all the shared domain requests.

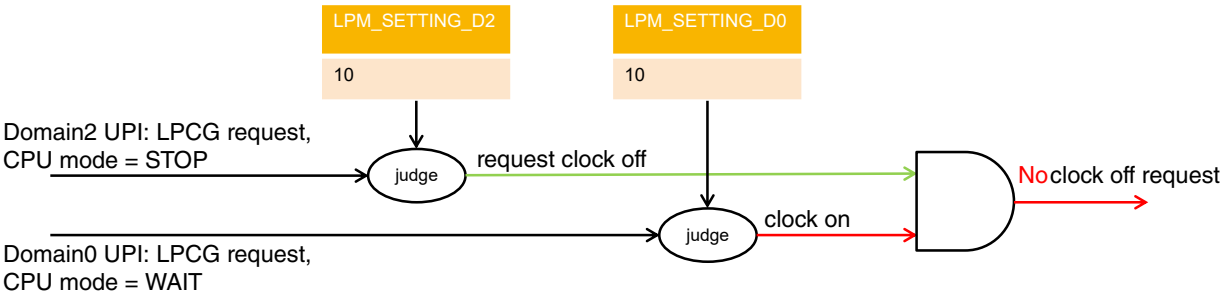


Figure 71. Shared Resource Hardware Control

31.2.7.3 Software Control

The resource behavior can be directly controlled by software. In this mode, the CCM register LPCGn_AUTHEN[CPULPM_MODE] must be set to 0.

31.2.7.4 Switch Between Hardware Control and Software Control

When switching from software control to hardware control, software needs to ensure that the resource is ON. LPCGn_LPMn should be configured to a common value, not always on or always off.

31.2.8 Clocks

This section describes clocks and special clocking requirements of the GPC module.

Table 149. Clocks

Clock	Description
ipg_clk	Main GPC clock. This clock originates from RCOSC 24 MHz and and can be turned off by setting GPC_ROSC_CTRL[ROSC_OFF_EN].
32k_clk	32 kHz clock used to wakeup GPC and RCOSC.

31.2.9 Interrupts

The interrupt inputs of each CMC are independent and the same as the CPU platforms they are connected to. Each CMC has up to 256 IRQ inputs to wake up the core or system.

31.3 External Signals

There are no external signals pinned out for the GPC.

31.4 Initialization

See Low Power Sequence for low power entry and exit flow and considerations.

31.4.1 CPU Sleep Hold (Cortex M-Core Only)

There is a certain delay between the time the core sleeps and when the core clock stops. If a wakeup IRQ appears inside this delay window, the core will process the IRQ and wake up, but the GPC sleep sequence is still shutting off the core clock. Since the sleep sequence is not interruptable, this scenario may lead to unpredictable behavior.

To avoid this scenario, a CPU Sleep Hold is implemented to hold the core in the sleep state until the sleep sequence is finished. The GPC will then send a signal to release the core. The CM_MISC[SLEEP_HOLD_EN] field is used to enable this function.

The core clock can also be gated by a WFI event without any GPC sequence. The GPC uses a signal to clear this gate when a wakeup trigger occurs.

31.5 Memory Map and register definition

This section includes the GPC module memory map and detailed descriptions of all registers.

NOTE

Write/Read to the Reserved locations will not generate a transfer error.

31.5.1 GPC_CPU_CTRL register descriptions

31.5.1.1 gpc_cpu_ctrl memory map

gpc_ctrl_ca55_0 base address: 4447_0800h

gpc_ctrl_ca55_1 base address: 4447_1000h

gpc_ctrl_ca55_cluster base address: 4447_1800h

gpc_ctrl_cm33 base address: 4447_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	CM Authentication Control (CM_AUTHEN_CTRL)	32	RW	FFFF_0000
C	Miscellaneous (CM_MISC)	32	RW	0000_000E
10	CPU mode control (CM_MODE_CTRL)	32	RW	0000_0000
14	CM CPU mode Status (CM_MODE_STAT)	32	R	0000_0000
18	CM pin Status (CM_PIN_STAT)	32	R	0000_0000
100	CM IRQ0~31 wakeup mask (CM_IRQ_WAKEUP_MASK_0)	32	RW	0000_0000
104	CM IRQ32~63 wakeup mask (CM_IRQ_WAKEUP_MASK_1)	32	RW	0000_0000
108	CM IRQ64~95 wakeup mask (CM_IRQ_WAKEUP_MASK_2)	32	RW	0000_0000
10C	CM IRQ96~127 wakeup mask (CM_IRQ_WAKEUP_MASK_3)	32	RW	0000_0000
110	CM IRQ128~159 wakeup mask (CM_IRQ_WAKEUP_MASK_4)	32	RW	0000_0000
114	CM IRQ160~191 wakeup mask (CM_IRQ_WAKEUP_MASK_5)	32	RW	0000_0000
118	CM IRQ192~223 wakeup mask (CM_IRQ_WAKEUP_MASK_6)	32	RW	0000_0000
11C	CM IRQ224~255 wakeup mask (CM_IRQ_WAKEUP_MASK_7)	32	RW	0000_0000
140	CM non-IRQ wakeup mask (CM_NON_IRQ_WAKEUP_MASK)	32	RW	0000_0001
150	CM IRQ0~31 wakeup status (CM_IRQ_WAKEUP_STAT_0)	32	R	0000_0000
154	CM IRQ32~63 wakeup status (CM_IRQ_WAKEUP_STAT_1)	32	R	0000_0000
158	CM IRQ64~95 wakeup status (CM_IRQ_WAKEUP_STAT_2)	32	R	0000_0000
15C	CM IRQ96~127 wakeup status (CM_IRQ_WAKEUP_STAT_3)	32	R	0000_0000
160	CM IRQ128~159 wakeup status (CM_IRQ_WAKEUP_STAT_4)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
164	CM IRQ160~191 wakeup status (CM_IRQ_WAKEUP_STAT_5)	32	R	0000_0000
168	CM IRQ192~223 wakeup status (CM_IRQ_WAKEUP_STAT_6)	32	R	0000_0000
16C	CM IRQ224~255 wakeup status (CM_IRQ_WAKEUP_STAT_7)	32	R	0000_0000
190	CM non-IRQ wakeup status (CM_NON_IRQ_WAKEUP_STAT)	32	R	0000_0000
200	CM sleep A55_HDSK control (CM_SLEEP_A55_HDSK_CTRL)	32	RW	0000_0004
208	CM sleep SSAR control (CM_SLEEP_SSAR_CTRL)	32	RW	0000_0004
210	CM sleep LPCG control (CM_SLEEP_LPCG_CTRL)	32	RW	0000_0004
218	CM sleep PLL control (CM_SLEEP_PLL_CTRL)	32	RW	0000_0004
220	CM sleep isolation control (CM_SLEEP_ISO_CTRL)	32	RW	0000_0004
228	CM sleep memory control (CM_SLEEP_MEM_CTRL)	32	RW	0000_0004
230	CM sleep reset control (CM_SLEEP_RESET_CTRL)	32	RW	0000_0004
238	CM sleep power control (CM_SLEEP_POWER_CTRL)	32	RW	0000_0004
240	CM sleep rsv2 control (CM_SLEEP_RSV2_CTRL)	32	RW	0000_0004
288	CM wakeup rsv2 control (CM_WAKEUP_RSV2_CTRL)	32	RW	0000_0004
290	CM wakeup power control (CM_WAKEUP_POWER_CTRL)	32	RW	0000_0004
298	CM wakeup memory control (CM_WAKEUP_MEM_CTRL)	32	RW	0000_0004
2A0	CM wakeup reset control (CM_WAKEUP_RESET_CTRL)	32	RW	0000_0004
2A8	CM wakeup isolation control (CM_WAKEUP_ISO_CTRL)	32	RW	0000_0004
2B0	CM wakeup PLL control (CM_WAKEUP_PLL_CTRL)	32	RW	0000_0004
2B8	CM wakeup LPCG control (CM_WAKEUP_LPCG_CTRL)	32	RW	0000_0004
2C0	CM wakeup MTR control (CM_WAKEUP_MTR_CTRL)	32	RW	0000_0004
2C8	CM wakeup SSAR control (CM_WAKEUP_SSAR_CTRL)	32	RW	0000_0004
2D0	CM wakeup A55_HDSK control (CM_WAKEUP_A55_HDSK_CTRL)	32	RW	0000_0004
380	CM system sleep control (CM_SYS_SLEEP_CTRL)	32	RW	0000_0004
390	CM debug (CM_DEBUG)	32	RW	0000_0000

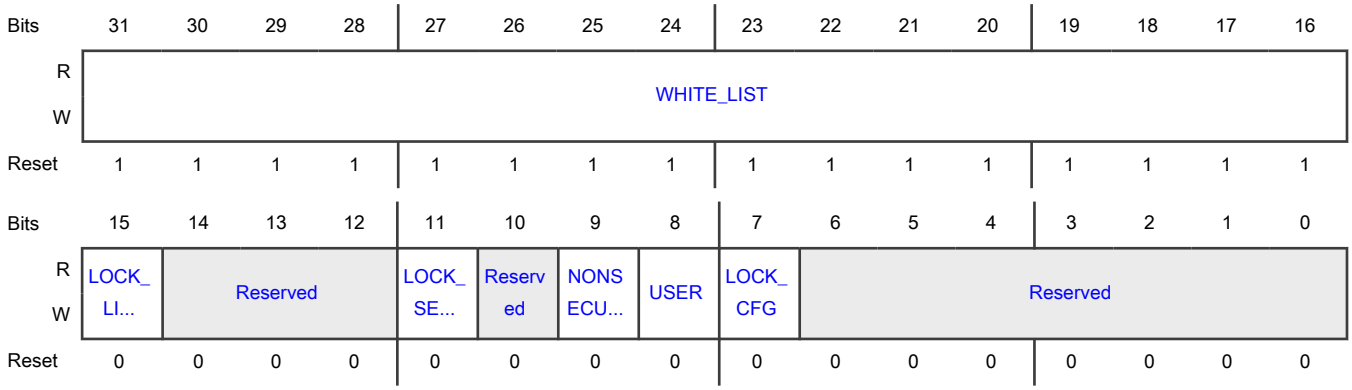
31.5.1.2 CM Authentication Control (CM_AUTHEN_CTRL)

Authentication control for CMC.

Offset

Register	Offset
CM_AUTHEN_CTRL	4h

Diagram



Fields

Field	Description
31-16 WHITE_LIST	Domain ID white list When bit is set to 1, the corresponding domain ID can access CPU mode control registers. bit[0] for Domain 0, bit[1] for Domain 1, bit[2] for Domain 2, and bit[3] for Domain 3. Multiple bits with a value of 1 is permitted.
15 LOCK_LIST	White list lock This bit locks the WHITE_LIST. When this bit is set, WHITE_LIST cannot be changed. Once this bit is set, it cannot be cleared until the next system reset. 0 - WHITE_LIST is not locked 1 - WHITE_LIST is locked
14-12 —	Reserved
11 LOCK_SETTING	Lock NONSECURE and USER This bit locks the NONSECURE and USER fields. When this bit is set, LOCK_SETTING cannot be changed. Once this bit is set, it cannot be cleared until the next system reset. 0 - NONSECURE and USER fields are not locked 1 - NONSECURE and USER fields are locked
10 —	Reserved
9 NONSECURE	Allow non-secure mode access 0 - Allow only secure mode to access CPU mode control 1 - Allow both secure and non-secure mode to access CPU mode control registers
8 USER	Allow user mode access 0 - Allow only privilege mode to access CPU mode control registers

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Allow both privilege and user mode to access CPU mode control registers
7 LOCK_CFG	Configuration lock This bit locks the value of low power configuration fields. Once this bit is set, it cannot be cleared until the next system reset. 0 - The value of low power configuration fields are not locked. 1 - The value of low power configuration fields are locked. It locks the CPUx_CM registers which are marked as "Locked by LOCK_CFG field" in the function field.
6-0 —	Reserved

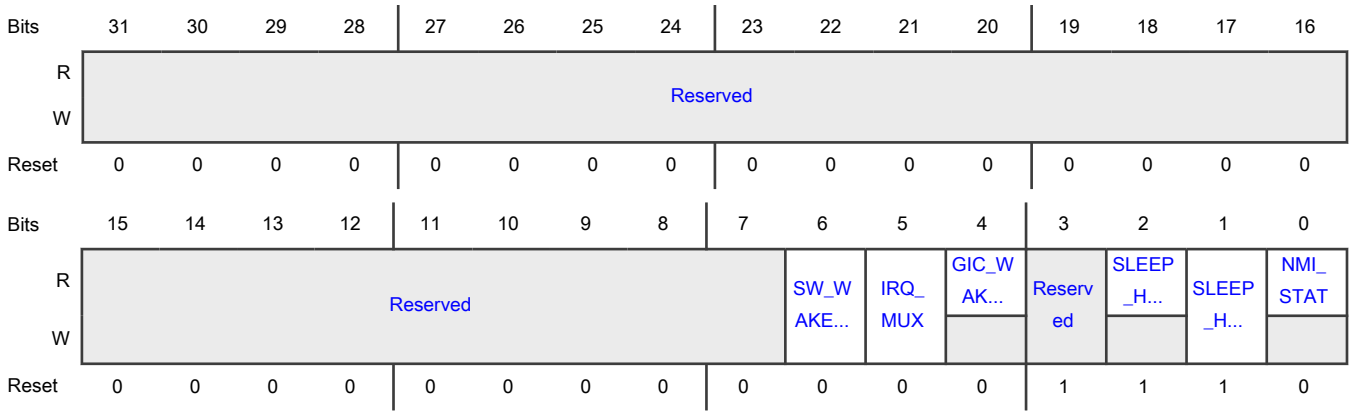
31.5.1.3 Miscellaneous (CM_MISC)

Authentication control for CMC.

Offset

Register	Offset
CM_MISC	Ch

Diagram



Fields

Field	Description
31-7 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
6 SW_WAKEUP	software wakeup. Used for CPU hotplug.
5 IRQ_MUX	IRQ select 0 - From raw IRQ 1 - From GIC
4 GIC_WAKEUP_STAT	GIC wakeup request status
3 —	Reserved
2 SLEEP_HOLD_STAT	Status of cpu_sleep_hold_ack_b 0 - CPU sleep hold is acknowledged 1 - CPU is not in sleep hold
1 SLEEP_HOLD_EN	Allow cpu_sleep_hold_req assert during CPU low power status 0 - Disable cpu_sleep_hold_req 1 - Allow cpu_sleep_hold_req to assert during CPU low power status
0 NMI_STAT	Non-masked interrupt status 0 - NMI is not asserted 1 - NMI is asserted

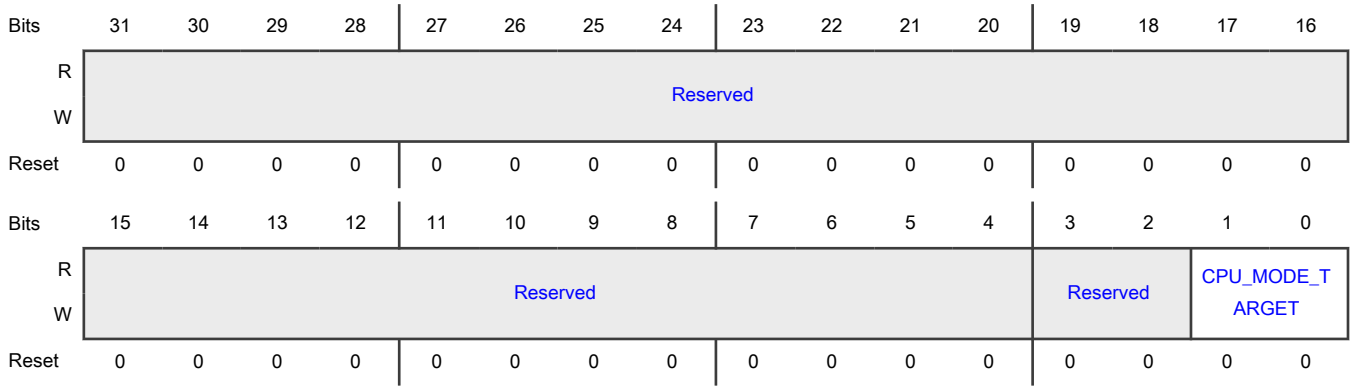
31.5.1.4 CPU mode control (CM_MODE_CTRL)

CPU mode control register.

Offset

Register	Offset
CM_MODE_CTRL	10h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-2 —	Reserved
1-0 CPU_MODE_TARGET	The CPU mode the CPU platform should transit to on next sleep event Important: Core MUST re-configure this field EACH TIME before entering sleep 00 - Stay in RUN mode 01 - Transit to WAIT mode 10 - Transit to STOP mode 11 - Transit to SUSPEND mode

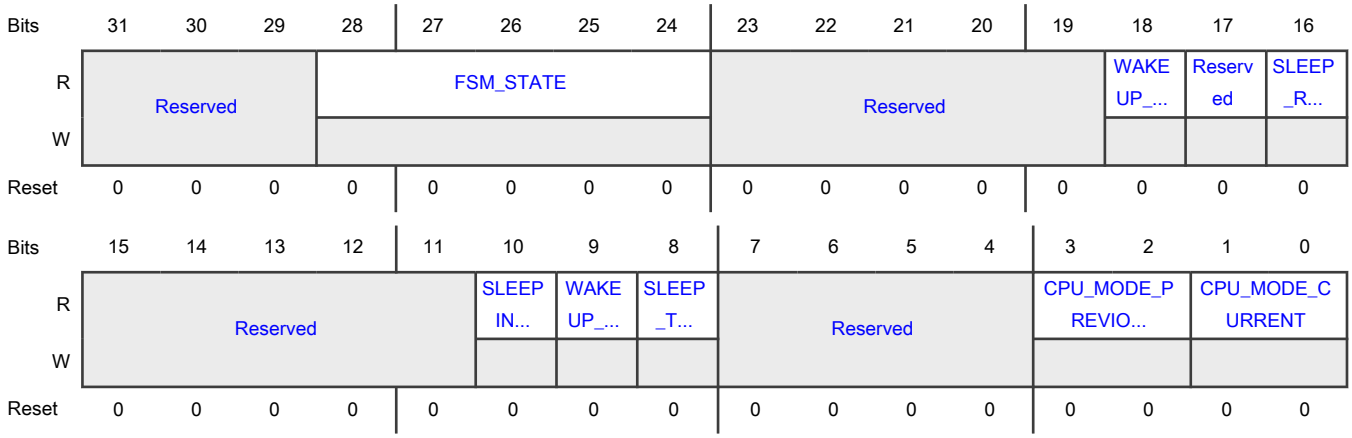
31.5.1.5 CM CPU mode Status (CM_MODE_STAT)

CPU mode status register.

Offset

Register	Offset
CM_MODE_STAT	14h

Diagram



Fields

Field	Description
31-29 —	Reserved
28-24 FSM_STATE	CPU mode trans FSM state.
23-19 —	Reserved
18 WAKEUP_REQUEST	"ORed" of all unmasked IRQ in.
17 —	Reserved
16 SLEEP_REQUEST	Status of sleep_request input port
15-11 —	Reserved
10 SLEEPING_IDLE	Completed CPU mode and set point transition of sleep sequence, in a sleeping_idle state.
9	Busy on CPU mode transition of wakeup, not include set point trans busy.

Table continues on the next page...

Table continued from the previous page...

Field	Description
WAKEUP_TRANS_BUSY	
8 SLEEP_TRANS_BUSY	Busy on CPU mode transition of sleep, not include set point trans busy.
7-4 —	Reserved
3-2 CPU_MODE_PREVIOUS	Previous CPU mode 00 - CPU was previously in RUN mode 01 - CPU was previously in WAIT mode 10 - CPU was previously in STOP mode 11 - CPU was previously in SUSPEND mode
1-0 CPU_MODE_CURRENT	Current CPU mode 00 - CPU is currently in RUN mode 01 - CPU is currently in WAIT mode 10 - CPU is currently in STOP mode 11 - CPU is currently in SUSPEND mode

31.5.1.6 CM pin Status (CM_PIN_STAT)

.

Offset

Register	Offset
CM_PIN_STAT	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEBUG_W...	CPU_MODE_STAT			Reserved			MTR_DON...	POWER_D...	RESET_D...	MEM_DON...	ISO_DON...	PLL_DON...	LPCG_DO...	SSAR_DO...	A55_HDS...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							MTR_REQ...	POWER_R...	RESET_R...	MEM_REQ...	ISO_REQ...	PLL_REQ...	LPCG_RE...	SSAR_RE...	A55_HDS...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 DEBUG_WAKEUP_ACK_STAT	debug wakeup acknowledge pin status
30-29 CPU_MODE_STAT	cpu_power_mode pin status
28-25 —	Reserved
24 MTR_DONE_STAT	cpu_mode_trans_mtr_done pin status
23 POWER_DONE_STAT	cpu_mode_trans_power_done pin status
22 RESET_DONE_STAT	cpu_mode_trans_reset_done pin status
21 MEM_DONE_STAT	cpu_mode_trans_mem_done pin status
20	cpu_mode_trans_iso_done pin status

Table continues on the next page...

Table continued from the previous page...

Field	Description
ISO_DONE_STAT	
19 PLL_DONE_STAT	cpu_mode_trans_pll_done pin status
18 LPCG_DONE_STAT	cpu_mode_trans_lpcg_done pin status
17 SSAR_DONE_STAT	cpu_mode_trans_ssar_done pin status
16 A55_HDSK_DONE_STAT	cpu_mode_trans_a55_hdsk_done pin status
15-9 —	Reserved
8 MTR_REQUEST_STAT	cpu_mode_trans_mtr_request pin status
7 POWER_REQUEST_STAT	cpu_mode_trans_power_request pin status
6 RESET_REQUEST_STAT	cpu_mode_trans_reset_request pin status
5 MEM_REQUEST_STAT	cpu_mode_trans_mem_request pin status
4 ISO_REQUEST_STAT	cpu_mode_trans_iso_request pin status
3	cpu_mode_trans_pll_request pin status

Table continues on the next page...

Table continued from the previous page...

Field	Description
PLL_REQUEST_STAT	
2 LPCG_REQUEST_STAT	cpu_mode_trans_lpcg_request pin status
1 SSAR_REQUEST_STAT	cpu_mode_trans_ssar_request pin status
0 A55_HDSK_REQUEST_STAT	cpu_mode_trans_a55_hdsk_request pin status

31.5.1.7 CM IRQ0~31 wakeup mask (CM_IRQ_WAKEUP_MASK_0)

IRQ0~31 mask register.

Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_0	100h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_0_31															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_0_31															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0	"1" means the IRQ cannot wakeup CPU platform

Table continues on the next page...

Field	Description
IRQ_WAKEUP_MASK_0_31	

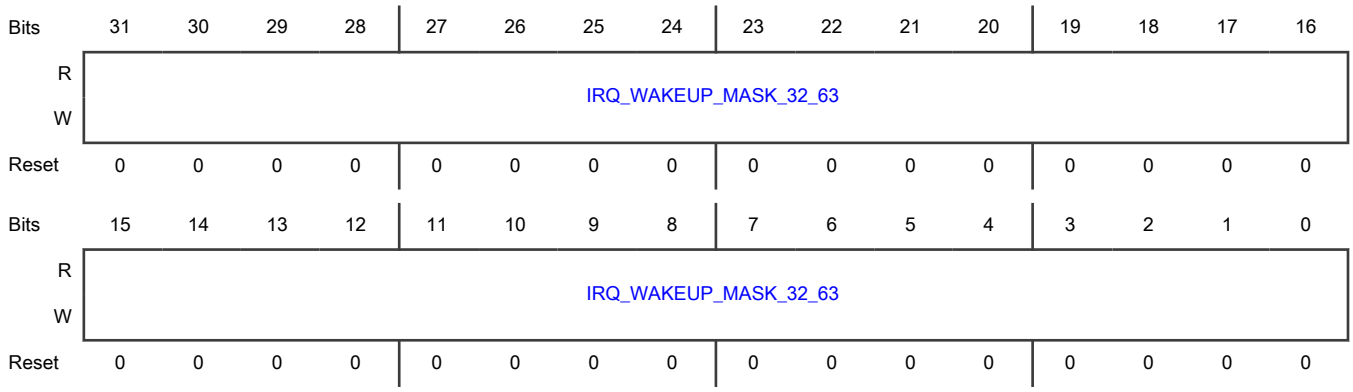
31.5.1.8 CM IRQ32~63 wakeup mask (CM_IRQ_WAKEUP_MASK_1)

IRQ32~63 mask register.

Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_1	104h

Diagram



Fields

Field	Description
31-0 IRQ_WAKEUP_MASK_32_63	"1" means the IRQ cannot wakeup CPU platform

31.5.1.9 CM IRQ64~95 wakeup mask (CM_IRQ_WAKEUP_MASK_2)

IRQ64~95 mask register.

Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_2	108h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_64_95															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_64_95															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 IRQ_WAKEUP_MASK_64_95	"1" means the IRQ cannot wakeup CPU platform

31.5.1.10 CM IRQ96~127 wakeup mask (CM_IRQ_WAKEUP_MASK_3)

IRQ96~127 mask register.

Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_3	10Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_96_127															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_96_127															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 IRQ_WAKEUP_MASK_96_127	"1" means the IRQ cannot wakeup CPU platform

31.5.1.11 CM IRQ128~159 wakeup mask (CM_IRQ_WAKEUP_MASK_4)

IRQ128~159 mask register.

Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_4	110h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_128_159															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_128_159															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 IRQ_WAKEUP_MASK_128_159	"1" means the IRQ cannot wakeup CPU platform

31.5.1.12 CM IRQ160~191 wakeup mask (CM_IRQ_WAKEUP_MASK_5)

IRQ160~191 mask register.

Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_5	114h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_160_191															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_160_191															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 IRQ_WAKEUP_MASK_160_191	"1" means the IRQ cannot wakeup CPU platform

31.5.1.13 CM IRQ192~223 wakeup mask (CM_IRQ_WAKEUP_MASK_6)

IRQ192~223 mask register.

Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_6	118h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_192_223															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_192_223															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 IRQ_WAKEUP_MASK_192_223	"1" means the IRQ cannot wakeup CPU platform

31.5.1.14 CM IRQ224~255 wakeup mask (CM_IRQ_WAKEUP_MASK_7)

IRQ224~255 mask register.

Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_7	11Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_224_255															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_224_255															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 IRQ_WAKEUP_MASK_224_255	"1" means the IRQ cannot wakeup CPU platform

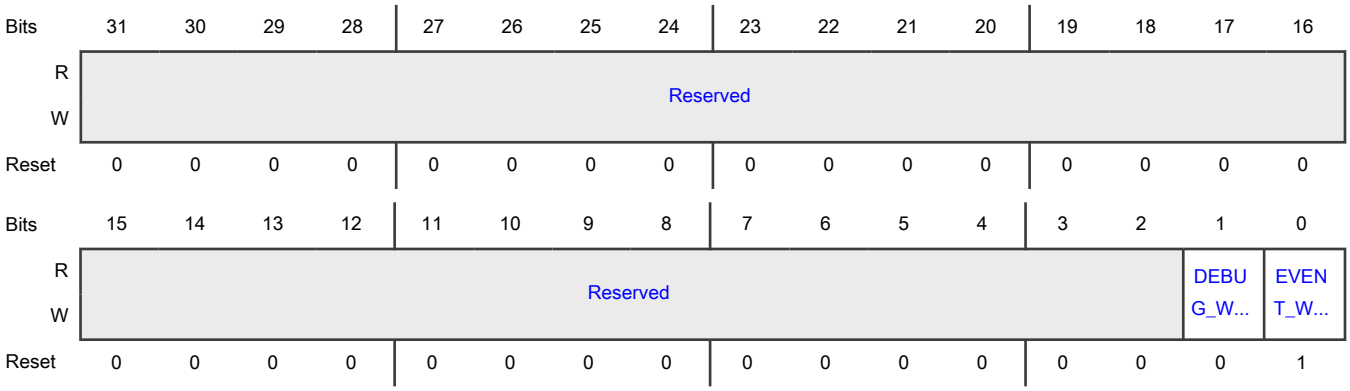
31.5.1.15 CM non-IRQ wakeup mask (CM_NON_IRQ_WAKEUP_MASK)

This register is used to mask wakeup events which are not interrupt.

Offset

Register	Offset
CM_NON_IRQ_WAKEUP_MASK	140h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 DEBUG_WAKEUP_MASK	"1" means the debug_wakeup_request cannot wakeup CPU platform
0 EVENT_WAKEUP_MASK	"1" means the event cannot wakeup CPU platform

31.5.1.16 CM IRQ0~31 wakeup status (CM_IRQ_WAKEUP_STAT_0)

Interrupt 0~31 status.

Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_0	150h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_0_31															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_0_31															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 IRQ_WAKEUP_STAT_0_31	IRQ status A bit set to 1 means the corresponding IRQ has a valid pending interrupt to wakeup the CPU platform.

31.5.1.17 CM IRQ32~63 wakeup status (CM_IRQ_WAKEUP_STAT_1)

Interrupt 32~63 status.

Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_1	154h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_32_63															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_32_63															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 IRQ_WAKEUP_STAT_32_63	IRQ status A bit set to 1 means the corresponding IRQ has a valid pending interrupt to wakeup the CPU platform.

31.5.1.18 CM IRQ64~95 wakeup status (CM_IRQ_WAKEUP_STAT_2)

Interrupt 64~95 status.

Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_2	158h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_64_95															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_64_95															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 IRQ_WAKEUP_STAT_64_95	IRQ status A bit set to 1 means the corresponding IRQ has a valid pending interrupt to wakeup the CPU platform.

31.5.1.19 CM IRQ96~127 wakeup status (CM_IRQ_WAKEUP_STAT_3)

Interrupt 96~127 status.

Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_3	15Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_96_127															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_96_127															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 IRQ_WAKEUP_STAT_96_127	IRQ status A bit set to 1 means the corresponding IRQ has a valid pending interrupt to wakeup the CPU platform.

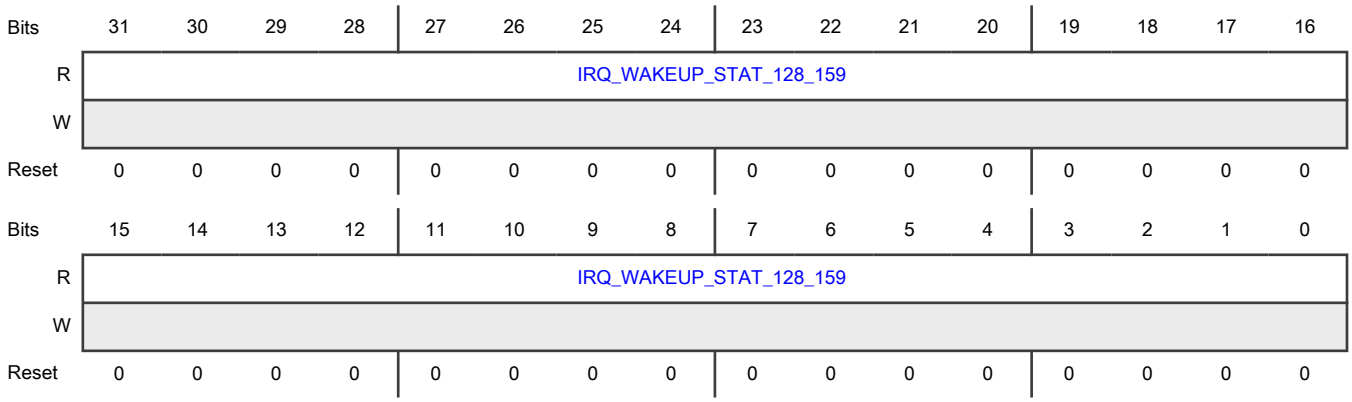
31.5.1.20 CM IRQ128~159 wakeup status (CM_IRQ_WAKEUP_STAT_4)

Interrupt 128~159 status.

Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_4	160h

Diagram



Fields

Field	Description
31-0 IRQ_WAKEUP_STAT_128_159	IRQ status A bit set to 1 means the corresponding IRQ has a valid pending interrupt to wakeup the CPU platform.

31.5.1.21 CM IRQ160~191 wakeup status (CM_IRQ_WAKEUP_STAT_5)

Interrupt 160~191 status.

Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_5	164h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_160_191															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_160_191															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 IRQ_WAKEUP_STAT_160_191	IRQ status A bit set to 1 means the corresponding IRQ has a valid pending interrupt to wakeup the CPU platform.

31.5.1.22 CM IRQ192~223 wakeup status (CM_IRQ_WAKEUP_STAT_6)

Interrupt 192~223 status.

Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_6	168h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_192_223															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_192_223															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 IRQ_WAKEUP_STAT_192_223	IRQ status A bit set to 1 means the corresponding IRQ has a valid pending interrupt to wakeup the CPU platform.

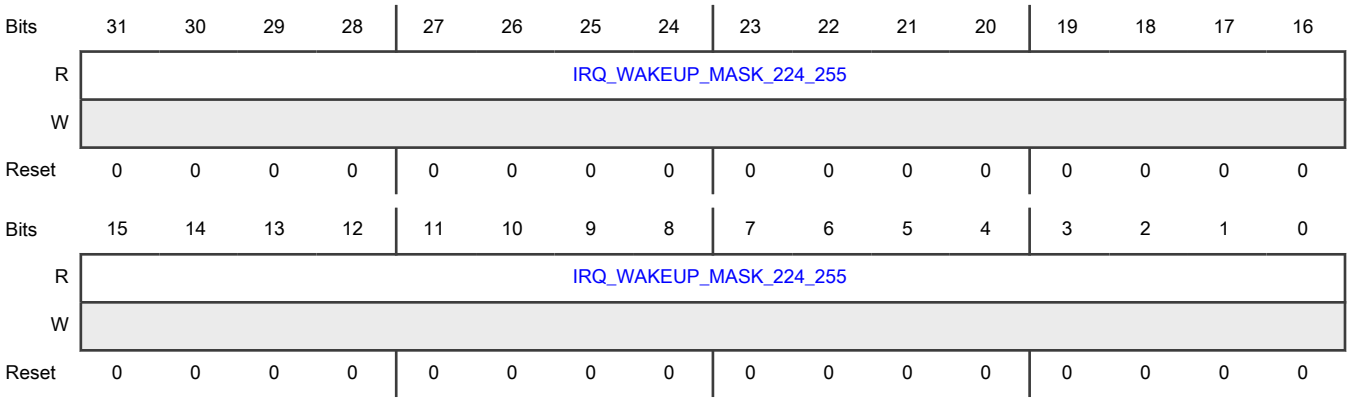
31.5.1.23 CM IRQ224~255 wakeup status (CM_IRQ_WAKEUP_STAT_7)

Interrupt 224~255 status.

Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_7	16Ch

Diagram



Fields

Field	Description
31-0 IRQ_WAKEUP_MASK_224_255	IRQ status A bit set to 1 means the corresponding IRQ has a valid pending interrupt to wakeup the CPU platform.

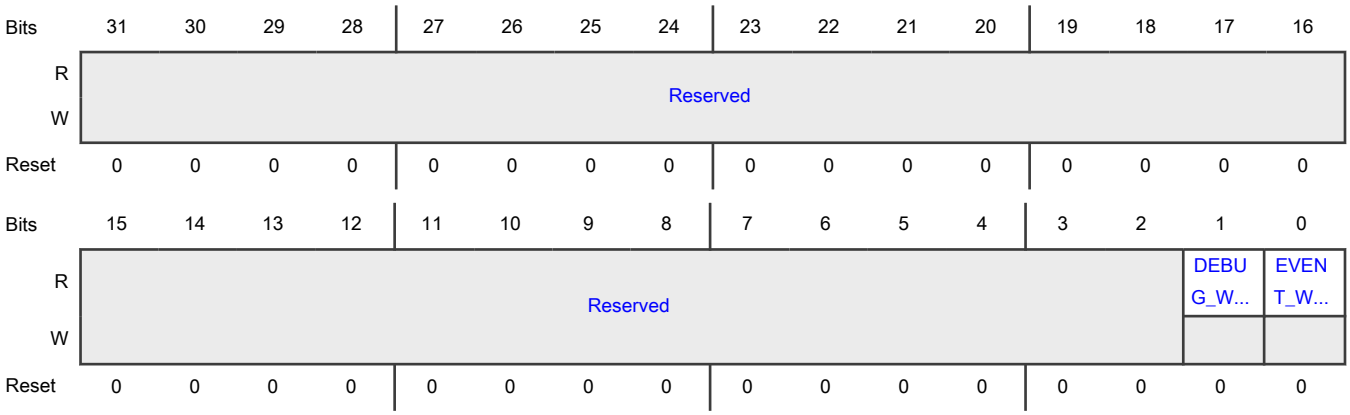
31.5.1.24 CM non-IRQ wakeup status (CM_NON_IRQ_WAKEUP_STAT)

Wakeup event status.

Offset

Register	Offset
CM_NON_IRQ_WAKEUP_STAT	190h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 DEBUG_WAKEUP_STAT	Debug wakeup status 0 - No debug wakeup is requested 1 - Debug wakeup is requested
0 EVENT_WAKEUP_STAT	Event wakeup status 0 - No event wakeup is requested 1 - Event wakeup is requested

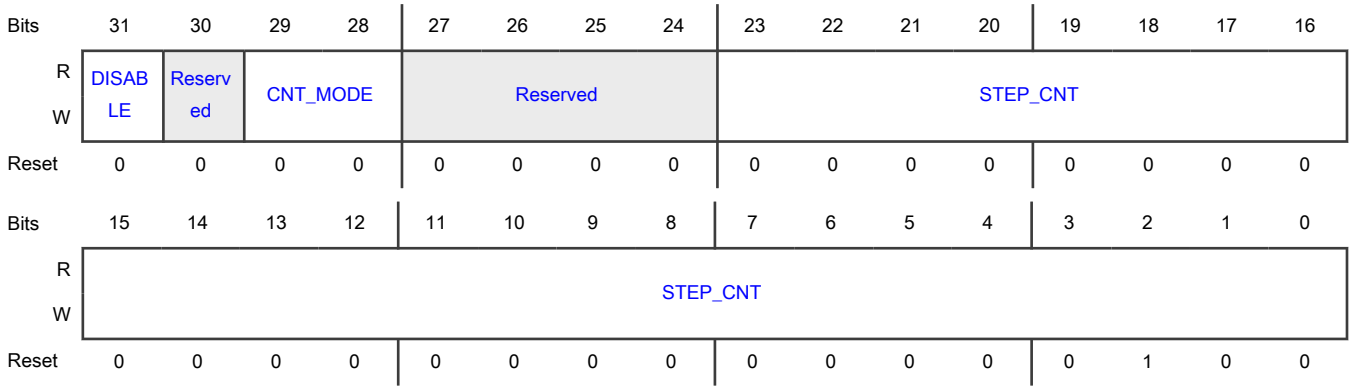
31.5.1.25 CM sleep A55_HDSK control (CM_SLEEP_A55_HDSK_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_SLEEP_A55_HDSK_CTRL	200h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE. Must be greater than 2 under CNT_MODE 2 and 3

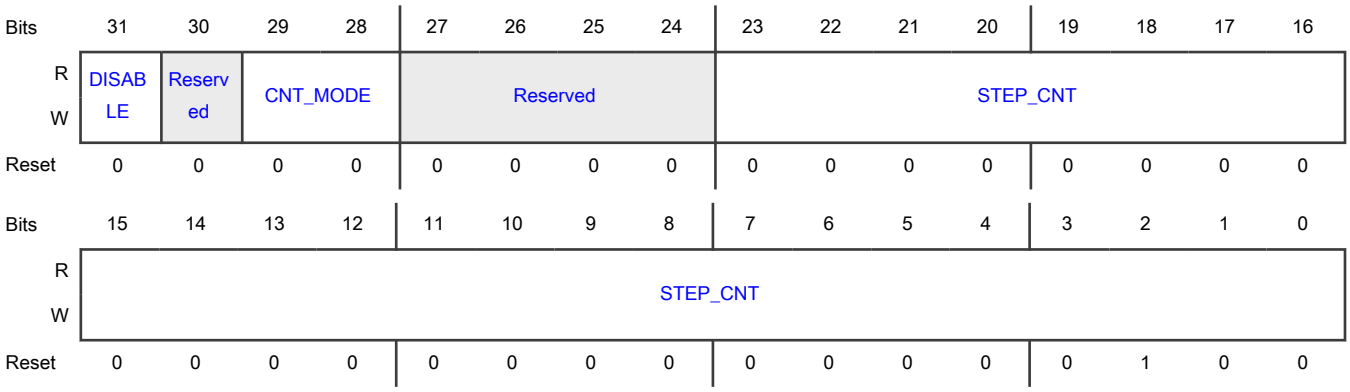
31.5.1.26 CM sleep SSAR control (CM_SLEEP_SSAR_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_SLEEP_SSAR_CTRL	208h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE. Must be greater than 2 under CNT_MODE 2 and 3

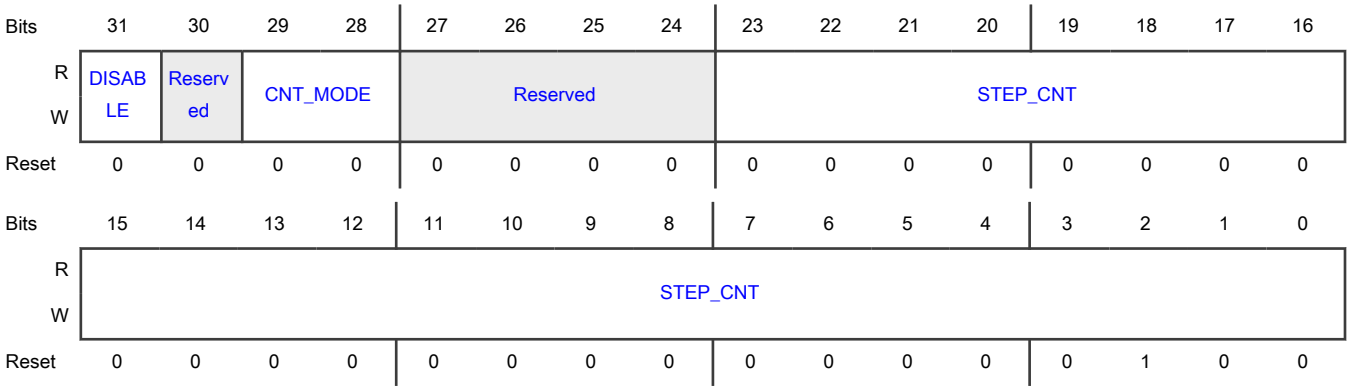
31.5.1.27 CM sleep LPCG control (CM_SLEEP_LPCG_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_SLEEP_LPCG_CTRL	210h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

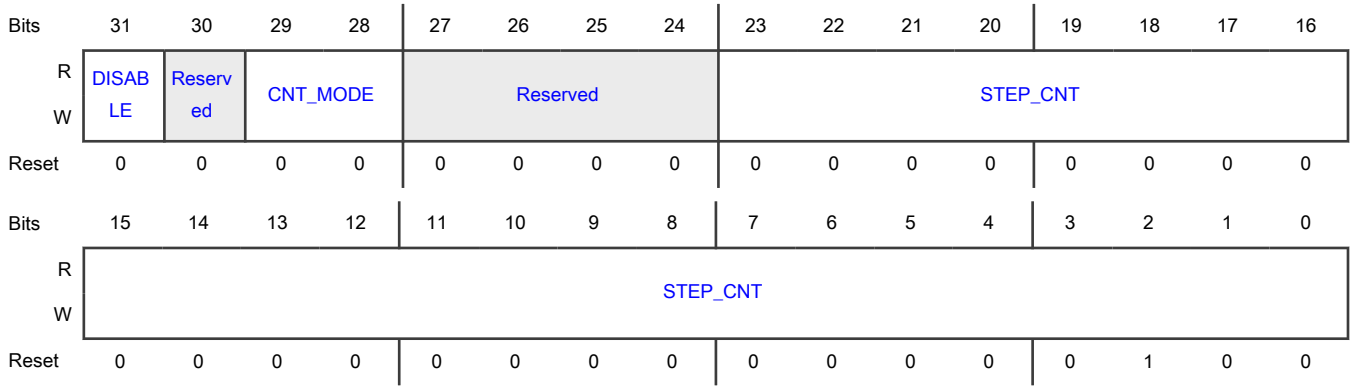
31.5.1.28 CM sleep PLL control (CM_SLEEP_PLL_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_SLEEP_PLL_CTRL	218h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT

Table continues on the next page...

Table continued from the previous page...

Field	Description
	10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

31.5.1.29 CM sleep isolation control (CM_SLEEP_ISO_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_SLEEP_ISO_CTRL	220h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DISABLE	Reserved	CNT_MODE		Reserved				STEP_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STEP_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

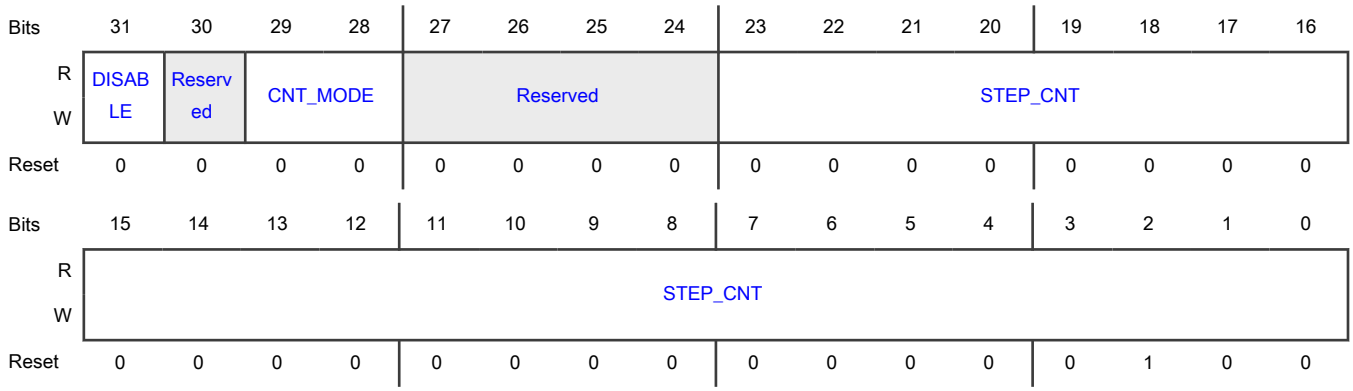
31.5.1.30 CM sleep memory control (CM_SLEEP_MEM_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_SLEEP_MEM_CTRL	228h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

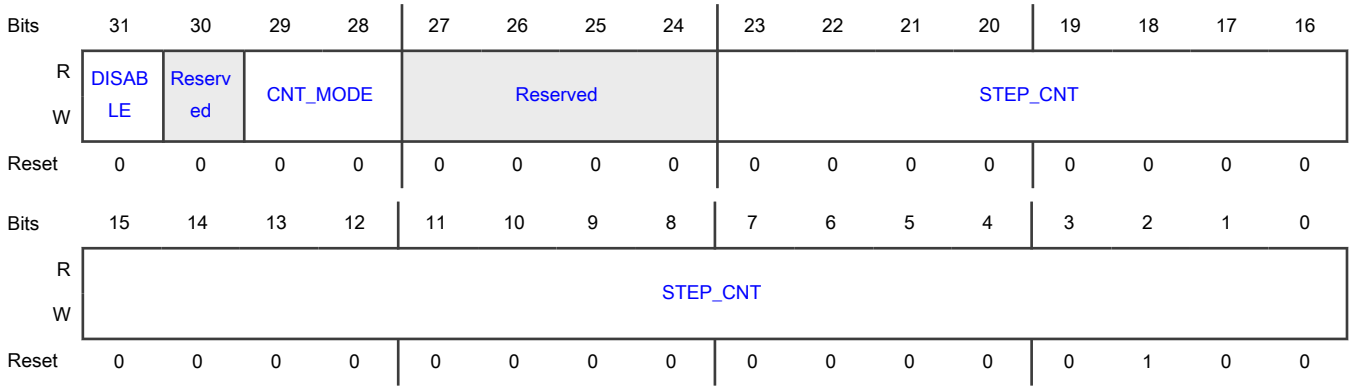
31.5.1.31 CM sleep reset control (CM_SLEEP_RESET_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_SLEEP_RESET_CTRL	230h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

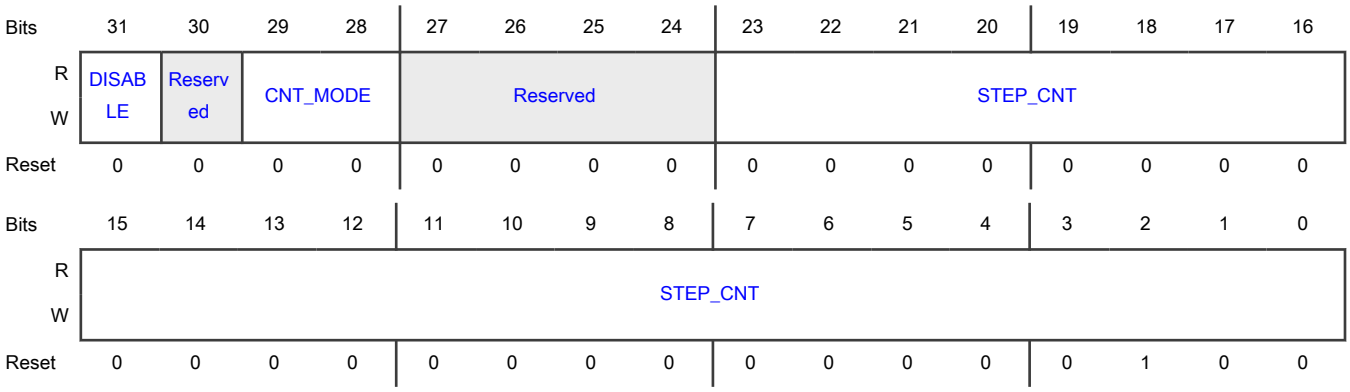
31.5.1.32 CM sleep power control (CM_SLEEP_POWER_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_SLEEP_POWER_CTRL	238h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

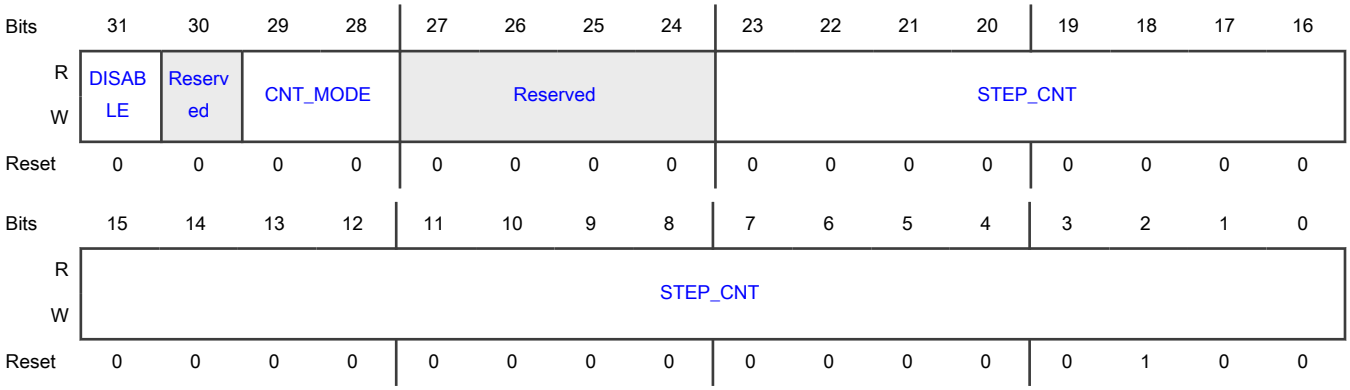
31.5.1.33 CM sleep rsv2 control (CM_SLEEP_RSV2_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_SLEEP_RSV2_CTRL	240h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

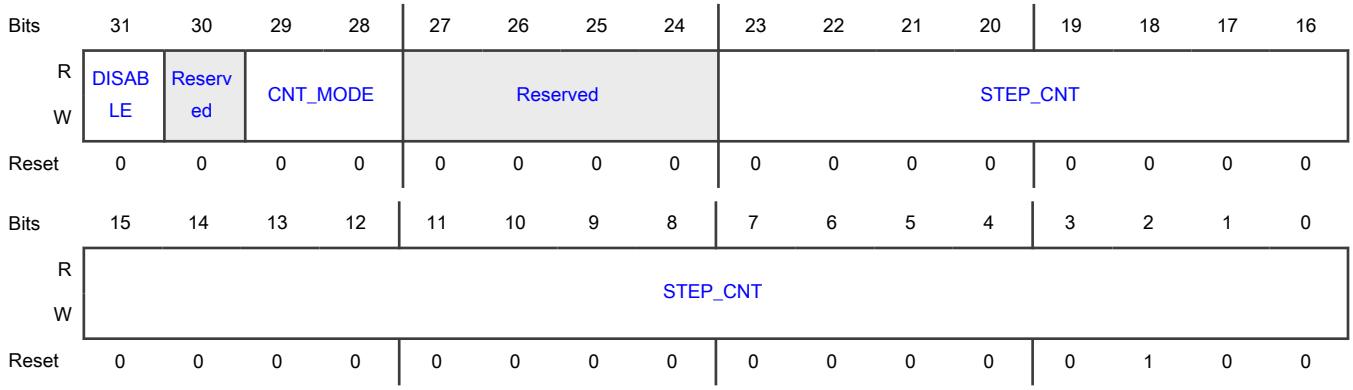
31.5.1.34 CM wakeup rsv2 control (CM_WAKEUP_RSV2_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_WAKEUP_RSV2_CTRL	288h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT

Table continues on the next page...

Table continued from the previous page...

Field	Description
	10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

31.5.1.35 CM wakeup power control (CM_WAKEUP_POWER_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_WAKEUP_POWER_CTRL	290h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DISABLE	Reserved	CNT_MODE		Reserved				STEP_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STEP_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.

Table continues on the next page...

Table continued from the previous page...

Field	Description
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

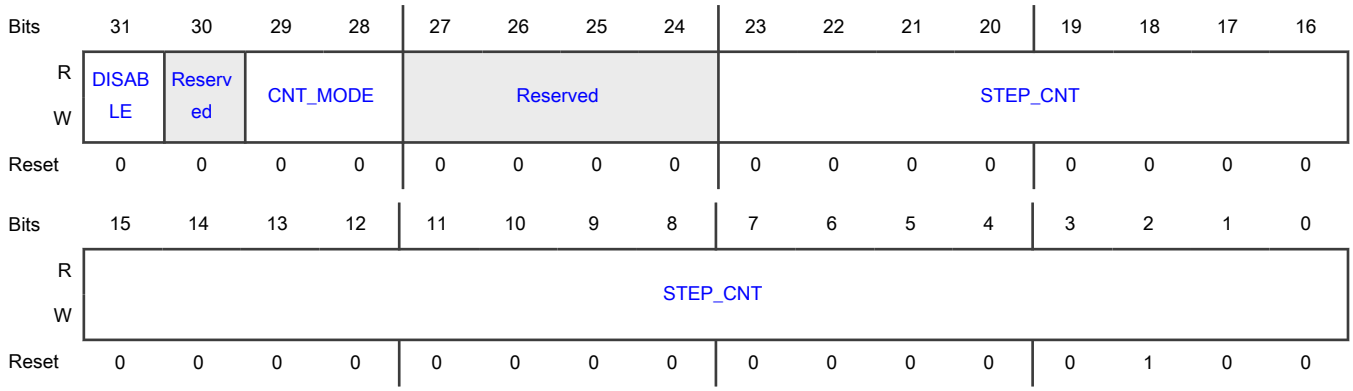
31.5.1.36 CM wakeup memory control (CM_WAKEUP_MEM_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_WAKEUP_MEM_CTRL	298h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

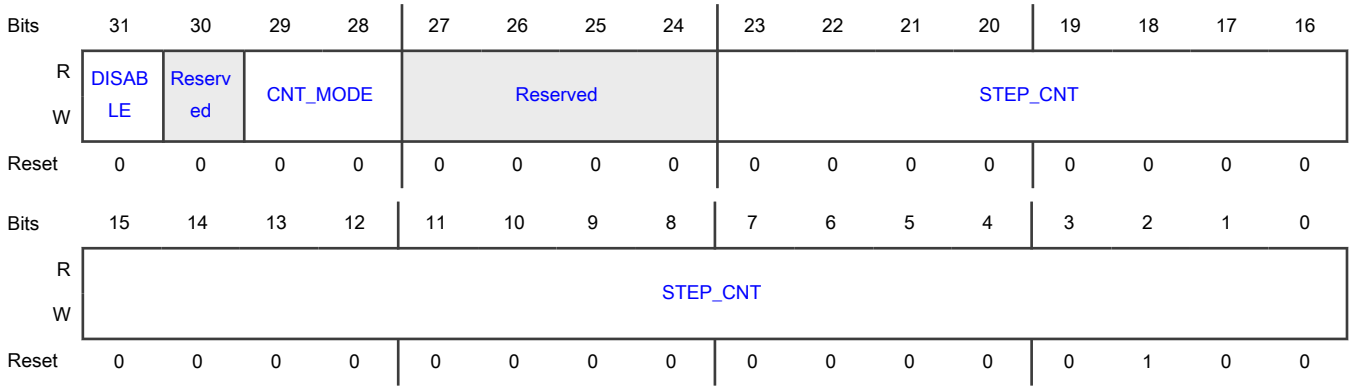
31.5.1.37 CM wakeup reset control (CM_WAKEUP_RESET_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_WAKEUP_RESET_CTRL	2A0h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

31.5.1.38 CM wakeup isolation control (CM_WAKEUP_ISO_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_WAKEUP_ISO_CTRL	2A8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DISABLE	Reserved	CNT_MODE		Reserved				STEP_CNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STEP_CNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

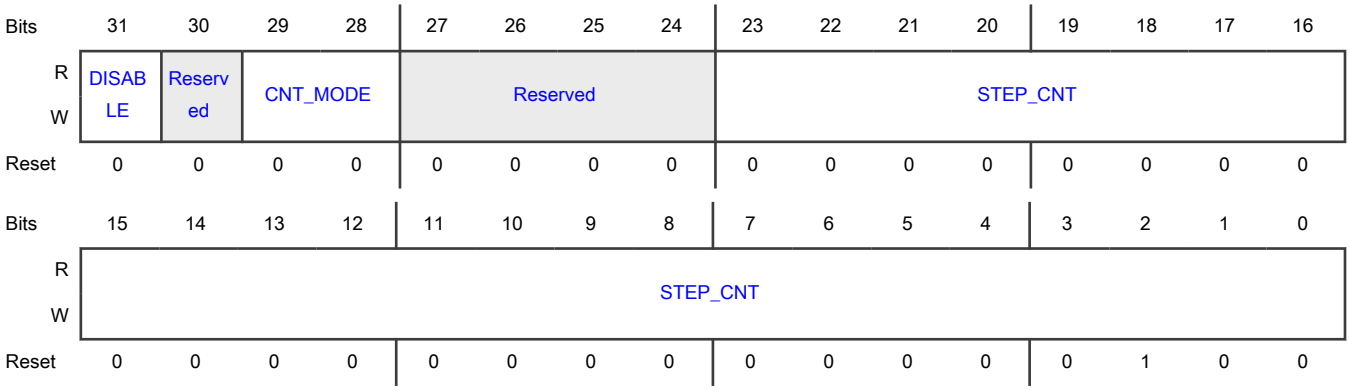
31.5.1.39 CM wakeup PLL control (CM_WAKEUP_PLL_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_WAKEUP_PLL_CTRL	2B0h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

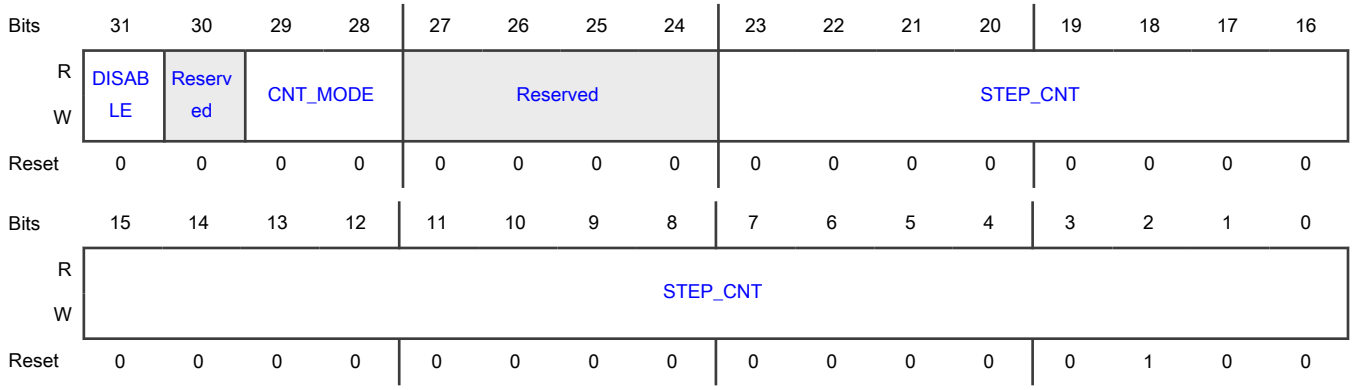
31.5.1.40 CM wakeup LPCG control (CM_WAKEUP_LPCG_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_WAKEUP_LPCG_CTRL	2B8h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT

Table continues on the next page...

Table continued from the previous page...

Field	Description
	10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

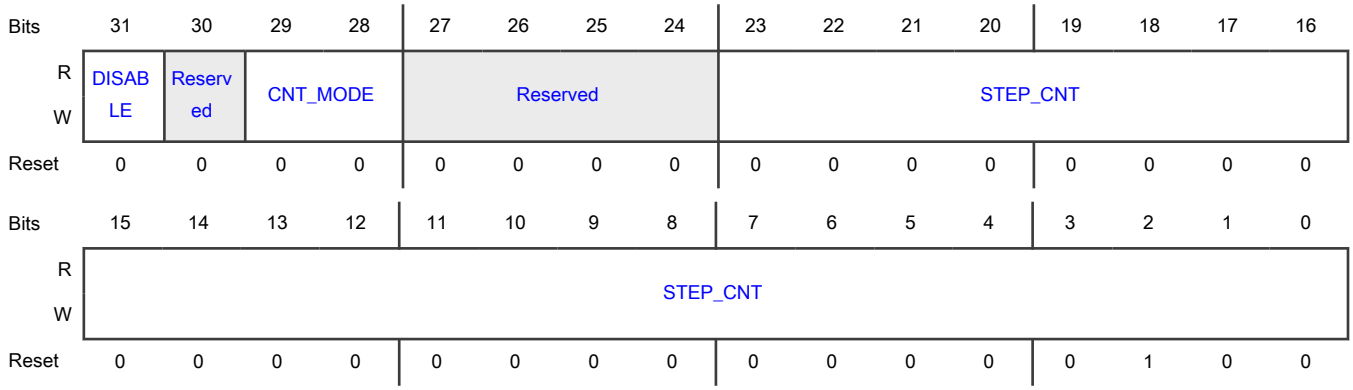
31.5.1.41 CM wakeup MTR control (CM_WAKEUP_MTR_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_WAKEUP_MTR_CTRL	2C0h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.

Table continues on the next page...

Table continued from the previous page...

Field	Description
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

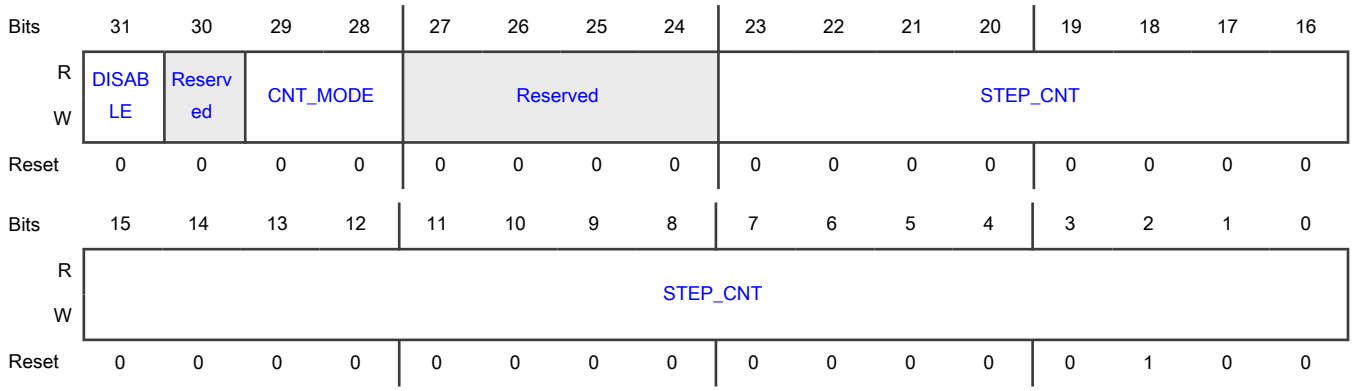
31.5.1.42 CM wakeup SSAR control (CM_WAKEUP_SSAR_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_WAKEUP_SSAR_CTRL	2C8h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

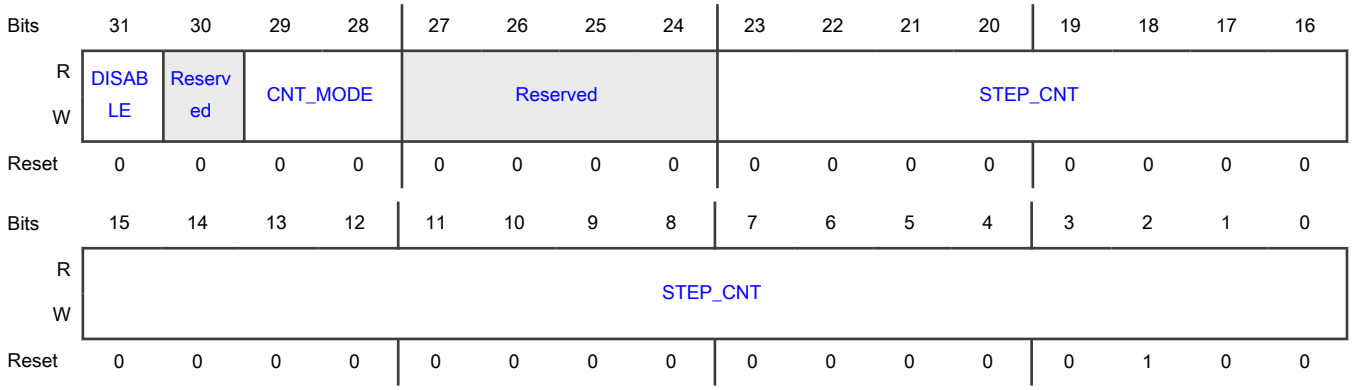
31.5.1.43 CM wakeup A55_HDSK control (CM_WAKEUP_A55_HDSK_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_WAKEUP_A55_HDSK_CTRL	2D0h

Diagram



Fields

Field	Description
31 DISABLE	Disable this step 0 - This step is enabled. 1 - This step is disabled. GPC will skip this step and not send any request.
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00 - Counter disable mode: not use step counter, step completes once receiving step_done 01 - Counter delay mode: delay after receiving step_done, delay cycle number is STEP_CNT 10 - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes 11 - Time out mode, the counter starts to count once step begins, the step completes when either step_done received or counting to STEP_CNT value
27-24 —	Reserved
23-0 STEP_CNT	Step count, usage depends on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

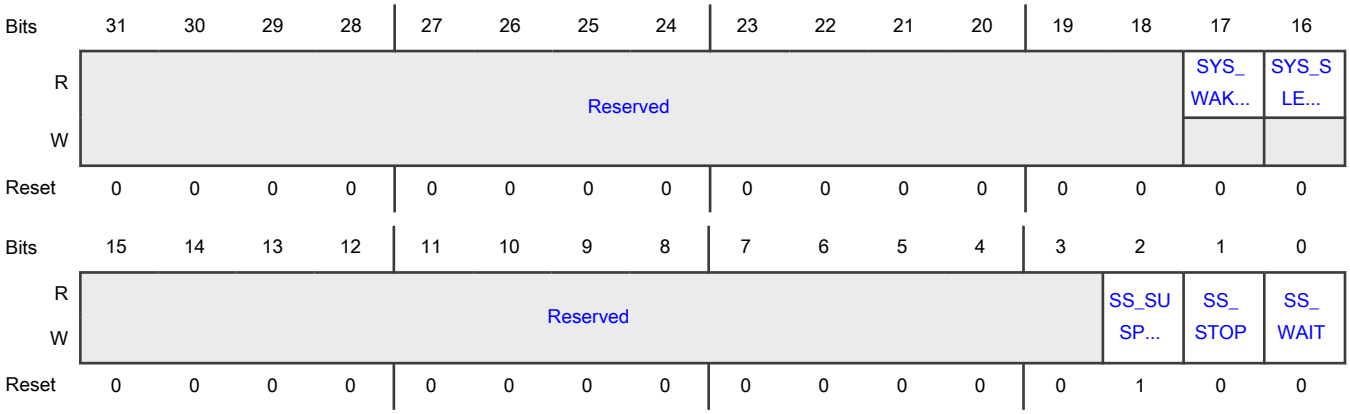
31.5.1.44 CM system sleep control (CM_SYS_SLEEP_CTRL)

Locked by LOCK_CFG field.

Offset

Register	Offset
CM_SYS_SLEEP_CTRL	380h

Diagram



Fields

Field	Description
31-18 —	Reserved
17 SYS_WAKEUP_BUSY	Indicates the CPU is busy exiting system sleep mode.
16 SYS_SLEEP_BUSY	Indicates the CPU is busy entering system sleep mode.
15-3 —	Reserved
2 SS_SUSPEND	Request system sleep when CPU is in SUSPEND mode 0 - Do not request system sleep when CPU is in SUSPEND mode 1 - Request system sleep when CPU is in SUSPEND mode
1 SS_STOP	Request system sleep when CPU is in STOP mode 0 - Do not request system sleep when CPU is in STOP mode 1 - Request system sleep when CPU is in STOP mode
0	Request system sleep when CPU is in WAIT mode

Table continues on the next page...

Table continued from the previous page...

Field	Description
SS_WAIT	0 - Do not request system sleep when CPU is in WAIT mode 1 - Request system sleep when CPU is in WAIT mode

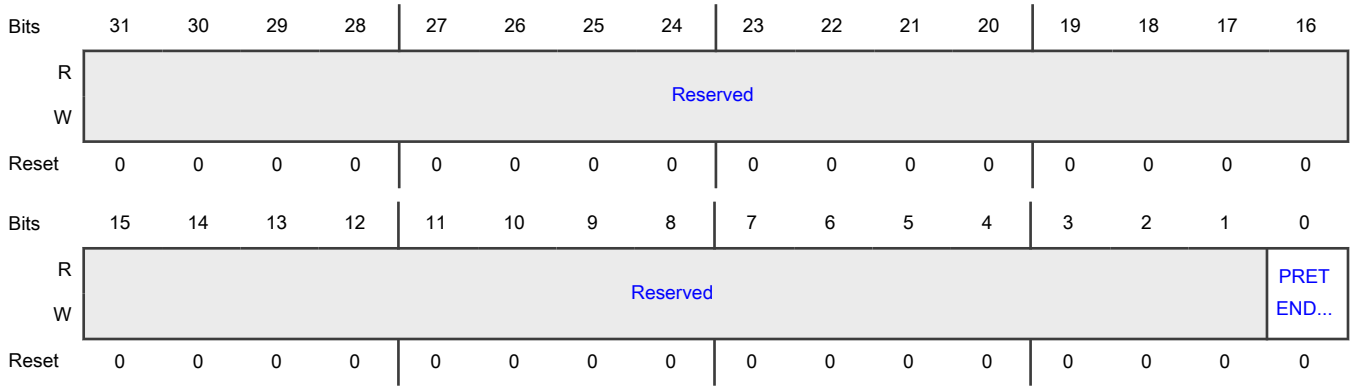
31.5.1.45 CM debug (CM_DEBUG)

.

Offset

Register	Offset
CM_DEBUG	390h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 PRETEND_SLEEP	Write 1 to force CMC into sleep. Used to debug GPC status. Locked by LOCK_CFG field.

31.5.2 GPC_GLOBAL register descriptions

31.5.2.1 gpc_global memory map

gpc_global base address: 4447_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	GPC Global Authentication Control (AUTHEN_CTRL)	32	RW	FFFF_0000
10	GPC domain assignment (GPC_DOMAIN)	32	RW	0000_0000
1C	GPC master CPU configuration (GPC_MASTER)	32	RW	0000_0000
40	GPC system sleep control (GPC_SYS_SLEEP)	32	RW	0000_0000
100	PMIC standby control (PMIC_CTRL)	32	RW	0000_0000
104	PMIC standby pre delay control (PMIC_PRE_DLY_CTRL)	32	RW	0000_0000
108	PMIC standby acknowledge control (PMIC_STBY_ACK_CTRL)	32	RW	8A00_8020
200	RCOSC control (GPC_ROSC_CTRL)	32	RW	0000_0000
204	AON Memory control (GPC_AON_MEM_CTRL)	32	RW	0000_0000
208	eFUSE control (GPC_EFUSE_CTRL)	32	RW	0000_0000

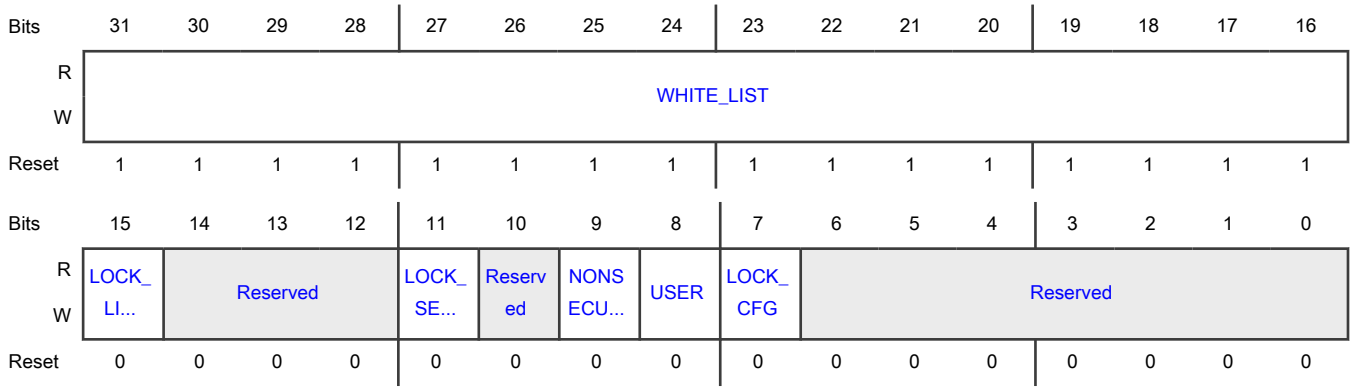
31.5.2.2 GPC Global Authentication Control (AUTHEN_CTRL)

Authentication control for GPC GLOBAL.

Offset

Register	Offset
AUTHEN_CTRL	4h

Diagram



Fields

Field	Description
31-16 WHITE_LIST	Domain ID white list

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When bit is set to 1, the corresponding domain ID can access CPU mode control registers. bit[0] for Domain 0, bit[1] for Domain 1, bit[2] for Domain 2, and bit[3] for Domain 3. Multiple bits with a value of 1 is permitted.
15 LOCK_LIST	White list lock This bit locks the WHITE_LIST. When this bit is set, WHITE_LIST cannot be changed. Once this bit is set, it cannot be cleared until the next system reset. 0 - WHITE_LIST is not locked 1 - WHITE_LIST is locked
14-12 —	Reserved
11 LOCK_SETTING	Lock NONSECURE and USER This bit locks the NONSECURE and USER fields. When this bit is set, LOCK_SETTING cannot be changed. Once this bit is set, it cannot be cleared until the next system reset. 0 - NONSECURE and USER fields are not locked 1 - NONSECURE and USER fields are locked
10 —	Reserved
9 NONSECURE	Allow non-secure mode access 0 - Allow only secure mode to access CPU mode registers 1 - Allow both secure and non-secure mode to access CPU mode control registers.
8 USER	Allow user mode access 0 - Allow only privilege mode to access CPU mode control registers 1 - Allow both privilege and user mode to access CPU mode control registers
7 LOCK_CFG	Configuration lock This bit locks the value of low power configuration fields. Once this bit is set, it cannot be cleared until the next system reset. 0 - The value of low power configuration fields are not locked. 1 - The value of low power configuration fields are locked. Refer to the function field of each gpc_global registers.
6-0 —	Reserved

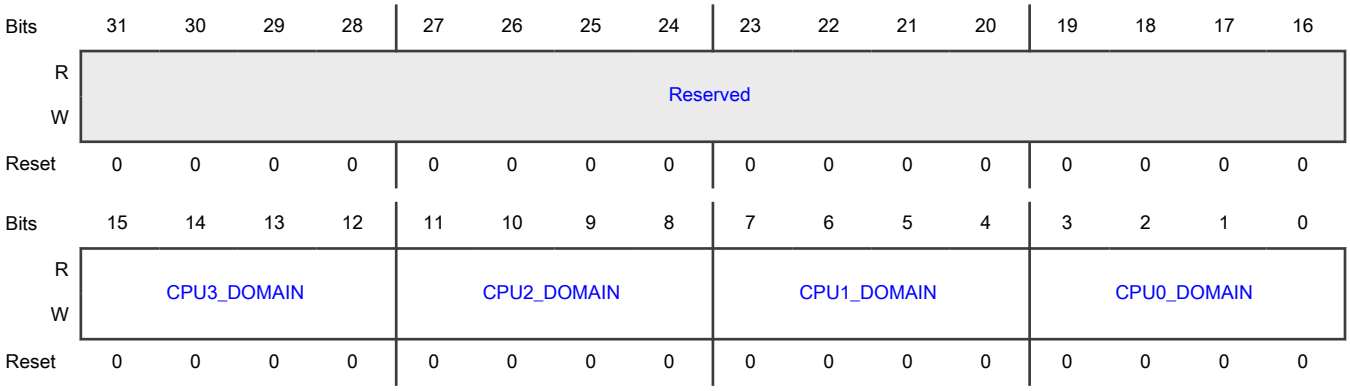
31.5.2.3 GPC domain assignment (GPC_DOMAIN)

Locked by LOCK_CFG field.

Offset

Register	Offset
GPC_DOMAIN	10h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-12 CPU3_DOMAIN	CPU3 domain assignment
11-8 CPU2_DOMAIN	CPU2 domain assignment
7-4 CPU1_DOMAIN	CPU1 domain assignment
3-0 CPU0_DOMAIN	CPU0 domain assignment

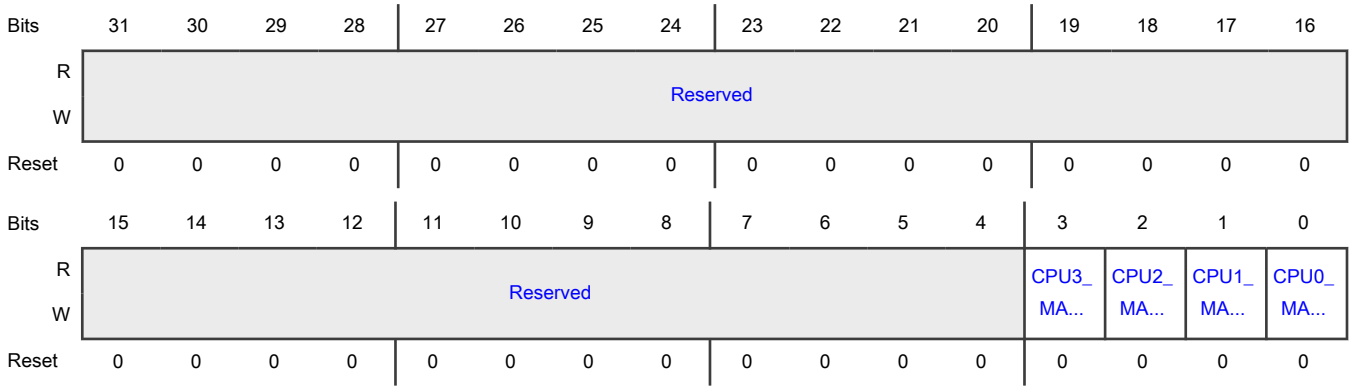
31.5.2.4 GPC master CPU configuration (GPC_MASTER)

Locked by LOCK_CFG field.

Offset

Register	Offset
GPC_MASTER	1Ch

Diagram



Fields

Field	Description
31-4 —	Reserved
3 CPU3_MASTER	Setting to 1 means CPU3 is the master CPU of its domain 0 - CPU3 is not the master CPU of its domain 1 - CPU3 is the master CPU of its domain
2 CPU2_MASTER	Setting to 1 means CPU2 is the master CPU of its domain 0 - CPU2 is not the master CPU of its domain 1 - CPU2 is the master CPU of its domain
1 CPU1_MASTER	Setting to 1 means CPU1 is the master CPU of its domain 0 - CPU1 is not the master CPU of its domain 1 - CPU1 is the master CPU of its domain
0 CPU0_MASTER	Setting to 1 means CPU0 is the master CPU of its domain 0 - CPU0 is not the master CPU of its domain 1 - CPU0 is the master CPU of its domain

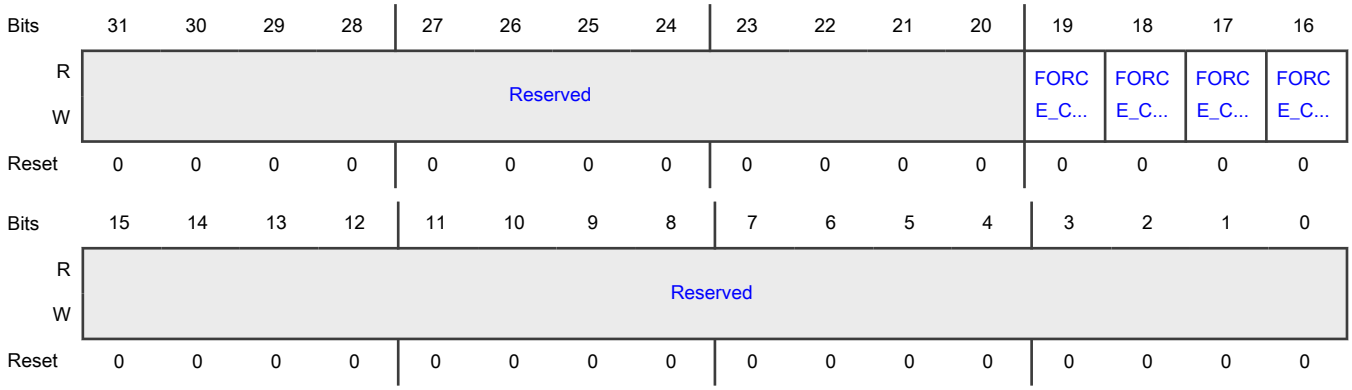
31.5.2.5 GPC system sleep control (GPC_SYS_SLEEP)

.

Offset

Register	Offset
GPC_SYS_SLEEP	40h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 FORCE_CPU3_DISABLE	Force CPU3 into a system sleep status
18 FORCE_CPU2_DISABLE	Force CPU2 into a system sleep status
17 FORCE_CPU1_DISABLE	Force CPU1 into a system sleep status
16 FORCE_CPU0_DISABLE	Force CPU0 into a system sleep status
15-0 —	Reserved

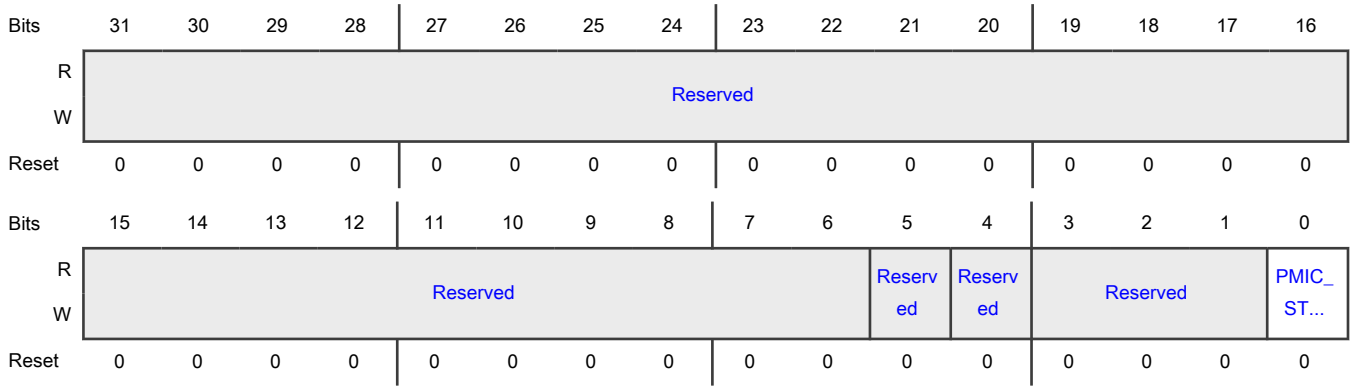
31.5.2.6 PMIC standby control (PMIC_CTRL)

.

Offset

Register	Offset
PMIC_CTRL	100h

Diagram



Fields

Field	Description
31-6 —	Reserved
5 —	Reserved
4 —	Reserved
3-1 —	Reserved
0 PMIC_STBY_EN	Assert the PMIC_STBY_REQ when system sleep

31.5.2.7 PMIC standby pre delay control (PMIC_PRE_DLY_CTRL)

.

Offset

Register	Offset
PMIC_PRE_DLY_CTRL	104h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLY_PRE_STBY_OFF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLY_PRE_STBY_ON															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 DLY_PRE_STBY_OFF	Delay before pmic_standby off. Locked by LOCK_CFG field.
15-0 DLY_PRE_STBY_ON	Delay before pmic_standby on. Locked by LOCK_CFG field.

31.5.2.8 PMIC standby acknowledge control (PMIC_STBY_ACK_CTRL)

.

Offset

Register	Offset
PMIC_STBY_ACK_CTRL	108h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STBY_OFF_CNT_MO...		Reserved		STBY_OFF_CNT_CFG											
Reset	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STBY_ON_CNT_MODE		Reserved		STBY_ON_CNT_CFG											
Reset	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Fields

Field	Description
31-30 STBY_OFF_CNT_MODE	PMIC standby off acknowledge count mode. Locked by LOCK_CFG field. 00 - Finish the process once pmic_standby signal changes 01 - Finish the process once getting acknowledge from PMIC 10 - Ignore PMIC acknowledge, the delay counter starts to count once pmic_standby changes 11 - Time out mode, the counter starts to count once pmic_standby changes, then finishes the process when either acknowledge received or counting to CNT_CFG value
29-28 —	Reserved
27-16 STBY_OFF_CNT_CFG	PMIC standby off acknowledge count configure. Usage depends on STBY_OFF_CNT_MODE. Locked by LOCK_CFG field.
15-14 STBY_ON_CNT_MODE	PMIC standby on acknowledge count mode. Locked by LOCK_CFG field. 00 - Finish the process once pmic_standby signal changes 01 - Finish the process once getting acknowledge from PMIC 10 - Ignore PMIC acknowledge, the delay counter starts to count once pmic_standby changes 11 - Time out mode, the counter starts to count once pmic_standby changes, then finishes the process when either acknowledge received or counting to CNT_CFG value
13-12 —	Reserved
11-0 STBY_ON_CNT_CFG	PMIC standby on acknowledge count configure. Usage depends on STBY_ON_CNT_MODE. Locked by LOCK_CFG field.

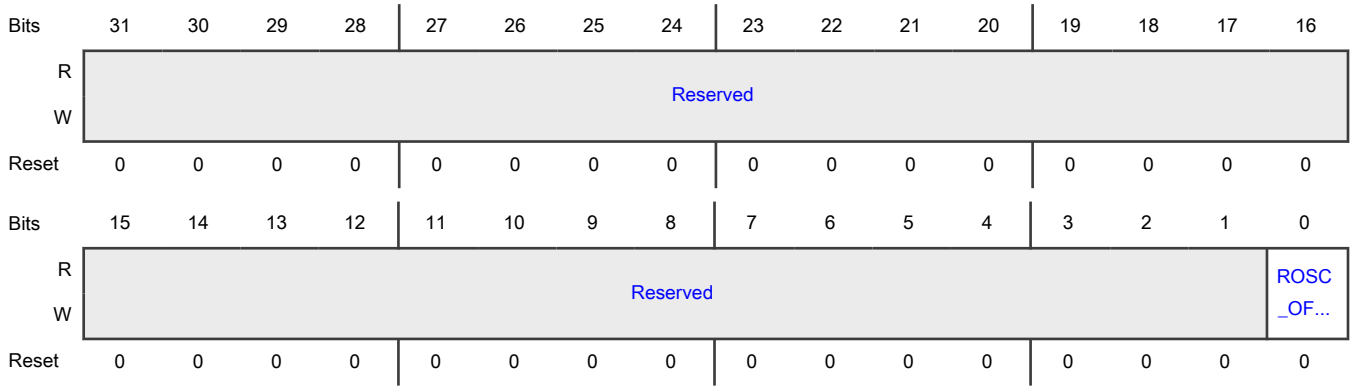
31.5.2.9 RCOSC control (GPC_ROSC_CTRL)

.

Offset

Register	Offset
GPC_ROSC_CTRL	200h

Diagram



Fields

Field	Description
31-1 —	Reserved 000_0000_0000_0000_0000_0000_0000_0000 - Keep 24 MHz ROSC clock running during system sleep 000_0000_0000_0000_0000_0000_0000_0001 - Shut off 24 MHz ROSC clock during system sleep
0 ROSC_OFF_EN	Shut off the 24 MHz RCOSC clock when system sleep

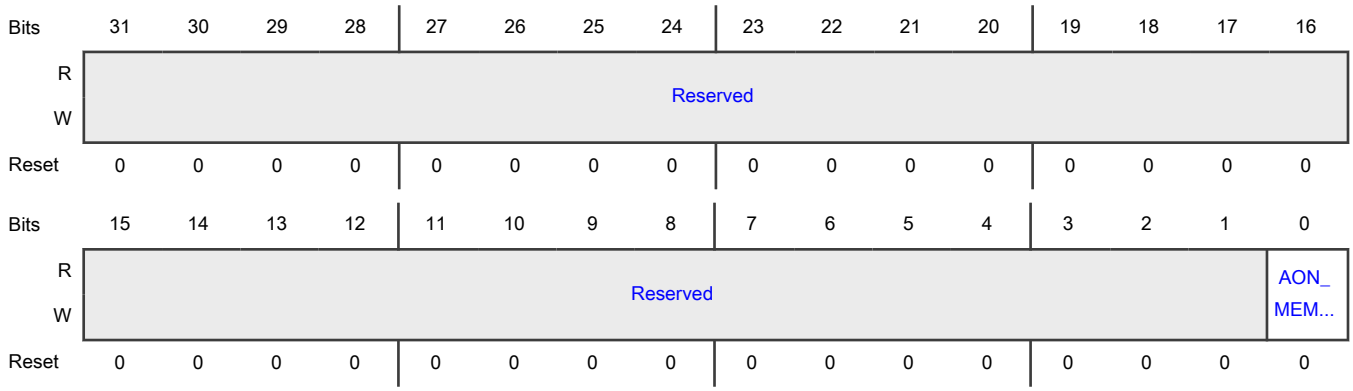
31.5.2.10 AON Memory control (GPC_AON_MEM_CTRL)

.

Offset

Register	Offset
GPC_AON_MEM_CTRL	204h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 AON_MEM_LP_EN	AON memory enter LP enable

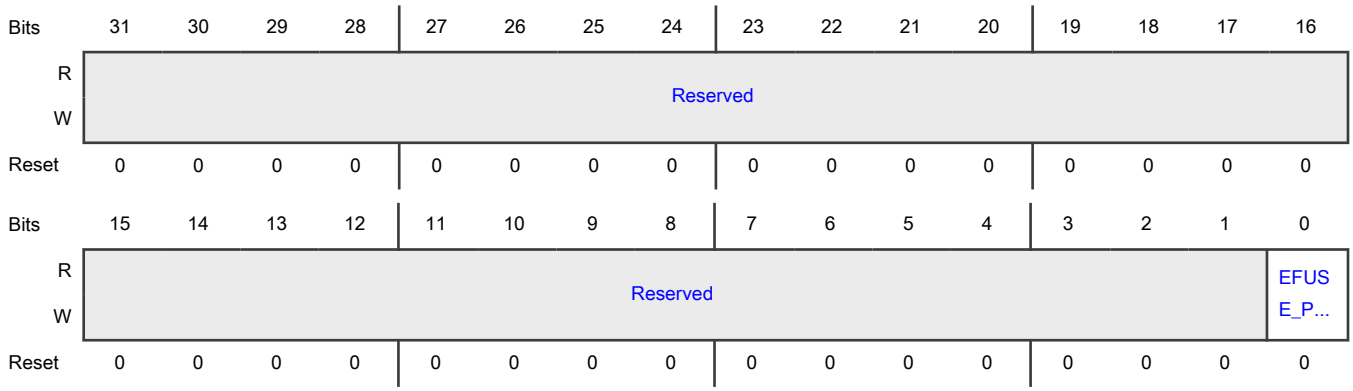
31.5.2.11 eFUSE control (GPC_EFUSE_CTRL)

.

Offset

Register	Offset
GPC_EFUSE_CTRL	208h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 EFUSE_PD_EN	eFUSE power down enable

Chapter 32
System Reset Controller (SRC)

32.1 Overview

The System Reset Controller (SRC) is responsible for the generation of all the system reset signals and boot argument latching. Its main functions are as follows:

- Deals with all global system reset sources from other modules, and generates global system reset.
- Responsible for power gating of MIXs (Slices) and their memory low power control.

The SRC takes the POR_B from the PAD and fuse bits to complete the boot sequence, and the GPC low power request to complete the power down/up sequence.

32.1.1 Block diagram

The SRC module consists of the following submodules:

- Cold reset control: This submodule is used to process the cold reset source, such as the watchdog, the tempsensor, or the cm33_lockup. These cold reset sources can assert system reset or single mix reset.
- src_signals_driver_ckil: This sub module uses POR_B from the pad and the jtag_rst_b signal to generate the system reset signal (src_system_early_rst_b) and the MIX SLICE reset signal (slice_early_rst_b).
- MIX SLICES: There are 13 MIX SLICES. Each slice corresponds to one resource on the chip. Each slice consists a set of controllers, such as the reset controller and the PG (power gating) controller. See Figure 72 for additional information.
- MEM SLICES: There are 13 MEM SLICES. Each slice corresponds to one memory on the chip.

The figure below shows a high-level block diagram of the SRC.

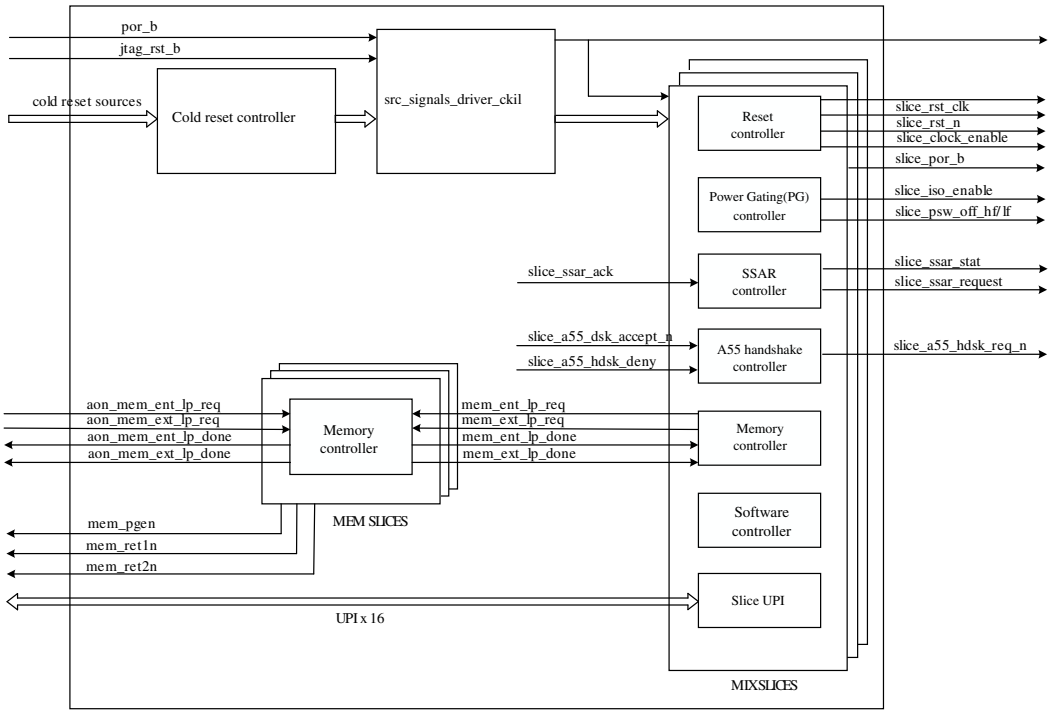


Figure 72. SRC block diagram

32.1.2 Features

The SRC includes the following features:

- Supports two boot modes
- Latches the SRC_BOOT_MODE pins and boot configuration signals from the on-chip Fuse
- Supports system cold reset
- Supports reset control, power control, and memory power control for all MIXs
- Supports handshake with EdgeLock Enclave before a sensitive MIXs reset/power status is changed
- Supports A55 MIX core0/1 and cluster dependency
- Supports A55 MIX power handshake
- Domain access control

32.2 Functional description

The reset control determines the source and type of reset, and performs the necessary reset qualification and stretching sequences. Based on the type of reset, the reset logic generates the reset sequence for the entire chip.

The boot argument latching includes the BOOT_MODE pin, when the system comes out of reset.

32.2.1 MIX and MEM SLICES

There are 13 MIX Slices and 13 MEM Slices. Each slice corresponds to one resource or memory on the chip. The slice and resource assignment is provided below.

Table 150. MIX Slice Assignment

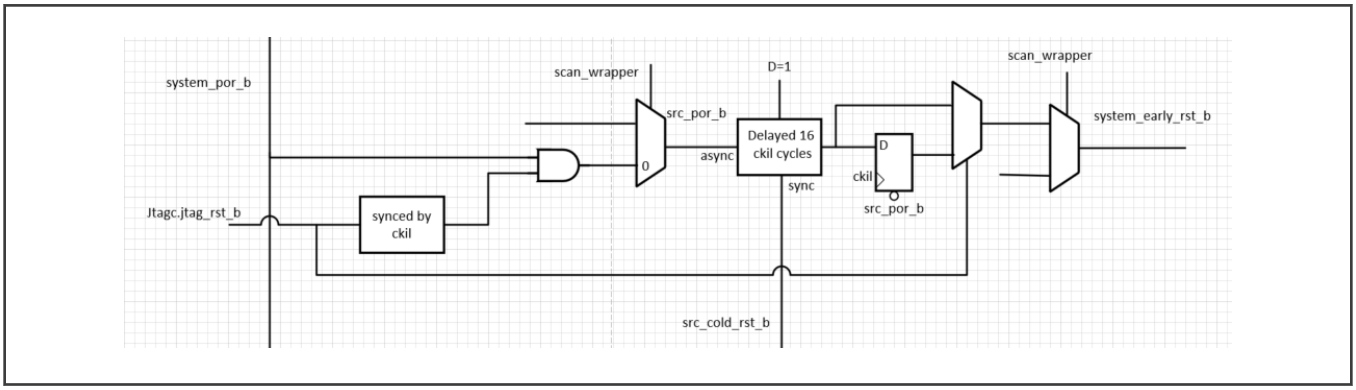
MIX SLICE index	Resource Name
0	EdgeLock Enclave
1	AONMIX
2	WAKEUPMIX
3	DDRMIX
4	DDRPHY
5	MLMIX
6	NICMIX
7	HSIOMIX
8	MEDIAMIX
9	CM33MIX
10	CA55 core 0
11	CA55 core 1
12	CA55 cluster

Table 151. MEM Slice Assignment

MEM SLICE Index	Memory Name
0	AONMIX memory
1	WAKEUPMIX memory
2	DDRMIX memory
3	DDRPHY memory
4	MLMIX
5	NICMIX memory
6	OCRAM
7	HSIOMIX memory
8	MEDIA memory
9	CA55 core 0 memory
10	CA55 core 1 memory
11	CA55 cluster memory
12	L3 cache

32.2.2 src_signals_driver_ckil

This submodule uses POR_B from the pad (system_por_b) and the jtag_rst_b signal to generate the system reset signal (system_early_rst_b).



32.2.3 Reset controller

The reset controller is used to generate reset signals for resources in the chip. It has four reset signals. The signals and the sequence is provided in the following figure.

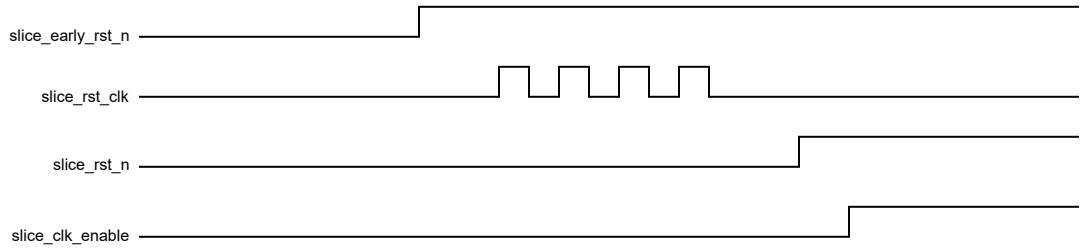


Figure 73. Reset Controller Sequence

Table 152. Reset Controller Signals

Signal	Description
slice_early_rst_n	This signal's release time is earlier than fuse load finished. Active low.
slice_rst_clk	Used for sync-reset logic.
slice_rst_n	Functional reset. Active low.
slice_clk_enable	Used for clock gating. It's release time is later than slice_rst_b, this means that there is no clock when slice_rst_b is releasing.

32.2.4 Power Gating (PG) controller

The power gating controller is used to generate power gating signals and isolation signals. The signals are described in the following table.

Table 153. PG Controller Signals

Signal	Description
slice_iso_enable	When it is high, the mix output signals are clamped to be default value.
slice_psw_off_hf/lf	Power switch gating signal. Active high.
slice_psw_ack_hf/lf	Power switch ack signal. Active high.

32.2.5 SSAR controller

The SRC supports the handshake mechanism with EdgeLock Enclave before executing power off, and after completing power up.

There are 3 signals for each slice. These signals are described in the following table.

Table 154. SSAR Controller Signals

Signal	Direction	Description
ssar_req	Output	Assert to send a notification or request to EdgeLock Enclave.
ssar_stat	Output	Indicates low power status: 0 - indicates power off, 1 - indicates power on
ssar_ack	Input	Assert to indicate that EdgeLock Enclave has gotten the notification or accepted the request.

The Waveform for one slice interface is provided below.

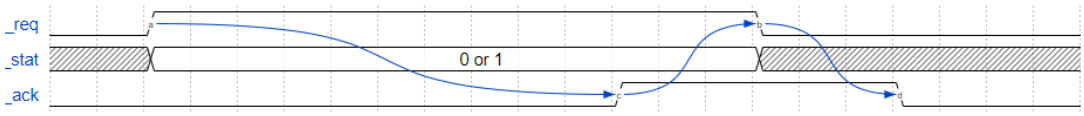


Figure 74. Waveform

32.2.6 A55 handshake controller

The SRC supports the handshake mechanism with A55 before executing A55 power off, and after completing A55 power up.

There are 3 signals for each slice and are described in the following table. These signals can only be sent out by the A55 core0, A55 core1, and A55 cluster slices.

Table 155. SSAR Controller Signals

Signal	Direction	Description
a55_hdsk_qreq_n	Output	Assert to send a notification or request to A55.
a55_hdsk_qaccept_n	Input	Assert to indicate that A55 has accepted the request.
a55_hdsk_qdeny	Input	Assert to indicate that A55 has rejected the request.

32.2.7 Memory controller

Each memory slice can control one memory block on the chip. There are two memory types.

- AON memories (AONMIX memory/ROM, HSIO memory, DDR PHY Memory): these are controlled by the GPC dsm.
- PSW memories: these are controlled by the GPC low power memory step.

32.2.8 Software controller

Software can control each SRC slice controller by configuring the SLICE_SW_CTRL[10:0] bits. SLICE_SW_CTRL[PDN_SOFT] is used for one-key power down/power up.

NOTE

Software access can be active only when AUTHEN_CTRL[LPM_MODE] = 0.

There are two methods to power up and power down a SRC slice.

1. One-key power down/ power up method
2. Step by step power down/ power up method

The one-key power down/power up method is provided below:

- To power off the SRC slice:
    1. Set MEM_CTRL[MEM_LP_EN] = 1, to enable MEM low power control.
    2. SET SLICE_SW_CTRL[PDN_SOFT] = 1, for software power down.
- To power on the SRC slice:
    1. Set MEM_CTRL[MEM_LP_EN] = 1, to enable MEM low power control.

2. Set SLICE_SW_CTRL[PDN_SOFT] = 0, for software power up.

NOTE

1. For initial power up, the register should be configured to an unsensitive value.
2. For initial power up of the CA55, SCR[BT_RELEASE_A55_PLAT] should set to 1.

The step by step power down/power up method is provided below:

- To power off the SRC slice:
  1. Set SLICE_SW_CTRL[ISO_CTRL_SOFT] = 1, for Software isolation on.
  2. Set MEM_CTRL[MEM_LP_EN] = 1, to enable MEM low power control.
  3. Set MEM_CTRL[SW_MEM_CTRL] = 1, for software control MEM to enter low power.
  4. Set SLICE_SW_CTRL[RST_CTRL_SOFT] = 1, reset and clock gate enable.
  5. Set SLICE_SW_CTRL[PSW_CTRL_S OFT] = 1, for software power switch off.
- To power on the SRC slice:
  1. SLICE_SW_CTRL[PSW_CTRL_S OFT] = 0, for software power switch on.
  2. MEM_CTRL[MEM_LP_EN] = 1, to enable MEM low power control.
  3. MEM_CTRL[SW_MEM_CTRL] = 0, for software control MEM to exit low power.
  4. SLICE_SW_CTRL[RST_CTRL_SO FT] = 0, reset release and clock gate disable.
  5. SLICE_SW_CTRL[ISO_CTRL_SO FT] = 0, for software isolation off.

The MIX slice status can be monitored via the Function Status (FUNC_STAT) register. The MEM slice status can be monitored via the MEM Status (MEM_STAT) register. The status must be monitored at each step when using the "step by step power down/power up method" listed above.

Software can control the memories power by configuring the MEM Low Power Control (MEM_CTRL) register.

32.2.9 Switching between hardware control and software control

When switching from software control to hardware control, software needs to ensure that the resource is ON. LPM_SETTING should be configured to common value, not always on or always off.

32.2.10 Cold reset controller

The table below shows the cold reset sources. Each cold reset source can assert system reset or single MIX reset via the SRMRn registers, and can be masked via the SRMASK register. The are two trigger modes available and can be selected through the SRTMR register.

Table 156. Cold Reset Sources

Reset sources	Description
csu_reset_b	Asserts when security violation happens
ipp_user_reset_b	Asserts when the ONOFF pad is pressed down in test mode
wdog1_cm33_rst_b	Asserts when the peripheral WDOG1 timeouts
wdog2_cm33_rst_b	Asserts when the peripheral WDOG2 timeouts
wdog3_wakeupmix_rst_b	Asserts when the peripheral WDOG3 timeouts
wdog4_ca55_rst_b	Asserts when the peripheral WDOG4 timeouts

Table continues on the next page...

Table 156. Cold Reset Sources (continued)

Reset sources	Description
wdog5_ca55_rst_b	Asserts when the peripheral WDOG5 timeouts
tempsense_reset	Asserts when the Temper Sensor requests a reset
m33_lockup	Asserts when CM33 lock up
m33_sysrstreq	Asserts when CM33 system resets
Jtag_sw_rst	Assert when the debugger asserts software reset by JTAG
jtag_rst_b	Asserts when the JTAG requests a reset

32.2.11 Boot sequence

The Boot MODE from the pins can be latched in the SRC SBMR2 register. BOOT_MODE[3] indicates the boot core. If BOOT_MODE[3]=1, CM33 is the boot core. If BOOT_MODE[3]=0, CA55 is the boot core. The MIXs boot status varies depending on the boot core selected. Refer to the table below for details.

MIX SLICE Index	Resource Name	CM33 boot	CA55 boot
0	EdgeLock Enclave	ON	ON
1	AONMIX	ON	ON
2	WAKEUPMIX	ON	ON
3	DDRMIX	OFF	ON
4	DDRPHY	ON	ON
5	MLMIX	OFF	OFF
6	NICMIX	ON	ON
7	HSIOMIX	ON	ON
8	MEDIAMIX	OFF	OFF
9	CM33MIX	ON	ON
10	CA55 Core 0	OFF	ON
11	CA55 Core 1	OFF	ON
12	CA55 Cluster	OFF	ON

All MIXs are by default OFF before the fuse loading is completed. After fuse loading is completed, the MIXs will power up or still be in OFF state depending on the FUSE or BOOT_MODE pin values. The power up sequence is as follows:

1. MIX power on
2. Memory power on
3. Reset release
4. iso_en = 0
5. Clock enable
6. EdgeLock Enclave handshake
7. System boot up

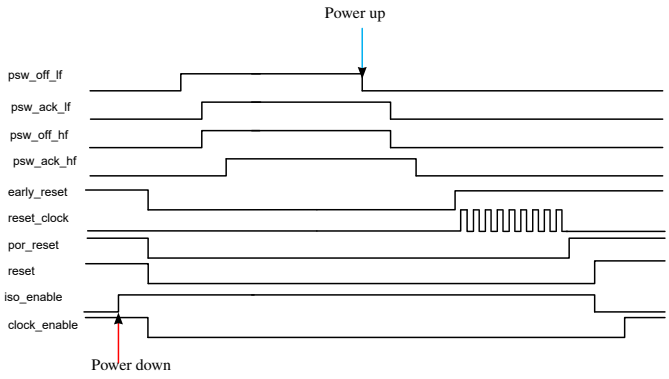


Figure 75. Power Up and Power down Sequence

32.2.12 Reset control

32.2.12.1 CM33 safe reset

There are two CM33 safe reset methods:

- Method 1: Safe reset by HW. The process is as follows.
  1. (SW) Set the M33_CFG[WAIT] bit in blk_ctrl_s_aonmix block. Refer to blk_ctrl_s_aonmix chapter for register details.
  2. (SW) Configure one of the SRC SRMRn register to 'h0100 for CM33
  3. Cold reset occurs
  4. (HW) SRC performs handshake with EgdeLock Enclave through the lp_hdsk_gasket.
  5. (HW) SRC performs safe reset handshake with CM33
  6. (HW) SRC asserts CM33 reset
  7. (CA55 SW) Check CM33 SRC slice FUNC_STAT[RST_STAT] bit for reset state.
  8. (CA55 SW) Load image to the TCM of the CM33
  9. (CA55 SW) Clear M33_CFG[WAIT] bit in the blk_ctrl_s_aonmix block. Refer to blk_ctrl_s_aonmix chapter for register details.
- Method 2: Safe reset by SW access safe reset register. The process is as follows:
  1. (SW) Set the M33_CFG[WAIT] bit in blk_ctrl_s_aonmix block. Refer to blk_ctrl_s_aonmix chapter for register details.
  2. (SW) Configure the SRC SRMRn register to 'h0100 for CM33
  3. (SW) SRC masks cold reset
  4. Cold reset occurs and generates an interrupt to the CA55
  5. (SW) CA55 performs handshake with Edgelock Enclave through the MU
  6. (SW) Set SRC CM_QUIESCE[SAFE_CM33_AHB_REQ] bit to request CM33 platform to be reset at safe state
  7. (SW) Wait for SRC CM_QUIESCE[SAFE_CM33_AHB_ACK_STAT] CM33 safe reset ack
  8. (SW) Set CM33 SRC slice SLICE_SW_CTRL[RST_CTRL_SOFT] bit for software reset assert
  9. (SW) Check CM33 SRC slice FUNC_STAT[RST_STAT] bit for reset state
  10. (SW) Clear SRC CM_QUIESCE[SAFE_CM33_AHB_REQ] bit
  11. (SW) Check SRC CM_QUIESCE[SAFE_CM33_AHB_ACK_STAT] CM33 safe reset state

12. (SW) Clear CM33 SRC slice SLICE_SW_CTRL[RST_CTRL_SOFT] reset bit
13. (SW) Check CM33 SRC slice FUNC_STAT[RST_STAT] bit for reset state
14. (CA55 SW) Load image to the TCM of the CM33
15. (CA55 SW) Clear M33_CFG[WAIT] bit in the blk_ctrl_s_aonmix block. Refer to blk_ctrl_s_aonmix chapter for register details.

32.2.12.2 Reset CA55 except debug component

The CA55 cannot be warm reset. The following power down and power up sequence is required to reset the CA55.

1. Power down CA55.
2. Set CA55_CPUWAIT[CPUn_WAIT] bit to 1 in the blk_ctrl_s_aonmix block. Once the CPUn_WAIT bit is set to 1, the CA55 core stops working.
3. Power up CA55.
4. The CA55 core will continue to stop working, until the WAIT reset is released.
5. Set CA55_CPUWAIT[CPUn_WAIT] bit to 0 in the blk_ctrl_s_aonmix block. Once the CPUn_WAIT bit is set to 0, the CA55 core works normally.

32.2.13 Low-power mode transition

All SRC slices can be powered off. A fixed power off/on sequence is adopted in this chip. Refer to Figure 75 for the power up and power down sequence.

32.2.13.1 Control of slice power-off

There are two ways to power off a slice.

- GPC mode (by setting AUTHEN_CTRL[LPM_MODE] = 1)
- Software (SW) mode (by setting AUTHEN_CTRL[LPM_MODE] = 0). Refer to Software controller for additional details.

NOTE

Do not change AUTHEN_CTRL[LPM_MODE] value, when the slice is in power off/on sequence or is already in power off state.

In the GPC mode, the SRC powers off the slice under the control of the GPC. Each Core can be mapped to any of the 16 domains by the TRDC via GPC_CPUn_DOMAIN_ASSIGNMENT register. In SRC, each slice can be assigned to any of the 16 domains via AUTHEN_CTRL[WHITE_LIST] bits. When the bit is set, the domain's mapped Core's power state transition is considered as a low power trigger, else the transition is ignored.

NOTE

The CPU3_domain cannot assign to CA55 core 0 slice and CA55 core 1 slice domain.

Because a slice can be assigned to 16 domains at most, a unified arbitration rule is defined for each slice.

- The slice enters power-off sequence with the last active Core power state transition, if several cores (of domain's assigned domains' mapping Cores) request the slice to power off.
- The slice enters power-on sequence with the first active Core power state transition, if any one core (of domain's assigned domains' mapping Cores) requests the slice to power on while the slice is in power-off status.

For example, the MLMIX slice can be assigned to the CM33 Core. After CM33 core requests domain to power off/on, CM33 domain enters power off/on sequence with CM33 Core power state transition. For comparison, the NICMIX slice can be assigned to both CM55 Core0 and CM33 Core. If only one Core requests the slice to power off, the NICMIX slice keeps in power on status. If CM55 Core0 requests the slice to power off, and CM33 requests the slice to power off later, then the NICMIX slice enters power-off sequence with CM33 Core's power state transition. When NICMIX slice is already in power-off status, after any Core requests the slice to power on, the NICMIX slice enters power-on sequence with the Core's power state transition.

NOTE

If the slice enters power-off/on sequence, it goes through all the steps[1] and cannot be interrupted. So during the power-off sequence, any Core's power-on request has to be delayed until the completion of slice's power-off sequence. During the power-on sequence, any Core's power-off request has to be delayed until the completion of slice's power-on sequence.

32.2.14 Clocks

The table below describes the relevant SRC clock sources.

Table 157. Clocks

Clock	Description
ipg_clk	Main SRC clock. This clock originates from the 24MHz RCOSC and can be turned off.
32k_clk	32KHz clock.

32.3 External Signals

This table describes the SRC external signals.

Table 158. External Signals

Signal	Direction	Description
POR_B	I	Power on reset input pin, active low.
BOOT_MODE[3:0]	I	Boot mode selection pins. The values are sampled during system reset and can be read in the SBMR2 register. See the boot chapter for specific boot configuration settings, and the pin mux chapter for pin assignments.

32.4 Register access permissions

The SRC uses the AUTHEN_CTRL register to control register access permissions.

- The AUTHEN_CTRL[WHITE_LIST] field is used to configure which domain IDs can write to the registers.
- The AUTHEN_CTRL[TZ_USER] and AUTHEN_CTRL[TZ_NS] fields configure privilege and security permissions.
- The AUTHEN_CTRL[LOCK_CFG] field locks configuration fields which should only be set at initialization.

32.5 Memory Map and register definition

This section includes the SRC module memory map and detailed descriptions of all registers.

32.5.1 SRC GENERAL register descriptions

32.5.1.1 SRC General Registers memory map

SRC General Registers

src_general_reg base address: 4446_0000h

[1] See the "Low-power mode transition" section.

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	Authentication Control (AUTHEN_CTRL)	32	RW	FFFF_0000
10	SRC Control Register (SCR)	32	RW	0000_0000
14	SRC RESET TRIGGER MODE REGISTER (SRTMR)	32	RW	0000_0000
18	SRC RESET TRIGGER MASK REGISTER (SRMASK)	32	RW	0000_0FBF
20	SRC RESET MODE REGISTER 1 (SRMR1)	32	RW	0000_0000
24	SRC RESET MODE REGISTER 2 (SRMR2)	32	RW	0000_0000
28	SRC RESET MODE REGISTER 3 (SRMR3)	32	RW	0000_0000
2C	SRC RESET MODE REGISTER 4 (SRMR4)	32	RW	0000_0000
30	SRC RESET MODE REGISTER 5 (SRMR5)	32	RW	0000_0000
40	SRC Boot Mode Register 1 (SBMR1)	32	R	0000_0000
44	SRC Boot Mode Register 2 (SBMR2)	32	R	0000_0000
50	SRC Reset Status Register (SRSR)	32	RW	0000_0000
54	SRC General Purpose Register 1 (GPR1)	32	RW	0000_0000
58	SRC General Purpose Register 2 (GPR2)	32	RW	0000_0000
5C	SRC General Purpose Register 3 (GPR3)	32	RW	0000_0000
60	SRC General Purpose Register 4 (GPR4)	32	RW	0000_0000
64	SRC General Purpose Register 5 (GPR5)	32	RW	0000_0000
68	SRC General Purpose Register 6 (GPR6)	32	RW	0000_0000
6C	SRC General Purpose Register 7 (GPR7)	32	RW	0000_0000
70	SRC General Purpose Register 8 (GPR8)	32	RW	0000_0000
74	SRC General Purpose Register 9 (GPR9)	32	RW	0000_0000
78	SRC General Purpose Register 10 (GPR10)	32	RW	0000_0000
7C	SRC General Purpose Register 11 (GPR11)	32	RW	0000_0000
80	SRC General Purpose Register 12 (GPR12)	32	RW	0000_0000
84	SRC General Purpose Register 13 (GPR13)	32	RW	0000_0000
88	SRC General Purpose Register 14 (GPR14)	32	RW	0000_0000
8C	SRC General Purpose Register 15 (GPR15)	32	RW	0000_0000
90	SRC General Purpose Register 16 (GPR16)	32	RW	0000_0000
94	SRC General Purpose Register 17 (GPR17)	32	RW	0000_0000
98	SRC General Purpose Register 18 (GPR18)	32	RW	0000_0000
9C	SRC General Purpose Register 19 (GPR19)	32	RW	0000_0000
100	SRC General Purpose Register 20 (GPR20)	32	RW	0000_0600

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
104	SRC_CORTEX_M_QUIESCE (CM_QUIESCE)	32	RW	0000_0000
108	Cold reset SSAR acknowledge control (COLD_RESET_SSAR_ACK_CTRL)	32	RW	0000_2EE0
10C	SRC special ISO Control (SP_ISO_CTRL)	32	RW	0000_000F
110	ROM Low Power Control (ROM_LP_CTRL)	32	RW	0000_0000
114	A55 Q_Channel Deny Status (A55_DENY_STAT)	32	R	0000_0000

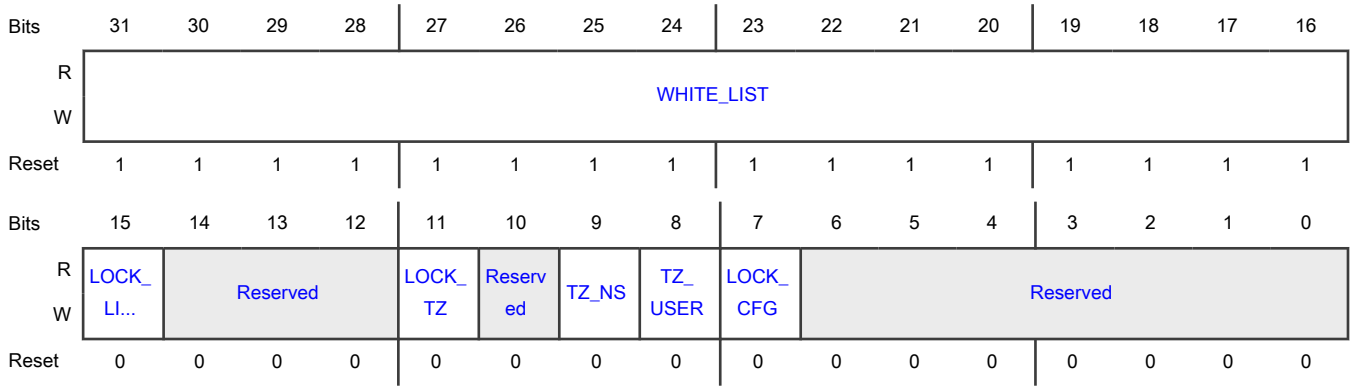
32.5.1.2 Authentication Control (AUTHEN_CTRL)

Authentication Control

Offset

Register	Offset
AUTHEN_CTRL	4h

Diagram



Fields

Field	Description
31-16 WHITE_LIST	Domain ID white list Each master(Core or non-Core) can be assigned to any domains among 0-15 through TRDC and through GPC domain assignment register. Once its domains are assigned in TRDC, then its accesses will have its unique domain ID. Each bit in this field represent for one domain. Bit 16-31 represent domains 0-15 respectively. Only the TRDC domains whose corresponding bits are 1 can write to this register. Write violation will fail without any exception or information. Read violation can still get read

Table continues on the next page...

Table continued from the previous page...

Field	Description
	data. In LPM mode, only the GPC domains whose corresponding bits are 1 can trigger the powering down of the given power slice. 0000_0000_0000_0001 - Core with domain ID=0 can write General registers. 0000_0000_0000_0010 - Core with domain ID=1 can write General registers. 0000_0000_0000_0100 - Core with domain ID=2 can write General registers. 0000_0000_0000_1000 - Core with domain ID=3 can write General registers. 0000_0000_0001_0000 - Core with domain ID=4 can write General registers. 0000_0000_0010_0000 - Core with domain ID=5 can write General registers. 0000_0000_0100_0000 - Core with domain ID=6 can write General registers. 0000_0000_1000_0000 - Core with domain ID=7 can write General registers. 0000_0001_0000_0000 - Core with domain ID=8 can write General registers. 0000_0010_0000_0000 - Core with domain ID=9 can write General registers. 0000_0100_0000_0000 - Core with domain ID=10 can write General registers. 0000_1000_0000_0000 - Core with domain ID=11 can write General registers. 0001_0000_0000_0000 - Core with domain ID=12 can write General registers. 0010_0000_0000_0000 - Core with domain ID=13 can write General registers. 0100_0000_0000_0000 - Core with domain ID=14 can write General registers. 1000_0000_0000_0000 - Core with domain ID=15 can write General registers.
15 LOCK_LIST	White list lock This field can lock itself and WHITE_LIST. 0 - WHITE_LIST value can be changed. 1 - LOCK_LIST and WHITE_LIST value cannot be changed.
14-12 —	Reserved
11 LOCK_TZ	Lock TZ_NS and TZ_USER bits This field can lock itself, TZ_NS and TZ_USER. 0 - TZ_NS and TZ_USER value can be changed. 1 - LOCK_TZ, TZ_NS and TZ_USER value cannot be changed.
10 —	Reserved
9 TZ_NS	Allow non-secure mode access Allow either secure mode or non-secure mode to write general registers.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - General registers can only be written in secure mode. 1 - General registers can be written either in secure mode or non-secure mode.
8 TZ_USER	Allow user mode access Allow either privilege mode or user mode to write general registers. 0 - General registers can only be written in privilege mode. 1 - General registers can be written either in privilege mode or user mode.
7 LOCK_CFG	Configuration lock This field can lock itself and some registers in this General registers. They SRTMR, SRMASK, and SRMRn. 0 - General registers are not locked. 1 - LOCK_CFG and registers in the list are locked.
6-0 —	Reserved

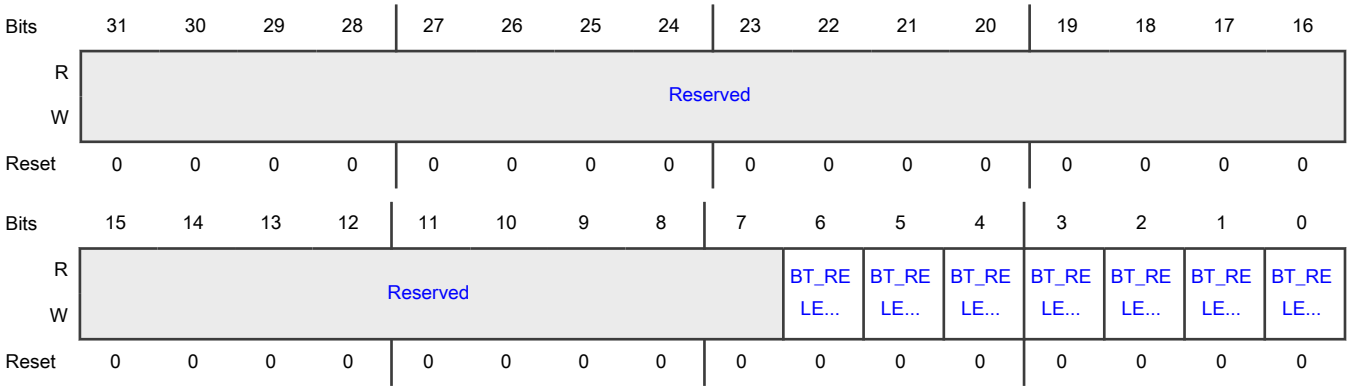
32.5.1.3 SRC Control Register (SCR)

The Reset control register (SCR), contains bits that control operation of the reset controller.

Offset

Register	Offset
SCR	10h

Diagram



Fields

Field	Description
31-7 —	Reserved
6 BT_RELEASE_DDR	If M33 boot, the DDRMIX reset will be held until boot core writes this bit to 1. Once this bit is set to 1, it will be locked. 0 - Hold reset. 1 - Release reset.
5 BT_RELEASE_MEDIA	MEDIAMIX reset will be held until boot core writes this bit to 1. Once this bit is set to 1, it will be locked. 0 - Hold reset. 1 - Release reset.
4 BT_RELEASE_ML	MLMIX reset will be held until boot core writes this bit to 1. Once this bit is set to 1, it will be locked. 0 - Hold reset. 1 - Release reset.
3 BT_RELEASE_A55_PLAT	If M33 boot, the CA55 platform reset will be held until boot core writes this bit to 1. Once this bit is set to 1, it will be locked. 0 - Hold reset. 1 - Release reset.
2 BT_RELEASE_A55_CPU1	If M33 boot, the CA55 CPU1 reset will be held until boot core writes this bit to 1. Once this bit is set to 1, it will be locked. 0 - Hold reset. 1 - Release reset.
1 BT_RELEASE_A55_CPU0	If M33 boot, the CA55 CPU0 reset will be held until boot core writes this bit to 1. Once this bit is set to 1, it will be locked. 0 - Hold reset. 1 - Release reset.
0 BT_RELEASE_M33	If A55 boot, the CM33 reset will be held until boot core writes this bit to 1. Once this bit is set to 1, it will be locked. 0 - Hold reset. 1 - Release reset.

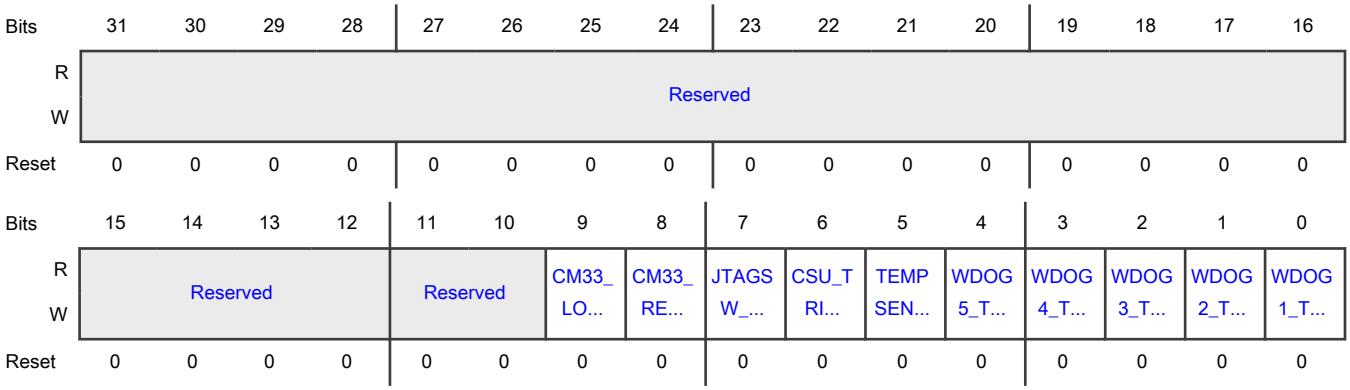
32.5.1.4 SRC RESET TRIGGER MODE REGISTER (SRTMR)

The Reset Trigger Mode Register (SRTMR). Reset by POR not system early reset.

Offset

Register	Offset
SRTMR	14h

Diagram



Fields

Field	Description
31-12 —	Reserved
11-10 —	Reserved
9 CM33_LOCKUP_TRIG_MODE	CM33 lockup trigger mode configuration,locked by LOCK_CFG field 0 - Level-sensitive: System stays in reset until the reset source deasserts. 1 - Edge-sensitive: System resets once, even if the reset source remains asserted.
8 CM33_RESET_TRIG_MODE	CM33 reset trigger mode configuration,locked by LOCK_CFG field 0 - Level-sensitive: System stays in reset until the reset source deasserts. 1 - Edge-sensitive: System resets once, even if the reset source remains asserted.
7 JTAGSW_TRIG_MODE	JTAGSW reset trigger mode configuration,locked by LOCK_CFG field 0 - Level-sensitive: System stays in reset until the reset source deasserts. 1 - Edge-sensitive: System resets once, even if the reset source remains asserted.
6 CSU_TRIG_MODE	CSU reset trigger mode configuration,locked by LOCK_CFG field 0 - Level-sensitive: System stays in reset until the reset source deasserts. 1 - Edge-sensitive: System resets once, even if the reset source remains asserted.
5	Tempsense reset trigger mode configuration,locked by LOCK_CFG field 0 - Level-sensitive: System stays in reset until the reset source deasserts.

Table continues on the next page...

Table continued from the previous page...

Field	Description
TEMPSENSE_TRIG_MODE	1 - Edge-sensitive: System resets once, even if the reset source remains asserted.
4 WDOG5_TRIG_MODE	WDOG5 reset trigger mode configuration,locked by LOCK_CFG field 0 - Level-sensitive: System stays in reset until the reset source deasserts. 1 - Edge-sensitive: System resets once, even if the reset source remains asserted.
3 WDOG4_TRIG_MODE	WDOG4 reset trigger mode configuration,locked by LOCK_CFG field 0 - Level-sensitive: System stays in reset until the reset source deasserts. 1 - Edge-sensitive: System resets once, even if the reset source remains asserted.
2 WDOG3_TRIG_MODE	WDOG3 reset trigger mode configuration,locked by LOCK_CFG field 0 - Level-sensitive: System stays in reset until the reset source deasserts. 1 - Edge-sensitive: System resets once, even if the reset source remains asserted.
1 WDOG2_TRIG_MODE	WDOG2 reset trigger mode configuration,locked by LOCK_CFG field 0 - Level-sensitive: System stays in reset until the reset source deasserts. 1 - Edge-sensitive: System resets once, even if the reset source remains asserted.
0 WDOG1_TRIG_MODE	WDOG1 reset trigger mode configuration,locked by LOCK_CFG field 0 - Level-sensitive: System stays in reset until the reset source deasserts. 1 - Edge-sensitive: System resets once, even if the reset source remains asserted.

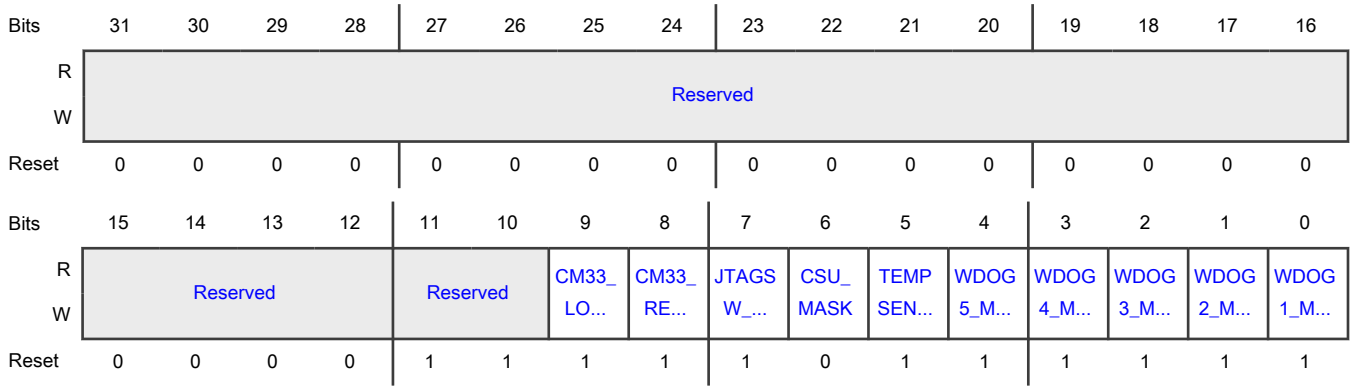
32.5.1.5 SRC RESET TRIGGER MASK REGISTER (SRMASK)

The Reset Trigger Mask Register (SRMASK). Reset by POR not system early reset

Offset

Register	Offset
SRMASK	18h

Diagram



Fields

Field	Description
31-12 —	Reserved
11-10 —	Reserved
9 CM33_LOCKUP_MASK	CM33 lockup mask, locked by LOCK_CFG field. 0 - The cold reset source can work. 1 - The cold reset source is masked and cannot work.
8 CM33_RESET_MASK	CM33 reset mask, locked by LOCK_CFG field. 0 - The cold reset source can work. 1 - The cold reset source is masked and cannot work.
7 JTAGSW_MASK	JTAG SW reset mask, locked by LOCK_CFG field. 0 - The cold reset source can work. 1 - The cold reset source is masked and cannot work.
6 CSU_MASK	CSU reset mask, locked by LOCK_CFG field. 0 - The cold reset source can work. 1 - The cold reset source is masked and cannot work.
5 TEMPSENSE_MASK	Tempsense reset mask, locked by LOCK_CFG field. 0 - The cold reset source can work. 1 - The cold reset source is masked and cannot work.
4 WDOG5_MASK	WDOG5 reset mask, locked by LOCK_CFG field. 0 - The cold reset source can work. 1 - The cold reset source is masked and cannot work.

Table continues on the next page...

Table continued from the previous page...

Field	Description
3 WDOG4_MASK	WDOG4 reset mask, locked by LOCK_CFG field. 0 - The cold reset source can work. 1 - The cold reset source is masked and cannot work.
2 WDOG3_MASK	WDOG3 reset mask, locked by LOCK_CFG field. 0 - The cold reset source can work. 1 - The cold reset source is masked and cannot work.
1 WDOG2_MASK	WDOG2 reset mask, locked by LOCK_CFG field. 0 - The cold reset source can work. 1 - The cold reset source is masked and cannot work.
0 WDOG1_MASK	WDOG1 reset mask, locked by LOCK_CFG field. 0 - The cold reset source can work. 1 - The cold reset source is masked and cannot work.

32.5.1.6 SRC RESET MODE REGISTER 1 (SRMR1)

The Reset Mode Register (SRMR), contains bits that control reset behavior of reset sources. Reset by POR not system early reset. Multiple domains can be reset by setting the bits to 1.

Offset

Register	Offset
SRMR1	20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WDOG2_RESET_MODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WDOG1_RESET_MODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 WDOG2_RESET_MODE	WDOG2 reset mode configuration, locked by LOCK_CFG field. 0000_0000_0000_0000 - system 0000_0000_0000_0001 - bit0: AONMIX 0000_0000_0000_0010 - bit1: WAKEUPMIX 0000_0000_0000_0100 - bit2: DDRMIX 0000_0000_0000_1000 - bit3: DRAM PHY 0000_0000_0001_0000 - MLMIX 0000_0000_0010_0000 - bit5: NICMIX 0000_0000_0100_0000 - bit6: HSIOMIX 0000_0000_1000_0000 - bit7: MEDIAMIX 0000_0001_0000_0000 - bit8: CM33_Platform 0000_0010_0000_0000 - bit9: CA55_CPU0 0000_0100_0000_0000 - bit10: CA55_CPU1 0000_1000_0000_0000 - bit11: CA55_Platform
15-0 WDOG1_RESET_MODE	WDOG1 reset mode configuration, locked by LOCK_CFG field. 0000_0000_0000_0000 - system 0000_0000_0000_0001 - bit0: AONMIX 0000_0000_0000_0010 - bit1: WAKEUPMIX 0000_0000_0000_0100 - bit2: DDRMIX 0000_0000_0000_1000 - bit3: DRAM PHY 0000_0000_0001_0000 - MLMIX 0000_0000_0010_0000 - bit5: NICMIX 0000_0000_0100_0000 - bit6: HSIOMIX 0000_0000_1000_0000 - bit7: MEDIAMIX 0000_0001_0000_0000 - bit8: CM33_Platform 0000_0010_0000_0000 - bit9: CA55_CPU0 0000_0100_0000_0000 - bit10: CA55_CPU1 0000_1000_0000_0000 - bit11: CA55_Platform

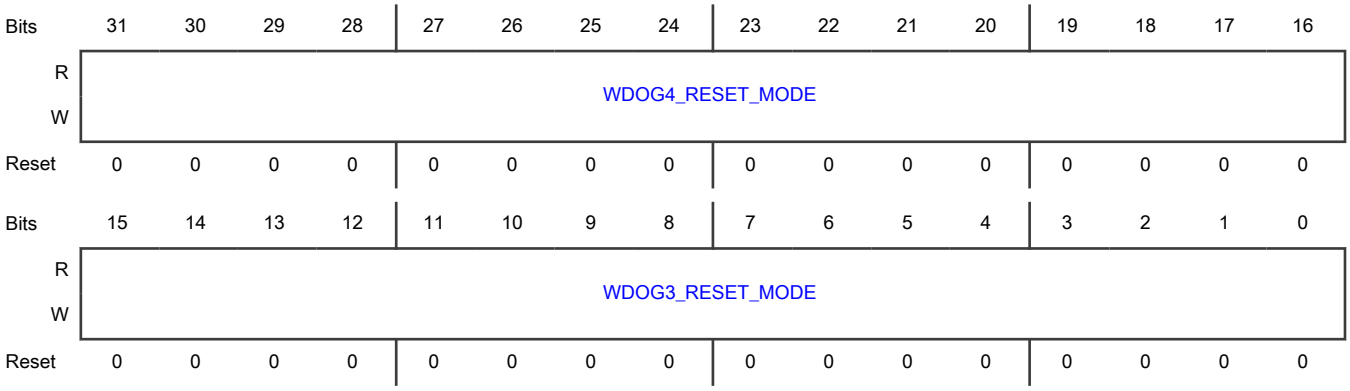
32.5.1.7 SRC RESET MODE REGISTER 2 (SRMR2)

The Reset Mode Register (SRMR), contains bits that control reset behavior of reset sources. Reset by POR not system early reset. Multiple domains can be reset by setting the bits to 1.

Offset

Register	Offset
SRMR2	24h

Diagram



Fields

Field	Description
31-16 WDOG4_RESET_MODE	WDOG4 reset mode configuration, locked by LOCK_CFG field. 0000_0000_0000_0000 - system 0000_0000_0000_0001 - bit0: AONMIX 0000_0000_0000_0010 - bit1: WAKEUPMIX 0000_0000_0000_0100 - bit2: DDRMIX 0000_0000_0000_1000 - bit3: DRAM PHY 0000_0000_0001_0000 - MLMIX 0000_0000_0010_0000 - bit5: NICMIX 0000_0000_0100_0000 - bit6: HSIOMIX 0000_0000_1000_0000 - bit7: MEDIAMIX 0000_0001_0000_0000 - bit8: CM33_Platform 0000_0010_0000_0000 - bit9: CA55_CPU0 0000_0100_0000_0000 - bit10: CA55_CPU1 0000_1000_0000_0000 - bit11: CA55_Platform
15-0 WDOG3_RESET_MODE	WDOG3 reset mode configuration, locked by LOCK_CFG field. 0000_0000_0000_0000 - system 0000_0000_0000_0001 - bit0: AONMIX 0000_0000_0000_0010 - bit1: WAKEUPMIX 0000_0000_0000_0100 - bit2: DDRMIX

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0000_0000_0000_1000 - bit3: DRAM PHY 0000_0000_0001_0000 - MLMIX 0000_0000_0010_0000 - bit5: NICMIX 0000_0000_0100_0000 - bit6: HSIOMIX 0000_0000_1000_0000 - bit7: MEDIAMIX 0000_0001_0000_0000 - bit8: CM33_Platform 0000_0010_0000_0000 - bit9: CA55_CPU0 0000_0100_0000_0000 - bit10: CA55_CPU1 0000_1000_0000_0000 - bit11: CA55_Platform

32.5.1.8 SRC RESET MODE REGISTER 3 (SRMR3)

The Reset Mode Register (SRMR), contains bits that control reset behavior of reset sources. Reset by POR not system early reset. Multiple domains can be reset by setting the bits to 1.

Offset

Register	Offset
SRMR3	28h

Diagram

Fields

Field	Description
31-16 TEMPSENSE_RESET_MODE	Tempsense reset mode configuration, locked by LOCK_CFG field. 0000_0000_0000_0000 - system 0000_0000_0000_0001 - bit0: AONMIX

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0000_0000_0000_0010 - bit1: WAKEUPMIX 0000_0000_0000_0100 - bit2: DDRMIX 0000_0000_0000_1000 - bit3: DRAM PHY 0000_0000_0001_0000 - MLMIX 0000_0000_0010_0000 - bit5: NICMIX 0000_0000_0100_0000 - bit6: HSIOMIX 0000_0000_1000_0000 - bit7: MEDIAMIX 0000_0001_0000_0000 - bit8: CM33_Platform 0000_0010_0000_0000 - bit9: CA55_CPU0 0000_0100_0000_0000 - bit10: CA55_CPU1 0000_1000_0000_0000 - bit11: CA55_Platform
15-0 WDOG5_RESET_MODE	WDOG5 reset mode configuration, locked by LOCK_CFG field. 0000_0000_0000_0000 - system 0000_0000_0000_0001 - bit0: AONMIX 0000_0000_0000_0010 - bit1: WAKEUPMIX 0000_0000_0000_0100 - bit2: DDRMIX 0000_0000_0000_1000 - bit3: DRAM PHY 0000_0000_0001_0000 - MLMIX 0000_0000_0010_0000 - bit5: NICMIX 0000_0000_0100_0000 - bit6: HSIOMIX 0000_0000_1000_0000 - bit7: MEDIAMIX 0000_0001_0000_0000 - bit8: CM33_Platform 0000_0010_0000_0000 - bit9: CA55_CPU0 0000_0100_0000_0000 - bit10: CA55_CPU1 0000_1000_0000_0000 - bit11: CA55_Platform

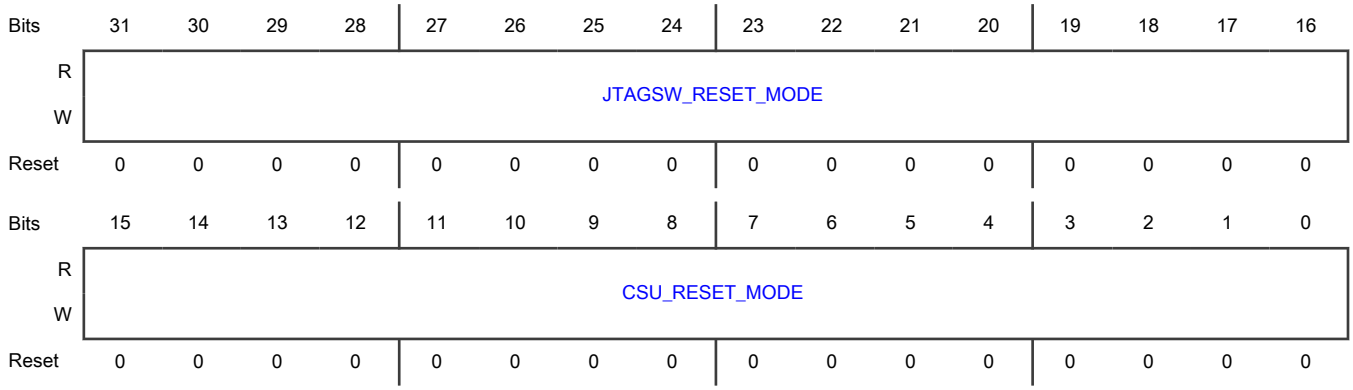
32.5.1.9 SRC RESET MODE REGISTER 4 (SRMR4)

The Reset Mode Register (SRMR), contains bits that control reset behavior of reset sources. Reset by POR not system early reset. Multiple domains can be reset by setting the bits to 1.

Offset

Register	Offset
SRMR4	2Ch

Diagram



Fields

Field	Description
31-16 JTAGSW_RESET_MODE	JTAG SW reset mode configuration, locked by LOCK_CFG field. 0000_0000_0000_0000 - system 0000_0000_0000_0001 - bit0: AONMIX 0000_0000_0000_0010 - bit1: WAKEUPMIX 0000_0000_0000_0100 - bit2: DDRMIX 0000_0000_0000_1000 - bit3: DRAM PHY 0000_0000_0001_0000 - MLMIX 0000_0000_0010_0000 - bit5: NICMIX 0000_0000_0100_0000 - bit6: HSIOMIX 0000_0000_1000_0000 - bit7: MEDIAMIX 0000_0001_0000_0000 - bit8: CM33_Platform 0000_0010_0000_0000 - bit9: CA55_CPU0 0000_0100_0000_0000 - bit10: CA55_CPU1 0000_1000_0000_0000 - bit11: CA55_Platform
15-0 CSU_RESET_MODE	CSU reset mode configuration, locked by LOCK_CFG field. 0000_0000_0000_0000 - system 0000_0000_0000_0001 - bit0: AONMIX 0000_0000_0000_0010 - bit1: WAKEUPMIX 0000_0000_0000_0100 - bit2: DDRMIX 0000_0000_0000_1000 - bit3: DRAM PHY 0000_0000_0001_0000 - MLMIX 0000_0000_0010_0000 - bit5: NICMIX 0000_0000_0100_0000 - bit6: HSIOMIX

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0000_0000_1000_0000 - bit7: MEDIAMIX 0000_0001_0000_0000 - bit8: CM33_Platform 0000_0010_0000_0000 - bit9: CA55_CPU0 0000_0100_0000_0000 - bit10: CA55_CPU1 0000_1000_0000_0000 - bit11: CA55_Platform

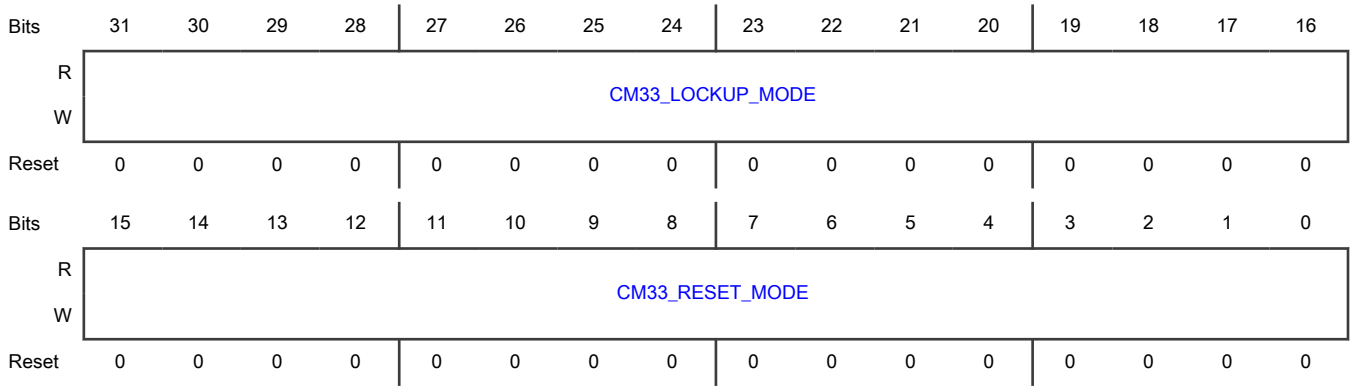
32.5.1.10 SRC RESET MODE REGISTER 5 (SRMR5)

The Reset Mode Register (SRMR), contains bits that control reset behavior of reset sources. Reset by POR not system early reset. Multiple domains can be reset by setting the bits to 1.

Offset

Register	Offset
SRMR5	30h

Diagram



Fields

Field	Description
31-16 CM33_LOCKUP_MODE	CM33 lockup reset mode configuration, locked by LOCK_CFG field. 0000_0000_0000_0000 - system 0000_0000_0000_0001 - bit0: AONMIX 0000_0000_0000_0010 - bit1: WAKEUPMIX 0000_0000_0000_0100 - bit2: DDRMIX 0000_0000_0000_1000 - bit3: DRAM PHY 0000_0000_0001_0000 - MLMIX

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0000_0000_0010_0000 - bit5: NICMIX 0000_0000_0100_0000 - bit6: HSIOMIX 0000_0000_1000_0000 - bit7: MEDIAMIX 0000_0001_0000_0000 - bit8: CM33_Platform 0000_0010_0000_0000 - bit9: CA55_CPU0 0000_0100_0000_0000 - bit10: CA55_CPU1 0000_1000_0000_0000 - bit11: CA55_Platform
15-0 CM33_RESET_MODE	CM33 sysrstreq reset mode configuration, locked by LOCK_CFG field. 0000_0000_0000_0000 - system 0000_0000_0000_0001 - bit0: AONMIX 0000_0000_0000_0010 - bit1: WAKEUPMIX 0000_0000_0000_0100 - bit2: DDRMIX 0000_0000_0000_1000 - bit3: DRAM PHY 0000_0000_0001_0000 - MLMIX 0000_0000_0010_0000 - bit5: NICMIX 0000_0000_0100_0000 - bit6: HSIOMIX 0000_0000_1000_0000 - bit7: MEDIAMIX 0000_0001_0000_0000 - bit8: CM33_Platform 0000_0010_0000_0000 - bit9: CA55_CPU0 0000_0100_0000_0000 - bit10: CA55_CPU1 0000_1000_0000_0000 - bit11: CA55_Platform

32.5.1.11 SRC Boot Mode Register 1 (SBMR1)

The Boot Mode register (SBMR) contains bits that reflect the status of Boot Mode Pins of the chip and some boot related fuse bits. Reset by POR not system early reset.

Offset

Register	Offset
SBMR1	40h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BOOT_CFG0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BOOT_CFG0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BOOT_CFG0	This bit field stores the BOOT_CFG8 fuse values. Please refer to the Fusemap for the fuse details.

32.5.1.12 SRC Boot Mode Register 2 (SBMR2)

The Boot Mode register (SBMR) contains bits that reflect the status of Boot Mode Pins of the chip and some boot related fuse bits. Reset by POR not system early reset.

Offset

Register	Offset
SBMR2	44h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DIT_BT_...	Reserved	IPP_BOOT_MODE						Reserved				Reserved		Reserved	SDP_DIS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BOOT_CFG1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 DIT_BT_DIS	Please see the fusemap for fuse details.
30 —	Reserved
29-24 IPP_BOOT_MODE	Boot mode from pins IPP_BOOT_MODE[3:0] shows the latched state of the BOOT_MODE[3:0] signals on the rising edge of POR_B. See the Boot Mode pins section of the System Boot chapter for additional details.
23-21 —	Reserved
20-18 —	Reserved
17 —	Reserved
16 SDP_DIS	Please see the fusemap for fuse details.
15-0 BOOT_CFG1	This bit field stores the BOOT_CFG0[15:0] fuse values. Please refer to the Fusemap for the fuse details.

32.5.1.13 SRC Reset Status Register (SRSR)

The SRSR is a write to one clear register which records the source of the reset events for the chip. Writing zero does not have any effect. Writing one will clear the corresponding bit. Reset by POR not system early reset.

Offset

Register	Offset
SRSR	50h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				A55PMIX...	A55C1MI...	A55C0MI...	M33PMIX...	MEDIAMI...	HSIOMIX...	NICMIX_...	MLMIX_R...	DDRPHY_...	DDRMIX_...	WAKEUPM...	AONMIX_...
W					W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved		CM33_LO...	CM33_RE...	JTAG_SW...	CSU_RES...	TEMPSEN...	WDOG5_R...	WDOG4_R...	WDOG3_R...	WDOG2_R...	WDOG1_R...	IPP_USE...	JTAG_RS...	SRC_POR...
W				W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27 A55PMIX_RST	Indicates whether a A55PMIX slice reset happens 0 - Power on reset or system cold reset happens. 1 - A55PMIX slice reset happens.
26 A55C1MIX_RST	Indicates whether a A55C1MIX slice reset happens 0 - Power on reset or system cold reset happens. 1 - A55C1MIX slice reset happens.
25 A55C0MIX_RST	Indicates whether a A55C0MIX slice reset happens 0 - Power on reset or system cold reset happens. 1 - A55C0MIX slice reset happens.
24 M33PMIX_RST	Indicates whether a M33PMIX slice reset happens 0 - Power on reset or system cold reset happens. 1 - M33PMIX slice reset happens.
23 MEDIAMIX_RST	Indicates whether a MEDIAMIX slice reset happens 0 - Power on reset or system cold reset happens. 1 - MEDIAMIX slice reset happens.
22 HSIOMIX_RST	Indicates whether a HSIOMIX slice reset happens 0 - Power on reset or system cold reset happens. 1 - HSIOMIX slice reset happens.

Table continues on the next page...

Table continued from the previous page...

Field	Description
21 NICMIX_RST	Indicates whether a NICMIX slice reset happens 0 - Power on reset or system cold reset happens. 1 - NICMIX slice reset happens.
20 MLMIX_RST	Indicates whether a MLMIX slice reset happens 0 - Power on reset or system cold reset happens. 1 - MLMIX slice reset happens.
19 DDRPHY_RST	Indicates whether a DDRPHY slice reset happens 0 - Power on reset or system cold reset happens. 1 - DDRPHY slice reset happens.
18 DDRMIX_RST	Indicates whether a DDRMIX slice reset happens 0 - Power on reset or system cold reset happens. 1 - DDRMIX slice reset happens.
17 WAKEUPMIX_RST	Indicates whether a WAKEUPMIX slice reset happens 0 - Power on reset or system cold reset happens. 1 - WAKEUPMIX slice reset happens.
16 AONMIX_RST	Indicates whether a AONMIX slice reset happens 0 - Power on reset or system cold reset happens. 1 - AONMIX slice reset happens.
15 —	Reserved
14-13 —	Reserved
12 CM33_LOCKUP	Indicates a reset has been caused by CM33 CPU lockup 0 - Reset is not caused by CM33 lockup. 1 - Reset is caused by CM33 lockup.
11 CM33_REQUEST	Indicates whether reset was the result of CM33 reset request 0 - Reset is not caused by CM33 reset request. 1 - Reset is caused by CM33 reset request.
10 JTAG_SW_RST	JTAG software reset. Indicates whether the reset was the result of JTAG_SW_RST. 0 - Reset is not caused by JTAG_SW_RST. 1 - Reset is caused by JTAG_SW_RST.

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 CSU_RESET_B	Indicates whether the reset was the result of the csu_reset_b input. 0 - Reset is not caused by the csu_reset_b event. 1 - Reset is caused by the csu_reset_b event.
8 TEMPSENSE_RST_B	Temper Sensor software reset. Indicates whether the reset was the result of Temperature Sensor. 0 - Reset is not caused by Temperature Sensor. 1 - Reset is caused by Temperature Sensor.
7 WDOG5_RST_B	Time-out reset. Indicates whether the reset was the result of the watchdog5 time-out 0 - Reset is not caused by the watchdog5 time-out event. 1 - Reset is caused by the watchdog5 time-out event.
6 WDOG4_RST_B	Time-out reset. Indicates whether the reset was the result of the watchdog4 time-out 0 - Reset is not caused by the watchdog4 time-out event. 1 - Reset is caused by the watchdog4 time-out event.
5 WDOG3_RST_B	Time-out reset. Indicates whether the reset was the result of the watchdog3 time-out 0 - Reset is not caused by the watchdog3 time-out event. 1 - Reset is caused by the watchdog3 time-out event.
4 WDOG2_RST_B	Time-out reset. Indicates whether the reset was the result of the watchdog2 time-out event. 0 - Reset is not caused by the watchdog2 time-out event. 1 - Reset is caused by the watchdog2 time-out event.
3 WDOG1_RST_B	Time-out reset. Indicates whether the reset was the result of the watchdog1 time-out event. 0 - Reset is not caused by the watchdog1 time-out event. 1 - Reset is caused by the watchdog1 time-out event.
2 IPP_USER_RESET_B	Indicates whether the reset was the result of ipp_user_reset_b. 0 - Reset is not caused by IPP_USER_RESET_B. 1 - Reset is caused by IPP_USER_RESET_B.
1 JTAG_RST_B	Indicates whether the reset was the result of jtag_rst_b. 0 - Reset is not caused by JTAG_RST_B. 1 - Reset is caused by JTAG_RST_B.
0 SRC_POR_B	Indicates whether the reset was the result of the system_por_b or jtag_rst_b. 0 - Reset is not caused by system_por_b or jtag_rst_b event. 1 - Reset is caused by system_por_b or jtag_rst_b event.

32.5.1.14 SRC General Purpose Register 1 (GPR1)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

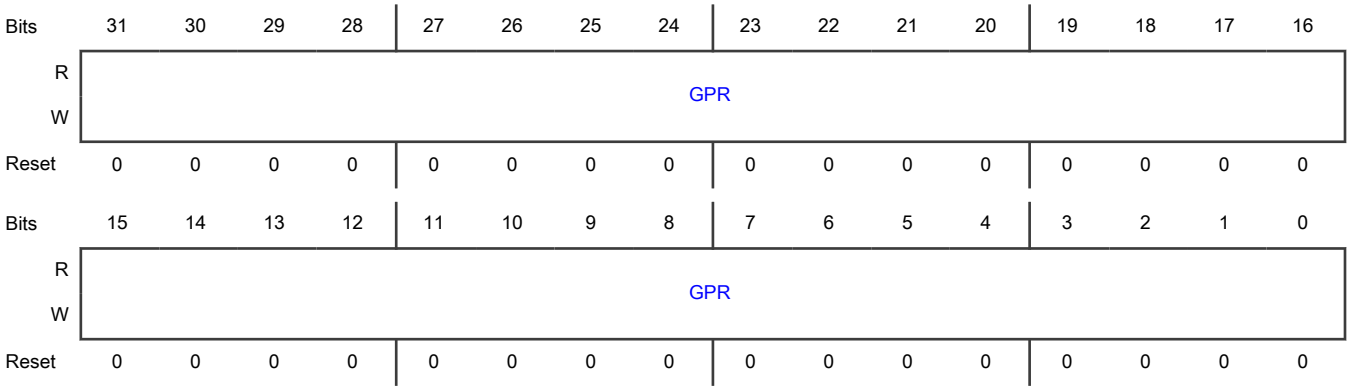
NOTE

This register is not in user space

Offset

Register	Offset
GPR1	54h

Diagram



Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.15 SRC General Purpose Register 2 (GPR2)

This register is in user space and can be used by application software.

Offset

Register	Offset
GPR2	58h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.16 SRC General Purpose Register 3 (GPR3)

This register is in user space and can be used by application software.

Offset

Register	Offset
GPR3	5Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GPR	General Purpose

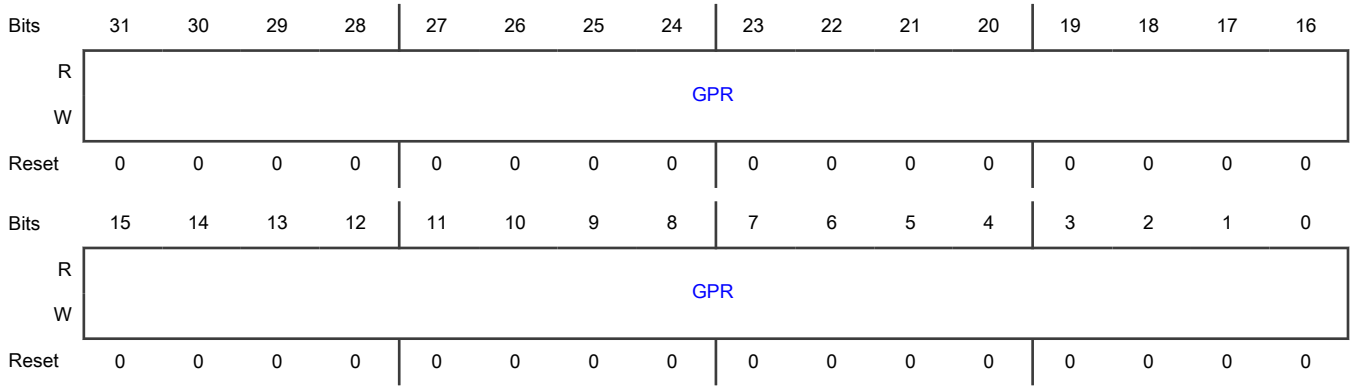
32.5.1.17 SRC General Purpose Register 4 (GPR4)

This register is in user space and can be used by application software.

Offset

Register	Offset
GPR4	60h

Diagram



Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.18 SRC General Purpose Register 5 (GPR5)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR5	64h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.19 SRC General Purpose Register 6 (GPR6)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR6	68h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GPR	General Purpose

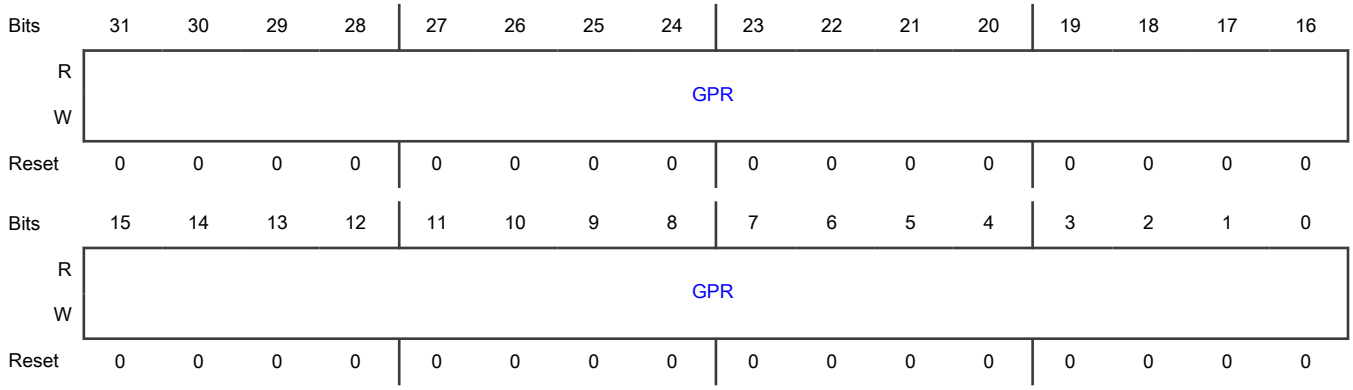
32.5.1.20 SRC General Purpose Register 7 (GPR7)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR7	6Ch

Diagram



Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.21 SRC General Purpose Register 8 (GPR8)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR8	70h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.22 SRC General Purpose Register 9 (GPR9)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR9	74h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GPR	General Purpose

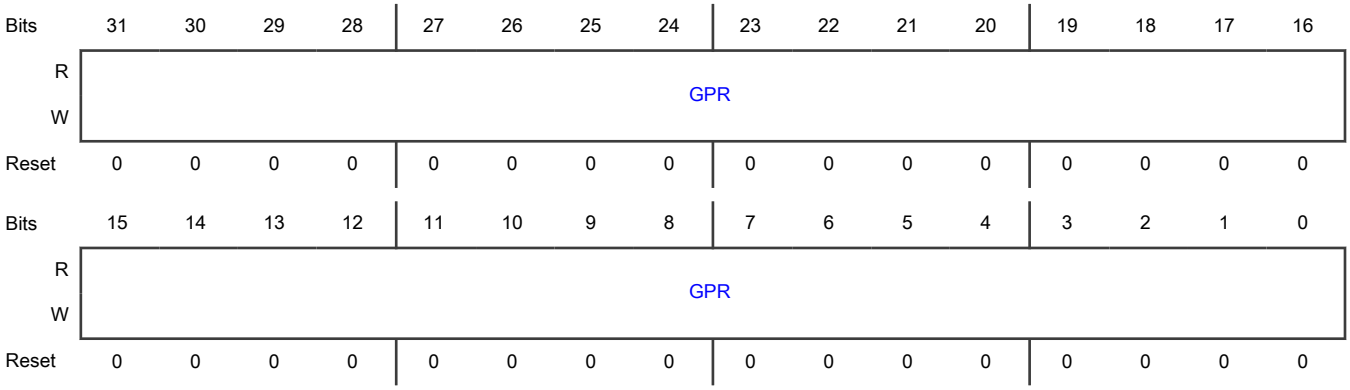
32.5.1.23 SRC General Purpose Register 10 (GPR10)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR10	78h

Diagram



Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.24 SRC General Purpose Register 11 (GPR11)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR11	7Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.25 SRC General Purpose Register 12 (GPR12)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR12	80h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GPR	General Purpose

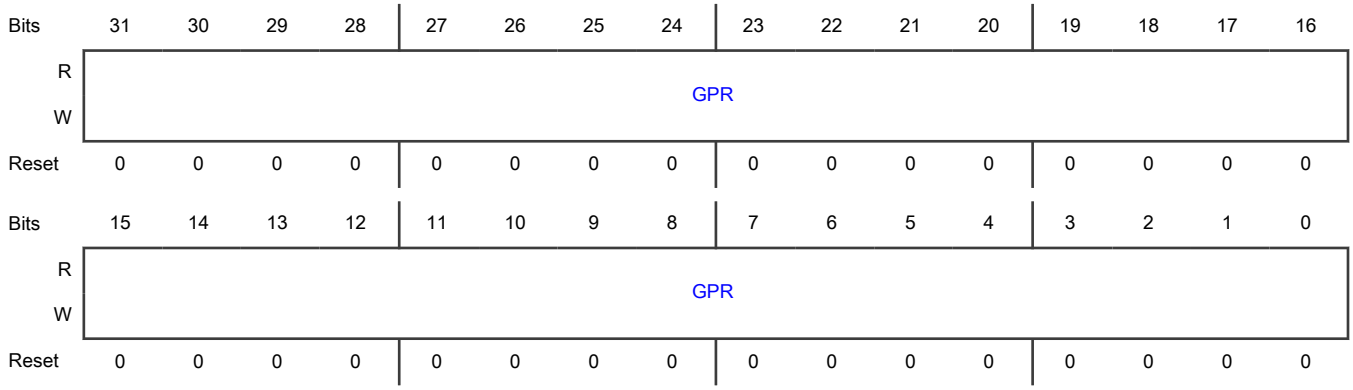
32.5.1.26 SRC General Purpose Register 13 (GPR13)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR13	84h

Diagram



Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.27 SRC General Purpose Register 14 (GPR14)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR14	88h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.28 SRC General Purpose Register 15 (GPR15)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR15	8Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GPR	General Purpose

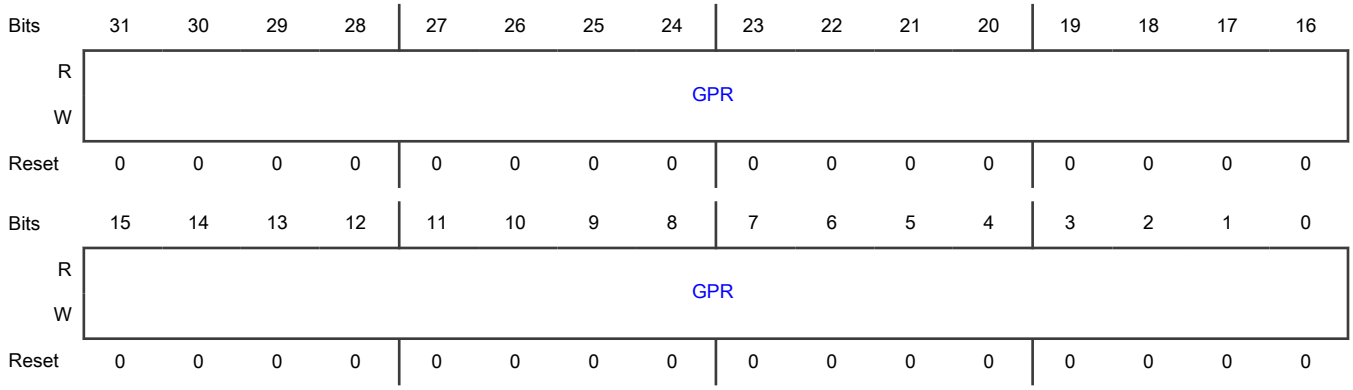
32.5.1.29 SRC General Purpose Register 16 (GPR16)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR16	90h

Diagram



Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.30 SRC General Purpose Register 17 (GPR17)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR17	94h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.31 SRC General Purpose Register 18 (GPR18)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR18	98h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GPR	General Purpose

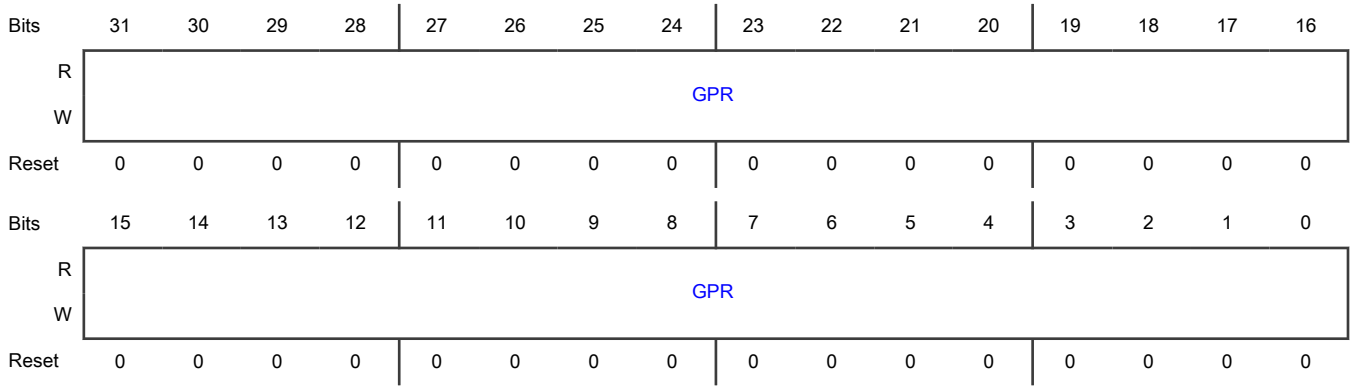
32.5.1.32 SRC General Purpose Register 19 (GPR19)

This register is used by the ROM code and should not be used by application software. Reset by POR not system early reset.

Offset

Register	Offset
GPR19	9Ch

Diagram



Fields

Field	Description
31-0 GPR	General Purpose

32.5.1.33 SRC General Purpose Register 20 (GPR20)

anamix_clkin_mux = gpr_src_general_reg_gpr20_gpr[6:5] default 0

anamix_clkin_mux2 = gpr_src_general_reg_gpr20_gpr[8:7] default 0

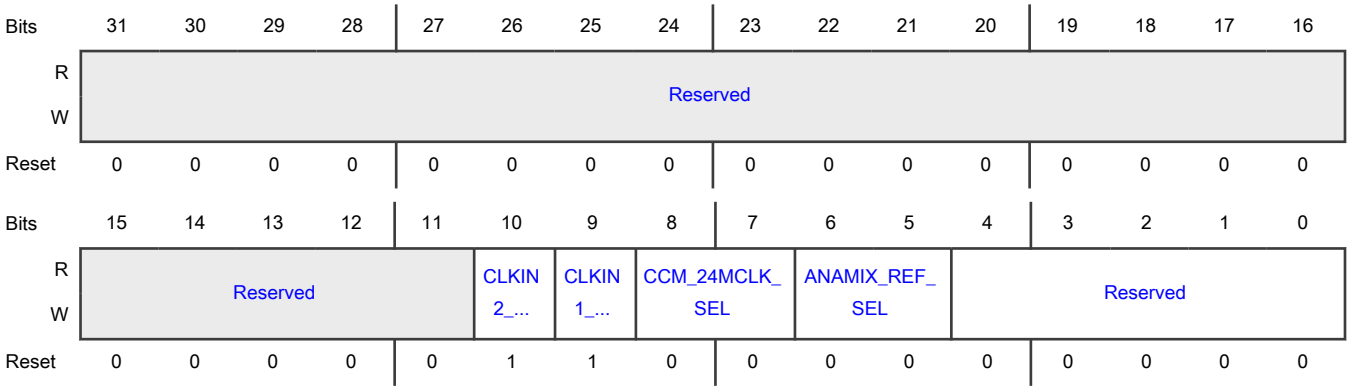
anamix_clkin1_ipp_ibe = gpr_src_general_reg_gpr20_gpr[9] default 1

anamix_clkin2_ipp_ibe = gpr_src_general_reg_gpr20_gpr[10] default 1

Offset

Register	Offset
GPR20	100h

Diagram



Fields

Field	Description
31-11 —	Reserved
10 CLKIN2_ENABLE	CLKIN2 input buffer enable 0 - Input buffer disable 1 - Input buffer enabled. Needed to enable CLKIN2 as an input.
9 CLKIN1_ENABLE	CLKIN1 input buffer enable 0 - Input buffer disable 1 - Input buffer enabled. Needed to enable CLKIN1 as an input.
8-7 CCM_24MCLK_SEL	ANAMIX reference clock select These two bits select which source will drive 24MHz clock root to the CCM. For any CCM clock root, if the 24MHz source is selected, then this is the source for the clock domain. 00 - OSC24M 01 - CLKIN1 10 - CLKIN2 11 - CLKIN1 or CLKIN2
6-5 ANAMIX_REF_SEL	ANAMIX reference clock select These two bits select which source will drive the reference clock to the ANAMIX. For example, this source would be used as the reference for the PLLs in ANAMIX.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	00 - OSC24M 01 - CLKIN1 10 - CLKIN2 11 - Reserved
4-0 —	Reserved

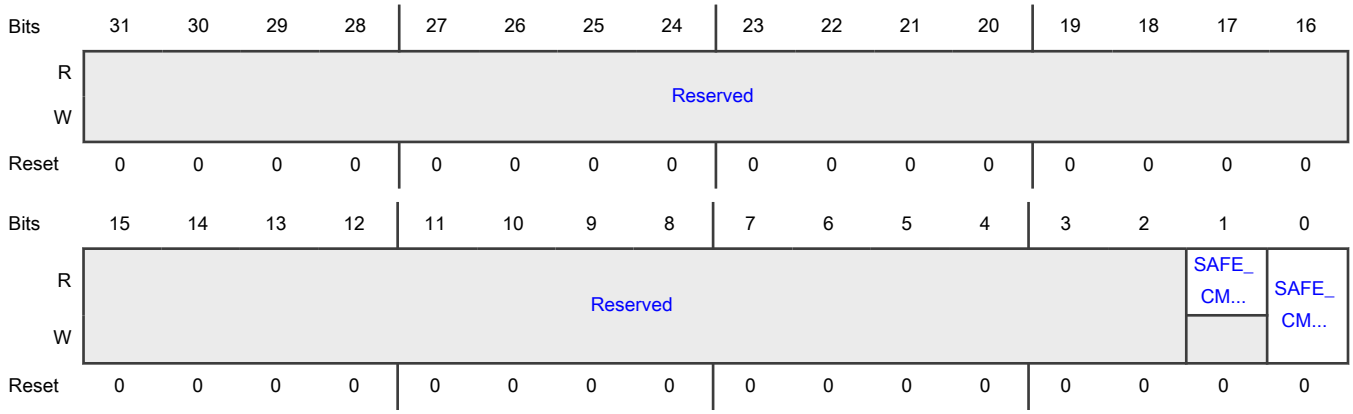
32.5.1.34 SRC_CORTEX_M_QUIESCE (CM_QUIESCE)

This register is used to safe reset Cortex-M paltform. Reset by POR not system early reset

Offset

Register	Offset
CM_QUIESCE	104h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 SAFE_CM33_AHB_ACK_STAT	CM33 platform safe reset ahb ack status 0 - No CM33 platform safe reset req or it could not be placed in safe state for reset sequencing 1 - CM33 platform is in safe state for reset sequencing

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 SAFE_CM33_AHB_REQ	CM33 platform safe reset req control 0 - No effect 1 - request CM33 platform to be reset at safe state

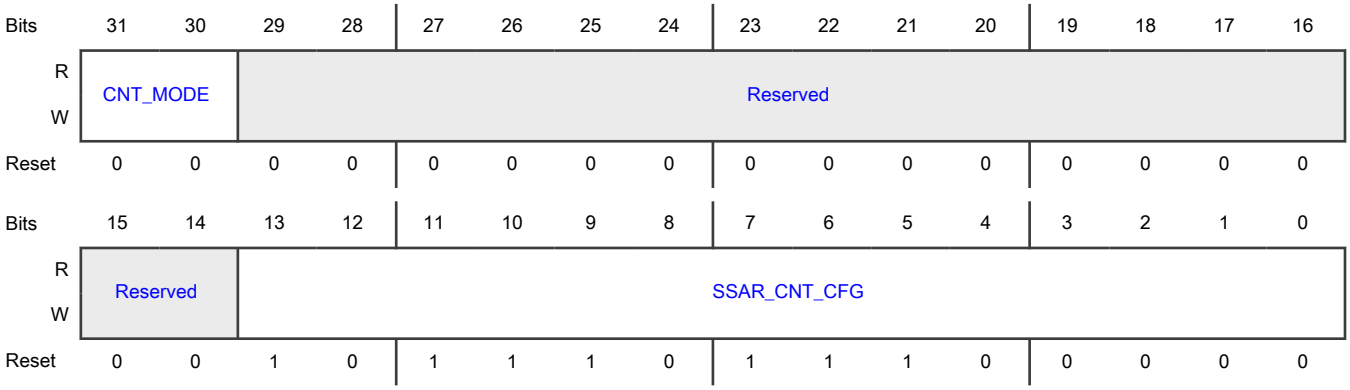
32.5.1.35 Cold reset SSAR acknowledge control (COLD_RESET_SSAR_ACK_CTRL)

Cold reset SSAR acknowledge control

Offset

Register	Offset
COLD_RESET_SSAR_ACK_CTRL	108h

Diagram



Fields

Field	Description
31-30 CNT_MODE	Configure the acknowledge counter working mode. Locked by LOCK_CFG field 00 - Not use counter, raise done to cold_reset_controller once get EdgeLock Enclave ack 01 - Delay after receiving EdgeLock Enclave ack, delay cycle number is CNT_CFG 10 - Ignore EdgeLock Enclave ack, raise done to cold_reset_controller when counting to CNT_CFG value 11 - Time out mode, raise done to cold_reset_controller when either EdgeLock Enclave ack received or counting to CNT_CFG value
29-14 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
13-0 SSAR_CNT_CFG	ssar count configure. Usage depends on CNT_MODE, locked by LOCK_CFG field

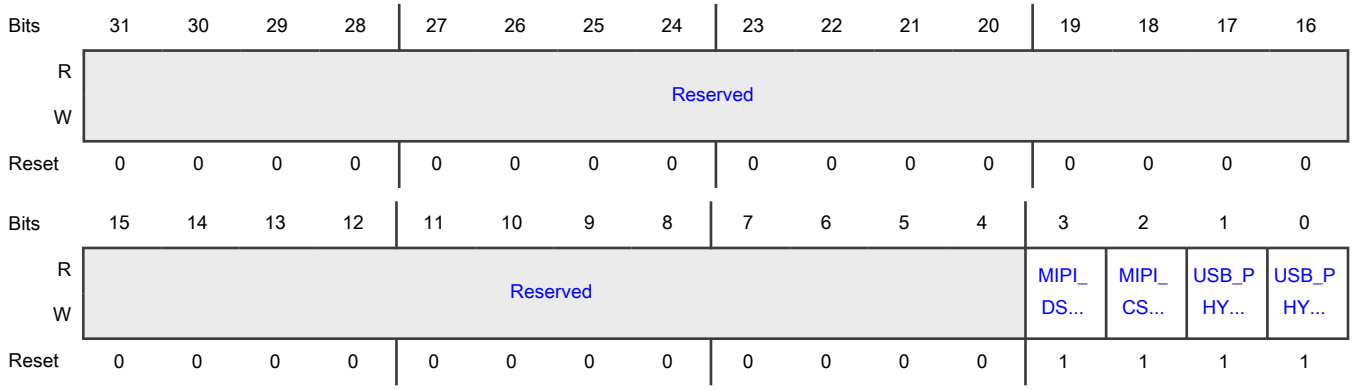
32.5.1.36 SRC special ISO Control (SP_ISO_CTRL)

USB PHY1/2 and MIPI DSI/CSI isolation enable control

Offset

Register	Offset
SP_ISO_CTRL	10Ch

Diagram



Fields

Field	Description
31-4 —	Reserved
3 MIPI_DSI_ISO	Software control MIPI DSI isolation enable The PHY is isolated by default 0 - Isolation disable 1 - Isolation enable
2 MIPI_CSI_ISO	Software control MIPI CSI isolation enable The PHY is isolated by default 0 - Isolation disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Isolation enable
1 USB_PHY2_ISO	Software control USB PHY2 isolation enable The PHY is isolated by default 0 - Isolation disable 1 - Isolation enable
0 USB_PHY1_ISO	Software control USB PHY1 isolation enable The PHY is isolated by default 0 - Isolation disable 1 - Isolation enable

32.5.1.37 ROM Low Power Control (ROM_LP_CTRL)

Offset

Register	Offset
ROM_LP_CTRL	110h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															AONMIX_...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-1 —	Reserved
0	ROM in AONMIX low power control enable

Table continues on the next page...

Table continued from the previous page...

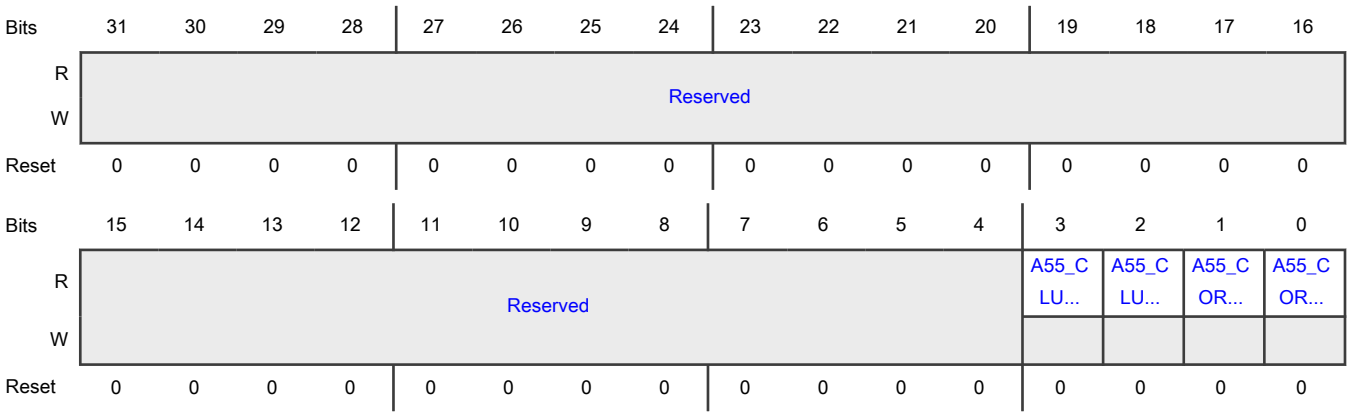
Field	Description
AONMIX_ROM_LP_EN	

32.5.1.38 A55 Q_Channel Deny Status (A55_DENY_STAT)

Offset

Register	Offset
A55_DENY_STAT	114h

Diagram



Fields

Field	Description
31-4 —	Reserved
3 A55_CLUSTER_CLKOFF_DENY_STAT	A55 Cluster Q_Channel clockoff deny status
2 A55_CLUSTER_PWRDN_DENY_STAT	A55 Cluster Q_Channel pwrdn deny status
1	A55 CORE 1 Q_Channel pwrdn deny status

Table continues on the next page...

Table continued from the previous page...

Field	Description
A55_CORE1_PWRDN_DENY_STAT	
0 A55_CORE0_PWRDN_DENY_STAT	A55 CORE 0 Q_Channel pwrdn deny status

32.5.2 SRC MEM SLICE register descriptions

32.5.2.1 src_mem memory map

src_a55_l3_mem base address: 4446_6800h

src_a55_scu_mem base address: 4446_6400h

src_a55c0_mem base address: 4446_5C00h

src_a55c1_mem base address: 4446_6000h

src_aon_mem base address: 4446_3800h

src_ddr_mem base address: 4446_4000h

src_dphy_mem base address: 4446_4400h

src_hsio_mem base address: 4446_5400h

src_media_mem base address: 4446_5800h

src_ml_mem base address: 4446_4800h

src_nic_mem base address: 4446_4C00h

src_nic_ocram base address: 4446_5000h

src_wkup_mem base address: 4446_3C00h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	MEM Low Power Control (MEM_CTRL)	32	RW	00A0_A000
8	MEM Low Power Control_0 (MEMLP_CTRL_0)	32	RW	0000_0001
10	MEM Low Power Control_1 (MEMLP_CTRL_1)	32	RW	0000_0001
14	MEM Low Power Control_2 (MEMLP_CTRL_2)	32	RW	0000_0001
18	MEM Status (MEM_STAT)	32	R	0000_0030

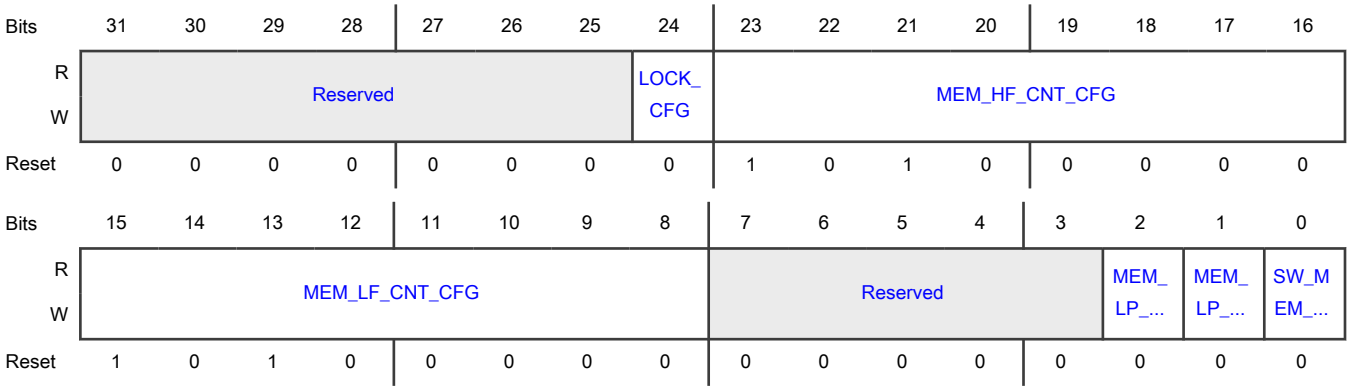
32.5.2.2 MEM Low Power Control (MEM_CTRL)

.

Offset

Register	Offset
MEM_CTRL	4h

Diagram



Fields

Field	Description
31-25 —	Reserved
24 LOCK_CFG	Configuration lock This field is write once and can lock the value of the low power configuration fields. 0 - Not locked. 1 - Locked.
23-16 MEM_HF_CNT_CFG	MEM power up counter Default value : • Non-core:'h1 (3.125us) • core:'ha0 (0.5ms)
15-8 MEM_LF_CNT_CFG	MEM power up counter Default value: • Non-core:'h1 (3.125us) • core:'ha0 (0.5ms)
7-3 —	Reserved
2	Enable MEM low power control. Locked by LOCK_CFG field

Table continues on the next page...

Table continued from the previous page...

Field	Description
MEM_LP_EN	
1 MEM_LP_MODE	MEM low power mode. Locked by LOCK_CFG field. 0 - Power down mode 1 - Retention mode
0 SW_MEM_CTRL	Software control MEM low power 0 - software control MEM to exit low power 1 - software control MEM to enter low power

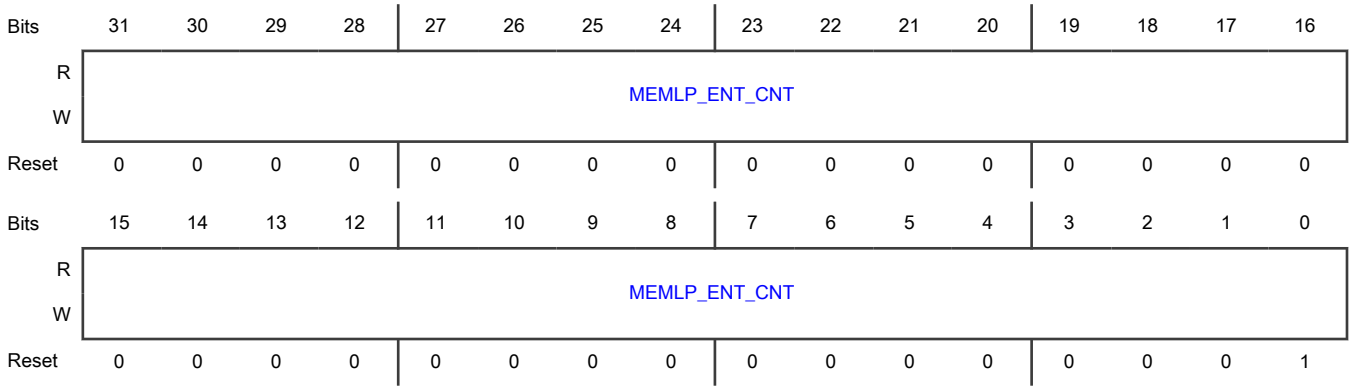
32.5.2.3 MEM Low Power Control_0 (MEMLP_CTRL_0)

.

Offset

Register	Offset
MEMLP_CTRL_0	8h

Diagram



Fields

Field	Description
31-0 MEMLP_ENT_CNT	Delay counter to start entering to memory low power mode. Locked by LOCK_CFG field

32.5.2.4 MEM Low Power Control_1 (MEMLP_CTRL_1)

.

Offset

Register	Offset
MEMLP_CTRL_1	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MEMLP_RET_PGEN_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MEMLP_RET_PGEN_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31-0 MEMLP_RET_PGEN_CNT	Delay counter to interval for retn to pgen. Locked by LOCK_CFG field

32.5.2.5 MEM Low Power Control_2 (MEMLP_CTRL_2)

.

Offset

Register	Offset
MEMLP_CTRL_2	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MEMLP_EXT_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MEMLP_EXT_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31-0 MEMLP_EXT_CNT	Delay counter to start exiting from memory low power mode. Locked by LOCK_CFG field

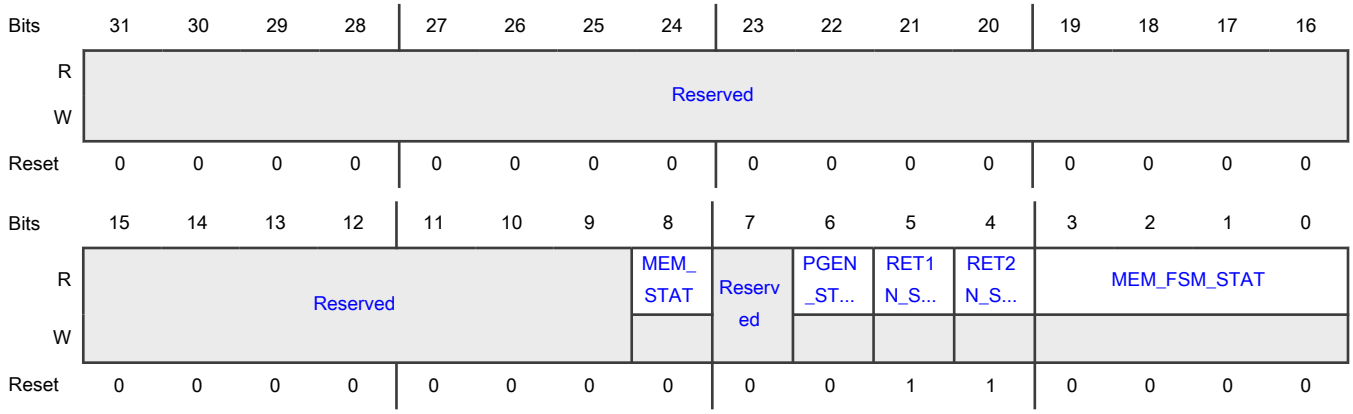
32.5.2.6 MEM Status (MEM_STAT)

.

Offset

Register	Offset
MEM_STAT	18h

Diagram



Fields

Field	Description
31-9 —	Reserved
8 MEM_STAT	MEM status 0 - MEM exit low power 1 - MEM enter low power - rentention1 mode or power down mode
7 —	Reserved
6	PGEN status

Table continues on the next page...

Table continued from the previous page...

Field	Description
PGEN_STAT	
5 RET1N_STAT	RET1N status
4 RET2N_STAT	RET2N status
3-0 MEM_FSM_STAT	MEM FSM status

32.5.3 SRC MIX SLICE register descriptions

32.5.3.1 src mix slice register description memory map

src_a55c0_slice base address: 4446_2C00h

src_a55c1_slice base address: 4446_3000h

src_a55p_slice base address: 4446_3400h

src_aon_slice base address: 4446_0800h

src_ddr_slice base address: 4446_1000h

src_dphy_slice base address: 4446_1400h

src_edgelock_slice base address: 4446_0400h

src_hsio_slice base address: 4446_2000h

src_m33p_slice base address: 4446_2800h

src_media_slice base address: 4446_2400h

src_ml_slice base address: 4446_1800h

src_nic_slice base address: 4446_1C00h

src_wkup_slice base address: 4446_0C00h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	Authentication Control (AUTHEN_CTRL)	32	RW	FFFF_0000
10	Low power mode setting (LPM_SETTING_0)	32	RW	0000_0003
14	Low power mode setting (LPM_SETTING_1)	32	RW	3333_3333
18	Low power mode setting (LPM_SETTING_2)	32	RW	3333_3333
20	Slice software control (SLICE_SW_CTRL)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
24	Single reset by software control (SINGLE_RESET_SW_CTRL)	32	RW	0000_0555
40	A55 handshake acknowledge control (A55_HDSK_ACK_CTRL)	32	RW	0000_0000
44	A55 handshake acknowledge status (A55_HDSK_ACK_STAT)	32	R	0000_0000
50	SSAR acknowledge control (SSAR_ACK_CTRL)	32	RW	0000_2EE0
54	SSAR acknowledge status (SSAR_ACK_STAT)	32	R	0000_0000
5C	iso off delay control when por (ISO_OFF_DLY_POR)	32	RW	0000_001F
60	iso on delay control (ISO_ON_DLY)	32	RW	0000_0000
64	iso off delay control (ISO_OFF_DLY)	32	RW	0000_0024
68	psw off lf delay control (PSW_OFF_LF_DLY)	32	RW	0000_0000
70	psw off hf delay control (PSW_OFF_HF_DLY)	32	RW	0000_0000
74	psw on lf delay control (PSW_ON_LF_DLY)	32	RW	0000_0000
78	psw on hf delay control (PSW_ON_HF_DLY)	32	RW	0000_0000
80	Power switch acknowledge control (PSW_ACK_CTRL_0)	32	RW	10A0_00A0
84	Power switch acknowledge control (PSW_ACK_CTRL_1)	32	RW	0001_0001
88	PSW acknowledge status (PSW_ACK_STAT)	32	R	0000_0000
90	MTR acknowledge control (MTR_ACK_CTRL)	32	RW	0000_0000
94	MTR acknowledge status (MTR_ACK_STAT)	32	R	0000_0000
A0	UPI status 0 (UPI_STAT_0)	32	R	0000_0000
A4	UPI status 1 (UPI_STAT_1)	32	R	0000_0000
A8	UPI status 2 (UPI_STAT_2)	32	R	0000_0000
AC	UPI status 3 (UPI_STAT_3)	32	R	0000_0000
B0	FSM status (FSM_STAT)	32	R	0000_0000
B4	function status (FUNC_STAT)	32	R	0000_0000

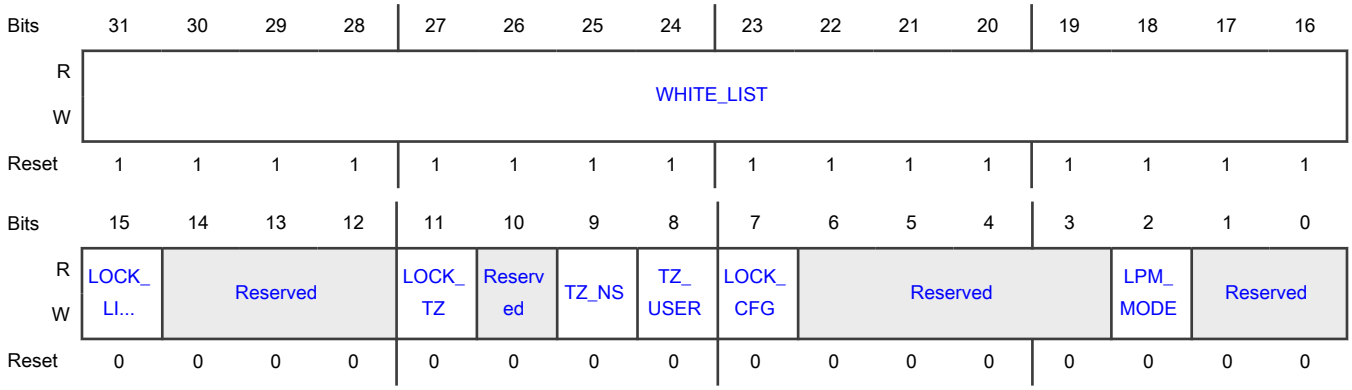
32.5.3.2 Authentication Control (AUTHEN_CTRL)

When GPC_DOMAIN[CPUx_DOMAIN] bit is 1, the MIX SLICE will be power switch with the corresponding CPU.

Offset

Register	Offset
AUTHEN_CTRL	4h

Diagram



Fields

Field	Description
31-16 WHITE_LIST	Domain ID white list Each master(Core or non-Core) can be assigned to any domains among 0-15 through TRDC and through GPC domain assignment register. Once its domains are assigned in TRDC, then its accesses will have its unique domain ID. Each bit in this field represent for one domain. Bit 16-31 represent domains 0-15 respectively. Only the TRDC domains whose corresponding bits are 1 can write this General registers. Write violation will fail without any exception or information. Read violation can still get read data. In LPM mode, only the GPC domains whose corresponding bits are 1 can trigger the powering down of the given power slice. 0000_0000_0000_0001 - Core with domain ID=0 can write SRC MIX SLICE registers. 0000_0000_0000_0010 - Core with domain ID=1 can write SRC MIX SLICE registers. 0000_0000_0000_0100 - Core with domain ID=2 can write SRC MIX SLICE registers. 0000_0000_0000_1000 - Core with domain ID=3 can write SRC MIX SLICE registers. 0000_0000_0001_0000 - Core with domain ID=4 can write SRC MIX SLICE registers. 0000_0000_0010_0000 - Core with domain ID=5 can write SRC MIX SLICE registers. 0000_0000_0100_0000 - Core with domain ID=6 can write SRC MIX SLICE registers. 0000_0000_1000_0000 - Core with domain ID=7 can write SRC MIX SLICE registers. 0000_0001_0000_0000 - Core with domain ID=8 can write SRC MIX SLICE registers. 0000_0010_0000_0000 - Core with domain ID=9 can write SRC MIX SLICE registers. 0000_0100_0000_0000 - Core with domain ID=10 can write SRC MIX SLICE registers. 0000_1000_0000_0000 - Core with domain ID=11 can write SRC MIX SLICE registers. 0001_0000_0000_0000 - Core with domain ID=12 can write SRC MIX SLICE registers. 0010_0000_0000_0000 - Core with domain ID=13 can write SRC MIX SLICE registers. 0100_0000_0000_0000 - Core with domain ID=14 can write SRC MIX SLICE registers. 1000_0000_0000_0000 - Core with domain ID=15 can write SRC MIX SLICE registers.
15	White list lock

Table continues on the next page...

Table continued from the previous page...

Field	Description
LOCK_LIST	This field is write once and can lock the value of the WHITE_LIST field. 0 - WHITE_LIST value can be changed. 1 - WHITE_LIST value cannot be changed.
14-12 —	Reserved
11 LOCK_TZ	Lock NONSECURE and USER This field is write once and can lock the TZ_NS and TZ_USER fields. 0 - TZ_NS and TZ_USER value can be changed. 1 - TZ_NS and TZ_USER value cannot be changed.
10 —	Reserved
9 TZ_NS	Allow non-secure mode access Allow either secure mode or non-secure mode to write this MIX SLICE registers. 0 - This MIX SLICE registers can only be written in secure mode. 1 - This MIX SLICE registers can be written either in secure mode or non-secure mode.
8 TZ_USER	Allow user mode access Allow either privilege mode or user mode to write this MIX SLICE registers. 0 - This MIX SLICE registers can only be written in privilege mode. 1 - This MIX SLICE registers can be written either in privilege mode or user mode.
7 LOCK_CFG	Configuration lock This field is write once and can lock value of the low power configuration fields. 0 - Low power configuration fields are not locked. 1 - Low power configuration fields are locked.
6-3 —	Reserved
2 LPM_MODE	Low power control mode **NOTE** This bit cannot be written for SRC.src_edgelock_slice. 0 - Low power mode controlled by software 1 - Low power mode controlled by GPC hardware
1-0	Reserved

Table continues on the next page...

Table continued from the previous page...

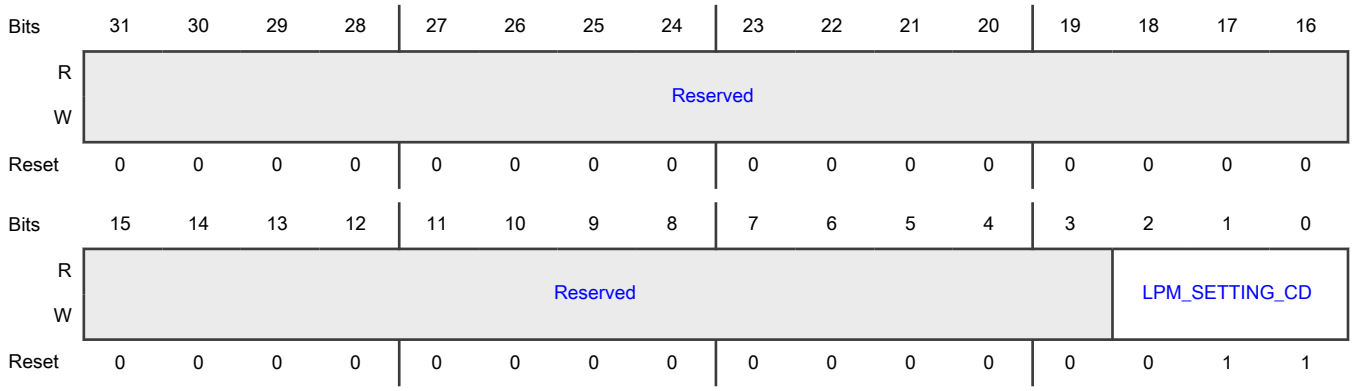
Field	Description
—	

32.5.3.3 Low power mode setting (LPM_SETTING_0)

Offset

Register	Offset
LPM_SETTING_0	10h

Diagram



Fields

Field	Description
31-3 —	Reserved
2-0 LPM_SETTING_CD	LPM setting of current domain This register can only be updated while MIX SLICE AUTHEN_CTRL[LPM_MODE]=1. When writing to this register, the data will update related domain register based on ips_domain_id. 000 - Power always off 001 - Power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - Power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - Power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - Power always on

32.5.3.4 Low power mode setting (LPM_SETTING_1)

This is only valid if the corresponding WHITELIST bit is set.

Offset

Register	Offset
LPM_SETTING_1	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	LPM_SETTING_D7			Reserved	LPM_SETTING_D6			Reserved	LPM_SETTING_D5			Reserved	LPM_SETTING_D4		
W																
Reset	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	LPM_SETTING_D3			Reserved	LPM_SETTING_D2			Reserved	LPM_SETTING_D1			Reserved	LPM_SETTING_D0		
W																
Reset	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

Fields

Field	Description
31 —	Reserved
30-28 LPM_SETTING_D7	LPM setting of domain 7 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
27 —	Reserved
26-24 LPM_SETTING_D6	LPM setting of domain 6 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
23 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
22-20 LPM_SETTING_D5	LPM setting of domain 5 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
19 —	Reserved
18-16 LPM_SETTING_D4	LPM setting of domain 4 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
15 —	Reserved
14-12 LPM_SETTING_D3	LPM setting of domain 3 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
11 —	Reserved
10-8 LPM_SETTING_D2	LPM setting of domain 2 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
7 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
6-4 LPM_SETTING_D1	LPM setting of domain 1 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
3 —	Reserved
2-0 LPM_SETTING_D0	LPM setting of domain 0 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on

32.5.3.5 Low power mode setting (LPM_SETTING_2)

This is only valid if the corresponding WHITELIST bit is set.

Offset

Register	Offset
LPM_SETTING_2	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	LPM_SETTING_D15			Reserved	LPM_SETTING_D14			Reserved	LPM_SETTING_D13			Reserved	LPM_SETTING_D12		
Reset	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	LPM_SETTING_D11			Reserved	LPM_SETTING_D10			Reserved	LPM_SETTING_D9			Reserved	LPM_SETTING_D8		
Reset	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

Fields

Field	Description
31 —	Reserved
30-28 LPM_SETTING_D15	LPM setting of domain 15 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
27 —	Reserved
26-24 LPM_SETTING_D14	LPM setting of domain 14 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
23 —	Reserved
22-20 LPM_SETTING_D13	LPM setting of domain 13 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
19 —	Reserved
18-16 LPM_SETTING_D12	LPM setting of domain 12 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on

Table continues on the next page...

Table continued from the previous page...

Field	Description
15 —	Reserved
14-12 LPM_SETTING_D11	LPM setting of domain 11 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
11 —	Reserved
10-8 LPM_SETTING_D10	LPM setting of domain 10 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
7 —	Reserved
6-4 LPM_SETTING_D9	LPM setting of domain 9 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on
3 —	Reserved
2-0 LPM_SETTING_D8	LPM setting of domain 8 000 - Not used. Do not write this value. 001 - power on when domain n is in RUN, off in WAIT/STOP/SUSPEND 010 - power on when domain n is in RUN/WAIT, off in STOP/SUSPEND 011 - power on when domain n is in RUN/WAIT/STOP, off in SUSPEND 100-111 - power always on

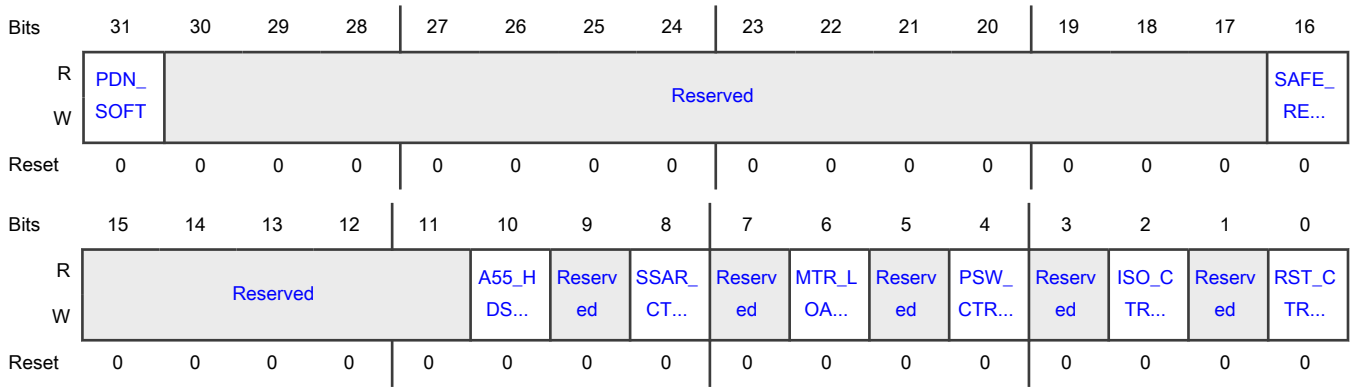
32.5.3.6 Slice software control (SLICE_SW_CTRL)

this register controls whole slice

Offset

Register	Offset
SLICE_SW_CTRL	20h

Diagram



Fields

Field	Description
31 PDN_SOFT	Software power trans control, including reset, iso, and power switch. Locked by LPM_MODE field. 0 - No effect or software power up 1 - Software power down
30-17 —	Reserved
16 SAFE_RESET	Only for CM33 safe reset feature 0 - No effect or software CM33 safe reset deassert 1 - Software CM33 safe reset assert
15-11 —	Reserved
10 A55_HDSK_CTRL_SOFT	Software A55 handshake control. Locked by LPM_MODE field. 0 - No effect or software notify A55 power up info 1 - Software notify A55 power down info
9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 SSAR_CTRL_SOFT	Software SSAR control. Locked by LPM_MODE field. 0 - No effect or software SSAR restore 1 - Software SSAR save
7 —	Reserved
6 MTR_LOAD_SOFT	Software control MTR repair load. Locked by LPM_MODE field. 0 - No effect 1 - Software load MTR repair
5 —	Reserved
4 PSW_CTRL_SOFT	Software power switch control. Locked by LPM_MODE field. 0 - No effect or software power switch on 1 - Software power switch off
3 —	Reserved
2 ISO_CTRL_SOFT	Software isolation control. Locked by LPM_MODE field. 0 - No effect or software iso off 1 - Software iso on
1 —	Reserved
0 RST_CTRL_SOFT	Software reset control. Locked by LPM_MODE field. 0 - No effect or software reset deassert 1 - Software reset assert

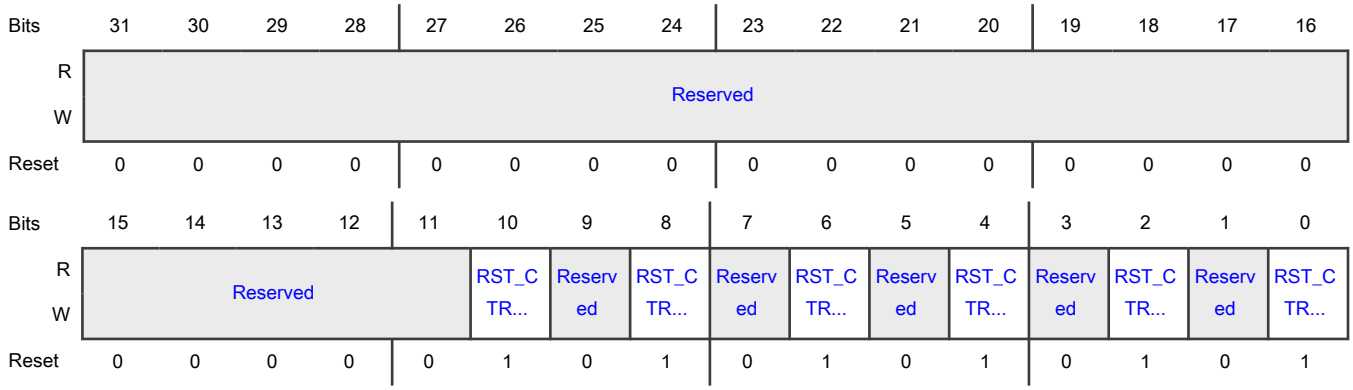
32.5.3.7 Single reset by software control (SINGLE_RESET_SW_CTRL)

The SRC only provides a reset for a slice. If a slice has multiple reset signals requirement, every single reset can be soft controlled by this register.

Offset

Register	Offset
SINGLE_RESET_SW_CTRL	24h

Diagram



Fields

Field	Description
31-11 —	Reserved
10 RST_CTRL_SOFT_5	Locked by LPM_MODE field. reset_5 : For A55 Plat, it is nPERIPHRESET 0 - software reset assert 1 - No effect or software reset deassert
9 —	Reserved
8 RST_CTRL_SOFT_4	Locked by LPM_MODE field. reset_4 : For A55 Plat, it is nGICRESET 0 - Software reset assert 1 - No effect or software reset deassert
7 —	Reserved
6 RST_CTRL_SOFT_3	Locked by LPM_MODE field. reset_3 : For A55 Plat, it is nATRESET 0 - Software reset assert 1 - No effect or software reset deassert
5 —	Reserved
4	Locked by LPM_MODE field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
RST_CTRL_SOFT_2	reset_2 : For A55 Plat, it is nSRESET 0 - Software reset assert 1 - No effect or software reset deassert
3 —	Reserved
2 RST_CTRL_SOFT_1	Locked by LPM_MODE field. reset_1: For A55 CPU, it is nCORERESET. For A55 Plat, it is nSPORESET. • 0b - Software reset assert • 1b - No effect or software reset deassert For DDRPHY, it is phy_reset_n. • 0b - No effect or software reset deassert • 1b - Software reset assert
1 —	Reserved
0 RST_CTRL_SOFT_0	Locked by LPM_MODE field. reset_0 : For A55 CPU, it is nCPUPORRESET. For A55 Plat, it is nPRESET. • 0b - Software reset assert • 1b - No effect or software reset deassert For DDRPHY, it is PwrOKIn. • 0b - No effect or software reset deassert • 1b - Software reset assert

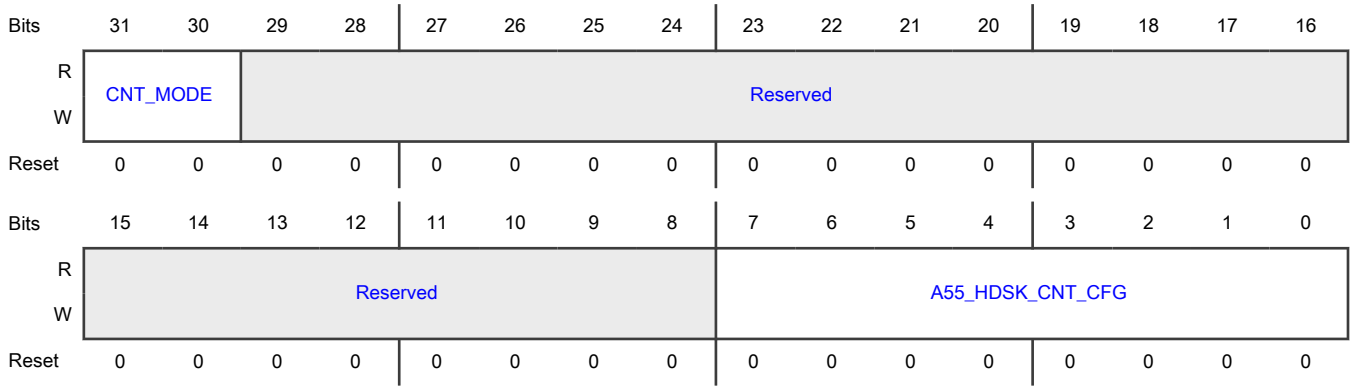
32.5.3.8 A55 handshake acknowledge control (A55_HDSK_ACK_CTRL)

A55 handshake acknowledge control

Offset

Register	Offset
A55_HDSK_ACK_CTRL	40h

Diagram



Fields

Field	Description
31-30 CNT_MODE	Configure the acknowledge counter working mode. Locked by LOCK_CFG field. 00 - Not use counter, raise a55_hdsk done to GPC once get A55 ack 01 - Delay after receiving a55 ack, delay cycle number is CNT_CFG 10 - Ignore A55 ack, raise a55_hdsk done to GPC when counting to CNT_CFG value 11 - Time out mode, raise a55_hdsk done to GPC when either A55 ack received or counting to CNT_CFG value
29-8 —	Reserved
7-0 A55_HDSK_CNT_CFG	A55 handshake count configure. Usage depends on CNT_MODE. Locked by LOCK_CFG field.

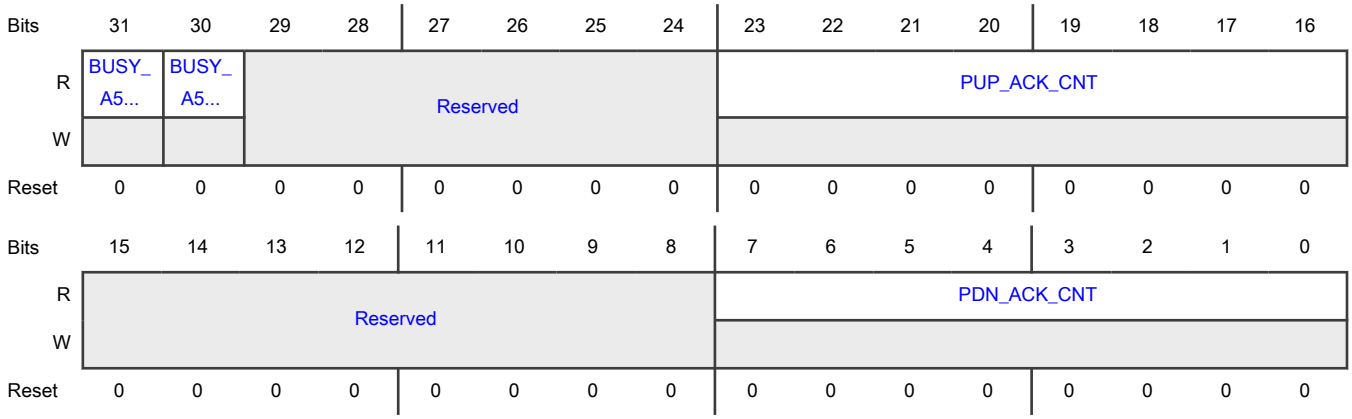
32.5.3.9 A55 handshake acknowledge status (A55_HDSK_ACK_STAT)

A55 handshake acknowledge status

Offset

Register	Offset
A55_HDSK_ACK_STAT	44h

Diagram



Fields

Field	Description
31 BUSY_A55_PUP_HDSK	Busy requesting A55 power up handshake
30 BUSY_A55_PDN_HDSK	Busy requesting A55 power down handshake
29-24 —	Reserved
23-16 PUP_ACK_CNT	A55 handshake power up acknowledge count Record the delay from stat change to acknowledge received.
15-8 —	Reserved
7-0 PDN_ACK_CNT	A55 handshake power down acknowledge count. Record the delay from stat change to acknowledge received.

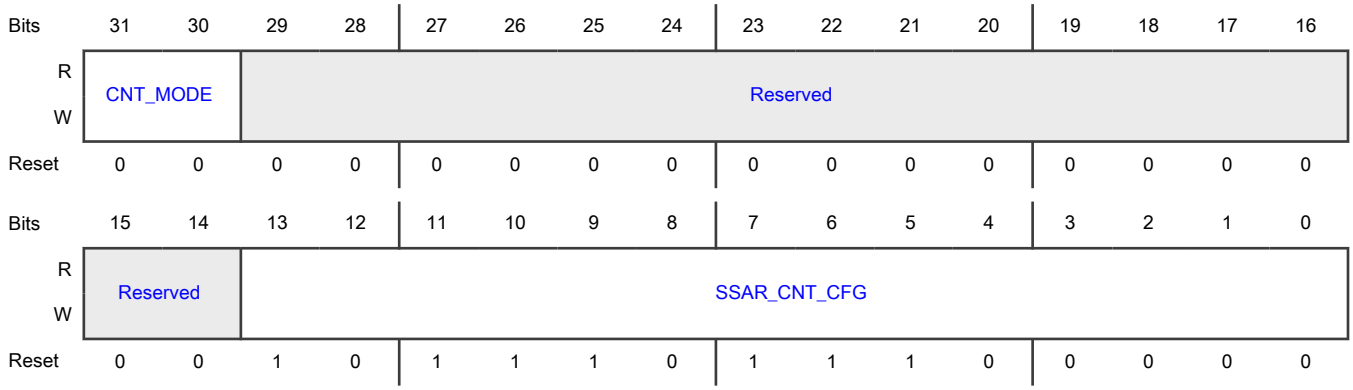
32.5.3.10 SSAR acknowledge control (SSAR_ACK_CTRL)

SSAR acknowledge control

Offset

Register	Offset
SSAR_ACK_CTRL	50h

Diagram



Fields

Field	Description
31-30 CNT_MODE	Configure the acknowledge counter working mode. Locked by LOCK_CFG field. 00 - Not use counter, raise ssar_save/restore done to GPC once get Edgelock Enclave ack 01 - Delay after receiving Edgelock Enclave ack, delay cycle number is CNT_CFG 10 - Ignore Edgelock Enclave ack, raise ssar_save/restore done to GPC when counting to CNT_CFG value 11 - Time out mode, raise ssar_save/restore done to GPC when either Edgelock Enclave ack received or counting to CNT_CFG value
29-14 —	Reserved
13-0 SSAR_CNT_CFG	ssar count configure. Usage depends on CNT_MODE. Locked by LOCK_CFG field.

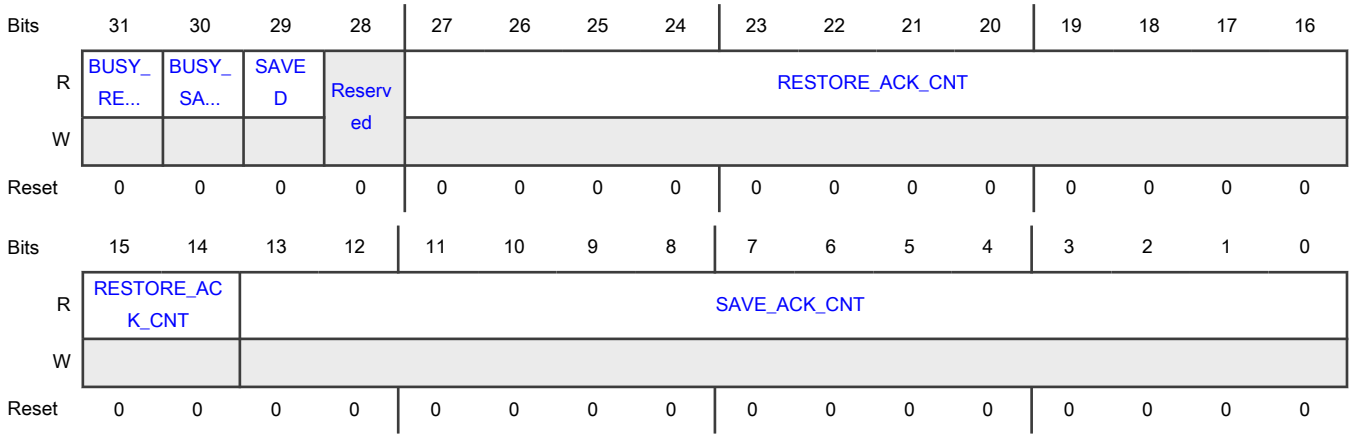
32.5.3.11 SSAR acknowledge status (SSAR_ACK_STAT)

SSAR acknowledge status

Offset

Register	Offset
SSAR_ACK_STAT	54h

Diagram



Fields

Field	Description
31 BUSY_RESTORE	Busy requesting SSAR restore
30 BUSY_SAVED	Busy requesting SSAR save
29 SAVED	Indicate this mix power down info have accepted Edgelock Enclave ack
28 —	Reserved
27-14 RESTORE_ACK_CNT	RESTORE acknowledge count, record the delay from stat change to acknowledge received
13-0 SAVE_ACK_CNT	SAVE acknowledge count, record the delay from stat change to acknowledge received

32.5.3.12 iso off delay control when por (ISO_OFF_DLY_POR)

.

Offset

Register	Offset
ISO_OFF_DLY_POR	5Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLY_PRE_ISO_OFF_POR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLY_PRE_ISO_OFF_POR															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Fields

Field	Description
31-0 DLY_PRE_ISO_OFF_POR	Delay from receiving iso off request to isolation disable. Locked by LOCK_CFG field. Delay to release iso same time with reset.

32.5.3.13 iso on delay control (ISO_ON_DLY)

.

Offset

Register	Offset
ISO_ON_DLY	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLY_PRE_ISO_ON															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLY_PRE_ISO_ON															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 DLY_PRE_ISO_ON	Delay from receiving iso_on request to isolation enable. Locked by LOCK_CFG field.

32.5.3.14 iso off delay control (ISO_OFF_DLY)

.

Offset

Register	Offset
ISO_OFF_DLY	64h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLY_PRE_ISO_OFF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLY_PRE_ISO_OFF															
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

Fields

Field	Description
31-0 DLY_PRE_ISO_OFF	Delay from receiving iso off request to isolation disable. Locked by LOCK_CFG field. Delay to release iso same time with reset.

32.5.3.15 psw off lf delay control (PSW_OFF_LF_DLY)

.

Offset

Register	Offset
PSW_OFF_LF_DLY	68h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLY_PRE_PSW_OFF_LF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLY_PRE_PSW_OFF_LF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 DLY_PRE_PSW_OFF_LF	Delay from receiving power off lf request to power switch shut off. Locked by LOCK_CFG field.

32.5.3.16 psw off hf delay control (PSW_OFF_HF_DLY)

.

Offset

Register	Offset
PSW_OFF_HF_DLY	70h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLY_PRE_PSW_OFF_HF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLY_PRE_PSW_OFF_HF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 DLY_PRE_PSW_OFF_HF	Delay from receiving power off hf request to power switch shut off. Locked by LOCK_CFG field.

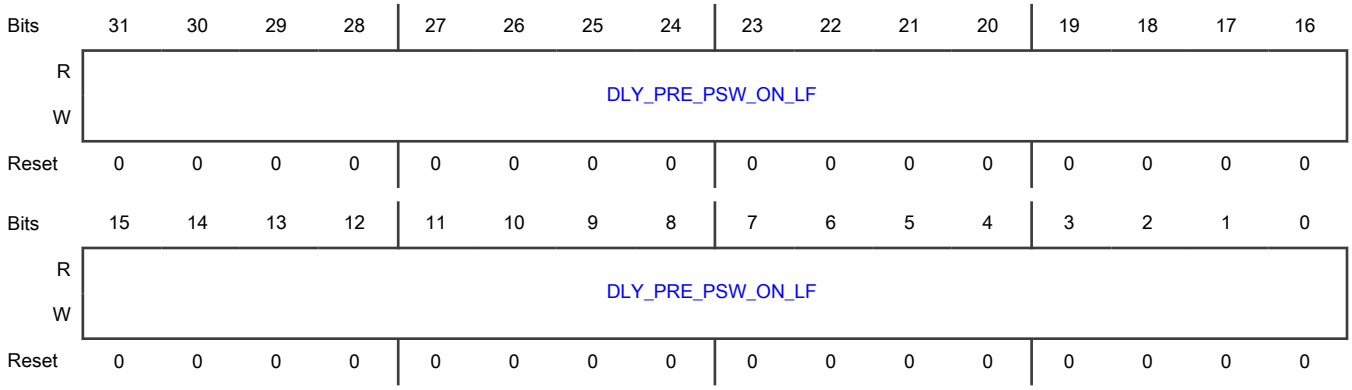
32.5.3.17 psw on lf delay control (PSW_ON_LF_DLY)

.

Offset

Register	Offset
PSW_ON_LF_DLY	74h

Diagram



Fields

Field	Description
31-0 DLY_PRE_PSW_ON_LF	Delay from receiving power on lf request to power switch turns on. Locked by LOCK_CFG field.

32.5.3.18 psw on hf delay control (PSW_ON_HF_DLY)

.

Offset

Register	Offset
PSW_ON_HF_DLY	78h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLY_PRE_PSW_ON_HF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLY_PRE_PSW_ON_HF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 DLY_PRE_PSW_ON_HF	Delay from receiving power on lf request to power switch turns on. Locked by LOCK_CFG field.

32.5.3.19 Power switch acknowledge control (PSW_ACK_CTRL_0)

.

Offset

Register	Offset
PSW_ACK_CTRL_0	80h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HF_ACK_...	LF_ACK_...	CNT_MODE		Reserved		PUP_HF_CNT_CFG									
Reset	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						PUP_LF_CNT_CFG									
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

Fields

Field	Description
31 HF_ACK_INVERT	HF Acknowledge value is inverted from power switch control. Locked by LOCK_CFG field.
30 LF_ACK_INVERT	LF Acknowledge value is inverted from power switch control. Locked by LOCK_CFG field.
29-28 CNT_MODE	Configure the acknowledge counter working mode. Locked by LOCK_CFG field. 00 - Not use counter, raise power_on/off done to GPC once get psw ack 01 - Delay after receiving psw ack, delay cycle number is CNT_CFG 10 - Ignore psw ack, raise power_on/off done to GPC when counting to CNT_CFG value 11 - Time out mode, raise power_on/off done to GPC when either psw ack received or counting to CNT_CFG value
27-26 —	Reserved
25-16 PUP_HF_CNT_CFG	PUP HF Count configure. Usage depends on CNT_MODE. Locked by LOCK_CFG field. default value: • Non-core:'h1 (3.125us) • core:'ha0 (0.5ms)
15-10 —	Reserved
9-0 PUP_LF_CNT_CFG	PUP LF Count configure. Usage depends on CNT_MODE. Locked by LOCK_CFG field. default value: • Non-core:'h1 (3.125us) • core:'ha0 (0.5ms)

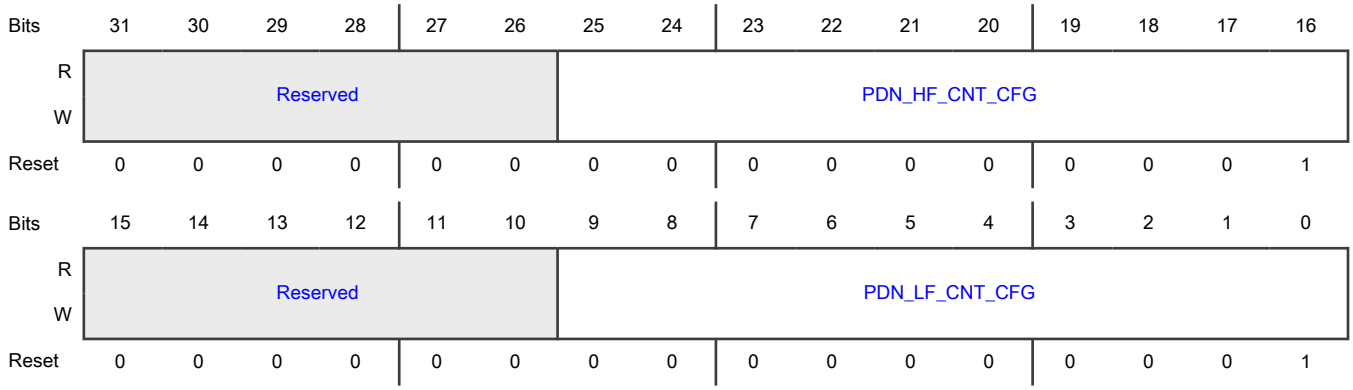
32.5.3.20 Power switch acknowledge control (PSW_ACK_CTRL_1)

.

Offset

Register	Offset
PSW_ACK_CTRL_1	84h

Diagram



Fields

Field	Description
31-26 —	Reserved
25-16 PDN_HF_CNT_CFG	PDN HF Count configure. Usage depends on CNT_MODE. Locked by LOCK_CFG field.
15-10 —	Reserved
9-0 PDN_LF_CNT_CFG	PDN LF Count configure. Usage depends on CNT_MODE. Locked by LOCK_CFG field.

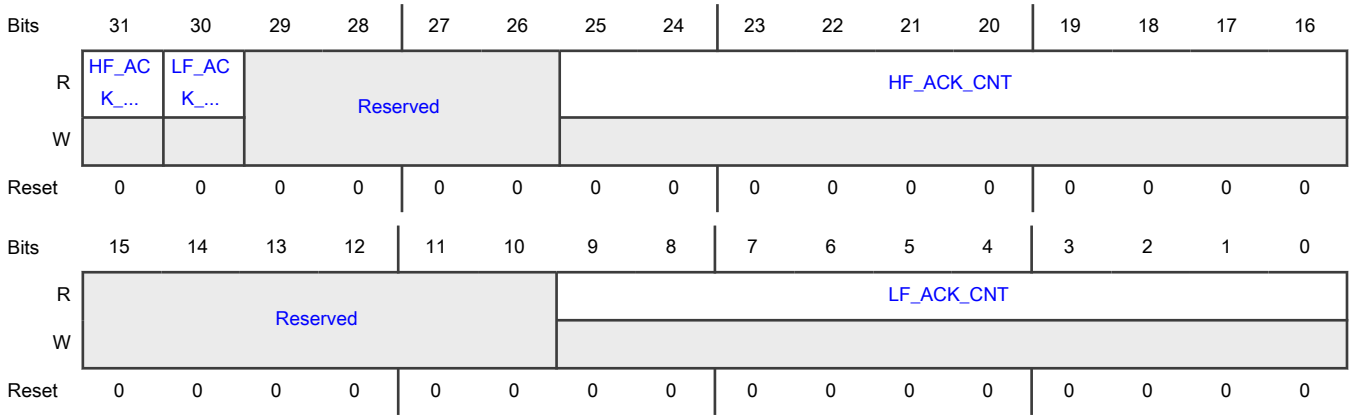
32.5.3.21 PSW acknowledge status (PSW_ACK_STAT)

.

Offset

Register	Offset
PSW_ACK_STAT	88h

Diagram



Fields

Field	Description
31 HF_ACK_STAT	HF PSW acknowledge status
30 LF_ACK_STAT	LF PSW acknowledge status
29-26 —	Reserved
25-16 HF_ACK_CNT	HF PSW acknowledge count Record the delay from power switch change to acknowledge received
15-10 —	Reserved
9-0 LF_ACK_CNT	LF PSW acknowledge count Record the delay from power switch change to acknowledge received.

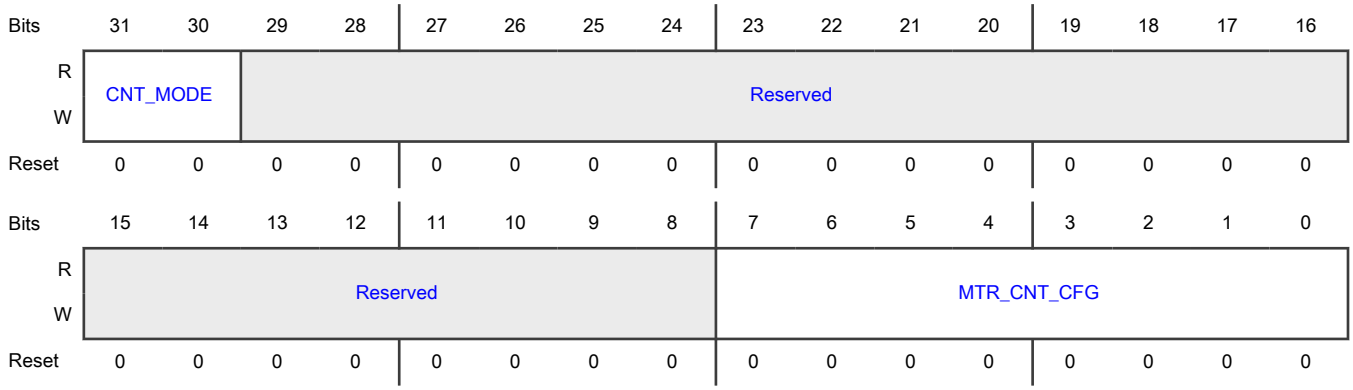
32.5.3.22 MTR acknowledge control (MTR_ACK_CTRL)

MTR acknowledge control

Offset

Register	Offset
MTR_ACK_CTRL	90h

Diagram



Fields

Field	Description
31-30 CNT_MODE	Configure the acknowledge counter working mode. Locked by LOCK_CFG field. 00 - Not use counter, raise mtr done to GPC once get MTR ack 01 - Delay after receiving MTR ack, delay cycle number is CNT_CFG 10 - Ignore MTR ack, raise mtr done to GPC when counting to CNT_CFG value 11 - Time out mode, raise mtr done to GPC when either MTR ack received or counting to CNT_CFG value
29-8 —	Reserved
7-0 MTR_CNT_CFG	MTR count configure. Usage depends on CNT_MODE. Locked by LOCK_CFG field.

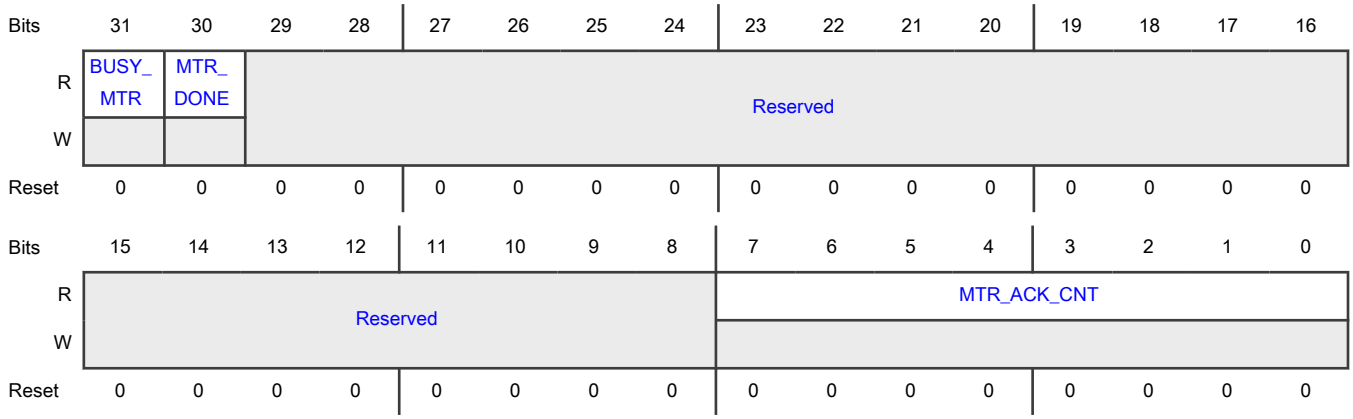
32.5.3.23 MTR acknowledge status (MTR_ACK_STAT)

MTR acknowledge status

Offset

Register	Offset
MTR_ACK_STAT	94h

Diagram



Fields

Field	Description
31 BUSY_MTR	Busy requesting MTR
30 MTR_DONE	Indicate MTR load repair finished
29-8 —	Reserved
7-0 MTR_ACK_CNT	MTR acknowledge count, record the delay from stat change to acknowledge received

32.5.3.24 UPI status 0 (UPI_STAT_0)

.

Offset

Register	Offset
UPI_STAT_0	A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	UPI_POWER_REQUEST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UPI_ISO_REQUEST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 UPI_POWER_REQUEST	CPU mode trans power request of 16 domains
15-0 UPI_ISO_REQUEST	CPU mode trans iso request of 16 domains

32.5.3.25 UPI status 1 (UPI_STAT_1)

.

Offset

Register	Offset
UPI_STAT_1	A4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	UPI_SSAR_REQUEST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UPI_RESET_REQUEST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 UPI_SSAR_REQUEST	CPU mode trans ssar request of 16 domains
15-0 UPI_RESET_REQUEST	CPU mode trans reset request of 16 domains

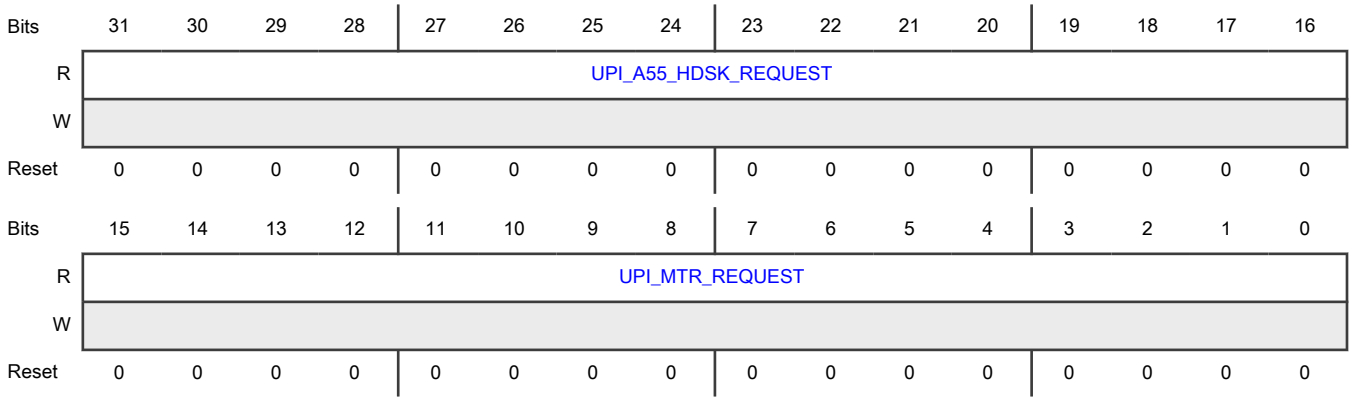
32.5.3.26 UPI status 2 (UPI_STAT_2)

.

Offset

Register	Offset
UPI_STAT_2	A8h

Diagram



Fields

Field	Description
31-16 UPI_A55_HDSK_REQUEST	CPU mode trans A55 handshake request of 16 domains
15-0 UPI_MTR_REQUEST	CPU mode trans mtr request of 16 domains

32.5.3.27 UPI status 3 (UPI_STAT_3)

.

Offset

Register	Offset
UPI_STAT_3	ACh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UPI_MEM_REQUEST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 UPI_MEM_REQUEST	CPU mode trans mem request of 16 domains

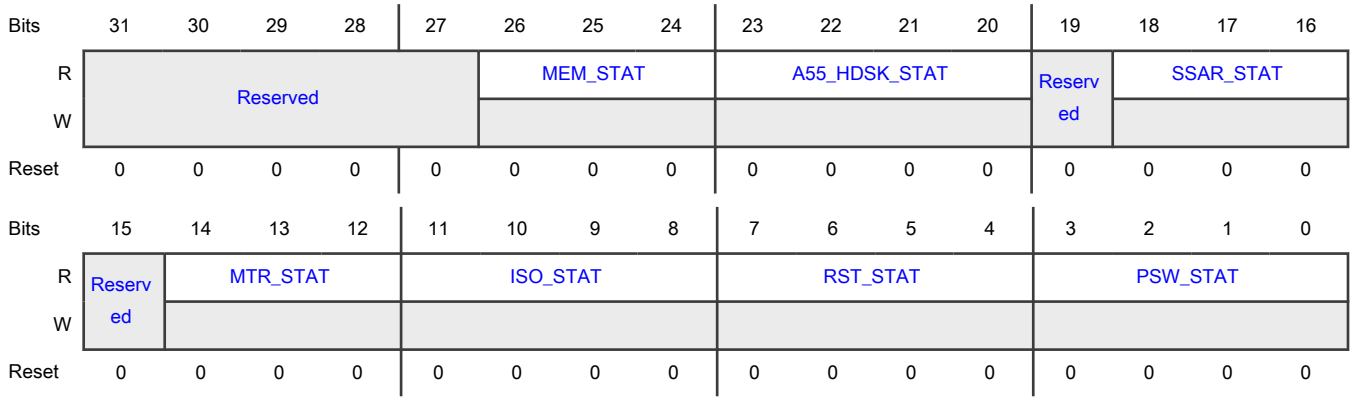
32.5.3.28 FSM status (FSM_STAT)

.

Offset

Register	Offset
FSM_STAT	B0h

Diagram



Fields

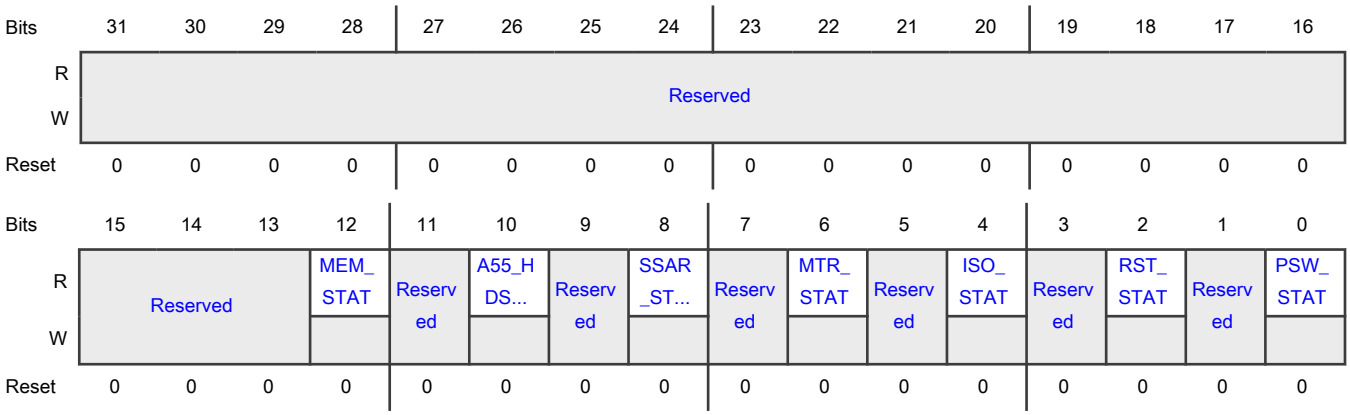
Field	Description
31-27 —	Reserved
26-24 MEM_STAT	Memory FSM status
23-20 A55_HDSK_STAT	A55 handshake FSM status
19 —	Reserved
18-16 SSAR_STAT	SSAR FSM status
15 —	Reserved
14-12 MTR_STAT	MTR FSM status
11-8 ISO_STAT	Isolation FSM status
7-4 RST_STAT	Reset FSM status
3-0 PSW_STAT	Power switch FSM status

32.5.3.29 function status (FUNC_STAT)

Offset

Register	Offset
FUNC_STAT	B4h

Diagram



Fields

Field	Description
31-13 —	Reserved.
12 MEM_STAT	Memory w/ status 0 - No effect or memory w/ exit LP done 1 - Memory w/ enter LP done
11 —	Reserved.
10 A55_HDSK_STAT	A55 handshake status 0 - No effect or power up handshake with A55 done(just for A55 SLICE) 1 - Power down handshake with A55 done(just for A55 SLICE)
9 —	Reserved.
8 SSAR_STAT	ssar status 0 - No effect or power up handshake with Edgelock Enclave done 1 - Power down handshake with Edgelock Enclave done

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 —	Reserved
6 MTR_STAT	MTR status 0 - No effect 1 - Memory repair/trim done
5 —	Reserved.
4 ISO_STAT	Isolation status 0 - Isolation off 1 - Isolation on
3 —	Reserved.
2 RST_STAT	Reset status 0 - Reset assert 1 - Reset release
1 —	Reserved.
0 PSW_STAT	Power switch status 0 - Power switch on 1 - Power switch off

Chapter 33
Battery-Backed Non-Secure Module (BBNSM)

33.1 Chip-specific BBNSM information

Table 159. Reference links to related information

Topic	Related module	Reference
Full description	Arm Cortex CM33 Platform (CM33)	CM33 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The chip includes one instance of BBNSM module.

NOTE

The BBNSM output “pmic_en_b” is active-low, but PMIC_ON_REQ PAD is active-high. There is an inverter between them. PMIC_ON_REQ starts low but is driven high after NVCC_BBSM_1P8 is turned on.

NOTE

Accessing BBNSM registers will block the access of AONMIX peripherals(SAI1, MQS1, CAN1, LPSPI1/LPSPI2, LPI2C1/2, I3C1, TPM1/2) temporarily. For peripherals that need real time, should avoid accessing BBNSM registers when they are working.

33.2 Overview

BBNSM serves as non-volatile logic and storage for the system.

33.2.1 Block diagram

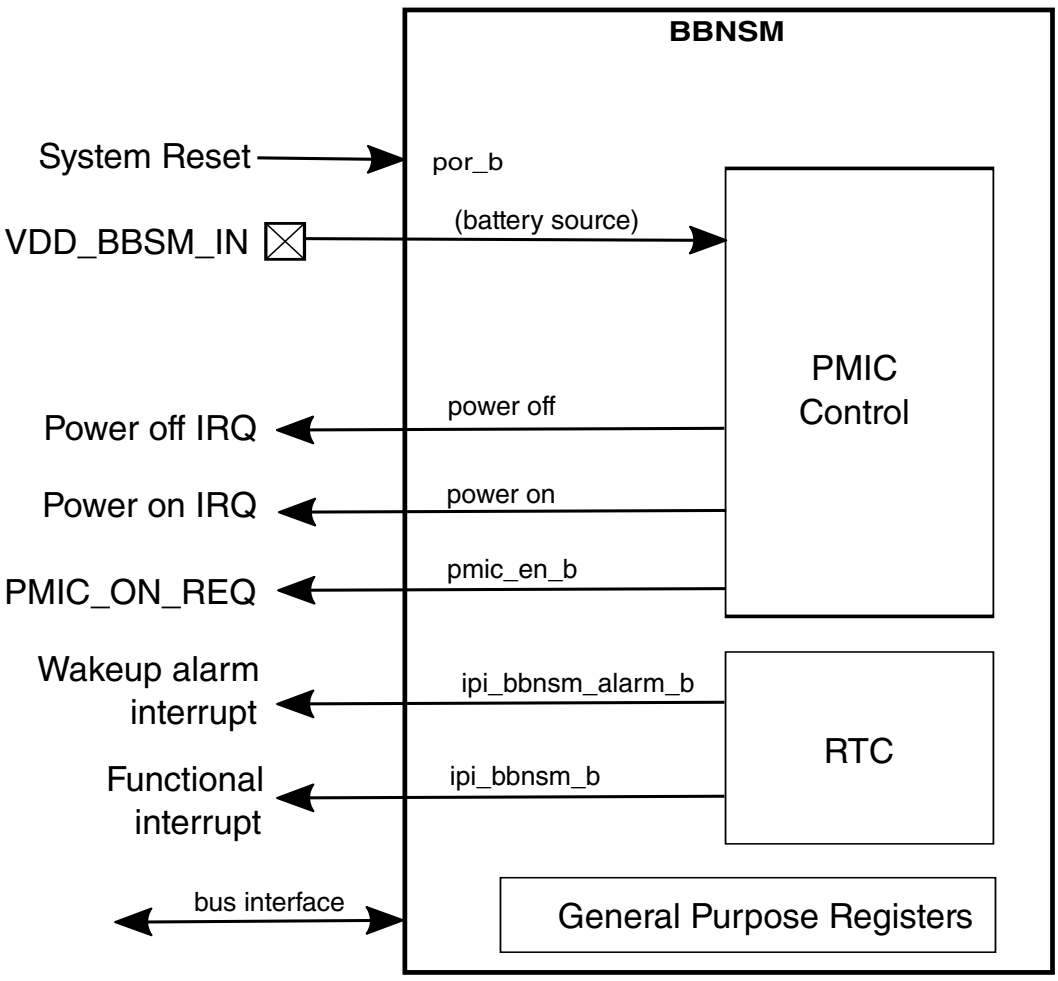


Figure 76. BBNSM block diagram

33.2.2 Features

BBNSM implements the following features:

Table 160. BBNSM feature list

Feature	Description
Non-volatile storage and logic (PMIC logic)	• If the BBNSM power input is connected to an uninterrupted power supply: — BBNSM register values are maintained even when the main SoC is powered off — BBNSM logic continues to operate even when the main SoC is powered off
General-purpose register	• The general-purpose register is available to software to store data. It is 8 registers of 32 bits each by default.

Table continues on the next page...

Table 160. BBNSM feature list (continued)

Feature	Description
Real-time counter (RTC)	• Triggers an interrupt if the RTC reaches its maximum value. • A time alarm can be programmed to generate an interrupt at a specific time.
On/Off Control	• Allows connectivity to a Power Management Integrated Circuit (PMIC) or other voltage regulator devices. • Configurable debounce support for button presses.

33.3 Functional description

The following sections describe functional details of the BBNSM module.

33.3.1 Real-Time Counter (RTC)

BBNSM implements a real-time counter that:

- Retains its state and continues counting even when the main chip is powered down. The BBNSM power input is connected to an uninterrupted power supply
- Generates an interrupt when it reaches the maximum value of all ones. It will then rollover and continue to count
- Generates a functional interrupt request at a particular time by setting the desired time in the Time Alarm register and setting BBNSM_CTRL[TA_EN]
- Can be written at any time
- Can be calibrated so that its count frequency tracks some other time source
- Can be configured to capture the time of a security violation

33.3.1.1 Time Counter Calibration

The real-time counter accuracy may suffer from a drift in the clock, which is used to increment the RTC register. To compensate for this drift, a clock calibration mechanism can adjust the RTC value. It is up to the system processor to decide whether calibration is required or not. If RTC correction is required, enable the mechanism and set the calibration value in the control register. The calibration value (BBNSM_CTRL[CAL_VAL]) includes the sign bit, which is implemented in twos' complement.

If the calibration mechanism is enabled, the calibration value is added or subtracted from the RTC on a periodic basis, once per 32768 cycles of the RTC clock. Positive calibration values will not work with the time alarm feature.

The following table shows the available correction range.

Table 161. Real-time counter calibration settings

Calibration setting (BBNSM_CTRL[CAL_VAL])	Correction in counts per 32768 cycles of the counter clock
01111	+15
...	...
00010	+2
00001	+1
00000	0

Table continues on the next page...

Table 161. Real-time counter calibration settings (continued)

Calibration setting (BBNSM_CTRL[CAL_VAL])	Correction in counts per 32768 cycles of the counter clock
11111	-1
11110	-2
...	...
10001	-15
10000	-16

33.3.1.2 Time Alarm

A time alarm is generated once the most significant 32 bits of the real-time counter match the value in the Time Alarm register. The time alarm can generate interrupts to alert the host processor and can wake up the host processor from one of its low-power modes. This alarm can also wake up the entire system in the power-down mode by asserting the wake-up external output signal.

33.3.2 On/Off Control

The on/off logic inside the BBNSM allows for connecting directly to a PMIC or other voltage regulator device. The module has a button input signal and a wakeup request input signal. It also has two interrupts (power off and power on) and an active-low PMIC enable (pmic_en_b) output. The module can function in two different modes of operation (Dumb PMIC mode and Smart PMIC mode).

33.3.2.1 Smart PMIC Mode

In Smart PMIC mode, BBNSM can generate interrupt outputs (power off and power on) like in Dumb PMIC mode. However, the pmic_en_b signal is a pulse and active-high signal in Smart PMIC mode instead of a level and active-low signal in Dumb PMIC mode. The only configuration option available for this mode is the debounce configuration, which is used for the power off and power on interrupts in the same way as Dumb PMIC mode.

33.3.2.1.1 Smart PMIC Enable Control

The active-high pmic_en_b output asserts if either condition occurs:

- the button input signal asserts
- the wakeup alarm triggers when system power has failed (isolation enable is asserted)

33.3.2.2 Dumb PMIC Mode

In Dumb PMIC mode, BBNSM can generate interrupt outputs (power off and power on) and control the active-low PMIC enable output (pmic_en_b).

33.3.2.2.1 Dumb PMIC Enable Control

The pmic_en_b signal is active (ON state) at reset. In the ON state, the pmic_en_b is deactivated (to OFF state) if either condition occurs:

- the button input signal asserts longer than the BBNSM_CTRL[BTN_TIMEOUT] configuration
- the BBNSM_CTRL[TOSP] bit is set

In the OFF state, the pmic_en_b is activated (to ON state) if either condition occurs:

- the button input signal asserts longer than the BBNSM_CTRL[TURN_ON_TIME] configuration
- the external wakeup request signal asserts

- the wakeup alarm triggers from programming the time alarm and enabling the real-time counter

The behavior of the pmic_en_b signal is described in the state machine below.

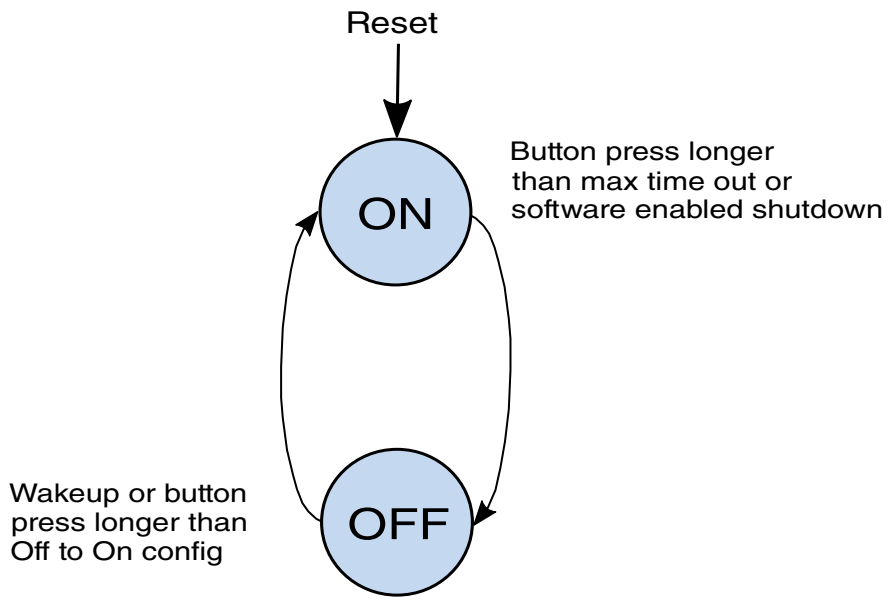


Figure 77. PMIC Enable State Machine (Dumb PMIC Mode)

33.3.3 Clocking

Table 162. BBNSM Clock

Clock	Description
32kHZ RTC clock	Clock is used by BBNSM real-time counter. This clock does not need to be synchronous with other clocks

33.3.4 Reset

Table 163. BBNSM Reset

Reset	Description
por_b	Power-on reset

33.3.5 Interrupts and Alarms

BBNSM provides the following interrupts and alarms:

Table 164. Interrupts and alarms summary

Interrupt/Violation	Event	Default configuration[1]	Configuration options
Functional interrupt (ipi_bbnsm_b)	RTC rollover	Disable	Enable/disable

Table continues on the next page...

Table 164. Interrupts and alarms summary (continued)

Interrupt/Violation	Event	Default configuration[1]	Configuration options
Wakeup alarm interrupt (ipi_bbnsm_alarm_b)	RTC time alarm	Disable	Enable/disable
Power off interrupt	Button input signal	50 msec debounce	Debounce time
Power on interrupt	Button input signal	50 msec debounce	Debounce time

1. Default configuration refers to the state after reset.

33.3.5.1 PMIC Interrupts

The interrupts behave identically in Smart and Dumb PMIC modes.

The power on asserts if:

- system power has failed (isolation enable is asserted), **and**
- the button input signal asserts longer than the BBNSM_CTRL[DEBOUNCE] configuration

The power off asserts if:

- system power has not failed (isolation enable is not asserted), **and**
- the button input signal asserts longer than the BBNSM_CTRL[DEBOUNCE] configuration in the condition of no alarm from BBNSM and tamper event asserted

33.4 External Signals

The table below lists the external signals to the BBNSM.

Table 165. BBNSM external signals

Signal	Description	Direction
ONOFF (btn)	Active-high external on/off button	Input
PMIC_ON_REQ (pmic_en_b)	ONOFF button/PAD has two functions: 1. It can impact the PMIC_ON_REQ value via BBNSM ON-OFF FSM in Dumb PMIC mode or directly pass through the ONOFF button/PAD in Smart PMIC mode. 2. It can generate IRQs which are routed to M33/A55/M7 cores, and also routed to GPC to wake up the system. When PMIC_ON_REQ=1, 1. If ONOFF button was pressed for a "short" time, it can trigger IRQ "set_pwr_off_irq" if it's pressed longer than debounce time the debounce time which is configured in BBNSM_CTRL[19:16]. 2. If you want to toggle PMIC_ON_REQ, then you need to press it longer, the press time length is configured in BBNSM_CTRL[17:16]. When PMIC_ON_REQ=0,	Output

Table continues on the next page...

Table 165. BBNSM external signals (continued)

Signal	Description	Direction
	1. If ONOFF button was pressed for a "short" time, it can trigger IRQ "set_pwr_on_irq" if it's pressed longer than debounce time. The debounce time is configured in BBNSM_CTRL[19:18]. 2. If you want to toggle PMIC_ON_REQ, then you need to press it longer, the press time length is configured in BBNSM_CTRL[21:20].	

33.5 Initialization

There are no initialization steps.

33.6 Memory Map and register definition

This section includes the BBNSM module memory map and detailed descriptions of all registers.

33.6.1 BBNSM register descriptions

33.6.1.1 BBNSM memory map

BBNSM base address: 4444_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	BBNSM Version ID Register (BBNSM_VID)	32	R	0001_013E
4	BBNSM Features Register (BBNSM_FEATURES)	32	R	0000_0020
8	BBNSM Control Register (BBNSM_CTRL)	32	RW	0100_0005
10	BBNSM Interrupt Enable Register (BBNSM_INT_EN)	32	RW	0000_0005
14	BBNSM Events Register (BBNSM_EVENTS)	32	RW	0000_0005
24	BBNSM External Pad Control Register (BBNSM_PAD_CTRL)	32	RW	0000_0000
40	BBNSM Real-Time Counter LS Register (BBNSM_RTC_LS)	32	RW	0000_0000
44	BBNSM Real-Time Counter MS Register (BBNSM_RTC_MS)	32	RW	0000_0000
50	BBNSM Time Alarm Register (BBNSM_TA)	32	RW	0000_0000
300 - 31C	General Purpose Register Word a (GPR0 - GPR7)	32	RW	0000_0000

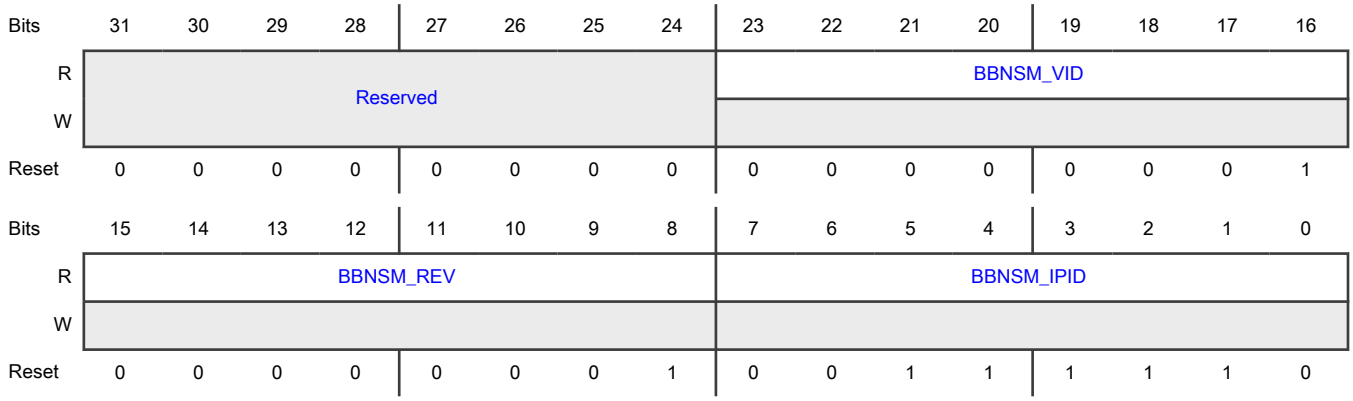
33.6.1.2 BBNSM Version ID Register (BBNSM_VID)

The BBNSM Version ID register can be used by software to differentiate between different versions of the BBNSM and determine what options are implemented in this version of BBNSM. The bit assignments of this register appear below.

Offset

Register	Offset
BBNSM_VID	0h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 BBNSM_VID	BBNSM Version ID Version identifier or major version ID of the BBNSM.
15-8 BBNSM_REV	BBNSM Revision Revision number or minor version ID of the BBNSM.
7-0 BBNSM_IPID	BBNSM IP ID Identifier for the Battery-Backed Non-Secure Module.

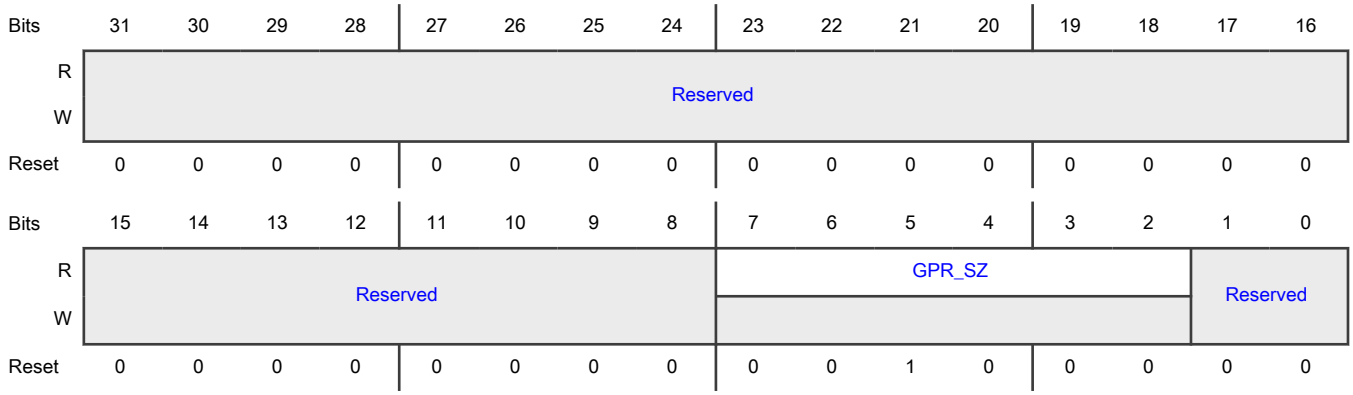
33.6.1.3 BBNSM Features Register (BBNSM_FEATURES)

The BBNSM Features register indicates which optional features are implemented in this version of BBNSM. The bit assignments of this register appear below.

Offset

Register	Offset
BBNSM_FEATURES	4h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-2 GPR_SZ	GPR Register Array Size BBNSM implements a register array of 32-bit words that ELKe firmware can use to retain the GPR values across ELKe power cycles. 00_0000 - This version of BBNSM does not implement a general-purpose register array. All other values - The number of 32-bit words implemented in the general-purpose register array.
1-0 —	Reserved

33.6.1.4 BBNSM Control Register (BBNSM_CTRL)

The BBNSM Control register is used to enable features such as the real-time counter and the time alarm. It can also be used to calibrate the RTC or control the on/off logic for connecting to a PMIC.

Offset

Register	Offset
BBNSM_CTRL	8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				Reserved		TOSP	DP_EN	PK_OVR	PK_EN	TURN_ON_TIME		DEBOUNCE		BTN_TIMEOUT	
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			CAL_VAL					Reserved			CAL_EN	TA_EN		RTC_EN	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-29 —	Reserved
28-26 —	Reserved
25 TOSP	Turn Off System Power When set, a signal is sent to the PMIC to turn off the system power. This bit will clear once power is off. This bit is only valid when Dumb PMIC is enabled. 0 - Leave system power on. 1 - Turn off system power when Dumb PMIC is enabled.
24 DP_EN	Dumb PMIC Enable When set, software can control the system power. When cleared, the system requires a Smart PMIC to automatically turn power off. 0 - Smart PMIC is enabled. 1 - Dumb PMIC is enabled.
23 PK_OVR	PMIC On Request Override The value written to PK_OVR will be asserted on output signal bbnsm_pk_override. That signal is used to override the IOMUX control for the PMIC I/O pad. 0 - PMIC On Request Override is disabled. 1 - PMIC On Request Override is enabled.
22 PK_EN	PMIC On Request Enable The value written to PK_EN will be asserted on output signal bbnsm_pk_en. That signal is used to turn off the pullup/pulldown circuitry in the PMIC I/O pad. 0 - PMIC On Request is disabled.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - PMIC On Request is enabled.
21-20 TURN_ON_TIME	Turn-On Time Amount of time that button is pressed before pmic_en_b is asserted to turn on the SoC power. 00 - 500 milliseconds. 01 - 50 milliseconds. 10 - 100 milliseconds. 11 - 0 milliseconds.
19-18 DEBOUNCE	Debounce Time Amount of debounce time for the BTN input signal. 00 - 50 milliseconds. 01 - 100 milliseconds. 10 - 500 milliseconds. 11 - 0 milliseconds.
17-16 BTN_TIMEOUT	Button Press Timeout Time it takes to hold the button to request a power down in the condition of no alarm from BBNSM and tamper event asserted. 00 - 5 seconds. 01 - 10 seconds. 10 - 15 seconds. 11 - Timeout disabled. Long button presses will not request a power down.
15-13 —	Reserved
12-8 CAL_VAL	Calibration Value Defines signed calibration value for the RTC. This field can be programmed only when RTC calibration is disabled. This is a 5-bit twos' complement value. Hence, the allowable calibration values are in the range from -16 to +15 counts per 32768 ticks of the counter clock. Positive calibration values will not work with the time alarm feature. 0_0000 - +0 counts per each 32768 ticks of the counter clock. 0_0001 - +1 counts per each 32768 ticks of the counter clock. 0_0010 - +2 counts per each 32768 ticks of the counter clock. 0_1111 - +15 counts per each 32768 ticks of the counter clock. 1_0000 - -16 counts per each 32768 ticks of the counter clock. 1_0001 - -15 counts per each 32768 ticks of the counter clock.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1_1110 - -2 counts per each 32768 ticks of the counter clock. 1_1111 - -1 counts per each 32768 ticks of the counter clock.
7-5 —	Reserved
4 CAL_EN	Calibration Enable Enables the RTC calibration mechanism when set. This bit cannot be changed once set. 0 - RTC Time calibration is disabled. 1 - RTC Time calibration is enabled.
3-2 TA_EN	Time Alarm Enable 01 - Disable the time alarm. 10 - Enable the time alarm. A time alarm event occurs if the value in the real-time counter register is equal to the value in the time alarm register.
1-0 RTC_EN	Real-Time Counter Enable 01 - Disable the real-time counter. 10 - Enable the real-time counter.

33.6.1.5 BBNSM Interrupt Enable Register (BBNSM_INT_EN)

The BBNSM Interrupt Enable Register is used to enable interrupt generation when a status event has occurred in BBNSM. Interrupts supported include RTC rollover and time alarm. The bit assignments of this register appear below.

Offset

Register	Offset
BBNSM_INT_EN	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											Reserved	TA_INT_EN		RTC_INT_EN	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-5 —	Reserved
4 —	Reserved
3-2 TA_INT_EN	Time Alarm Interrupt Enable **NOTE** To use TA_INT_EN, enable the RTC_INT_EN. 01 - Do not issue an interrupt when RTC has reached alarm time. The interrupt is cleared when this value is written. 10 - Issue an interrupt when RTC has reached alarm time.
1-0 RTC_INT_EN	Real-Time Counter Rollover Interrupt Enable 01 - Do not issue an interrupt when RTC has rolled over. The interrupt is cleared when this value is written. 10 - Issue an interrupt when RTC has rolled over.

33.6.1.6 BBNSM Events Register (BBNSM_EVENTS)

The BBNSM Events Register indicates whether any features internal to BBNSM is reporting an event. The bit assignments of this register appear below.

Offset

Register	Offset
BBNSM_EVENTS	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						Reserved	Reserved	Reserved	PWR_ON	PWR_OFF	EMG_OFF	TA		RTC_ROLL	
W										W1C	W1C	W1C				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-10 —	Reserved
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 PWR_ON	Set Power On Event This bit is set when the power on interrupt is triggered, which happens when the power button is pressed longer than the configured debounce time (when system power is off). 0 - The power on interrupt has not been requested. 1 - The power on interrupt has been requested.
5 PWR_OFF	Set Power Off Event This bit is set when the power off interrupt is triggered, which happens when the power button is pressed longer than the configured debounce time (when system power is on). 0 - The power off interrupt has not been requested. 1 - The power off interrupt has been requested.
4 EMG_OFF	Emergency Off Event This bit is set when the TOSP bit in BBNSM_CTRL is written or when the power button is pressed longer than the configured button timeout time. 0 - An emergency power off has not been requested. 1 - An emergency power off has been requested.
3-2 TA	Time Alarm Event TA is only cleared when 0x10 is written. 01 - The real-time counter has not reached the alarm time. 10 - The real-time counter has reached the alarm time.
1-0 RTC_ROLL	Real-Time Counter Rollover Event RTC_ROLL is only cleared when 0x10 is written. 01 - The real-time counter has not rolled over. 10 - The real-time counter has rolled over.

33.6.1.7 BBNSM External Pad Control Register (BBNSM_PAD_CTRL)

The BBNSM External Pad Control Register allows configuration and control of I/O pads. The bit assignments of this register appear below.

Offset

Register	Offset
BBNSM_PAD_CTRL	24h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...	PAD_CTR...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 PAD_CTRLn	Control I/O Pads General purpose outputs for configuring and controlling I/O pads. 0 - Deasserts bit n in bbnsm_pad_ctrl[n] 1 - Assert bit n in bbnsm_pad_ctrl[n]

33.6.1.8 BBNSM Real-Time Counter LS Register (BBNSM_RTC_LS)

The BBNSM Real-Time Counter LSB register contains the 32 least-significant bits of the real-time counter.

Offset

Register	Offset
BBNSM_RTC_LS	40h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RTC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RTC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RTC	Real-time Counter The least-significant 32 bits of the RTC.

33.6.1.9 BBNSM Real-Time Counter MS Register (BBNSM_RTC_MS)

The BBNSM Real-Time Counter MS register contains the 15 most-significant bits of the real-time counter.

Offset

Register	Offset
BBNSM_RTC_MS	44h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	RTC														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-15 —	Reserved
14-0 RTC	Real-Time Counter The most-significant 15 bits of the RTC.

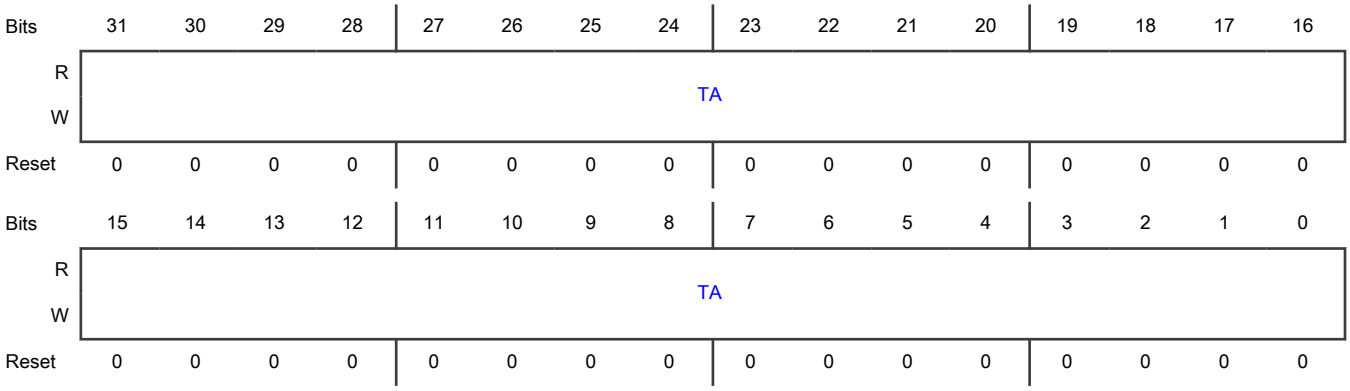
33.6.1.10 BBNSM Time Alarm Register (BBNSM_TA)

The BBNSM Time Alarm register contains the 32-bit value of the time alarm.

Offset

Register	Offset
BBNSM_TA	50h

Diagram



Fields

Field	Description
31-0 TA	Time Alarm Value The 32 bits of the time alarm value.

33.6.1.11 General Purpose Register Word a (GPR0 - GPR7)

The GPR register array holds data whose value is retained across power cycles of the SoC. The words of the array can be used to store general purpose data.

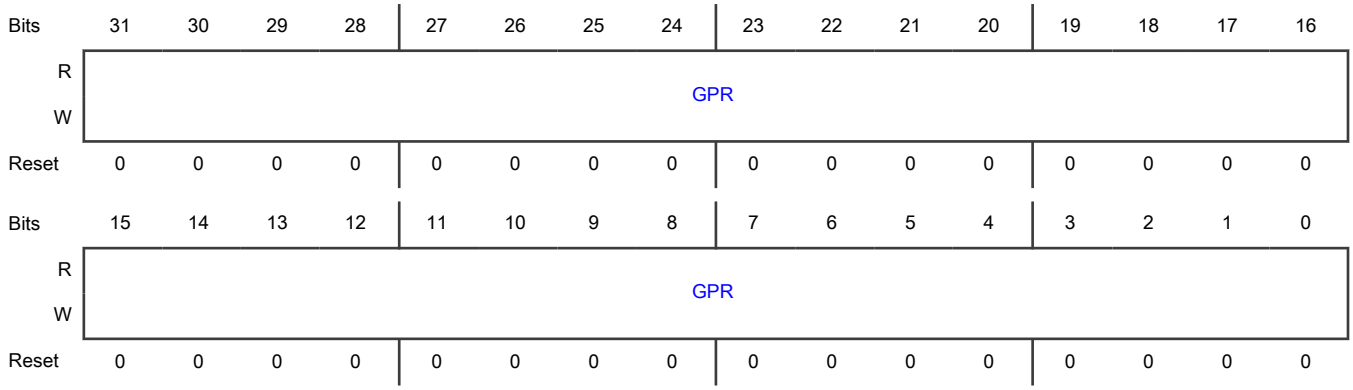
It is up to ELKe firmware to determine how the space in the array is allocated.

Offset

For a = 0 to 7:

Register	Offset
GPRa	300h + (a × 4h)

Diagram



Fields

Field	Description
31-0 GPR	32 bits of the GPR.

Chapter 34
External Memories

34.1 Overview

This chip has these external memory interfaces and controllers:

- High Performance DRAM Interface
- SD/eMMC/SDIO Interface
- Flexible Serial Peripheral Interface (FlexSPI)

34.2 High Performance DRAM Interface (DDR)

The chip includes one 16-bit DRAM controller with LPDDR4x memory technology. Maximum supported capacity is 2GByte.

34.3 eMMC/SD/SDIO

This chip has three Ultra Secured Digital Host Controller (uSDHC) modules for SD/eMMC interface with simultaneous support for all three of 1x eMMC, 1x SD Card and 1x SDIO (for WIFI connectivity). It provides the interface between the host system and the SD/SDIO/eMMC cards.

The key features include:

- Support SD/SDIO standard, up to version 3.0
- Support eMMC standard, up to version 5.1
- Support 3.3V and 1.8V

34.4 Flexible Serial Peripheral Interface

The chip has one FlexSPI block that acts as an interface to one external Octal SPI or Quad SPI serial flash device, each with up to four bidirectional data lines. Both Serial NOR flash and Serial NAND flash are supported (both for boot and runtime operation).

The key features include:

- Channel A can be configured as 1/2/4/8-bit operation
- Support both dual-channel or single-channel operation
- Support both SDR mode and DDR mode
- Support up to 166MHz SDR Mode and 166MHz DDR Mode (with external Flash device DQS input)
- Support up to 133MHz SDR Mode and 66MHz DDR Mode (with internal DQS loopback through DQS pad mode)
- Support On-The-Fly AES Decryption (OTFAD) with 4 keys.

Chapter 35
DDR Memory Controller (DDRC)

35.1 Chip-specific DDR information

Table 166. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing

The chip includes one instance of DDR Controller. DDR4 and DDR3L are not supported. DDR Scrubbing is not supported.

Table 167. DDR Signal Mapping

DDR Signal Name	Signal Name on the chip
MCAA[5:0]	DRAM_CA_A[5:0]
MCK[0]	DRAM_CK_T_A
MCK[0]_b	DRAM_CK_C_A
MCKE[0:1]	DRAM_CKE[0:1]_A
MCS[0:1]_b	DRAM_CS[0:1]_A
MDM[0:1]	DRAM_DMI[0:1]_A
MDQ[0:15]	DRAM_DQ[0:15]_A
DQS[0:1]	DRAM_DQS[0:1]_T_A
MDQS[0:1]_b	DRAM_DQS[0:1]_C_A
ZN	DRAM_ZQ

Table 168. LPDDR4/4X - maximum supported densities

Max Data bus width	Maximum density	Assumed memory organization	Notes
16-bit	16 Gb/2GB	single rank, single channel device with 17-row addresses (R0 - R16)	Byte-mode LPDDR4/4X devices (x16 channel internally split between two dies, x8 each) of any density are not supported therefore, the numbers are applicable only to devices

Table continues on the next page...

Table 168. LPDDR4/4X - maximum supported densities

Max Data bus width	Maximum density	Assumed memory organization	Notes
			with x16 internal organization (referred to as "standard" in the JEDEC specification). The chip also supports dual rank single channel devices therefore, 16Gb/2GB density can be also achieved by using a dual rank single channel device with 16-row addresses (R0 - R15).

35.2 Overview

DDRC includes:

- In-line ECC that detects and corrects single-bit errors for reliable memory operation.
- Dynamic Power Management and Global Auto-Precharge modes that simplify system memory design.
- A large set of special features, including ECC error injection and rapid system debug support.

35.2.1 Block diagram

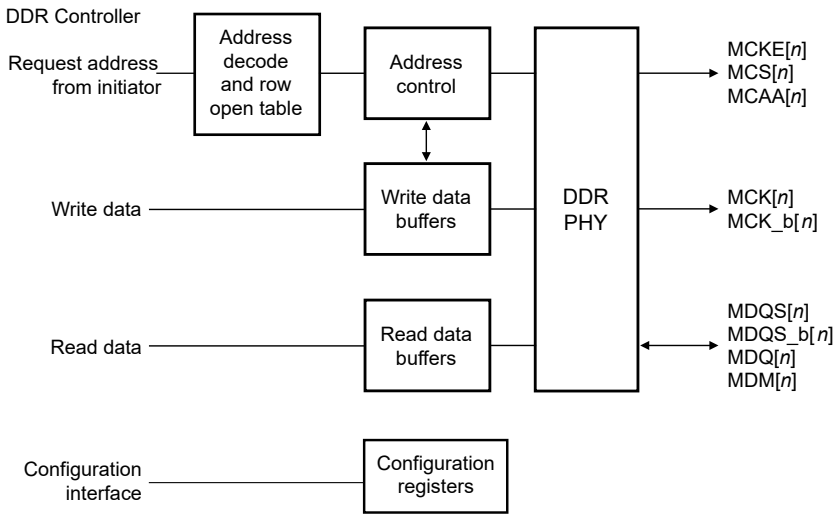


Figure 78. Block diagram

35.2.2 Features

- Support for DDR SDRAM standards:
  - LPDDR4

NOTE

DDRC is designed to support LPDDR4 SDRAMs that are compliant with JESD209-4B as of February 2017.

  - LPDDR4x

NOTE

DDRC is designed to support LPDDR4 SDRAMs that are compliant with JESD209-4B and JESD209-4-1 as of February 2017.

— LPDDR4X

NOTE

DDRC is designed to support LPDDR4X SDRAMs that are compliant with JESD209-4C, January 2020 and JESD209-4-1 as of January 2017.

- Support for the following data-bus widths:
  — 16-bit
- Programmable settings to meet all DDR SDRAM timing parameters
- Support for the following DDR SDRAM configurations:

NOTE

Please refer Chip-specific DDR information section for details on the configurations supported.

  — As many as two ranks, each rank independently addressable. Rank interleaving is also supported.
  — 1-Gbit to 16-Gbit for single channel LPDDR4 devices depending on the internal device configuration
  — 1-Gbit to 16-Gbit for single channel, 32-Gbit for dual channel LPDDR4x devices depending on the internal device configuration
  — x16 data ports
- Data mask signals and read-modify-write operations for subdoubleword writes (a read-modify-write sequence is only necessary when in-line ECC is enabled)
- SECDED in-line ECC
- Open-page management (dedicated entry for each logical bank)
- Automatic DDR SDRAM initialization sequence
- Automatic DDR SDRAM data initialization via the PHY
- Sleep power management
- Error injection
- Performance monitor counters that track the number of occurrences of specific events and generate an interrupt when an event count threshold is exceeded

35.3 Functional description

35.3.1 Overview

DDRC controls processor and I/O interactions with system memory. The memory system allows a wide range of memory devices to be mapped to any arbitrary rank, and provides support for discrete DRAMs .

DDRC receives read and write requests and decodes the transaction address to generate the rank, bank, row, and column addresses. DDRC compares the request's decoded address with values in the row-open table to determine if the address maps to an open page. If the transaction does not map to an open page, DDRC issues an ACTIVATE command.

Programmable parameters allow for a variety of memory organizations and timings.

Optional ECC protection is provided for the DDR SDRAM data bus. DDRC uses ECC to:

- Detect and correct all single-bit errors within the data bus.
- Detect all double-bit errors within the data bus.

DDRC allows you to open a dedicated page for each available DDR SDRAM bank. You can use DDR_SDRAM_INTERVAL[BSTOPRE] to control the amount of time (in clock cycles) that pages remain open.

Error management explains how DDRC handles errors.

35.3.2 Modes of operation

Table 169. Modes of operation

Mode	Description
Dynamic Power Management	In this mode, DDRC can reduce power consumption by negating the DDR SDRAM MCKE signal when no transactions are pending to the DDR SDRAM. This applies to LPDDR4, LPDDR4x only.
Global Auto-Precharge	In this mode, DDRC issues an auto-precharge command with every read or write transaction. To enter this mode: • Write 0 to DDR_SDRAM_INTERVAL[BSTOPRE]. • For separate ranks, write 1 to CS_CONFIG*n*[AP_EN]. See Page-related modes and rank retention for more details.
Open-Page	In this mode, DDRC retains the currently active DDR SDRAM page by not issuing a precharge command. See Page-related modes and rank retention for more details.

35.3.3 Remapping regions

You can use the REMAP_*n*A and REMAP_*n*B registers to remap four regions.

You cannot:

- Overlap regions using REMAP_*n*A and REMAP_*n*B.
- Overlap a mapped region using REMAP_*n*A and REMAP_*n*B with an unmapped region.
- Enable a region for a register if all the regions below that register are enabled as well. This means that:
  - If you write 1 to REMAP_1A[REG_1_REMAP_EN], you must also write 1 to REMAP_0A[REG_0_REMAP_EN].
  - If you write 1 to REMAP_2A[REG_2_REMAP_EN], you must also write 1 to REMAP_0A[REG_0_REMAP_EN] and REMAP_1A[REG_1_REMAP_EN].
  - If you write 1 to REMAP_3A[REG_3_REMAP_EN], you must also write 1 to REMAP_0A[REG_0_REMAP_EN], REMAP_1A[REG_1_REMAP_EN], and REMAP_2A[REG_2_REMAP_EN].

The comparison with Rank n Memory Bounds (CS0_BNDS - CS1_BNDS) occurs after DDRC remaps the AXI addresses mentioned in Remap Region 0B Configuration (REMAP_0B), Remap Region 1B Configuration (REMAP_1B), and Remap Region 2B Configuration (REMAP_2B) to the addresses mentioned in Remap Region 0A Configuration (REMAP_0A), Remap Region 1A Configuration (REMAP_1A) and Remap Region 2A Configuration (REMAP_2A) respectively. You must program Rank n Memory Bounds (CS0_BNDS - CS1_BNDS) to match the addresses after remapping. After remapping, you can access the addresses (programmed in the REMAP_[0-3]A registers) only via the AXI addresses captured in the REMAP_[0-3]B registers. Mapped and nonmapped regions cannot overlap.

ECC is generated based on the AXI address that DDRC receives. DDR_SDRAM_CFG_2[D_INIT] must be 0 so that there is no data initialization in progress, and none scheduled. After writing 1 to DDR_SDRAM_CFG[MEM_EN], execute a write-only memory test on the entire DDR memory region to initialize ECC. If the AXI addresses being remapped are noncontiguous, ensure that the memory test starts from the lowest address range.

For example, if Rank n Memory Bounds (CS0_BNDS - CS1_BNDS) indicates a memory range of 8000_0000h–81FF_FFFFh, program the REMAP registers for mapping 7A00_0000h–7BFF_FFFFh (REMAP_*n*B) to 8000_0000h–81FF_FFFFh (REMAP_*n*A). The ECC calculation uses the AXI address received (7A00_0000h–7BFF_FFFFh).

35.3.4 Timing parameters spanning multiple fields

35.3.4.1 Precharge-to-activate interval (t_{RP})

Calculate t_{RP}, in clock cycles, using this equation:

$$t_{RP} = \text{TIMING_CFG_3[EXT_PRETOACT]} \times 16d + \text{TIMING_CFG_1[PRETOACT]}$$

where TIMING_CFG_1[PRETOACT] must not be 0000

Equation 1. Precharge-to-activate interval (t_{RP})

35.3.4.2 Activate-to-precharge interval (t_{RAS})

Calculate t_{RAS}, in clock cycles, using the following table. The table demonstrates that the minimum value for t_{RAS} is 4.

Table 170. Activate-to-precharge interval (t_{RAS})

When TIMING_CFG_3[EXT_ACTTOPRE] is:	and TIMING_CFG_1[ACTTOPRE] is:	t_{RAS} equals:
000b	0000b	16
	0001b	17
	0010b	18
	0011b	19
	0100b–1111b	TIMING_CFG_1[ACTTOPRE]
001b–111b	0000b–1111b	TIMING_CFG_3[EXT_ACTTOPRE] × 16d + TIMING_CFG_1[ACTTOPRE]

35.3.4.3 Activate-to-read-or-write interval (t_{RCD})

Calculate t_{RCD}, in clock cycles, using this equation:

$$t_{RCD} = \text{TIMING_CFG_3[EXT_ACTTORW]} \times 16d + \text{TIMING_CFG_1[ACTTORW]}$$

where TIMING_CFG_1[ACTTORW] must not be 0

Equation 2. Activate-to-read-or-write interval (t_{RCD})

35.3.4.4 Refresh recovery time (t_{RFC})

Calculate t_{RFC}, in clock cycles, using this equation:

$$t_{RFC} = \text{TIMING_CFG_3[EXT_REFREC]} \times 16d + \text{TIMING_CFG_1[REFREC]} + 8$$

where TIMING_CFG_3[EXT_REFREC] must be less than 11_0000b

Equation 3. Refresh recovery time (t_{RFC})

35.3.4.5 CAS latency

CAS latency is the number of clock cycles between the registration of DDR SDRAM's READ command and the availability of the first output data. For example, if DDRC registers a READ command at clock edge *n* and the latency is *m* clocks, data is available nominally coincident with clock edge *n* + *m*. The largest-possible CAS latency is 44 clock cycles.

Calculate the latency using this equation:

$$\text{CAS latency} = \text{TIMING_CFG_3[EXT_CASLAT]} \times 8d + \text{TIMING_CFG_1[CASLAT]} + 1$$

Equation 4. CAS latency

35.3.4.6 Minimum write recovery interval (n_{WR})

n_{WR} specifies the number of clock cycles from the last data associated with a write command until a precharge command is allowed. Calculate n_{WR} using this equation:

$$n_{WR} = \text{TIMING_CFG_3[EXT_WRREC]} \times 16d + \text{TIMING_CFG_1[WRREC]}$$

Equation 5. Minimum write recovery interval (n_{WR})

35.3.4.7 Minimum MCKE pulse width (t_{CKE})

Calculate t_{CKE}, in clock cycles, using this equation:

$$t_{CKE} = \text{TIMING_CFG_3[EXT_CKE_PLS]} \times 8d + \text{TIMING_CFG_2[CKE_PLS]}$$

Equation 6. Minimum MCKE pulse width (t_{CKE})

35.3.4.8 Four-activate window (t_{FAW})

t_{FAW} is the time during which you can issue no more than four activate commands. Calculate t_{FAW}, in clock cycles, using this equation:

$$t_{FAW} = \text{TIMING_CFG_3[EXT_FOUR_ACT]} \times 64d + \text{TIMING_CFG_2[FOUR_ACT]}$$

where TIMING_CFG_2[FOUR_ACT] must not be 0

Equation 7. Four-activate window (t_{FAW})

35.3.4.9 Precharge power-down exit timing (t_{XP})

t_{XP} defines how long DDRC must wait after exiting precharge power-down before issuing a command to the DDR SDRAM. Calculate t_{XP}, in clock cycles, using this equation:

$$t_{XP} = \text{TIMING_CFG_0[EXT_PRE_PD_EXIT]} \times 16d + \text{TIMING_CFG_0[PRE_PD_EXIT]}$$

where TIMING_CFG_0[PRE_PD_EXIT] must not be 0

Equation 8. Precharge power-down exit timing (t_{XP})

35.3.4.10 Activate-to-activate interval (t_{RRD})

Calculate t_{RRD}, in clock cycles, using this equation:

$$t_{RRD} = \text{TIMING_CFG_3[EXT_ACTTOACT]} \times 16d + \text{TIMING_CFG_1[ACTTOACT]}$$

Equation 9. Activate-to-activate interval (t_{RRD})

where TIMING_CFG_1[ACTTOACT] must not be 0.

35.3.4.11 Write-to-read delay (t_{WTR})

Calculate t_{WTR}, in clock cycles, using this equation:

$$t_{WTR} = \text{TIMING_CFG_3[EXT_WRTORD]} \times 16d + \text{TIMING_CFG_1[WRTORD]}$$

Equation 10. Write-to-read delay (t_{WTR})

where TIMING_CFG_1[WRTORD] must not be 0.

35.3.4.12 Read-to-write turnaround

Calculate the read-to-write turnaround, in clock cycles, using this equation:

$$\text{Read-to-write turnaround} = \text{TIMING_CFG_0[RWT]} + \text{TIMING_CFG_4[EXT_RWT]} \times 4d + CL - WL + (BL \div 2) + 5$$

Equation 11. Read-to-write turnaround

where:

- CL = the CAS latency rounded up to the next integer
- WL = the programmed write latency
- BL = the burst length

35.3.4.13 Write-to-read turnaround

Calculate the write-to-read turnaround, in clock cycles, using this equation:

$$\text{Write-to-read turnaround} = \text{TIMING_CFG_0[WRT]} + \text{TIMING_CFG_4[EXT_RWT]} \times 4d + CL - WL + (BL \div 2) + 1$$

Equation 12. Write-to-read turnaround

where:

- CL = the CAS latency rounded up to the next integer
- WL = the programmed write latency
- BL = the burst length

35.3.4.14 Read-to-read turnaround

Calculate the read-to-read turnaround, in clock cycles, using this equation:

$$\text{Read-to-read turnaround} = \text{TIMING_CFG_0[RRT]} + \text{TIMING_CFG_4[EXT_RRT]} \times 4d + (BL \div 2) + 1$$

Equation 13. Read-to-read turnaround

where BL is the burst length.

35.3.4.15 Write-to-write turnaround

Calculate the write-to-write turnaround, in clock cycles, using this equation:

$$\text{Write-to-write turnaround} = \text{TIMING_CFG_0[WWT]} + \text{TIMING_CFG_4[EXT_WWT]} \times 4d + (BL \div 2)$$

Equation 14. Write-to-write turnaround

where BL is the burst length. The maximum value that can be added to (BL ÷ 2) is 21 clock cycles.

35.3.4.16 Total write latency

Calculate the total write latency, in clock cycles, using this equation:

$$\text{Total write latency} = \text{TIMING_CFG_3[EXT_WR_LAT_2]} \times 32d + \text{TIMING_CFG_2[EXT_WR_LAT]} \times 16d + \text{TIMING_CFG_2[WR_LAT]}$$

Equation 15. Total write latency

Follow these restrictions:

- The total write latency must be:
  - 34 cycles or less.
  - At least four clock cycles.
- Do not write 0b–101b to TIMING_CFG_2[WR_LAT] unless TIMING_CFG_2[EXT_WR_LAT] = 1.

35.3.5 Memory-test address restrictions

DDRC has restrictions for any individual memory test when you use the value of DDR_MTCR[MT_ADDR_EN] to limit the address range to be tested. In this case, neither of the following concatenated values (specified by curly brackets) can be less than the value used during the previous test:

- Starting address {DDR_MT_ST_EXT_ADDR[MT_ST_EXT_ADDR], DDR_MT_ST_ADDR[MT_ST_ADDR]}
- Ending address {DDR_MT_END_EXT_ADDR[MT_END_EXT_ADDR], DDR_MT_END_ADDR[MT_END_ADDR]}

35.3.6 Burst-oriented accesses

Read and write accesses to memory are burst-oriented—accesses start at a selected location and continue for a programmed burst (16 data beats). Accesses to closed pages begin with the registration of an ACTIVATE command, followed by a READ or WRITE command. To access an open page, you do not need an ACTIVATE command.

The address coincident with the ACTIVATE command specifies the logical bank and row to be accessed.

The address coincident with the READ or WRITE command specifies the logical bank and starting column for the burst access.

35.3.7 DDR SDRAM interface operation

DDRC supports the DDR SDRAM configurations shown in Supported DDR SDRAM configurations. The Command Address bus (CA) signals support numerous device densities. The CS signals (see External signals) support up to two ranks of memory. The data path to individual ranks is 16 bits wide.

DDRC supports ranks up to max 16 Gbit per device channel for LPDDR4, LPDDR4X.

The ranks can be constructed using memory devices that are internally organized as x16 memory devices. The MDM signals provide byte selection for memory accesses.

NOTE

A 16-bit DDR SDRAM device has two MDM signals associated with specific halves of the 16 data signals (DQ[0:7] and DQ[8:15]).

When ECC is enabled, DDRC performs all memory accesses on doubleword boundaries (that is, all MDM signals assert simultaneously). However, when ECC is disabled, the memory system uses the MDM signals for byte lane selection when using x16 devices.

35.3.7.1 Relationship of byte lanes to data for x16 devices

Table 171. Relationship of byte lanes to data for x16 devices

Data byte lane	Data bus mask	Data bus strobe	Data bus
0	MDM[0]	MDQS[0] and MDQS_B[0]	MDQ[0:7]
1	MDM[1]	MDQS[1] and MDQS_B[1]	MDQ[8:15]

35.3.7.2 Supported DDR SDRAM configurations

35.3.7.2.1 LPDDR4 LPDDR4x

NOTE

Byte-mode LPDDR4 LPDDR4x devices (x16 channel internally split between two dies, x8 each) of any density are not supported therefore, the numbers are applicable only to devices with x16 internal organization (referred to as "standard" in the JEDEC specification).

NOTE

Dual rank single channel devices are supported therefore, 16Gb/2GB density can be also achieved by using a dual rank single channel device with 16-row addresses (R0 - R15). The tables below always assume single rank configurations.

Table 172. LPDDR4, LPDDR4x x16 device, single 16-bit channel

Device capacity (Gbit)	Device configuration	Row × column × bank bits
1	64 Mbit × 16	13 × 10 × 3
2	128 Mbit × 16	14 × 10 × 3
4	256 Mbit × 16	15 × 10 × 3
8	512 Mbit × 16	16 × 10 × 3
16	1 Gbit × 16	17 × 10 × 3

35.3.7.2.2 Transaction request outside of the programmed address range

If DDRC receives a transaction request and the address does not lie within any of the programmed address ranges for an enabled rank, DDRC reports a memory-select error. See Error management for a detailed description of errors.

35.3.7.2.3 Firmware configures DDRC

System firmware uses the memory-boundary registers to configure DDRC to map the size of each rank in memory.

DDRC uses its bank map to assert the appropriate MCS*n*_b signal for memory accesses according to the provided rank starting and ending addresses. You are not required to map the ranks to a contiguous address space.

35.3.8 DDR SDRAM address multiplexing

This section shows the address bit encodings for the supported DDR SDRAM configurations.

The address and bank address are encoded within the DDR SDRAM command that is presented on the MCAA pins.

NOTE

For LPDDR4, LPDDR4x memories, the DDR SDRAM command includes the MA bits described in this section.

35.3.8.1 LPDDR4 LPDDR4X, 16-bit data bus, no interleaving

Table 173. LPDDR4 LPDDR4X, 16-bit data bus, no interleaving

		High bit	Address from core initiator																														Low bit
Row × col		32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
17 × 10 × 3	MCAA ROW			16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MCAA BA																				2	1	0											
	MCAA COL																							9	8	7	6	5	4	3	2	1	0	
16 × 10 × 3	MCAA ROW				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MCAA BA																				2	1	0											
	MCAA COL																							9	8	7	6	5	4	3	2	1	0	
15 × 10 × 3	MCAA ROW					14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MCAA BA																				2	1	0											
	MCAA COL																							9	8	7	6	5	4	3	2	1	0	
14 × 10 × 3	MCAA ROW						13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MCAA BA																				2	1	0											
	MCAA COL																							9	8	7	6	5	4	3	2	1	0	

Table continues on the next page...

Table 173. LPDDR4 LPDDR4X, 16-bit data bus, no interleaving (continued)

		High bit	Address from core initiator																													Low bit
Row × col		32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
13 x 10 x 3	MCAA ROW							12	11	10	9	8	7	6	5	4	3	2	1	0														
	MCAA BA																				2	1	0											
	MCAA COL																							9	8	7	6	5	4	3	2	1	0	

35.3.9 DDR SDRAM refresh

DDRC supports auto-refresh and self-refresh commands.

Normal operation uses auto-refresh. You control this command by using DDR_SDRAM_INTERVAL[REFINT].

Use self-refresh only when you configure DDRC to enter a sleep power-management state.

The REFINT value:

- Specifies the number of memory bus clock cycles between refresh cycles
- Must accommodate possible outstanding transactions to complete before a refresh request is sent to the memory after the REFINT value is reached

If a memory transaction is in progress when REFINT is reached, the refresh cycle waits for the transaction to complete. In the worst-case scenario, the refresh cycle must wait for the number of bus clock cycles that the longest programmed access requires. Therefore, NXP recommends that the programmed REFINT value be less than what the DDR SDRAM requires. This ensures that a memory transaction does not cause any latency that could violate the device refresh period,

When a refresh cycle is required, DDRC:

1. Completes all current memory requests.
2. Closes all open pages by issuing a PRECHARGE-ALL command to each DDR SDRAM bank with an open page (as indicated by the row open table).
3. Issues one or more auto-refresh commands to each of the DDR SDRAM banks, as identified by their MCS signals, to refresh one row in each bank of the selected rank.

DDRC staggers the auto-refresh commands across the possible banks to reduce the system's instantaneous power requirements. When the memory is populated with two ranks, DDRC issues two sets of auto-refresh commands on consecutive cycles. DDRC also staggers the initial PRECHARGE-ALL commands in two groups for convenience. For this entire refresh sequence, no cycle optimization occurs for the usual case where fewer than two ranks are installed.

When the refresh sequence completes, DDRC initiates any pending memory requests after an inactive period specified by TIMING_CFG_1[REFREC] and TIMING_CFG_3[EXT_REFREC]. DDRC also supports posted refreshes that allow you to use a longer refresh interval.

35.3.9.1 DDR SDRAM refresh timing

You specify the DDR SDRAM refresh timing using TIMING_CFG_1[REFREC].

After reset, you must configure the refresh timing (TIMING_CFG_1[REFREC]) and the extended refresh recovery time (TIMING_CFG_3[EXT_REFREC]) before attempting to access the DDR SDRAM.

35.3.9.2 DDR SDRAM self-refresh and power-saving modes

In normal operation, DDRC issues auto-refresh commands based on the refresh timing (TIMING_CFG_4[EXT_REFINT], DDR_SDRAM_INTERVAL[REFINT]). In low-power modes, DDRC must issue self-refresh commands to preserve the data. You control when DDRC performs a self-refresh by:

- Setting the self-refresh idle threshold (DDR_SR_CNTR[SR_IT]).
- Writing 1 to DDR_SDRAM_CFG_2[FRC_SR].

In the absence of refresh support, you must preserve DDR SDRAM data—for example, by copying the data to disk.

For LPDDR4 LPDDR4x, Dynamic Power Management mode offers tight control of the memory system's power consumption by trading power for performance. This mode uses the MCKE signal to dynamically power down the DDR SDRAM when there is no system memory activity. The MCKE signal deasserts when both of the following conditions are met:

- DDRC has not scheduled any memory refreshes.
- DDRC has not scheduled any memory accesses.

The MCKE signal asserts when a new access or refresh is scheduled. You control this mode by using DDR_SDRAM_CFG[DYN_PWR].

Powering up the DDR SDRAM when a new memory reference is scheduled causes an access latency penalty, depending on:

- Whether you use active or precharge powerdown.
- The settings of TIMING_CFG_0[ACT_PD_EXIT] and TIMING_CFG_0[PRE_PD_EXIT].

35.3.10 DDR SDRAM data beat ordering

DDRC always performs transfers to and from memory in 16-beat bursts. For transfer sizes other than 16 beats, the data transfers still operate as 16-beat bursts. If you enable ECC and either the access is not doubleword aligned or the size is not a multiple of a doubleword (8 byte), DDRC performs a full read-modify-write operation for a write to the DDR SDRAM. If you disable ECC or both accesses are doubleword-aligned with a size that is a multiple of a doubleword, you can use the data-mask signals to prevent the writing of unwanted data to the DDR SDRAM. DDRC also uses data masks to prevent the writing of all unintended full double words to the DDR SDRAM. For example, if you want a one-doubleword (8-byte) write transaction (in 16-bit channel), the remaining 12 beats of data are not written to the DDR SDRAM.

All writes to LPDDR4, LPDDR4x, memory align to beat 0 of the DDR SDRAM.

35.3.11 Page-related modes and rank retention

DDRC manages pages differently in Global Auto-Precharge mode and in Open-Page mode.

In Global Auto-Precharge mode, DDRC uses the DDR SDRAM auto-precharge feature. This feature allows DDRC to indicate that the DDR SDRAM must automatically close the page after the read or write access. This access is performed by using MA[10] of the address during the COMMAND phase of the access to enable auto-precharge. Auto-precharge is nonpersistent—you enable or disable it for each individual READ or WRITE command. You can, however, enable or disable auto-precharge separately for each rank. You disable Global Auto-Precharge mode by either of the following:

- Write 0 to DDR_SDRAM_INTERVAL[BSTOPRE]
- Write 1 to CS_CONFIG[AP_EN]

In Open-Page mode, DDRC allows an open page for each DDR SDRAM rank. In Open-Page mode, DDRC also retains the currently active DDR SDRAM page by not issuing a precharge command. The page remains opens until one of the following conditions occurs:

- The refresh interval completes.
- The time specified in DDR_SDRAM_INTERVAL[BSTOPRE] elapses.
- A bank row collision occurs with another transaction that DDRC just received.

Open-Page mode can dramatically reduce access latencies for page hits. Depending on the memory system design and timing parameters, using Open-Page mode can save several clock cycles for subsequent burst accesses that hit an active page. You can also obtain better performance by using more ranks.

35.3.12 ECC

DDRC supports ECC for the data path between the core and the system memory. The memory detects all double-bit errors and corrects all single-bit errors. It may detect other errors as well but not correct them. Double-bit errors are always reported when error reporting is enabled. When a single-bit error occurs, DDRC:

- Increments ERR_SBE[SBEC].
- Compares ERR_SBE[SBEC] to ERR_SBE[SBET].
- Reports an error if these values are equal.

You can program Single-Bit ECC Memory Error Management (ERR_SBE) in a way that DDRC corrects and ignores minor memory faults, but a catastrophic memory failure generates an interrupt.

For writes that are smaller than 64 bits, DDRC:

1. Performs a double-word read from system memory of the address for the write, checking for errors.
2. Merges the write data with the data read from memory.
3. Generates a new ECC code for the merged double word.
4. Writes the data and ECC code to memory.

If DDRC detects a multi-bit error after the read, the transaction completes the read-modify-write operation to prevent DDRC from hanging. However, the corrupt data is masked on the write, so the original contents in the DDR SDRAM remain unchanged.

35.3.13 Error management

35.3.13.1 Overview

DDRC detects these types of errors:

- Single-bit
- Multi-bit[2]
- Memory-select

This section and its subsections assume that you enabled all the relevant error detection, correction, and reporting functions as described in:

- Memory Error Address Capture (CAPTURE_ADDRESS)
- Memory Error Attributes Capture (CAPTURE_ATTRIBUTES)
- Memory Error Extended Address Capture (CAPTURE_EXT_ADDRESS)
- Memory Data Path Read Capture High (CAPTURE_DATA_HI)
- Memory Data Path Read Capture Low (CAPTURE_DATA_LO)
- Memory Data Path Read Capture ECC (CAPTURE_ECC)
- Memory Error Detect (ERR_DETECT)
- Memory Error Disable (ERR_DISABLE)
- Memory Error Interrupt Enable (ERR_INT_EN)

[2] DDRC uses SECDED ECC, which corrects single-bit errors and detects double-bit errors. Detection of errors of more than two bits is not guaranteed. In this chapter, "multi-bit errors" refers to double-bit errors.

35.3.13.2 Error summary

Table 174. Error summary

Category	Error	Description	Action
Notification	Single-bit ECC threshold	The number of ECC errors has reached the threshold specified in Single-Bit ECC Memory Error Management (ERR_SBE).	If you enable interrupts, DDRC reports the error by generating the appropriate interrupt.
Access error	Multi-bit ECC error	A multi-bit ECC error is detected during a read or read-modify-write memory operation.	
	Memory-select error	Read or write address does not fall within the address range of any of the memory banks.	

35.3.13.3 Single-bit errors

DDRC counts and reports single-bit errors depending on how you program Single-Bit ECC Memory Error Management (ERR_SBE). DDRC does this when it detects a single-bit error:

1. Corrects the data.
2. Increments ERR_SBE[SBEC].
3. Generates a critical interrupt if ERR_SBE[SBEC] = ERR_SBE[SBET].
4. Completes the transaction normally.

35.3.13.4 Multi-bit errors

DDRC does this when it detects a multi-bit error for a read operation:

1. Logs the error in ERR_DETECT[MBE].
2. Optionally generates the machine-check or critical interrupt (if you enable these as described in Memory Error Disable (ERR_DISABLE)).

35.3.13.5 Memory-select errors

These errors occurs when the address from the memory request does not fall into any of the enabled, programmed rank address ranges. DDRC does this when it detects a memory-select error:

1. Logs the error.
2. Optionally generates a critical interrupt (if you enable it as described in Memory Error Detect (ERR_DETECT)).

For all memory-select errors, DDRC does not issue any transactions onto the pins after the first read returns data strobes.Writes are not issued to the ADDR/CMD pins, but reads are issued to the first enabled rank. Write transaction is issued to the DRAM and the read transaction is not issued to the DRAM.

35.3.14 Performance monitoring

Performance monitor counters (PMC0–PMC10) are used to count events specified by the performance monitor local control registers.

- PMC0 (PMC0a and PMC0b) is a 64-bit counter that counts only clock cycles.
- PMC1–PMC10 are 32-bit counters that can monitor 64 counter-specific events in addition to counting 64 reference events.

The following figure shows the performance monitoring components, which consist of:

- One global control register (PMGC0)
- One 64-bit counter (PMC0)

- Ten 32-bit counters (PMC1–PMC10)
- Two local control registers for each counter

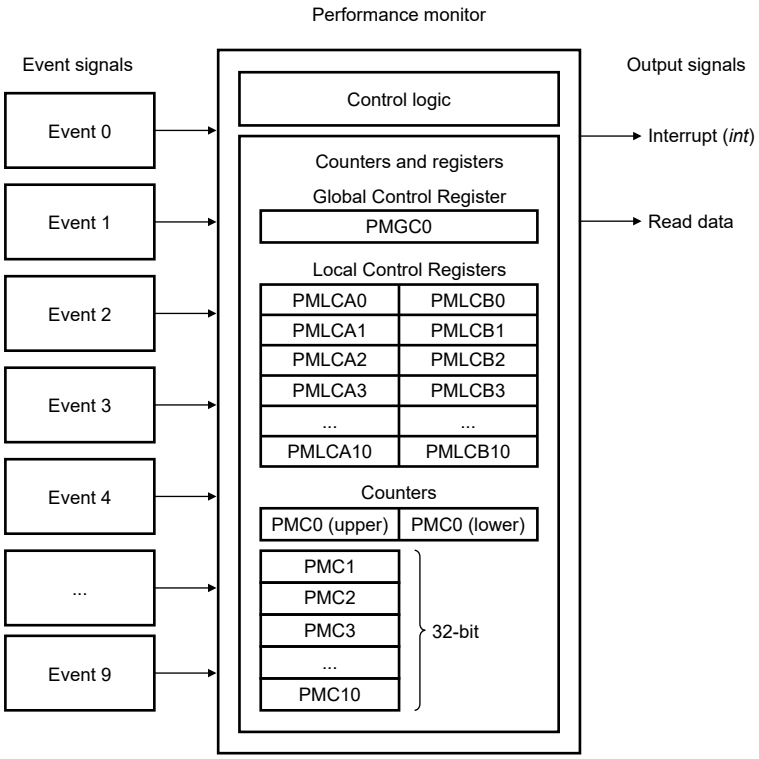


Figure 79. Performance monitor block diagram

The global control register, PMGC0, affects all counters and takes priority over local control registers.

The local control registers are divided into two groups:

- Local control A registers specify:
  - Counter freezing
  - Overflow condition enable
  - Event selection
  - Burstiness

  Local control register PMLCA0, which controls counter PMC0, does not contain event selection because PMC0 counts only cycles.
- Local control B registers:
  - Start and stop triggering
  - Contain the counters' threshold values
  - Contain the threshold multiplier

  Local control register PMLCB0, which controls PMC0, does not contain threshold information because PMC0 only counts cycles.

Performance monitor events are signaled by the functional blocks in DDRC and are selectively recorded in the counters.

- 64 of the events, called reference events, can be counted on any of the ten counters.
- The remaining events, called counter-specific events, can be counted only on the counter where the event is defined.

The performance monitor can generate an interrupt upon a counter overflow. The PMCs can also be programmed to freeze when an interrupt is signaled.

35.3.14.1 Performance monitor interrupt

PMCs can generate an interrupt on an overflow when the most significant bit of a counter transitions to 1. For the interrupt to be signaled, the condition enable field for the counter (PMLCA*n*[CE]) and performance monitor interrupt enable field (PMGC0[PMIE]) must both be 1.

When an interrupt is signaled and the freeze-counters-on-enabled-condition-or-event field (PMGC0[FCECE]) is 1, PMGC0[FAC] is set to 1 by hardware and all of registers are frozen. Software can clear the interrupt condition by resetting the performance monitor and writing 0 to the most significant bit of the counter that generated the overflow.

35.3.14.2 Event counting

PMC1–PMC10 can count the occurrences of specific events. PMC0 is designated to count only clock cycles. However, to provide flexibility, a total of 64 reference events can be counted on any of the 32-bit PMCs (PMC1–PMC10). Additionally, up to 64 unique events can be counted on each 32-bit counter.

You must reset the performance monitor before event-counting sequences. Reset the performance monitor by first freezing one or more counters and then clearing the freeze condition to enable the counters to count according to the register settings. You can freeze counters individually by writing 1 to PMLCA*n*[FC], or simultaneously by writing 1 to PMGC0[FAC]. Writing 0 to these fields enables the performance monitor to begin counting based on the register settings.

Using PMLCA*n*[FC] to reset the performance monitor resets only counter *n*.

35.3.14.3 Threshold events

The threshold feature allows characterization of events that can take a variable number of clock cycles to complete. A threshold event is counted only if the latency is greater than the threshold value specified in PMLCB*n*[THRESHOLD].

There are two types of threshold events:

- Duration threshold events
- Quantity threshold events

35.3.14.3.1 Duration threshold events

For duration threshold events, the 32-bit PMCs increment only when the duration of the event is equal to or greater than the threshold value. The threshold value is scaled by a multiple specified in PMLCB*n*[TBMULT].

A threshold event requires two signals:

- One signal indicates when a threshold event sequence begins.
- A second signal indicates when the event ends.

An internal counter determines when the threshold count is exceeded and when the PMC can increment. This internal counter decrements during a threshold event sequence until it reaches the value of one. A new sequence cannot begin until the current one completes.

Additional threshold start signals are ignored during a sequence until a threshold stop signal occurs. When both a start and stop signal are asserted during the same cycle in a current sequence, the stop terminates the current sequence and the start signals the beginning of a new one. However, if both signals are asserted during the same cycle while not in a current event sequence, both signals are ignored. The following figure is a timing diagram for duration threshold event counting.

Several illegal conditions for threshold counting cause undefined behavior. An illegal condition exists if the threshold value obtained from PMLCB*n*[THRESHOLD] and PMLCB*n*[TBMULT] is less than two. Under these conditions, the intent of threshold counting is ambiguous.

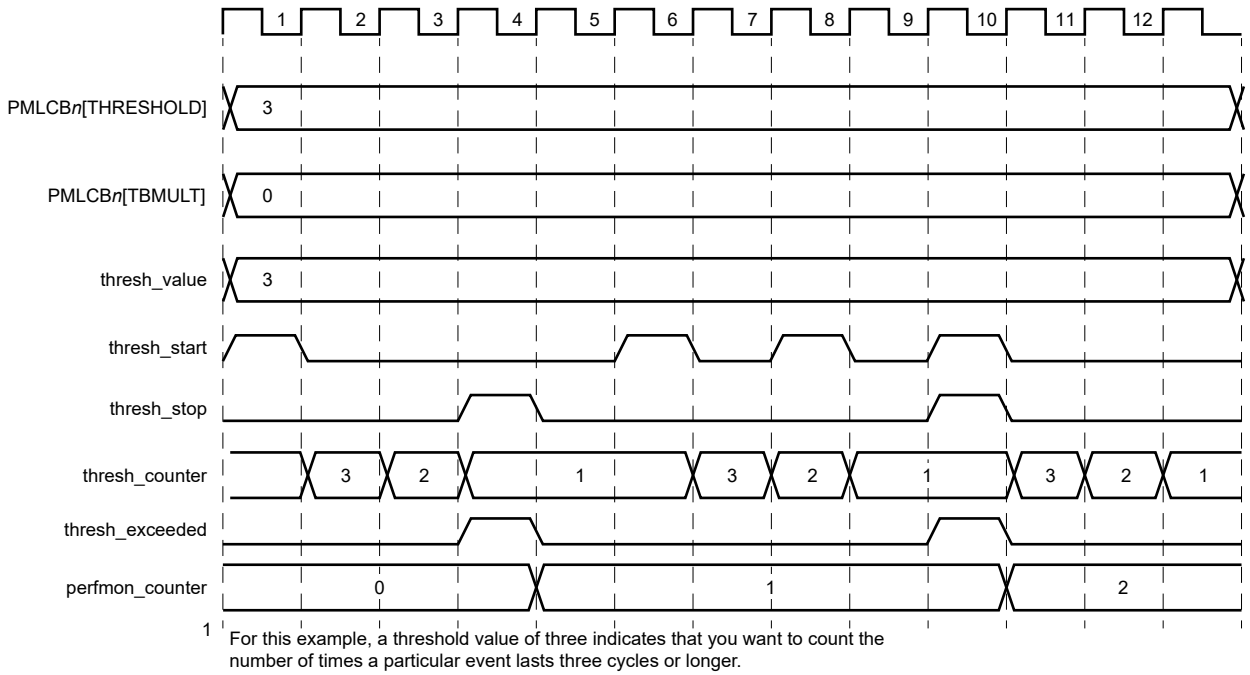
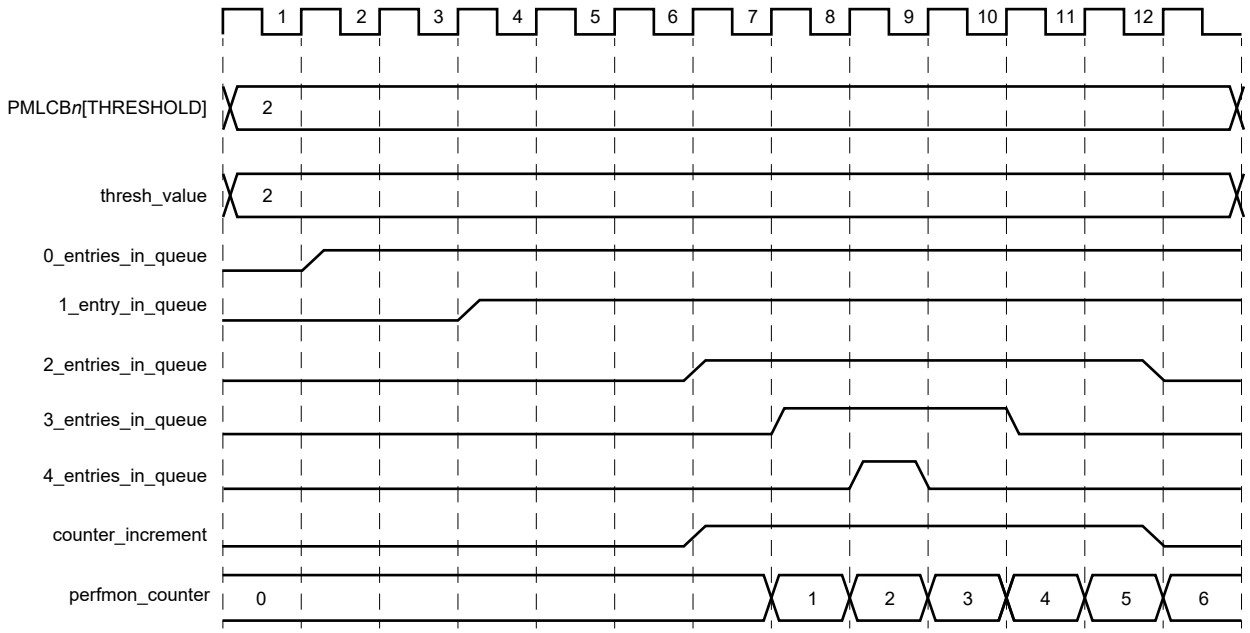


Figure 80. Duration threshold event sequence timing diagram

35.3.14.3.2 Quantity threshold events

For quantity threshold events, the performance monitor counter is only incremented when the specified threshold event exceeds the threshold value. These events do not use the multiplier register field (PMLCB*n*[TBMULT]) like the duration threshold events. This type of threshold event is generally used to monitor the usage of buffers and queues. For example, the usage of a specific queue could be characterized by measuring the amount of time the queue is completely full or partially full. For this example, the threshold field would be used to specify how many entries are required to be valid in the queue for that event to be counted. The following figure shows a timing diagram for quantity threshold event counting.



Note: For this example, a threshold value of two indicates that you want to count the number of cycles. There are at least two valid entries in a particular queue.

Figure 81. Quantity threshold event sequence timing diagram

35.3.14.4 Event signals table

Table 175. Event signals

Event ID	Event type	Counter(s)	PMLCA*n*[EVENT] value[1] (h)	Description
REF_12	Reference	C1–C10	0C	Number of read-modify-writes that are required due to ECC
REF_13	Reference	C1–C10	0D	Number of times a read is reordered ahead of another transaction
REF_14	Reference	C1–C10	0E	Number of times a write is reordered ahead of another transaction
REF_15	Reference	C1–C10	0F	Number of internal controller cycles the DDR controller samples data from the DDR PHY for reads
REF_17	Reference	C1–C10	11	Number of pipelined reads that were page misses
REF_18	Reference	C1–C10	12	Number of pipelined reads or writes that were page misses
REF_19	Reference	C1–C10	13	Number of non-pipelined reads that were page misses
REF_22	Reference	C1–C10	16	Number of non-pipelined reads or writes that were page misses
REF_23	Reference	C1–C10	17	Number of pipelined reads that were page hits
REF_24	Reference	C1–C10	18	Number of pipelined reads or writes that were page hits
REF_25	Reference	C1–C10	19	Number of non-pipelined reads that were page hits
REF_26	Reference	C1–C10	1A	Number of non-pipelined reads or writes that were page hits

Table continues on the next page...

Table 175. Event signals (continued)

Event ID	Event type	Counter(s)	PMLCA*n*[EVENT] value[1] (h)	Description
REF_27	Reference	C1–C10	1B	Number of precharges issued to DRAM that are not caused by a refresh
REF_28	Reference	C1–C10	1C	Number of page misses for oldest transaction
REF_31	Reference	C1–C10	1F	Number of page hits for oldest transaction
REF_59	Reference	C1–C10	3B	Number of precharges including PRE_ALL commands for refreshes
REF_61	Reference	C1–C10	3D	Number of precharges due to bank collision
REF_63	Reference	C1–C10	3F	Number of cycles that commands are stalled due to lack of resources for DRAM access timings
PMC2_72	Counter	C2	48	Number of cycles the scheduler is empty
PMC2_73	Counter	C2	49	Number of filtered write data beats
PMC3_72	Counter	C3	48	Number of cycles the scheduler is full
PMC3_73	Counter	C3	49	Number of filtered read data beats per mask 0
PMC4_72	Counter	C4	48	Number of transactions that load a read for a read-modify-write
PMC4_73	Counter	C4	49	Number filtered read data beats per mask 1
PMC5_72	Counter	C5	48	Number of times medium priority read is loaded to scheduler
PMC5_73	Counter	C5	49	Number of filtered read data beats per mask 2
PMC6_64	Counter	C6	40	Number of cycles interface is blocked for a refresh (only valid for all-bank refresh)
PMC6_72	Counter	C6	48	Number of times low-priority read is loaded to scheduler
PMC7_64	Counter	C7	40	Number of cycles the scheduler is half full
PMC7_65	Counter	C7	41	Number of times a write is loaded to scheduler
PMC8_64	Counter	C8	40	This signal asserts each time the read/write bias is switched for arbitration
PMC8_65	Counter	C8	41	Number of cycles the scheduler is 1/4 full
PMC9_65	Counter	C9	41	Number of cycles the scheduler is 3/4 full
PMC10_65	Counter	C10	41	Number of times high priority read is loaded to scheduler

1. All values not listed are reserved.

35.3.14.5 Chaining

By configuring one counter to increment each time another counter overflows, several counters can be chained together to provide event counts larger than 32 bits. Each counter in a chain adds 32 bits to the maximum count. The register chaining sequence is not arbitrary and is specified indirectly by selecting the register overflow event to be counted. Selecting an event has the effect of selecting a source register because all available chaining events, are dedicated to specific registers.

The chaining overflow event occurs when the counter reaches its maximum value and wraps, not when the register's most significant bit transitions to 1. For this overflow to occur, PMLCA*n*[CE] must be zero to avoid signaling an interrupt when the

counter's most significant bit transitions to 1. Several cycles may be required for the chained counters to reflect the true count because of the internal delay between when an overflow occurs and a counter increments.

35.3.14.6 Triggering

Triggering allows one counter to start or stop counting on the change of another counter or on the overflow of another counter. More specifically, if PMC1 is set to start or stop counting as a result of a change or overflow in counter PMC2, then counter PMC2 must be identified in the local control register of counter PMC1. This is done by appropriately setting the trigger-on select bit or trigger-off select bit (PMLCB1[TRIGONSEL] or PMLCB1[TRIGOFFSEL]). Additionally, the condition that triggers the counter must be selected by configuring the corresponding control bits (PMLCB1[TRIGONCNTL] or PMLCB1[TRIGOFFCNTL]). Assuming the counter is enabled by other control register settings, the counter increments (or freezes) when its specified event occurs after the trigger-on (or off) condition occurs.

When trigger on and trigger off are both selected, the trigger-off condition is ignored until the trigger-on condition has occurred. Furthermore, when a trigger-off condition occurs, the counter state is preserved; it is not restarted by subsequent trigger-on conditions.

Triggering is disabled when the counter's trigger-select bits specify itself as the trigger source. Similarly, triggering is disabled when the trigger control bits are cleared.

35.3.14.7 Burstiness counting

The burstiness counting feature makes it easier to characterize events that occur in rapid succession followed by a relatively long pause. As shown in the table below, event bursts are defined by size, granularity, and distance.

Table 176. Burstiness counting

Parameter	Description	Register field
Size	The minimum number of events constituting a burst	PMLCA*n*[BSIZE]
Granularity	The maximum time between individual events counted as members of the same burst	PMLCA*n*[BGRAN]
Distance	The minimum time between bursts	PMLCA*n*[BDIST] × PMLCB*n*[TBMULT]

The figure below shows the relationships between size, granularity, and distance. Burstiness counting can be performed for all events except threshold events.

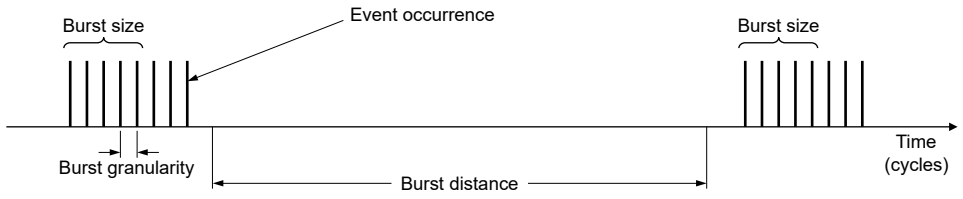


Figure 82. Burst size distance granularity and burstiness counting

The burstiness size field (PMLCA*n*[BSIZE]) specifies the minimum number of event occurrences that constitute a burst. A burst is identified when the number of event occurrences equals or exceeds PMLCA*n*[BSIZE]. Furthermore, these individual event occurrences must be separated by no more clock cycles than the value in the burstiness granularity field (PMLCA*n*[BGRAN]). Although a burst is identified when the minimum number of events occurs, it is not counted until the burst sequence has ended. A burst sequence ends when the specified burstiness granularity is exceeded, at which point the last valid event has occurred for that sequence.

PMLCA*n*[BGRAN] specifies the maximum number of cycles between individual events for them to qualify as members of the same burst sequence.

The burstiness distance field (PMLCA*n*[BDIST]) and threshold/burstiness multiplier field (PMLCB*n*[TBMULT]) specify the acceptable number of cycles between the end of a burst sequence and the beginning of a new sequence for a group of event occurrences to be counted as an individual burst.

The product of the burstiness distance field and the threshold/burstiness multiplier field determine the burstiness distance value used to determine when another burst sequence can begin. Note that the burst distance count begins when a new burst sequence ends and the PMC is incremented. No new burst sequence may begin until the burst distance count has reached zero. After the burst distance count reaches zero, it holds the zero value indicating that a new burst sequence can be counted. The burst distance count begins again when a new burst sequence is identified and counted.

Burstiness counting is disabled when the definition of a burst is ambiguous, that is, when the burst size field is less than two, or the burst distance is zero. When burstiness counting is disabled, regular counting is allowed.

Figure 83 shows that the burst distance is measured from the end of one burst sequence and that a new burst sequence may not begin until the burst distance count expires.

Three internal counters track the different values required for burstiness counting.

- Burstiness size is monitored by a counter. It is loaded with the value specified in the local control register when the burst granularity counter and the burst distance counters reach zero, and no new event is occurring. It always decrements when the following conditions occur: its value is not already zero, an event occurs, and the burst distance count equals zero.
- Burstiness granularity is monitored by a counter that is loaded with the specified value in the local control register on the rising edge of an event occurrence whenever the burst distance count equals zero. The granularity counter is decremented (if it has not already reached zero) when an event is not occurring and burst distance count equals zero.
- Burstiness distance is measured by a counter (burst_dist_ctr) that is loaded with the product of PMLCB*n*[BDIST] and PMLCB*n*[TBMULT] when a burst sequence has been identified and counted. This counter is decremented when burstiness counting is enabled (and the counter has not already reached zero).

A burst is counted at the end of a burst sequence when the three burst parameter counters are all zero. Figure 83 shows a burstiness counting example.

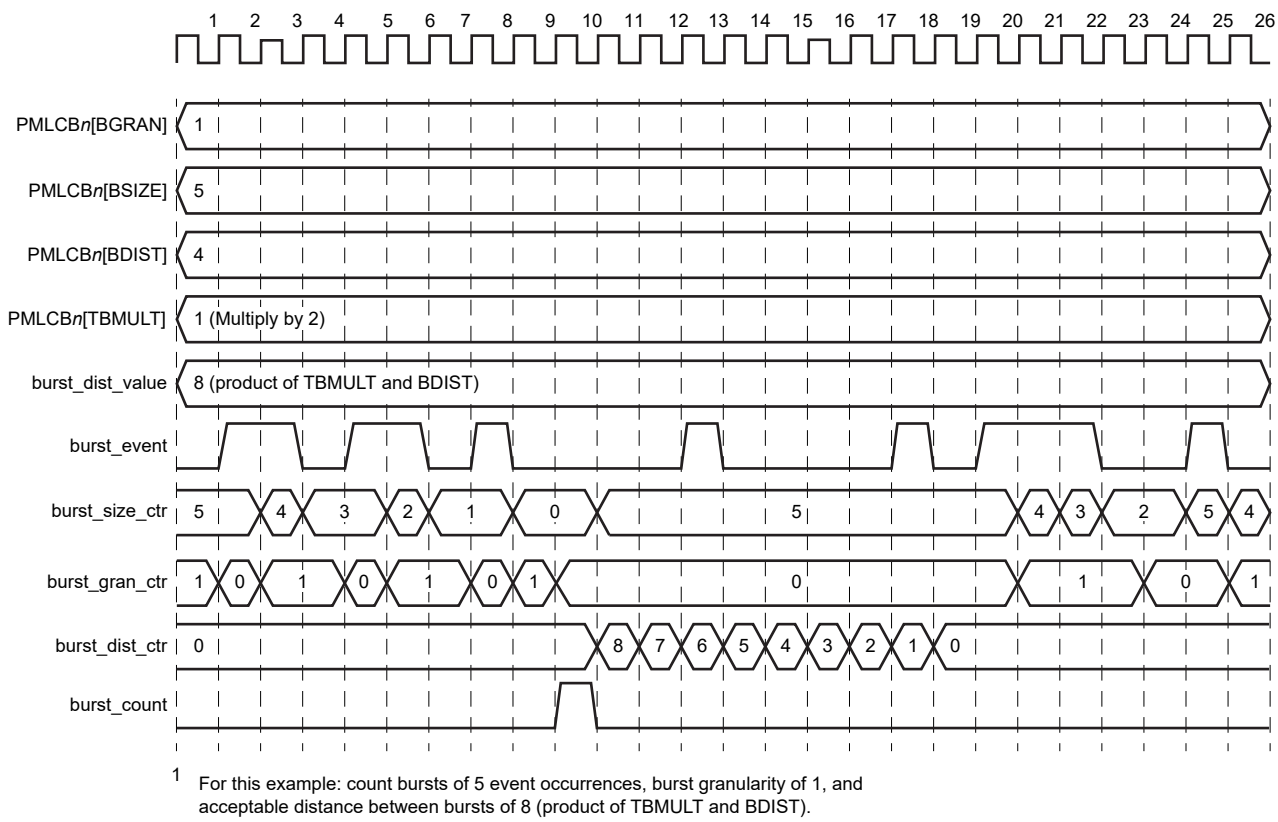


Figure 83. Burstiness counting timing diagram

35.3.15 Clocking

This module has no clocking considerations.

35.3.16 Interrupts

This section describes the interrupts generated by DDR.

Interrupts are generated in the following cases:

- Performance monitor counters generate an interrupt when an event count threshold is exceeded.
- Catastrophic memory failure.
- In case of a single-bit error if ERR_SBE[SBEC] = ERR_SBE[SBET], critical interrupt is generated. See Single-Bit ECC Memory Error Management (ERR_SBE) for details.
- In case of a multi-bit error if ERR_DISABLE[MBED] = 0, ERR_INT_EN[MBEE] and ERR_EN[ECC_EN] = 1, machine-check or critical interrupt is generated. See Memory Error Disable (ERR_DISABLE), Memory Error Interrupt Enable (ERR_INT_EN) and for details.
- In case of Memory-select errors if ERR_DETECT[MBE], ERR_INT_EN[MBEE] and ERR_EN[ECC_EN] = 1
- The performance monitor signals an interrupt when PMGC0[FCECE] = 1
- Writing 1 to PMCn[PMCn] may cause immediate interrupt
- ERR_INT_EN[MSEE, SBEE, MBEE, SSBE12E, REFRATEEE, UPDTMTEE,] enables ECC interrupts or memory-select error interrupts

35.4 External signals

This section provides a summary and details of DDRC external signals.

See the chip data sheet for:

- A pinout diagram showing pin numbers
- Electrical and mechanical specifications

35.4.1 Signal summary LPDDR4 LPDDR4X

Table 177. Signal summary

Name	Description	Reset	Pins	Input or output
MCAA[5:0]	LPDDR4, LPDDR4X address and command	All zeros	6	Output
MCK	DDR SDRAM clock output	0	1	Output
MCK_b	DDR SDRAM clock output (complement)	1	1	Output
MCKE[0:1]	DDR SDRAM clock enable		2	Output
MCS[1:0]_b	CS	All ones	2	Output
MDM[1:0]	Data mask	All zeros	2	Output
MDQ[15:0]	Data bus	All zeros	16	Input or output
MDQS[1:0]	Data strobes	All zeros	2	Input or output
MDQS[1:0]_b	Complement data strobes	All ones	2	Input or output
ZN	Driver impedance calibration	See the chip data sheet	1	Input or output

35.4.2 Signal details LPDDR4 LPDDR4X

Table 178. Signal details

Signals	Description	
LPDDR4 LPDDR4X address and command bus MCAA[5:0]	These signals control DDRC outputs for the DDR SDRAM. MA0 is the least-significant bit of the DDRC address output. A is for channel A.	
	Input	—
	Output	These signals represent the address that DDRC drives. They contain different portions of the address depending on the memory size and the DDR SDRAM command that DDRC issues. See DDR SDRAM address multiplexing for a complete mapping of these signals. When DDRC is enabled, it always drives the address, which is valid when a transaction is driven to DDR SDRAM. The signals are in the high-impedance state when DDRC is disabled.
DDR SDRAM clock outputs	These signals comply with the JEDEC DDR SDRAM specifications, which require true and complement clocks. The DDR SDRAM sees a clock edge when the true and complement signal waveforms cross.	

Table continues on the next page...

Table 178. Signal details (continued)

Signals	Description	
• MCK • MCK_b	Input	—
	Output	These signals provide the DDR SDRAM clock output and its complement.
Clock enable MCKEA	This signal enables the clock for the DDR SDRAM. You can deassert it to stop clocking the DDR SDRAM. You must connect this signal for the same memory rank as the corresponding MCK and MCK_b pair.	
	Input	—
	Output	Assertion of this signal indicates that clocking for the DDR SDRAM is enabled. Deassertion of this signal indicates that clocking for the DDR SDRAM is disabled and the DDR SDRAM must ignore MCK or MCK_b signal transitions. MCK and MCK_b become irrelevant during MCKE deassertion. This signal asserts when you write 1 to DDR_SDRAM_CFG[MEM_EN]. DDRC: • Negates this signal when DDRC enters Dynamic Power Management mode or performs a self-refresh. • Asserts this signal again when DDRC exits Dynamic Power Management mode or finishes a self-refresh.
Chip selects MCSA[1:0]_b	These signals select a physical DDR SDRAM rank to perform a memory operation as described in Rank n Memory Bounds (CS0_BNDS - CS1_BNDS) and Rank n Configuration (CS0_CONFIG - CS1_CONFIG).	
	Input	—
	Output	DDRC asserts these signals to begin a memory cycle and negates them when no DDR SDRAM action occurs during the current cycle. The transaction must follow the timing constraints defined in the TIMING_CFG_*n* registers.
DDR SDRAM data output masks MDM[1:0]	These signals mask unwanted bytes of data transferred during a write. You need these signals to support sub-burst-size transactions (such as single-byte writes) on DDR SDRAM where all input and output occurs in multi-byte bursts. MDM0 corresponds to the least-significant byte and MDM1 corresponds to the most-significant byte. Relationship of byte lanes to data for x16 devices shows byte lane encodings. The timing of these signals is the same as that of MDQ*n*.	
	Input	—
	Output	Asserting these signals prevents writing to the DDR SDRAM. Deasserting these signals allows you to read or write the corresponding byte in the DDR SDRAM.
Data bus MDQ[15:0]	These signals provide a bidirectional data bus for DDRC's input and output signals.	

Table continues on the next page...

Table 178. Signal details (continued)

Signals	Description	
	When these signals are in the high-impedance state, no READ or WRITE commands are in progress, and neither DDRC nor the DDR SDRAM drives any data.	
	Input	The signals represent the data that the external DDR SDRAMs drive during a READ transaction.
	Output	The signals represent the data that DDRC drives when writing to the DDR SDRAM.
Data strobes • MDQS[1:0] • MDQS[1:0]_b	These are differential signals that indicate when the MDQ signals convey valid data. See DDR SDRAM interface operation for byte-lane assignments.	
	Input	These signals are driven high when positive capture data is received and driven low when negative capture data is received. If a READ command is registered at clock edge *n*, and you program the latency in TIMING_CFG_1[CASLAT] to be *m* clocks, data strobes at the DDR SDRAM assert coincidentally with the data on clock edge *n*+*m*.
	Output	These signals are driven high when positive capture data is transmitted and driven low when negative capture data is transmitted. If a WRITE command is registered at clock edge *n*, data strobes at the DDR SDRAM assert centered in the data eye on clock edge *n*+1.
Driver impedance calibration ZN	This signal is used for automatic driver calibration. See the chip data sheet for the proper value and connection of the ZN resistor.	
	Input	DDRC drives this signal for four DDR SDRAM cycles at a time when DDRC executes the automatic driver compensation.
	Output	

35.5 Initialization

35.5.1 Fields to initialize at startup

Your boot code must configure the following fields at system reset or startup.

Table 179. Fields to initialize at startup

Register	Fields
Rank n Memory Bounds (CS0_BNDS - CS1_BNDS)	• SA • EA
Rank n Configuration (CS0_CONFIG - CS1_CONFIG)	• CS_EN • AP_EN • ROW_BITS_CS • BG_BITS_CS • COL_BITS_CS

Table continues on the next page...

Table 179. Fields to initialize at startup (continued)

Register	Fields
DDR SDRAM Timing Configuration 3 (TIMING_CFG_3)	• EXT_PRETOACT • EXT_ACTTOPRE • EXT_ACTTORW • EXT_REFREC • EXT_CASLAT • EXT_WRREC • EXT_ACTTOACT • EXT_WRTORD
DDR SDRAM Timing Configuration 0 (TIMING_CFG_0)	• RWT • WRT • RRT • WWT • ACT_PD_EXIT • PRE_PD_EXIT • MRS_CYC
DDR SDRAM Timing Configuration 1 (TIMING_CFG_1)	• PRETOACT • ACTTOPRE • ACTTORW • CASLAT • REFREC • WRREC • ACTTOACT • WRTORD
DDR SDRAM Timing Configuration 2 (TIMING_CFG_2)	• WR_LAT • RD_TO_PRE • CKE_PLS • FOUR_ACT
DDR SDRAM Control Configuration (DDR_SDRAM_CFG)	• SREN • ECC_EN • SDRAM_TYPE • DYN_PWR • DBW

Table continues on the next page...

Table 179. Fields to initialize at startup (continued)

Register	Fields
	• BA_INTLV_CTL • HSE • BI
DDR SDRAM Control Configuration 2 (DDR_SDRAM_CFG_2)	• NUM_PR • AP_EN • D_INIT
DDR SDRAM Interval Configuration (DDR_SDRAM_INTERVAL)	• REFINT • BSTOPRE
DDR SDRAM Data Initialization (DDR_DATA_INIT)	INIT_VALUE
DDR SDRAM Timing Configuration 4 (TIMING_CFG_4)	• RWT • WRT • RRT • WWT • EXT_RWT • EXT_WRT • EXT_RRT • EXT_WWT • EXT_REFINT • DLL_LOCK
DDR SDRAM Timing Configuration 7 (TIMING_CFG_7)	• CKE_RST • CKSRE • CKSRX
DDR SDRAM Timing Configuration 8 (TIMING_CFG_8)	• RWT_BG • WRT_BG • RRT_BG • WWT_BG • ACTTOACT_BG • WRTORD_BG • PRE_ALL_REC
DDR SDRAM ZQ Calibration Control (DDR_ZQ_CNTL)	• ZQ_EN • ZQINIT • ZQOPER

Table continues on the next page...

Table 179. Fields to initialize at startup (continued)

Register	Fields
	• ZQCS
DDR SDRAM Self-Refresh Counter (DDR_SR_CNTR)	SR_IT
DDR SDRAM Control Configuration 3 (DDR_SDRAM_CFG_3)	• DDRC_RST • WR_PRE • RD_PRE • DM_CFG

35.6 DDRC register descriptions

NOTE

Unimplemented registers within this address space read as zero and do not return an error. Writes have no effect.

35.6.1 DDRC memory map

ddr_ctrl.DDRC base address: 4E30_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Rank 0 Memory Bounds (CS0_BNDS)	32	RW	0000_0000
8	Rank 1 Memory Bounds (CS1_BNDS)	32	RW	0000_0000
20	Remap Region 0A Configuration (REMAP_0A)	32	RW	0000_0000
24	Remap Region 0B Configuration (REMAP_0B)	32	RW	0000_0000
28	Remap Region 1A Configuration (REMAP_1A)	32	RW	0000_0000
2C	Remap Region 1B Configuration (REMAP_1B)	32	RW	0000_0000
30	Remap Region 2A Configuration (REMAP_2A)	32	RW	0000_0000
34	Remap Region 2B Configuration (REMAP_2B)	32	RW	0000_0000
38	Remap Region 3A Configuration (REMAP_3A)	32	RW	0000_0000
3C	Remap Region 3B Configuration (REMAP_3B)	32	RW	0000_0000
40	DDRC Address Decode 0 (DDR_ADDR_DEC_0)	32	RW	0000_0000
44	DDRC Address Decode 1 (DDR_ADDR_DEC_1)	32	RW	0000_0000
48	DDRC Address Decode 2 (DDR_ADDR_DEC_2)	32	RW	0000_0000
4C	DDRC Address Decode 3 (DDR_ADDR_DEC_3)	32	RW	0000_0000
50	DDRC Address Decode 4 (DDR_ADDR_DEC_4)	32	RW	0000_0000
54	DDRC Address Decode 5 (DDR_ADDR_DEC_5)	32	RW	0000_0000
58	DDRC Address Decode 6 (DDR_ADDR_DEC_6)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
5C	DDRC Address Decode 7 (DDR_ADDR_DEC_7)	32	RW	0000_0000
60	DDRC Address Decode 8 (DDR_ADDR_DEC_8)	32	RW	0000_0000
64	DDRC Address Decode 9 (DDR_ADDR_DEC_9)	32	RW	0000_0000
80 - 84	Rank n Configuration (CS0_CONFIG - CS1_CONFIG)	32	RW	0000_0000
100	DDR SDRAM Timing Configuration 3 (TIMING_CFG_3)	32	RW	0000_0000
104	DDR SDRAM Timing Configuration 0 (TIMING_CFG_0)	32	RW	0011_0005
108	DDR SDRAM Timing Configuration 1 (TIMING_CFG_1)	32	RW	0000_0000
10C	DDR SDRAM Timing Configuration 2 (TIMING_CFG_2)	32	RW	0000_0000
110	DDR SDRAM Control Configuration (DDR_SDRAM_CFG)	32	RW	0700_0000
114	DDR SDRAM Control Configuration 2 (DDR_SDRAM_CFG_2)	32	RW	0000_0000
120	DDR SDRAM Mode Control (DDR_SDRAM_MD_CNTL)	32	RW	0000_0000
124	DDR SDRAM Interval Configuration (DDR_SDRAM_INTERVAL)	32	RW	0000_0000
128	DDR SDRAM Data Initialization (DDR_DATA_INIT)	32	RW	0000_0000
160	DDR SDRAM Timing Configuration 4 (TIMING_CFG_4)	32	RW	0000_0000
16C	DDR SDRAM Timing Configuration 7 (TIMING_CFG_7)	32	RW	0000_0000
170	DDR SDRAM ZQ Calibration Control (DDR_ZQ_CNTL)	32	RW	0000_0000
17C	DDR SDRAM Self-Refresh Counter (DDR_SR_CNTR)	32	RW	0000_0000
250	DDR SDRAM Timing Configuration 8 (TIMING_CFG_8)	32	RW	0000_0000
254	DDR SDRAM timing configuration 9 (TIMING_CFG_9)	32	RW	0000_0000
258	DDR SDRAM Timing Configuration 10 (TIMING_CFG_10)	32	RW	0000_0000
25C	DDR SDRAM Timing Configuration 11 (TIMING_CFG_11)	32	RW	0000_0000
260	DDR SDRAM Control Configuration 3 (DDR_SDRAM_CFG_3)	32	RW	0000_0000
264	DDR SDRAM Control Configuration 4 (DDR_SDRAM_CFG_4)	32	RW	0000_0000
2C0	DDR Refresh Rate (DDR_SDRAM_REF_RATE)	32	R	0000_0303
300	DDR SDRAM Timing Configuration 12 (TIMING_CFG_12)	32	RW	0000_0000
304	DDR SDRAM Timing Configuration 13 (TIMING_CFG_13)	32	RW	0000_0000
308	DDR SDRAM Timing Configuration 14 (TIMING_CFG_14)	32	RW	0000_0000
800	Transaction Configuration Register 1 (TX_CFG_1)	32	RW	43B3_0002
B24	DDR SDRAM Debug Status 2 (DDRDSR_2)	32	RW	C000_0000
BF8	DDRC Revision 1 (DDR_IP_REV1)	32	R	0002_050A
D00	DDR SDRAM Memory Test Control (DDR_MTCR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
D20 - D44	DDR SDRAM Memory Test Pattern n (DDR_MTP0 - DDR_MTP9)	32	RW	0000_0000
D60	DDR SDRAM Memory Test Start Extended Address (DDR_MT_ST_EXT_ADDR)	32	RW	0000_0000
D64	DDR SDRAM Memory Test Start Address (DDR_MT_ST_ADDR)	32	RW	0000_0000
D68	DDR SDRAM Memory Test End Extended Address (DDR_MT_END_EXT_ADDR)	32	RW	0000_0000
D6C	DDR SDRAM Memory Test End Address (DDR_MT_END_ADDR)	32	RW	0000_0000
E00	Performance Monitor Global Control (PMGC0)	32	RW	0000_0000
E10	Performance Monitor Local Control A0 (PMLCA0)	32	RW	0000_0000
E14	Performance Monitor Local Control B0 (PMLCB0)	32	RW	0000_0000
E18	PMC 0a (PMC0a)	32	RW	0000_0000
E1C	PMC 0b (PMC0b)	32	RW	0000_0000
E20	Performance Monitor Local Control A (PMLCA1)	32	RW	0000_0000
E24	Performance Monitor Local Control B (PMLCB1)	32	RW	0000_0000
E28	Performance Monitor Counter (PMC1)	32	RW	0000_0000
E30	Performance Monitor Local Control A (PMLCA2)	32	RW	0000_0000
E34	Performance Monitor Local Control B (PMLCB2)	32	RW	0000_0000
E38	Performance Monitor Counter (PMC2)	32	RW	0000_0000
E40	Performance Monitor Local Control A (PMLCA3)	32	RW	0000_0000
E44	Performance Monitor Local Control B (PMLCB3)	32	RW	0000_0000
E48	Performance Monitor Counter (PMC3)	32	RW	0000_0000
E50	Performance Monitor Local Control A (PMLCA4)	32	RW	0000_0000
E54	Performance Monitor Local Control B (PMLCB4)	32	RW	0000_0000
E58	Performance Monitor Counter (PMC4)	32	RW	0000_0000
E60	Performance Monitor Local Control A (PMLCA5)	32	RW	0000_0000
E64	Performance Monitor Local Control B (PMLCB5)	32	RW	0000_0000
E68	Performance Monitor Counter (PMC5)	32	RW	0000_0000
E70	Performance Monitor Local Control A (PMLCA6)	32	RW	0000_0000
E74	Performance Monitor Local Control B (PMLCB6)	32	RW	0000_0000
E78	Performance Monitor Counter (PMC6)	32	RW	0000_0000
E80	Performance Monitor Local Control A (PMLCA7)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
E84	Performance Monitor Local Control B (PMLCB7)	32	RW	0000_0000
E88	Performance Monitor Counter (PMC7)	32	RW	0000_0000
E90	Performance Monitor Local Control A (PMLCA8)	32	RW	0000_0000
E94	Performance Monitor Local Control B (PMLCB8)	32	RW	0000_0000
E98	Performance Monitor Counter (PMC8)	32	RW	0000_0000
EA0	Performance Monitor Local Control A (PMLCA9)	32	RW	0000_0000
EA4	Performance Monitor Local Control B (PMLCB9)	32	RW	0000_0000
EA8	Performance Monitor Counter (PMC9)	32	RW	0000_0000
EB0	Performance Monitor Local Control A (PMLCA10)	32	RW	0000_0000
EB4	Performance Monitor Local Control B (PMLCB10)	32	RW	0000_0000
EB8	Performance Monitor Counter (PMC10)	32	RW	0000_0000
1000	Error Enable (ERR_EN)	32	RW	0000_0000
1100	Memory Data Path Error Injection Mask High (DATA_ERR_INJECT_HI)	32	RW	0000_0000
1104	Memory Data Path Error Injection Mask Low (DATA_ERR_INJECT_LO)	32	RW	0000_0000
1108	Memory Data Path Error Injection Mask ECC (ERR_INJECT)	32	RW	0000_0000
110C	Address Error Inject (ADDR_ERR_INJ)	32	RW	0000_0000
1118	Memory Extended Data Path Read Capture High (CAPTURE_EXT_DATA_HI)	32	RW	0000_0000
111C	Memory Extended Data Path Read Capture Low (CAPTURE_EXT_DATA_LO)	32	RW	0000_0000
1120	Memory Data Path Read Capture High (CAPTURE_DATA_HI)	32	RW	0000_0000
1124	Memory Data Path Read Capture Low (CAPTURE_DATA_LO)	32	RW	0000_0000
1128	Memory Data Path Read Capture ECC (CAPTURE_ECC)	32	RW	0000_0000
1140	Memory Error Detect (ERR_DETECT)	32	RW	0000_0000
1144	Memory Error Disable (ERR_DISABLE)	32	RW	0000_0000
1148	Memory Error Interrupt Enable (ERR_INT_EN)	32	RW	0000_0000
114C	Memory Error Attributes Capture (CAPTURE_ATTRIBUTES)	32	RW	0000_0000
1150	Memory Error Address Capture (CAPTURE_ADDRESS)	32	RW	0000_0000
1154	Memory Error Extended Address Capture (CAPTURE_EXT_ADDRESS)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1158	Single-Bit ECC Memory Error Management (ERR_SBE)	32	RW	0000_0000
1240	ECC Region 0 Configuration (ECC_REG_0)	32	RW	0000_0000
1244	ECC Region 1 Configuration (ECC_REG_1)	32	RW	0000_0000
1248	ECC Region 2 Configuration (ECC_REG_2)	32	RW	0000_0000
124C	ECC Region 3 Configuration (ECC_REG_3)	32	RW	0000_0000
1250	ECC Region 4 Configuration (ECC_REG_4)	32	RW	0000_0000
1254	ECC Region 5 Configuration (ECC_REG_5)	32	RW	0000_0000
1258	ECC Region 6 Configuration (ECC_REG_6)	32	RW	0000_0000
125C	ECC Region 7 Configuration (ECC_REG_7)	32	RW	0000_0000

35.6.2 Rank n Memory Bounds (CS0_BNDS - CS1_BNDS)

Defines the starting and ending address of the memory space that corresponds to rank *n*.

When writing to this register, you must ensure that:

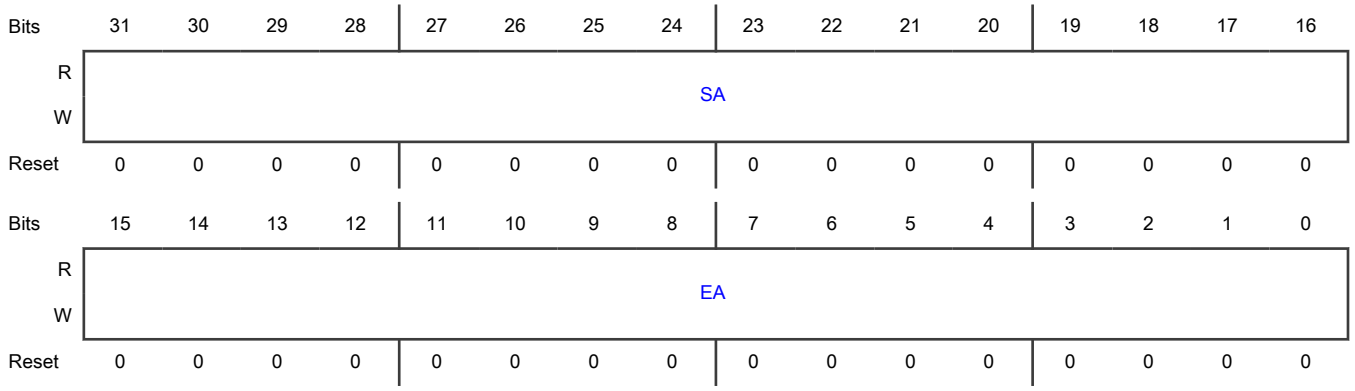
- The memory-space size you specify must equal the size of the physical DDR SDRAM.
- EA is greater than or equal to SA.

When you enable rank interleaving, DDRC uses only the fields in the lower interleaved rank to define the memory space, and does not use the other ranks' CS*n*_BNDS registers. For example, if ranks 0 and 1 are interleaved, DDRC uses CS0_BNDS but does not use CS1_BNDS.

Offset

Register	Offset
CS0_BNDS	0h
CS1_BNDS	8h

Diagram



Fields

Field	Description
31-16 SA	Starting Address Specifies the starting address for rank *n*. DDRC compares the lower 8 bits of this field with the 8 most-significant bits of the 32-bit address. You must write 0b to the upper 8 bits of this field.
15-0 EA	Ending Address Specifies the ending address for rank *n*. DDRC compares the lower 8 bits of this field with the 8 most-significant bits of the 32-bit address. You must write 0b to the upper 8 bits of this field.

35.6.3 Remap Region 0A Configuration (REMAP_0A)

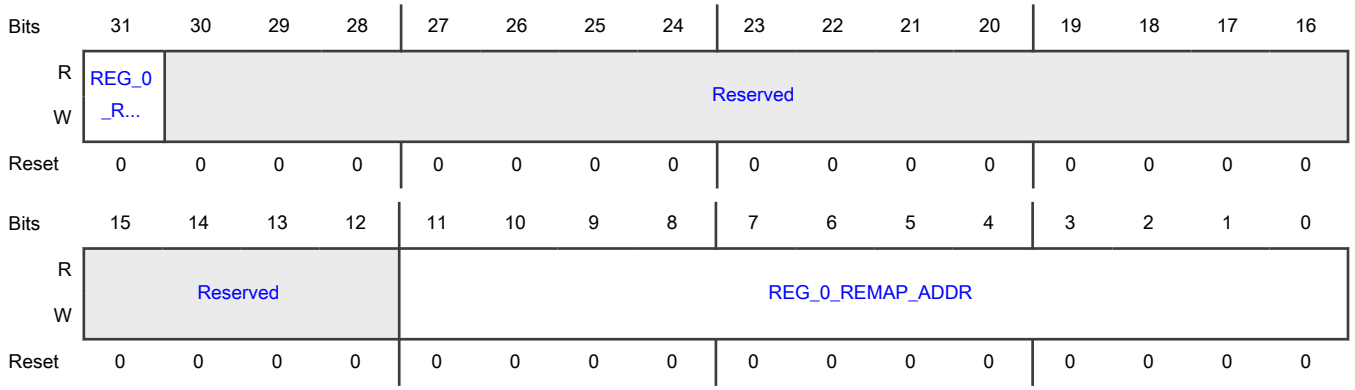
Works with Remap Region 0B Configuration (REMAP_0B) to enable a region that you can remap to a different memory region as DDRC receives the remap address. See Remapping regions for what you can and cannot do when remapping.

The comparison with Rank n Memory Bounds (CS0_BNDS - CS1_BNDS) occurs after DDRC remaps the AXI address mentioned in REMAP_[0-3]B registers on the address mentioned in REMAP_[0-3]A registers. You must program Rank n Memory Bounds (CS0_BNDS - CS1_BNDS) to match the address after remapping. The addresses after remapping (programmed in REMAP_[0-3]A registers) can be accessed only via AXI addresses captured in REMAP_[0-3]B registers. Mapped and nonmapped regions cannot overlap.

Offset

Register	Offset
REMAP_0A	20h

Diagram



Fields

Field	Description
31 REG_0_REMAP_EN	Region 0 Remap Enable Enables the remapping of region 0. When enabled, DDRC remaps any addresses within the range defined by Remap Region 0B Configuration (REMAP_0B) to a new region starting at the 1 MB boundary defined by REG_0_REMAP_ADDR. 0 - Disables 1 - Enables
30-12 —	Reserved
11-0 REG_0_REMAP_ADDR	Region 0 Remap Starting Address Specifies the starting address of this region when you enable remapping.

35.6.4 Remap Region 0B Configuration (REMAP_0B)

Works with Remap Region 0A Configuration (REMAP_0A) to enable a region that you can remap to a different memory region as DDRC receives the remap address. See Remapping regions for what you can and cannot do when remapping.

Offset

Register	Offset
REMAP_0B	24h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				REG_0_SA											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				REG_0_EA											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
27-16 REG_0_SA	Region 0 Starting Address Specifies the value to compare with the upper 12 bits of a 32-bit address when determining whether an address is within the to-be-remapped region.
15-12 —	Reserved
11-0 REG_0_EA	Region 0 Ending Address Specifies the value to compare with the upper 12 bits of a 32-bit address when determining whether an address is within the to-be-remapped region.

35.6.5 Remap Region 1A Configuration (REMAP_1A)

Works with Remap Region 1B Configuration (REMAP_1B) to enable a region that you can remap to a different memory region as DDRC receives the remap address. See Remapping regions for what you can and cannot do when remapping.

Offset

Register	Offset
REMAP_1A	28h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REG_1_R...				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				REG_1_REMAP_ADDR											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31	Region 1 Remap Enable Enables the remapping of region 1.

Table continues on the next page...

Table continued from the previous page...

Field	Description
REG_1_REMAP_EN	When enabled, DDRC remaps any addresses within the range defined by Remap Region 1B Configuration (REMAP_1B) to a new region starting at the 1 MB boundary defined by REG_1_REMAP_ADDR. 0 - Disables 1 - Enables
30-12 —	Reserved
11-0 REG_1_REMAP_ADDR	Region 1 Remap Starting Address Specifies the starting address of this region when you enable remapping.

35.6.6 Remap Region 1B Configuration (REMAP_1B)

Works with Remap Region 1A Configuration (REMAP_1A) to enable a region that you can remap to a different memory region as DDRC receives the remap address. See Remapping regions for what you can and cannot do when remapping.

Offset

Register	Offset
REMAP_1B	2Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				REG_1_SA											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				REG_1_EA											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
27-16 REG_1_SA	Region 1 Starting Address Specifies the value to compare with the upper 12 bits of a 32-bit address when determining whether an address is within the to-be-remapped region.
15-12 —	Reserved
11-0 REG_1_EA	Region 1 Ending Address Specifies the value to compare with the upper 12 bits of a 32-bit address when determining whether an address is within the to-be-remapped region.

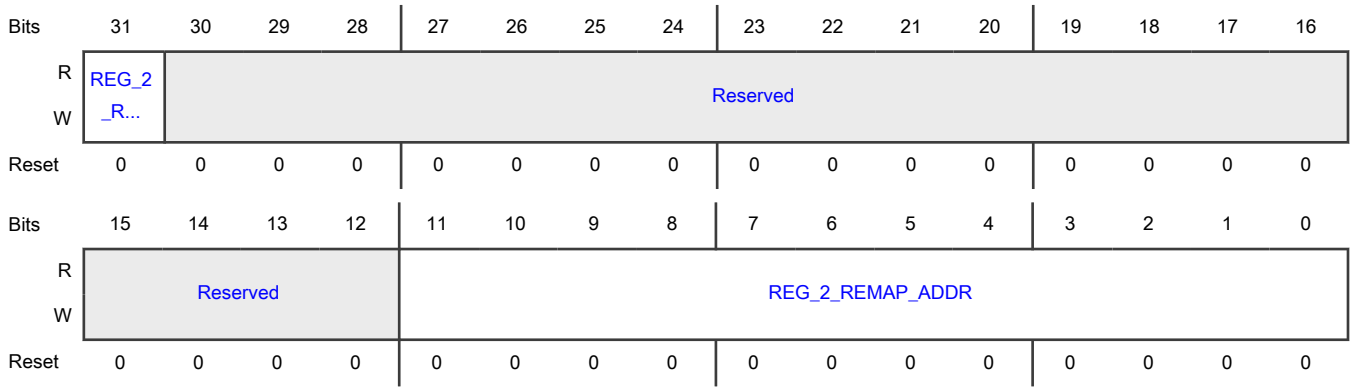
35.6.7 Remap Region 2A Configuration (REMAP_2A)

Works with Remap Region 2B Configuration (REMAP_2B) to enable a region that you can remap to a different memory region as DDRC receives the remap address. See Remapping regions for what you can and cannot do when remapping.

Offset

Register	Offset
REMAP_2A	30h

Diagram



Fields

Field	Description
31 REG_2_REMAP_EN	Region 2 Remap Enable Enables the remapping of region 2. When enabled, DDRC remaps any addresses within the range defined by Remap Region 2B Configuration (REMAP_2B) to a new region starting at the 1 M granule defined by REG_2_REMAP_ADDR.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Disables 1 - Enables
30-12 —	Reserved
11-0 REG_2_REMAP_ADDR	Region 2 Remap Starting Address Specifies the starting address of this region when you enable remapping.

35.6.8 Remap Region 2B Configuration (REMAP_2B)

Works with Remap Region 2A Configuration (REMAP_2A) to enable a region that you can remap to a different memory region as DDRC receives the remap address. See Remapping regions for what you can and cannot do when remapping.

Offset

Register	Offset
REMAP_2B	34h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				REG_2_SA											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				REG_2_EA											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 REG_2_SA	Region 2 Starting Address

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Specifies the value to compare with the upper 12 bits of a 32-bit address when determining whether an address is within the to-be-remapped region.
15-12 —	Reserved
11-0 REG_2_EA	Region 2 Ending Address Specifies the value to compare with the upper 12 bits of a 32-bit address when determining whether an address is within the to-be-remapped region.

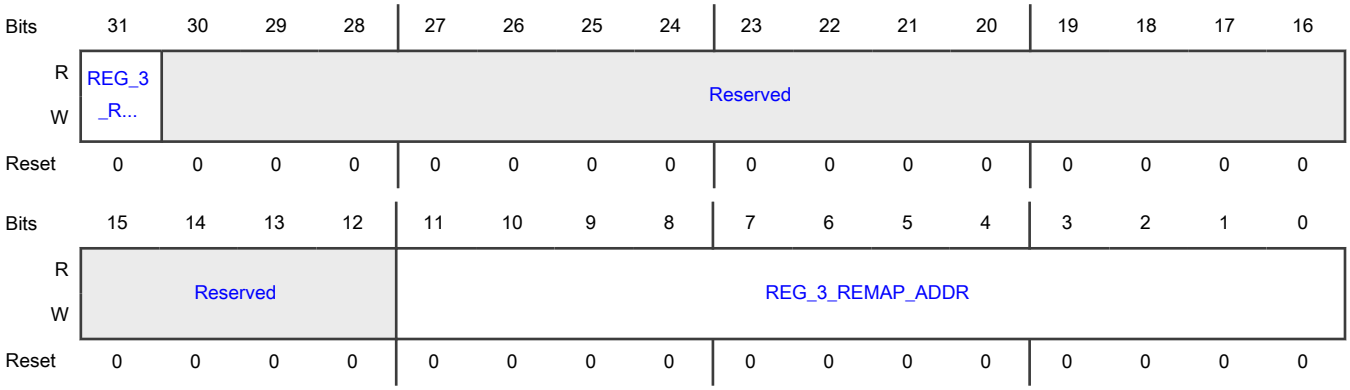
35.6.9 Remap Region 3A Configuration (REMAP_3A)

Works with Remap Region 3B Configuration (REMAP_3B) to enable a region that you can remap to a different memory region as DDRC receives the remap address. See Remapping regions for what you can and cannot do when remapping.

Offset

Register	Offset
REMAP_3A	38h

Diagram



Fields

Field	Description
31 REG_3_REMAP_EN	Region 3 Remap Enable Enables the remapping of region 3. When enabled, DDRC remaps any addresses within the range defined by Remap Region 3B Configuration (REMAP_3B) to a new region starting at the 1 M granule defined by REG_3_REMAP_ADDR. 0 - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enables
30-12 —	Reserved
11-0 REG_3_REMAP_ADDR	Region 3 Remap Starting Address Specifies the starting address of this region when you enable remapping.

35.6.10 Remap Region 3B Configuration (REMAP_3B)

Works with Remap Region 3A Configuration (REMAP_3A) to enable a region that you can remap to a different memory region as DDRC receives the remap address. See Remapping regions for what you can and cannot do when remapping.

Offset

Register	Offset
REMAP_3B	3Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				REG_3_SA											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				REG_3_EA											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 REG_3_SA	Region 3 Starting Address Specifies the value to compare with the upper 12 bits of a 32-bit address when determining whether an address is within the to-be-remapped region.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-12 —	Reserved
11-0 REG_3_EA	Region 3 Ending Address Specifies the value to compare with the upper 12 bits of a 32-bit address when determining whether an address is within the to-be-remapped region.

35.6.11 DDRC Address Decode 0 (DDR_ADDR_DEC_0)

Defines how the specified bits of the DDR SDRAM address are mapped to the original address of the transaction when you enable address decode overrides via DDR_ADDR_DEC_9[ADDR_DEC_OVRD].

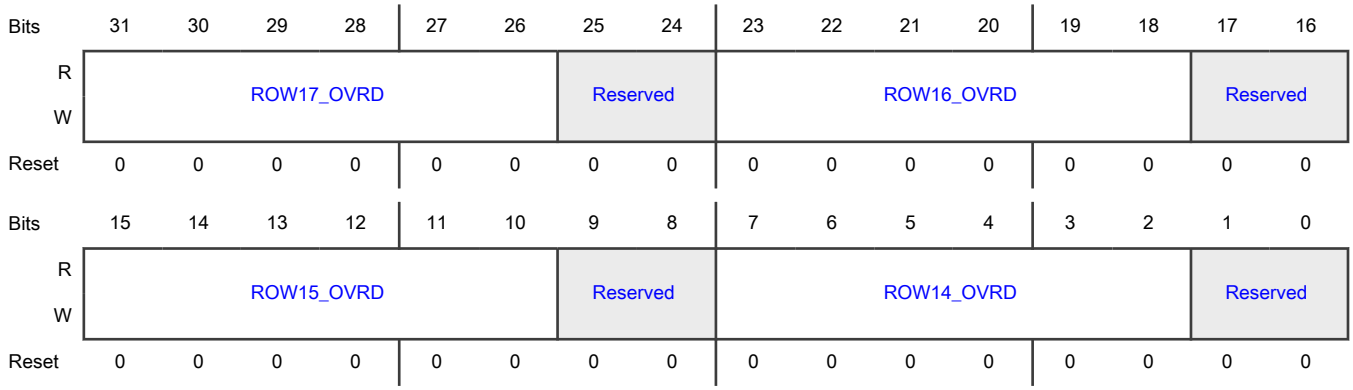
NOTE

When inline ECC is used for LPDDR4, LPDDR4x, we cannot override, lowest 7 Column bits and upper 3 row bits when ECC is enabled.

Offset

Register	Offset
DDR_ADDR_DEC_0	40h

Diagram



Fields

Field	Description
31-26 ROW17_OVRD	Row 17 Override Indicates the placement of row bit 17 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
25-24	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
23-18 ROW16_OVRD	Row 16 Override Indicates the placement of row bit 16 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
17-16 —	Reserved
15-10 ROW15_OVRD	Row 15 Override Indicates the placement of row bit 15 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
9-8 —	Reserved
7-2 ROW14_OVRD	Row 14 Override Indicates the placement of row bit 14 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
1-0 —	Reserved

35.6.12 DDRC Address Decode 1 (DDR_ADDR_DEC_1)

Defines how the specified bits of the DDR SDRAM address are mapped to the original address of the transaction when you enable address decode overrides via DDR_ADDR_DEC_9[ADDR_DEC_OVRD].

NOTE

When inline ECC is used for LPDDR4, LPDDR4x, we cannot override, lowest 7 Column bits and upper 3 row bits when ECC is enabled.

Offset

Register	Offset
DDR_ADDR_DEC_1	44h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ROW13_OVRD						Reserved		ROW12_OVRD						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ROW11_OVRD						Reserved		ROW10_OVRD						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-26 ROW13_OVRD	Row 13 Override Indicates the placement of row bit 13 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
25-24 —	Reserved
23-18 ROW12_OVRD	Row 12 Override Indicates the placement of row bit 12 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
17-16 —	Reserved
15-10 ROW11_OVRD	Row 11 Override Indicates the placement of row bit 11 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
9-8 —	Reserved
7-2 ROW10_OVRD	Row 10 Override Indicates the placement of row bit 10 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
1-0 —	Reserved

35.6.13 DDRC Address Decode 2 (DDR_ADDR_DEC_2)

Defines how the specified bits of the DDR SDRAM address are mapped to the original address of the transaction when you enable address decode overrides via DDR_ADDR_DEC_9[ADDR_DEC_OVRD].

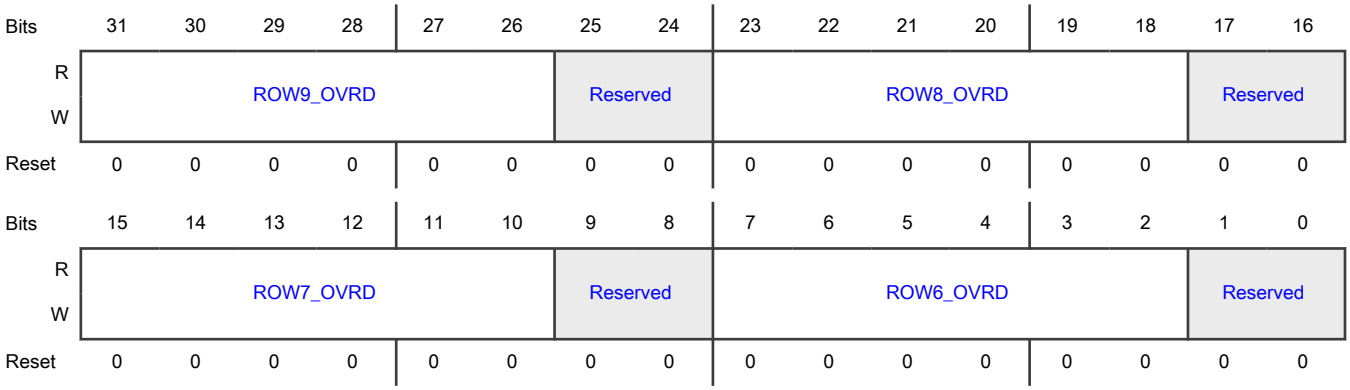
NOTE

When inline ECC is used for LPDDR4, LPDDR4x, we cannot override, lowest 7 Column bits and upper 3 row bits when ECC is enabled.

Offset

Register	Offset
DDR_ADDR_DEC_2	48h

Diagram



Fields

Field	Description
31-26 ROW9_OVRD	Row 9 Override Indicates the placement of row bit 9 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
25-24 —	Reserved
23-18 ROW8_OVRD	Row 8 Override Indicates the placement of row bit 8 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
17-16 —	Reserved
15-10 ROW7_OVRD	Row 7 Override Indicates the placement of row bit 7 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
9-8 —	Reserved
7-2 ROW6_OVRD	Row 6 Override Indicates the placement of row bit 6 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
1-0 —	Reserved

35.6.14 DDRC Address Decode 3 (DDR_ADDR_DEC_3)

Defines how the specified bits of the DDR SDRAM address are mapped to the original address of the transaction when you enable address decode overrides via DDR_ADDR_DEC_9[ADDR_DEC_OVRD].

NOTE

When inline ECC is used for LPDDR4, LPDDR4x, we cannot override, lowest 7 Column bits and upper 3 row bits when ECC is enabled.

Offset

Register	Offset
DDR_ADDR_DEC_3	4Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ROW5_OVRD						Reserved		ROW4_OVRD				Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ROW3_OVRD						Reserved		ROW2_OVRD				Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-26 ROW5_OVRD	Row 5 Override Indicates the placement of row bit 5 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
25-24 —	Reserved
23-18 ROW4_OVRD	Row 4 Override Indicates the placement of row bit 4 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
17-16 —	Reserved
15-10 ROW3_OVRD	Row 3 Override Indicates the placement of row bit 3 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
9-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7-2 ROW2_OVRD	Row 2 Override Indicates the placement of row bit 2 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
1-0 —	Reserved

35.6.15 DDRC Address Decode 4 (DDR_ADDR_DEC_4)

Defines how the specified bits of the DDR SDRAM address are mapped to the original address of the transaction when you enable address decode overrides via DDR_ADDR_DEC_9[ADDR_DEC_OVRD].

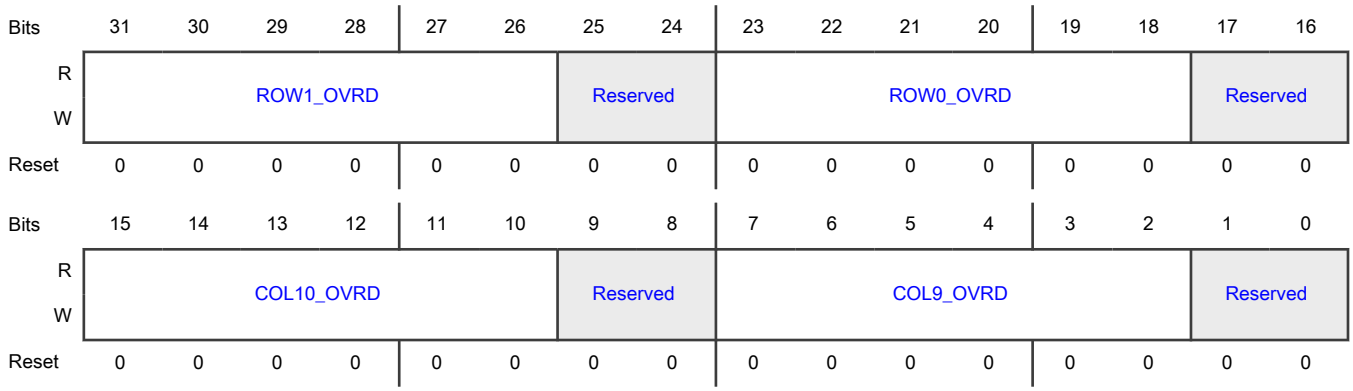
NOTE

When inline ECC is used for LPDDR4, LPDDR4x, we cannot override, lowest 7 Column bits and upper 3 row bits when ECC is enabled.

Offset

Register	Offset
DDR_ADDR_DEC_4	50h

Diagram



Fields

Field	Description
31-26 ROW1_OVRD	Row 1 Override Indicates the placement of row bit 1 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
25-24	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
23-18 ROW0_OVRD	Row 0 Override Indicates the placement of row bit 0 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
17-16 —	Reserved
15-10 COL10_OVRD	Col 10 Override Indicates the placement of col bit 10 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
9-8 —	Reserved
7-2 COL9_OVRD	Col 9 Override Indicates the placement of col bit 9 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
1-0 —	Reserved

35.6.16 DDRC Address Decode 5 (DDR_ADDR_DEC_5)

Defines how the specified bits of the DDR SDRAM address are mapped to the original address of the transaction when you enable address decode overrides via DDR_ADDR_DEC_9[ADDR_DEC_OVRD].

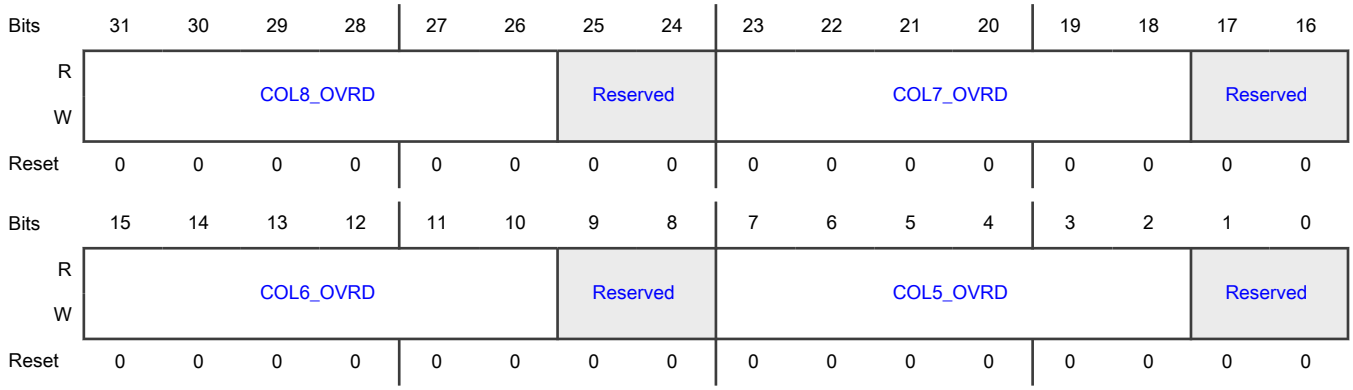
NOTE

When inline ECC is used for LPDDR4, LPDDR4x, we cannot override, lowest 7 Column bits and upper 3 row bits when ECC is enabled.

Offset

Register	Offset
DDR_ADDR_DEC_5	54h

Diagram



Fields

Field	Description
31-26 COL8_OVRD	Col 8 Override Indicates the placement of col bit 8 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
25-24 —	Reserved
23-18 COL7_OVRD	Col 7 Override Indicates the placement of col bit 7 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
17-16 —	Reserved
15-10 COL6_OVRD	Col 6 Override Indicates the placement of col bit 6 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
9-8 —	Reserved
7-2 COL5_OVRD	Col 5 Override Indicates the placement of col bit 5 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
1-0 —	Reserved

35.6.17 DDRC Address Decode 6 (DDR_ADDR_DEC_6)

Defines how the specified bits of the DDR SDRAM address are mapped to the original address of the transaction when you enable address decode overrides via DDR_ADDR_DEC_9[ADDR_DEC_OVRD].

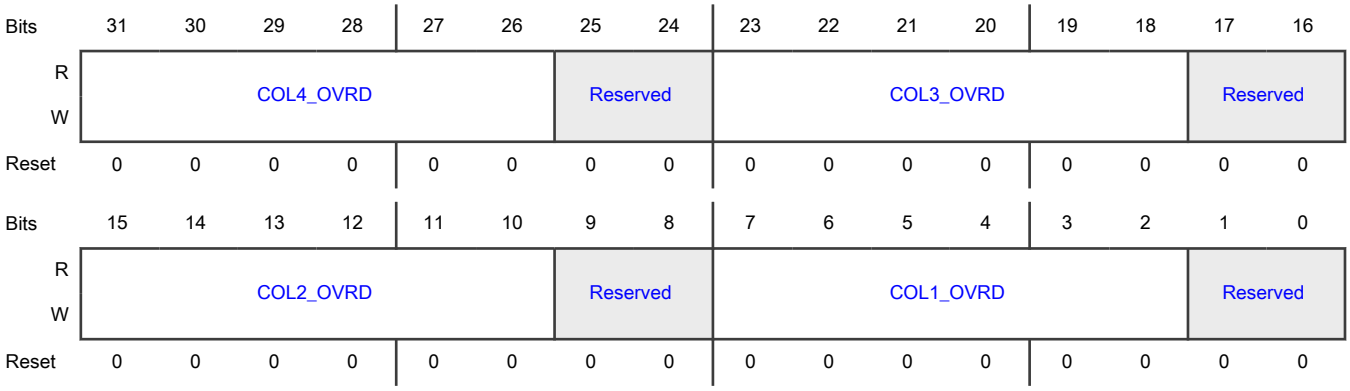
NOTE

When inline ECC is used for LPDDR4, LPDDR4x, we cannot override, lowest 7 Column bits and upper 3 row bits when ECC is enabled.

Offset

Register	Offset
DDR_ADDR_DEC_6	58h

Diagram



Fields

Field	Description
31-26 COL4_OVRD	Col 4 Override Indicates the placement of col bit 4 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1. You must write 100b to this field for a 64-bit bus, and 11b for a 32-bit bus.
25-24 —	Reserved
23-18 COL3_OVRD	Col 3 Override Indicates the placement of col bit 3 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1. You must write 101b to this field for a 32-bit bus.
17-16 —	Reserved
15-10 COL2_OVRD	Col 2 Override Indicates the placement of col bit 2 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1. You must write 101b to this field for a 64-bit bus, and 100b for a 32-bit bus.
9-8 —	Reserved
7-2 COL1_OVRD	Col 1 Override Indicates the placement of col bit 1 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You must write 100b to this field for a 64-bit bus, and 11b for a 32-bit bus.
1-0 —	Reserved

35.6.18 DDRC Address Decode 7 (DDR_ADDR_DEC_7)

Defines how the specified bits of the DDR SDRAM address are mapped to the original address of the transaction when you enable address decode overrides via DDR_ADDR_DEC_9[ADDR_DEC_OVRD].

NOTE

When inline ECC is used for LPDDR4, LPDDR4x, we cannot override, lowest 7 Column bits and upper 3 row bits when ECC is enabled.

Offset

Register	Offset
DDR_ADDR_DEC_7	5Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	COL0_OVRD						Reserved		BA1_OVRD						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BA0_OVRD						Reserved		CID1_OVRD						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-26 COL0_OVRD	Col 0 Override Indicates the placement of col bit 0 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1. You must write 11b to this field for a 64-bit bus, and 10b for a 32-bit bus.
25-24 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
23-18 BA1_OVRD	Bank 1 Override Indicates the placement of bank bit 1 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
17-16 —	Reserved
15-10 BA0_OVRD	Bank 0 Override Indicates the placement of bank bit 0 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
9-8 —	Reserved
7-2 CID1_OVRD	CID 1 Override Indicates the placement of CID bit 1 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
1-0 —	Reserved

35.6.19 DDRC Address Decode 8 (DDR_ADDR_DEC_8)

Defines how the specified bits of the DDR SDRAM address are mapped to the original address of the transaction when you enable address decode overrides via DDR_ADDR_DEC_9[ADDR_DEC_OVRD].

NOTE

When inline ECC is used for LPDDR4, LPDDR4x, we cannot override, lowest 7 Column bits and upper 3 row bits when ECC is enabled.

Offset

Register	Offset
DDR_ADDR_DEC_8	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CID0_OVRD						Reserved		CS1_OVRD						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CS0_OVRD						Reserved		BG1_OVRD						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-26 CID0_OVRD	CID 0 Override Indicates the placement of CID bit 0 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
25-24 —	Reserved
23-18 CS1_OVRD	Interleaved Rank 1 Override Indicates the placement of interleaved rank bit 1 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
17-16 —	Reserved
15-10 CS0_OVRD	Interleaved Rank 0 Override Indicates the placement of interleaved rank bit 0 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
9-8 —	Reserved
7-2 BG1_OVRD	Bank Group 1 Override Indicates the placement of bank group bit 1 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
1-0 —	Reserved

35.6.20 DDRC Address Decode 9 (DDR_ADDR_DEC_9)

Defines how the specified bits of the DDR SDRAM address are mapped to the original address of the transaction when you enable address decode overrides via DDR_ADDR_DEC_9[ADDR_DEC_OVRD].

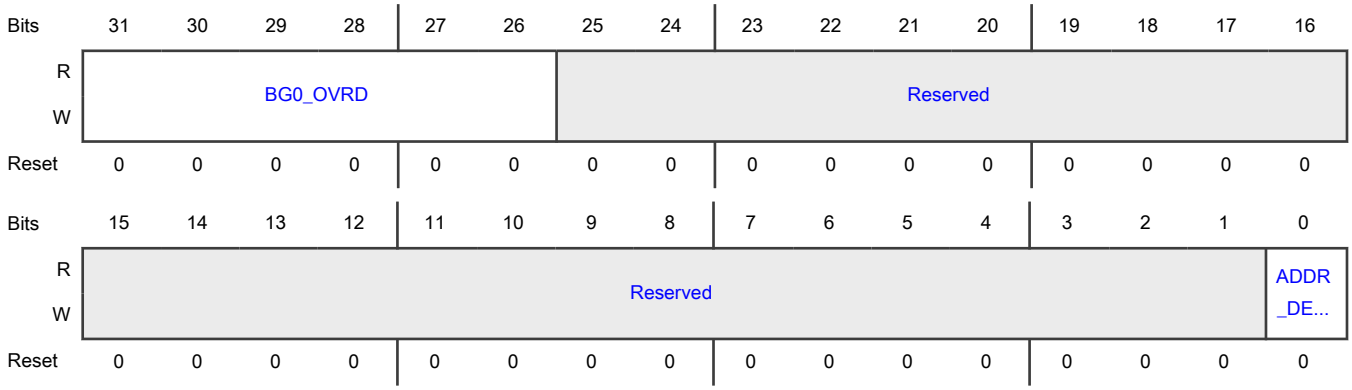
NOTE

When inline ECC is used for LPDDR4, LPDDR4x, we cannot override, lowest 7 Column bits and upper 3 row bits when ECC is enabled.

Offset

Register	Offset
DDR_ADDR_DEC_9	64h

Diagram



Fields

Field	Description
31-26 BG0_OVRD	Bank Group 0 Override Indicates the placement of bank group bit 0 in the address when DDR_ADDR_DEC_9[ADDR_DEC_OVRD] = 1.
25-1 —	Reserved
0 ADDR_DEC_OVRD	Address Decode Override Enables the DDR_ADDR_DEC_*n* registers to define the address decode scheme. If this field is enabled, you can implement all the fields of the DDR_ADDR_DEC_*n* registers while leaving a proper field for memory controller interleaving. When using a 16-bit bus, the least-significant bit of the address does not have an override bit mapped to it, and when using a 32-bit bus, the two least-significant bits of the address do not have an override bit mapped to them. When using inline ECC, you cannot alter the lower 7 column bits from their standard placement via the address decode registers.

35.6.21 Rank n Configuration (CS0_CONFIG - CS1_CONFIG)

Enables the DDR SDRAM rank and sets the number of row and column bits used for the rank.

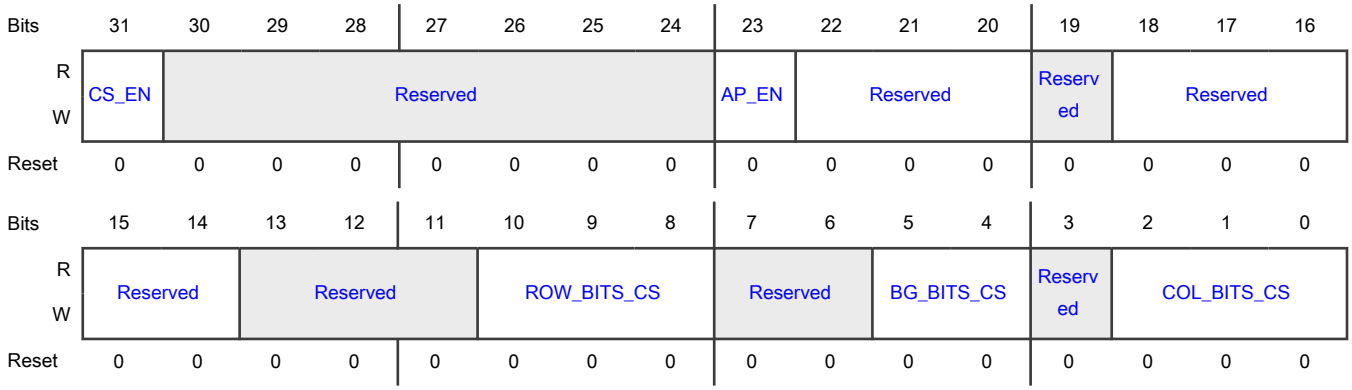
You must write the correct number of row and column bits for each DDR SDRAM to this register.

Because ROW_BITS_CS and COL_BITS_CS establish address multiplexing, you must ensure that you wrote the correct values.

Offset

Register	Offset
CS0_CONFIG	80h
CS1_CONFIG	84h

Diagram



Fields

Field	Description
31 CS_EN	Rank Enable Enables rank *n*. When CS_EN = 1, the rank assumes the state defined in Rank n Memory Bounds (CS0_BNDS - CS1_BNDS). The DDR PHY initialization begins when you write 1 to this field and completes when DDRC changes DDRDSR_2[PHY_INIT_CMPLT] to 1. You must write 1 to DDR_SDRAM_CFG[MEM_EN] after the initialization completes. 0 - Disables 1 - Enables
30-24 —	Reserved
23 AP_EN	Auto-Precharge Enable Enables the auto-precharge command for rank *n*. The rank: • Precharges automatically only if you enabled Global Auto-Precharge mode (DDR_SDRAM_INTERVAL[BSTOPRE] = 0). • Always issues an auto-precharge command for read and write transactions. 0 - Disables 1 - Enables
22-20 —	Reserved
19 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
18-16 —	Reserved
15-14 —	Reserved
13-11 —	Reserved
10-8 ROW_BITS_CS	Row Bits Specifies the number of row bits for the DDR SDRAM on rank *n*. If any rank uses 18 row bits, then all enabled ranks must use 18 row bits. 000 - 12 001 - 13 010 - 14 011 - 15 100 - 16 101 - 17 All other values are reserved.
7-6 —	Reserved
5-4 BG_BITS_CS	Bank Group Bits Specifies the number of bank group bits for DDR SDRAM on rank *n*. When using an LPDDR4, LPDDR4x compliant memory, you must write 1 to this field because the DDRC BG0 functionality controls the BA2 bit of LPDDR4. 00 - 0 01 - Must be set to 1 to enable the 3rd bank address bit for LPDDR4, LPDDR4x memories. 10 - Reserved for LPDDR4 11 - Reserved
3 —	Reserved
2-0 COL_BITS_CS	Column Bits Indicates the number of column bits for the DDR SDRAM on rank *n*. 000 - 8 001 - 9

Table continues on the next page...

Table continued from the previous page...

Field	Description
	010 - 10 011 - 11 111 - 7 All other values are reserved.

35.6.22 DDR SDRAM Timing Configuration 3 (TIMING_CFG_3)

Defines various timing and latency parameters.

Offset

Register	Offset
TIMING_CFG_3	100h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		EXT_PRETOACT		Reserved	EXT_ACTTOPRE			EXT_ACTTORW		EXT_REFREC					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	EXT_CASLAT			EXT_WR_...	Reserved	EXT_WRREC		Reserved		EXT_CKE_PLS		EXT_FOU...	Reserved	EXT_ACT...	EXT_WRT...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-30 —	Reserved
29-28 EXT_PRETOACT	Extended Precharge-To-Activate Time Enables you to add extra clock cycles that combine with TIMING_CFG_1[PRETOACT] to define t_{RP} (see Precharge-to-activate interval (t_{RP})).
27 —	Reserved
26-24	Extended Activate-To-Precharge Time

Table continues on the next page...

Table continued from the previous page...

Field	Description
EXT_ACTTOPRE	Enables you to add extra clock cycles that combine with TIMING_CFG_1[ACTTOPRE] to define t_{RAS} (see Activate-to-precharge interval (t_{RAS})).
23-22 EXT_ACTTORW	Extended Activate To Read Or Write Time Enables you to add extra clock cycles that combine with TIMING_CFG_1[ACTTORW] to define t_{RCD} (see Activate-to-read-or-write interval (t_{RCD})).
21-16 EXT_REFREC	Extended Refresh Recovery Enables you to add extra clock cycles that combine with TIMING_CFG_1[REFREC] to define t_{RFC} (see Refresh recovery time (t_{RFC})). Values greater than 00111111b are reserved.
15 —	Reserved
14-12 EXT_CASLAT	Extended CAS Latency Enables you to add extra clock cycles that combine with TIMING_CFG_1[CASLAT] to define the CAS latency (see CAS latency).
11 EXT_WR_LAT_2	Extended Write Latency 2 Enables you to add extra clock cycles that combine with TIMING_CFG_2[EXT_WR_LAT] TIMING_CFG_2[WR_LAT] to define the total write latency (see Total write latency).
10 —	Reserved
9-8 EXT_WRREC	Extended Write Recovery Enables you to add extra clock cycles that combine with TIMING_CFG_1[WRREC] to define tn_{WR} (see Minimum write recovery interval (n_{WR})). 11 - Reserved All other values - Clock cycles as defined in the description
7-6 —	Reserved
5-4 EXT_CKE_PLS	Extended MCKE Pulse Enables you to add extra clock cycles that combine with TIMING_CFG_2[CKE_PLS] to define t_{CKE} (see Minimum MCKE pulse width (t_{CKE})).
3 EXT_FOUR_ACT	Extended Four Activate Enables you to add extra clock cycles that combine with TIMING_CFG_2[FOUR_ACT] to define t_{FAW} (see Four-activate window (t_{FAW})).

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 —	Reserved
1 EXT_ACTTOACT	Extended Activate-To-Activate Time Enables you to add extra clock cycles that combine with TIMING_CFG_1[ACTTOACT] to define t_{RRD} (see Activate-to-activate interval (t_{RRD})).
0 EXT_WRTORD	Extended Write-To-Read Time Enables you to add extra clock cycles that combine with TIMING_CFG_1[WRTORD] to define t_{WTR} (see Write-to-read delay (t_{WTR})).

35.6.23 DDR SDRAM Timing Configuration 0 (TIMING_CFG_0)

Sets the number of clock cycles between various DDR SDRAM control commands.

Offset

Register	Offset
TIMING_CFG_0	104h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RWT		WRT		RRT		WWT		ACT_PD_EXIT				PRE_PD_EXIT			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EXT_PRE_PD_EXIT		Reserved	EXT_ACT...	Reserved						MRS_CYC					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-30 RWT	Read-To-Write Turnaround Specifies clock cycles that combine with TIMING_CFG_4[EXT_RWT] to define the read-to-write turnaround (see Read-to-write turnaround).
29-28	Write-To-Read Turnaround

Table continues on the next page...

Table continued from the previous page...

Field	Description
WRT	Specifies clock cycles that combine with TIMING_CFG_4[EXT_WRT] to define the read-to-write turnaround (see Write-to-read turnaround).
27-26 RRT	Read-To-Read Turnaround To Different Ranks Specifies clock cycles that combine with TIMING_CFG_4[EXT_RRT] to define the read-to-read turnaround (see Read-to-read turnaround) to different ranks.
25-24 WWT	Write-To-Write Turnaround To Different Ranks Specifies the number of extra cycles to be added between writes to different ranks. Specifies extra clock cycles that combine with TIMING_CFG_4[EXT_WWT] to define the write-to-write turnaround (see Write-to-write turnaround) to different ranks.
23-20 ACT_PD_EXIT	Active Powerdown Exit Specifies clock cycles that combine with TIMING_CFG_0[EXT_ACT_PD_EXIT] to wait after exiting an active power-down before issuing a command to the DDR SDRAM. 0000 - Reserved All other values - Clock cycles as defined in the description
19-16 PRE_PD_EXIT	Precharge Power-Down Exit Specifies clock cycles that combine with TIMING_CFG_0[EXT_PRE_PD_EXIT] to define t_{XP} (see Precharge power-down exit timing (t_{XP})). 0000 - Reserved All other values - Clock cycles as defined in the description
15-14 EXT_PRE_PD_EXIT	Extended Precharge Power-Down Exit Enables you to add extra clock cycles that combine with PRE_PD_EXIT to define t_{XP} (see Precharge power-down exit timing (t_{XP})).
13 —	Reserved
12 EXT_ACT_PD_EXIT	Extended Active Power-Down Exit Enables you to add extra clock cycles that combine with ACT_PD_EXIT to define t_{XP} (see Precharge power-down exit timing (t_{XP})).
11-6 —	Reserved
5-0 MRS_CYC	MRW Cycle Time Specifies the number of idle clock cycles that must pass between an MRW command and any other command. Write the greater of t_{MRD} and t_{MRW} to this field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	00_0000 - Reserved All other values - Clock cycles as defined in the description

35.6.24 DDR SDRAM Timing Configuration 1 (TIMING_CFG_1)

Sets the number of clock cycles between various DDR SDRAM control commands.

Offset

Register	Offset
TIMING_CFG_1	108h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PRETOACT				ACTTOPRE				ACTTORW				CASLAT			Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	REFREC				WRREC				ACTTOACT				WRTORD			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 PRETOACT	Precharge-To-Activate Time Specifies clock cycles that combine with TIMING_CFG_3[EXT_PRETOACT] to define t_{RP} (see Precharge-to-activate interval (t_{RP})). 0000 - Reserved All other values - Clock cycles as defined in the description
27-24 ACTTOPRE	Activate-To-Precharge Time Specifies clock cycles that combine with TIMING_CFG_3[EXT_ACTTOPRE] to define t_{RAS} (see Activate-to-precharge interval (t_{RAS})).
23-20 ACTTORW	Activate To Read Or Write Specifies clock cycles that combine with TIMING_CFG_3[EXT_ACTTORW] to define t_{RCD} (see Activate-to-read-or-write interval (t_{RCD})).

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0000 - Reserved All other values - Clock cycles as defined in the description
19-17 CASLAT	CAS Latency Specifies clock cycles that combine with TIMING_CFG_3[EXT_CASLAT] to define the CAS latency (see CAS latency).
16 —	Reserved
15-12 REFREC	Refresh Recovery Specifies clock cycles that combine with TIMING_CFG_3[EXT_REFREC] to define t_{RFC} (see Refresh recovery time (t_{RFC})).
11-8 WRREC	Write Recovery Specifies clock cycles that combine with TIMING_CFG_3[EXT_WRREC] to define n_{WR} (see Minimum write recovery interval (n_{WR})). The value 0b is reserved when TIMING_CFG_3[EXT_WRREC] = 0.
7-4 ACTTOACT	Activate-To-Activate Interval Specifies the number of clock cycles from an activate command until another activate command is allowed for a different logical bank in the same physical bank. This field indicates the activate-to-activate interval (t_{RRD}). 0000 - Reserved All other values - Clock cycles as defined in the description
3-0 WRTORD	Write-To-Read Interval Specifies the number of clock cycles between the last write data pair and the subsequent read command to the same physical bank. This field indicates the last write data pair to read command issue interval (t_{WTR}) for LPDDR4xLPDDR4. 0000 - Reserved All other values - Clock cycles as defined in the description

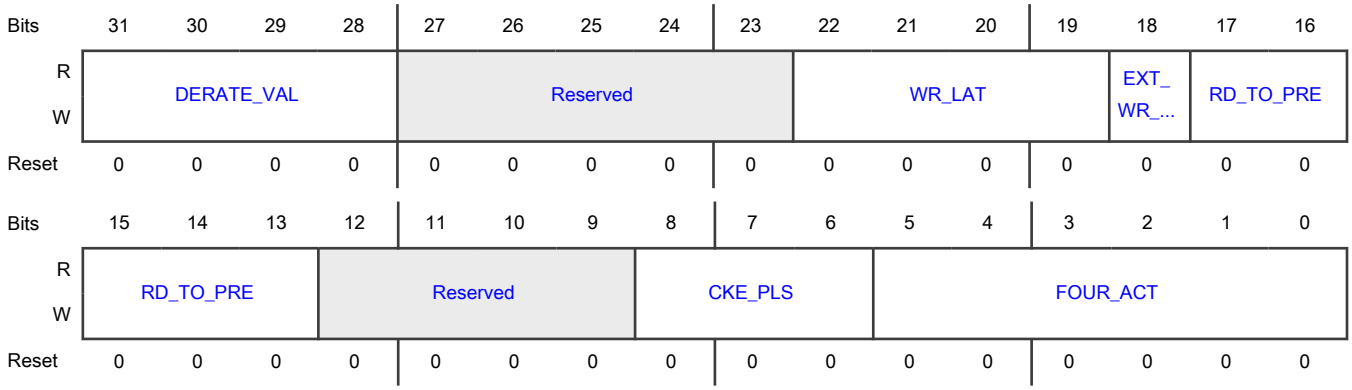
35.6.25 DDR SDRAM Timing Configuration 2 (TIMING_CFG_2)

Sets the clock delay to data for writes.

Offset

Register	Offset
TIMING_CFG_2	10Ch

Diagram



Fields

Field	Description
31-28 DERATE_VAL	Derate Value Enables you to specify the number of cycles used for de-rating DDR SDRAM timing parameters when you set the dynamic refresh rate DDR_SDRAM_CFG_3[DYN_REF_RATE_EN] and the monitored rate requires timing parameter de-rating.
27-23 —	Reserved
22-19 WR_LAT	Write Latency Specifies clock cycles that combine with TIMING_CFG_2[EXT_WR_LAT] and TIMING_CFG_3[EXT_WR_LAT_2] to define the total write latency (see Total write latency).
18 EXT_WR_LAT	Extended Write Latency Enables you to add extra clock cycles that combine with TIMING_CFG_2[WR_LAT] and TIMING_CFG_3[EXT_WR_LAT_2] to define the total write latency (see Total write latency).
17-13 RD_TO_PRE	Read-To-Precharge Time Specifies the read-to-precharge time n_{RTP}) in clock cycles for LPDDR4. Specifies the read-to-precharge time (t_{RTP}) in clock cycles for LPDDR4x. 0_0000 - Reserved All other values - Clock cycles as defined in the description
12-9 —	Reserved
8-6 CKE_PLS	MCKE Pulse Specifies clock cycles that combine with TIMING_CFG_3[EXT_CKE_PLS] to define t_{CKE} (see Minimum MCKE pulse width (t_{CKE})).
5-0	Four Activate

Table continues on the next page...

Table continued from the previous page...

Field	Description
FOUR_ACT	Specifies clock cycles that combine with TIMING_CFG_3[EXT_FOUR_ACT] to define t_{FAW} (see Four-activate window (t_{FAW})).

35.6.26 DDR SDRAM Control Configuration (DDR_SDRAM_CFG)

Enables:

- The DDR SDRAM interface logic.
- Operating features such as self-refresh, error check and correction, and dynamic power management.

Offset

Register	Offset
DDR_SDRAM_CFG	110h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MEM_EN	SREN	Reserved	Reserved	Reserved	SDRAM_TYPE			Reserved		DYN_PWR	DBW		Reserved	Reserved	Reserved
Reset	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	BA_INTLV_CTL							Reserved				Reserved	Reserved	MEM_HALT	BI
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 MEM_EN	DDRC Enable If DDRC is disabled, you must not enable it until you configure all other memory configuration parameters appropriately. 0 - Disables 1 - Enables
30 SREN	Self-Refresh Enable Enables the DDR SDRAM self-refresh function during sleep. 0 - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enables
29 —	Reserved
28 —	Reserved
27 —	Reserved
26-24 SDRAM_TYPE	DDR SDRAM Type Specifies the type of DDR SDRAM device to be used. Use this field to issue the automatic hardware initialization sequence to the DDR SDRAM via the MRW and extended-MRW commands. 100 - LPDDR4, LPDDR4x SDRAM All other values are reserved.
23-22 —	Reserved
21 DYN_PWR	Dynamic Power Management Controls whether DDRC is in Dynamic Power Management mode. In this mode, the DDR SDRAM CKE signal is negated if there is no ongoing memory activity. 0 - No 1 - Yes
20-19 DBW	DDR SDRAM Data Bus Width Specifies the DDR SDRAM data bus width. 10 - 16 bits All other values are reserved.
18 —	Reserved
17 —	Reserved
16 —	Reserved
15	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
14-8 BA_INTLV_CTL	Rank Interleaving Control Is used for rank interleaving and indicates whether external ranks are interleaved. Below, the unlisted field values are reserved, and an "x" denotes a don't-care value. 000_0000 - No external ranks are interleaved. 100_0000 - External ranks 0 and 1 are interleaved. All other values are reserved.
7-4 —	Reserved
3 —	Reserved
2 —	Reserved
1 MEM_HALT	DDRC Halt Specifies whether DDRC accepts new data read or write transactions. When this field is 1, DDRC does not accept any new transactions for DDR SDRAM until the field becomes 0. This field can be used when bypassing initialization and forcing MRW commands. 0 - Accepts new transactions 1 - Completes any remaining transactions and remains halted until you write 0 to this field
0 BI	Bypass Initialization Controls bypass initialization. You must write 1 to this field because the DDR PHY programs MRW commands. 0 - Reserved; do not use 1 - Initialization routine is bypassed

35.6.27 DDR SDRAM Control Configuration 2 (DDR_SDRAM_CFG_2)

Provides additional control configuration for DDRC.

Offset

Register	Offset
DDR_SDRAM_CFG_2	114h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FRC_SR	Reserved	Reserved		MCK_DIS				Reserved	Reserved	Reserved	Reserved				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_PR				Reserved	Reserved	Reserved	Reserved	Reserved	Reserved		D_INIT	Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 FRC_SR	Force Self-Refresh Specifies the mode in which DDRC operates or enters. 0 - Normal mode 1 - Self-Refresh mode
30 —	Reserved
29-28 —	Reserved
27-24 MCK_DIS	MCK Disable Can be used to disable up to 4 MCK clock pairs, if available, according to the following table.

When you have:	MCK_DIS bit 3:	MCK_DIS bit 2:	MCK_DIS bit 1:	MCK_DIS bit 0:
MCK0	Disables or enables the MCK0 clock pair	Does nothing		
MCK0 and MCK1		Disables or enables the MCK1 clock pair	Does nothing	
MCK0, MCK1, and MCK2			Disables or enables the MCK2 clock pair	Does nothing
MCK0, MCK1, MCK2, and MCK3				Disables or enables the MCK3 clock pair

NOTE

- MCK_DIS bit 0 is register bit 24, MCK_DIS bit 1 is register bit 25, and so on.
- In this chip, only MCK0 pair is available, hence only bit 3 is applicable.

0b - Enable the clock pair

1b - Disable the clock pair

Table continued from the previous page...

Field	Description
23 —	Reserved
22 —	Reserved
21 —	Reserved
20-16 —	Reserved
15-12 NUM_PR	Number Of Posted Refreshes Determines the number of posted refreshes (if any) that can be issued at a time. If you use posted refreshes, you must write to NUM_PR and DDR_SDRAM_INTERVAL[REFINT] in a way that does not violate the maximum t_{RAS} specification. **NOTE** These bits can be greater than 1 only for All Bank Refresh. For Per Bank refresh these bits should not be changed. 0000,0001 - 1 0010 - 2 0011 - 3 0100 - 4 0101 - 5 0110 - 6 0111 - 7 1000 - 8 All other values are reserved.
11 —	Reserved
10 —	Reserved
9 —	Reserved
8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7 —	Reserved
6-5 —	Reserved
4 D_INIT	DDR SDRAM Data Initialization Indicates the status of data initialization. You write 1 to this field, and DDRC writes 0 to it. If you write 1 before DDRC is enabled, DDRC automatically initializes the DDR SDRAM after DDRC is enabled. DDRC automatically writes 0 to the field after the initialization is complete. You can only write 1 to this field when DDRC is idle and DDR_MTCR[MT_EN] = 0. This field remains enabled until the initialization is complete. DDR SDRAM Data Initialization (DDR_DATA_INIT) initializes the DDR SDRAM. 0 - No data initialization in progress, and none scheduled 1 - DDRC to initialize the DDR SDRAM after DDRC is enabled
3 —	Reserved
2 —	Reserved
1 —	Reserved. NXP recommends writing 1 to this field.
0 —	Reserved

35.6.28 DDR SDRAM Mode Control (DDR_SDRAM_MD_CNTL)

Enables you to:

- Issue an MRW command to a particular rank.
- Force the MCKE signals to a specific value.

Table 180. DDR_SDRAM_MD_CNTL field settings

Field	MRW	Refresh	Precharge
MD_EN	1	0	0
SET_REF	0	1	0

Table continues on the next page...

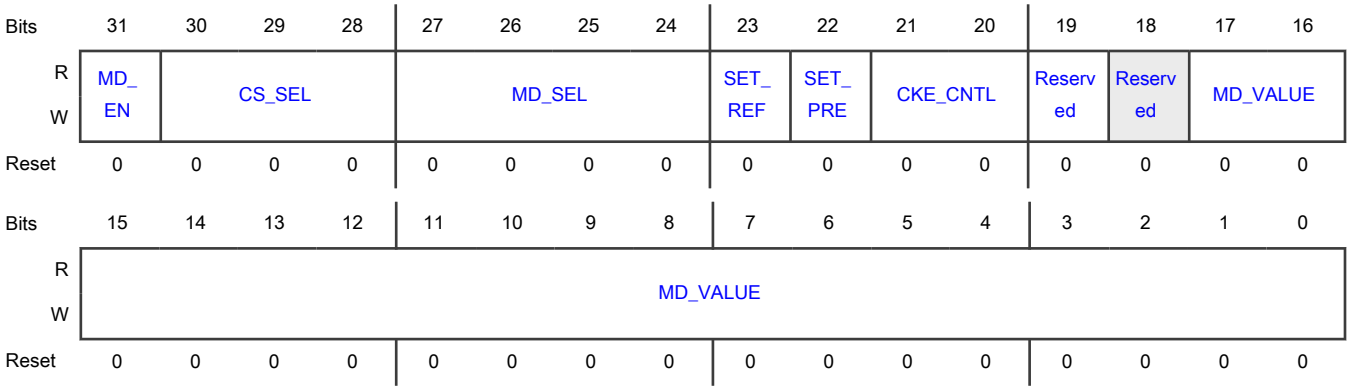
Table 180. DDR_SDRAM_MD_CNTL field settings (continued)

Field	MRW	Refresh	Precharge
SET_PRE	0	0	1
CS_SEL	Chooses rank		
MD_SEL	Selects the mode register	-	Selects a logical bank and bank group
MD_VALUE	Indicates the value written to the mode register	-	Only bit 7 is significant
CKE_CNTL	0	0	0

Offset

Register	Offset
DDR_SDRAM_MD_CNTL	120h

Diagram



Fields

Field	Description
31 MD_EN	Mode Enable This field specifies that valid data in DDR_SDRAM_MD_CNTL[MD_VALUE] is ready to be written to the DDR SDRAM via an MRW command. DDR_SDRAM_MD_CNTL[MD_SEL] selects the specific mode register. Additionally, you must select the rank by writing 1 to DDR_SDRAM_MD_CNTL[CS_SEL]. You write 1 to this field and DDRC writes 0 to it after the selected command is issued. **NOTE** You must not write 1 to this field unless DDR_SDRAM_CFG_3[DYN_REF_RATE_EN] is 0. The following procedure is required when disabling this field.

Table continues on the next page...

Field	Description
	Prior to disabling DDR_SDRAM_CFG_3[DYN_REF_RATE_EN], the following procedure is required to ensure any pending autonomous hardware based mode register operations are complete: 1. Clear DDR_SDRAM_MPR5[MPR_VLD] (set to 0) 2. Poll DDR_SDRAM_MPR5[MPR_VLD] until it is set to 1 which indicates that the latest dynamic refresh MR4 read has completed 3. Immediately clear (disable) DDR_SDRAM_CFG_3[DYN_REF_RATE_EN] Finally, poll DDR_SDRAM_MD_CNTL[MD_EN] to ensure it is 0 to ensure any pending software initiated mode register operations are complete prior to setting to 1. 0 - Does not need to be issued 1 - Valid data contained in the register ready to be issued as an MRW command
30-28 CS_SEL	Select Rank Specifies the rank that is driven active because of a command that you force in this register. 000 - 0 001 - 1 100 - 0 and 1 All other values are reserved.
27-24 MD_SEL	Mode Register Select Selects: • The DDR SDRAM mode register to be changed during a mode select command. • The DDR SDRAM logical bank to be precharged during a precharge command. The precharge-all and refresh commands ignore this field. MD_SEL[0:1] is not used with LPDDR4 LPDDR4x DDR SDRAM. 0000 - MR 0001 - EMR 0010 - EMR2 0011 - EMR3
23 SET_REF	Set Refresh Specifies whether a refresh command must be issued. Forces an immediate refresh to be issued to the rank, as specified by DDR_SDRAM_MD_CNTL[CS_SEL]. You write 1 to this field and DDRC writes 0 to it after the command is issued. LPDDR4 LPDDR4x DDR SDRAM does not support this field. 0 - No 1 - Yes
22 SET_PRE	Set Precharge Specifies whether a precharge command must be issued.

Table continues on the next page...

Field	Description
	Forces a precharge or precharge-all command (depending on the value of DDR_SDRAM_MD_CNTL[MD_VALUE]) to be issued to the rank, as specified by DDR_SDRAM_MD_CNTL[CS_SEL]. You write 1 to this field and DDRC writes 0 to it after the command is issued. LPDDR4 LPDDR4x DDR SDRAM does not support this field. 0 - No 1 - Yes
21-20 CKE_CNTL	Clock Enable Control Specifies the value to which MCKE signals are forced. This field enables you to globally assert or deassert all the MCKE signals issued to the DDR SDRAM. After you force a value to MCKE, DDRC continues to force that value until one of the following occurs: • You write 0 to this field. • Another event (for example, self-refresh entry or exit, or power-down entry or exit) causes the MCKE signals to change. 00 - Not forced 01 - Forced to a lower value 10 - Forced to a higher value 11 - Reserved
19 —	Reserved
18 —	Reserved
17-0 MD_VALUE	Mode Register Value Specifies the value that is presented on the DDRC memory-address pins during an MRW command. This field is significant only when this register is used to issue an MRW, precharge, or precharge-all command. The two most-significant bits do not affect the MRW commands to the DDR SDRAM because A[17:16] of the DDR SDRAM's MR registers are reserved. For an MRW command, this field contains the data to be written to the selected mode register. When using LPDDR4 LPDDR4x DDR SDRAM, the bits of this field, 17:0, map to the DDR SDRAM command as follows:

Table 181. LPDDR4 LPDDR4x MD_VALUE description

MD_VALUE bits	Mapped to
17–14	Unused
13–6	OP7–OP0
5–0	MA5–MA0

Table continues on the next page...

Field	Description
	<table><tr><th>MD_VALUE bits</th><th>Mapped to</th></tr><tr><td colspan="2">For a precharge command, only bit 5 of this field is significant.</td></tr></table>

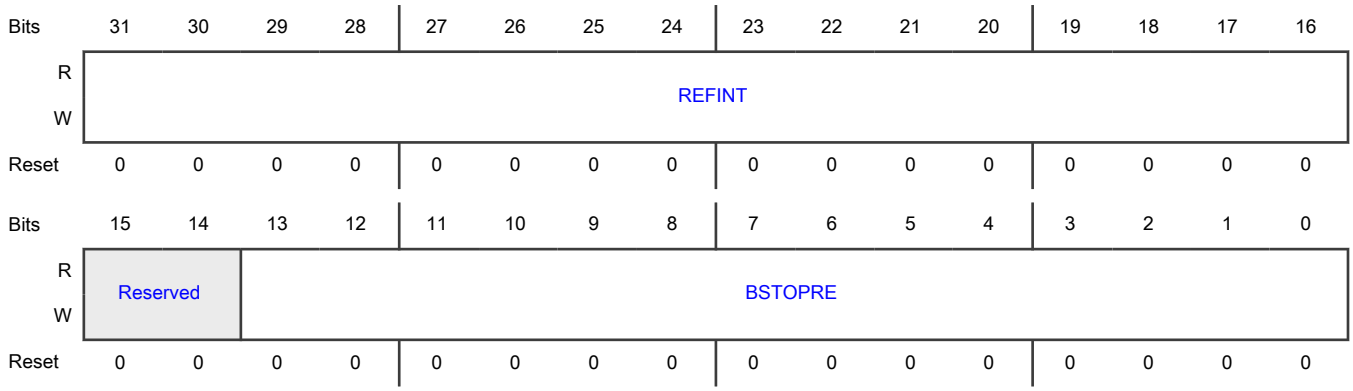
35.6.29 DDR SDRAM Interval Configuration (DDR_SDRAM_INTERVAL)

Specifies the number of DDR SDRAM clock cycles between refreshes issued to the DDR SDRAMs. Additionally, the register provides the number of DDR SDRAM cycles for which a page is retained after being accessed.

Offset

Register	Offset
DDR_SDRAM_INTERVAL	124h

Diagram



Fields

Field	Description
31-16 REFINT	Refresh Interval Represents the number of memory bus clock cycles between refresh cycles. This way, the field controls auto-refresh. Writing 0 to this field turns off auto-refresh. **NOTE** This field should be REFINT*NUM_PR when DDR_SDRAM_CFG2[NUM_PR] is > 1. Depending on DDR_SDRAM_CFG_2[NUM_PR], some quantity of rows is refreshed in each DDR SDRAM rank during each refresh cycle. The value of this field depends on the specific DDR SDRAM used and the interface clock frequency. This field is concatenated with TIMING_CFG_4[EXT_REFINT] to obtain a 17-bit value for the total refresh interval.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-14 —	Reserved
13-0 BSTOPRE	Precharge Interval Sets the duration (in memory bus clocks) that a page is retained after a DDR SDRAM access. If this field is 0, DDRC: • Leaves Open-Page mode. • Enters Global Auto-Precharge mode. • Uses auto-precharge read and write commands.

35.6.30 DDR SDRAM Data Initialization (DDR_DATA_INIT)

Provides the value that is used to initialize memory when DDR_SDRAM_CFG_2[D_INIT] = 1.

Offset

Register	Offset
DDR_DATA_INIT	128h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_VALUE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_VALUE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 INIT_VALUE	Initialization Value Represents the value that DDR SDRAM is initialized with when DDR_SDRAM_CFG_2[D_INIT] = 1.

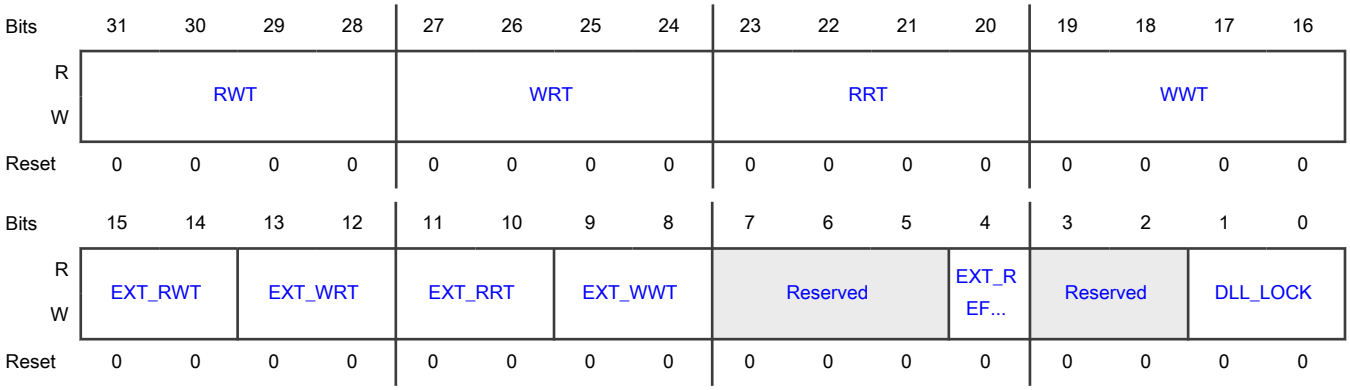
35.6.31 DDR SDRAM Timing Configuration 4 (TIMING_CFG_4)

Provides additional timing fields for turnaround clocks, refresh recovery, and (DLL) lock time.

Offset

Register	Offset
TIMING_CFG_4	160h

Diagram



Fields

Field	Description
31-28 RWT	Read-To-Write Turnaround For Same Rank Specifies the number of clock cycles to be added between a read-to-write turnaround for transactions to the same rank. When you select zero clock cycles, DDRC uses the value used for transactions to different ranks, as defined in TIMING_CFG_0[RWT], for calculating the number of additional clock cycles. Regardless of the value defined in this field, the value that TIMING_CFG_0[RWT] defines is also considered before issuing a write command.
27-24 WRT	Write-To-Read Turnaround For Same Rank Specifies the number of clock cycles to be added between a write-to-read turnaround for transactions to the same rank. When you select zero clock cycles, DDRC uses the value that is used for transactions to different ranks, as defined in TIMING_CFG_0[WRT], for calculating the number of additional clock cycles. TIMING_CFG_0[WRT] provides the minimum number of clock cycles that DDRC uses, regardless of its value.
23-20 RRT	Read-To-Read Turnaround For Same Rank Specifies the number of clock cycles to be added between reads to the same rank as follows: Number of cycles = BL/2 + RRT If you select zero clock cycles, DDRC uses the formula BL/2 to calculate the number of clock cycles required between read commands to the same rank. DDRC must wait for four clock cycles between read-to-read transactions to the same rank.

Table continues on the next page...

Table continued from the previous page...

Field	Description
19-16 WWT	Write-To-Write Turnaround For Same Rank Controls the number of clock cycles to be added between writes to the same rank as follows: Number of cycles = BL/2 + WWT If you select zero clock cycles, DDRC uses the formula BL/2 to calculate the number of clock cycles required between write commands to the same rank. DDRC must wait for four clock cycles between write-to-write transactions to the same rank.
15-14 EXT_RWT	Extended Read-To-Write Turnaround Specifies extra clock cycles that combine with TIMING_CFG_0[RWT] to define the read-to-write turnaround (see Read-to-write turnaround).
13-12 EXT_WRT	Extended Write-To-Read Turnaround Specifies extra clock cycles that combine with TIMING_CFG_0[WRT] to define the write-to-read turnaround (see Write-to-read turnaround).
11-10 EXT_RRT	Extended Read-To-Read Turnaround Specifies extra clock cycles that combine with TIMING_CFG_0[RRT] to define the read-to-read turnaround (see Read-to-read turnaround).
9-8 EXT_WWT	Extended Write-To-Write Turnaround Specifies extra clock cycles that combine with TIMING_CFG_0[WWT] to define the write-to-write turnaround (see Write-to-write turnaround).
7-5 —	Reserved
4 EXT_REFINT	Extended Refresh Interval Specifies the number of memory bus clock cycles between refresh cycles. Depending on the setting of DDR_SDRAM_CFG_2[NUM_PR], some number of rows are refreshed in each DDR SDRAM physical bank during each refresh cycle. EXT_REFINT is concatenated with DDR_SDRAM_INTERVAL[REFINT] to obtain a 17-bit value for the total refresh interval, {EXT_REFINT, REFINT}. This combined refresh interval depends on the specific SDRAMs used and the interface clock frequency. Refreshes are not issued when {EXT_REFINT, EFINT} = 0. 0 - 0 1 - 65,536
3-2 —	Reserved
1-0	DDR SDRAM DLL Lock Time

Table continues on the next page...

Table continued from the previous page...

Field	Description
DLL_LOCK	Specifies the number of clocks that it takes for the DDR SDRAM to be available after exiting Self-Refresh mode. DDRC waits for the number of cycles that this field specifies before issuing any commands after exiting POR or after a self-refresh completes (t_{XSR}). 10 - 1024 clocks 11 - 2048 clocks All other values are reserved.

35.6.32 DDR SDRAM Timing Configuration 7 (TIMING_CFG_7)

Provides required timing fields.

Offset

Register	Offset
TIMING_CFG_7	16Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		CKE_RST		CKSRE				CKSRX				Reserved	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved				Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-30 —	Reserved
29-28 CKE_RST	MCKE Reset Time Specifies the number of clock cycles for which DDRC must wait after a reset and assertion of MCKE until the first MRW command.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field indicates MCKE reset time (t_{XPR}), and determines t_{XS} when exiting Self-Refresh mode. Therefore, if t_{XPR} and t_{XS} are different for a particular memory, you must select the greater of these two values and write that value to this field. 00 - 200 01 - 256 10 - 512 11 - 4096
27-24 CKSRE	Clock After Self-Refresh Entry Specifies the number of clock cycles for which DDRC must drive valid MCK and MCK_b values after entering Self-Refresh mode before the clocks are allowed to stop. This field indicates the number of valid clock cycles after entering Self-Refresh mode (t_{CKSRE}). When using an LPDDR4, LPDDR4x memory, you must write t_{ESCKE} to this field. 0000,1010 - 15 0001 - 6 0010 - 7 0011 - 8 0100 - 9 0101 - 10 0110 - 11 0111 - 12 1000 - 13 1001 - 14 1011 - 16 1100 - 17 1101 - 18 1110 - 19 1111 - 32
23-20 CKSRX	Clock After Self-Refresh Exit Specifies the number of clock cycles during which DDRC must drive valid MCK and MCK_b values before exiting Self-Refresh mode and before the clocks are allowed to stop. This field indicates a valid number of clock cycles after exiting Self-Refresh mode (t_{CKSRX}). When using LPDDR4 LPDDR4x DDR SDRAM, you must write t_{SR} to this field. 0000,1010 - 15 0001 - 6

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0010 - 7 0011 - 8 0100 - 9 0101 - 10 0110 - 11 0111 - 12 1000 - 13 1001 - 14 1011 - 16 1100 - 17 1101 - 18 1110 - 19 1111 - 32
19 —	Reserved
18-16 —	Reserved
15-8 —	Reserved
7-4 —	Reserved
3-0 —	Reserved

35.6.33 DDR SDRAM ZQ Calibration Control (DDR_ZQ_CNTL)

Enables the controls required for ZQ calibration.

When using ZQ calibration, the following define the limitations for DDR SDRAM timing parameters:

- DDR_ZQ_CNTL[ZQOPER]
- DDR_ZQ_CNTL[ZQCS]
- TIMING_CFG_1[PRETOACT]
- TIMING_CFG_1[REFREC]
- DDR_SDRAM_INTERVAL[REFINT]
- Number of enabled ranks

A refresh command can be skipped after a self-refresh completes if this condition is true:

(ZQOPER + ZQCS) × (# of enabled ranks) + PRETOACT + REFREC + $2t_{ck}$ > REFINT

In this scenario, you could use either of these options:

- Posted refreshes to extend the refresh interval
- DDR SDRAM Mode Control (DDR_SDRAM_MD_CNTL) to force an extra refresh to each rank after exiting Self-Refresh mode

However, timing parameters for most chips and frequencies do not allow missed refreshes.

Offset

Register	Offset
DDR_ZQ_CNTL	170h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ZQ_EN	Reserved			ZQINIT				Reserved				ZQOPER			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				ZQCS				Reserved				ZQCS_INT			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 ZQ_EN	ZQ Calibration Enable Determines if ZQ calibration is used. If it is used, DDRC issues a ZQCL command after POR and anytime DDRC completes a self-refresh. A ZQCS command is issued after every 32 refresh sequences to account for VT variations. 0 - Not used 1 - Used
30-28 —	Reserved
27-24 ZQINIT	ZQ Calibration Initialization Time Determines the number of clock cycles allowed for DDR SDRAM ZQ calibration at POR. This field signifies POR ZQ calibration time (t_{ZQinit}).

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Each rank is calibrated separately, and this time must elapse after DDRC issues a ZQCL command to the DDR SDRAM for each rank before a separate command may be issued. When using LPDDR4x DDR SDRAM, you must write t_{ZQCAL} to this field. 0111 - 128 1000 - 256 1001 - 512 1010 - 1024 1011 - 2048
23-20 —	Reserved
19-16 ZQOPER	ZQ Calibration Operation Time Determines the number of clock cycles that DDR SDRAM requires to complete the ZQ calibration when exiting Self-Refresh mode. This field signifies normal operation full calibration time (t_{ZQoper}). Each CS is calibrated separately, and this time must elapse after DDRC issues a ZQCL command to the DDR SDRAM for each CS before a separate command may be issued. When using LPDDR4 ,LPDDR4x memories, you must write t_{ZQCAL} to this field. 0111 - 128 1000 - 256 1001 - 512 1010 - 1024 1011 - 2048
15-12 —	Reserved
11-8 ZQCS	ZQ Calibration Short Time Determines the number of clock cycles allowed for DDR SDRAM ZQ calibration during dynamic calibration, which is issued after every refresh sequence that DDR_ZQ_CNTL[ZQCS_INT] defines. This field signifies normal operation short calibration time (t_{ZQCS}). Each CS is calibrated separately, and this time must elapse after DDRC issues a ZQCS command for each CS before a separate command may be issued. When using LPDDR4 ,LPDDR4x DDR SDRAM, you must write t_{ZQLAT} to this field. 0000 - 1 0001 - 2 0010 - 4

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0011 - 8 0100 - 16 0101 - 32 0110 - 64 0111 - 128 1000 - 256 1001 - 512
7-4 —	Reserved
3-0 ZQCS_INT	ZQCS Interval Determines the number of refresh sequences that are to pass between each ZQCS calibration. 0000 - 32 0001 - 64 0010 - 128 0011 - 256 0100 - 512 0101 - 1024 0110 - 2048 0111 - 4096 1000 - 8192 1001 - 16384 1010 - 32768 1111 - ZQCS calibration disabled All other values are reserved.

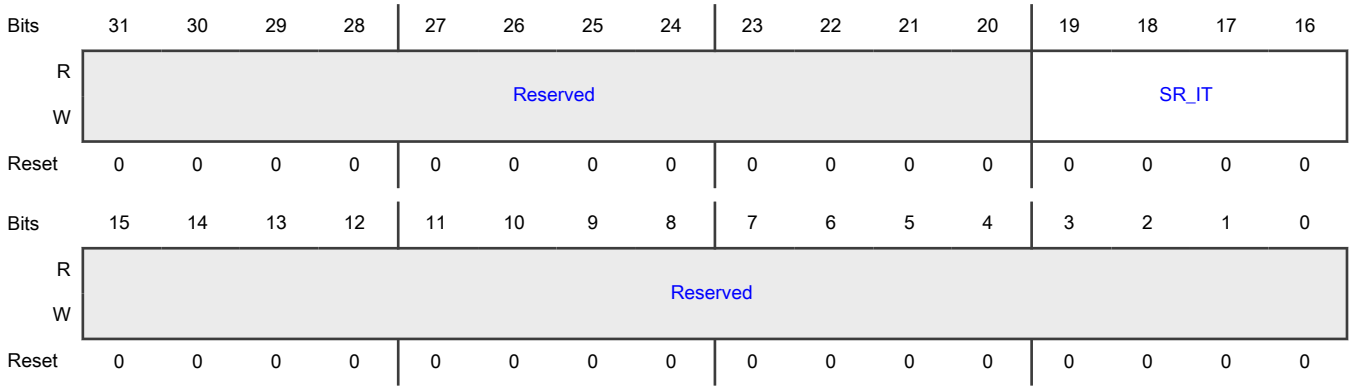
35.6.34 DDR SDRAM Self-Refresh Counter (DDR_SR_CNTR)

Enables you to force DDRC to perform a self-refresh after a predefined period of idle time.

Offset

Register	Offset
DDR_SR_CNTR	17Ch

Diagram



Fields

Field	Description
31-20 —	Reserved
19-16 SR_IT	Self-Refresh Idle Threshold Defines the number of DDR SDRAM clock cycles that must pass when DDRC is idle before it enters Self-Refresh mode, according to the following equation: Number of clock cycles = $2^{(8 + SR_IT \times 2)}$ **NOTE** SR_IT values 1-7 are not supported. Anytime you issue a transaction to DDRC, it exits Self-Refresh mode and resets its internal counter. • When this field is 0, this power-saving feature is disabled. • If you write a non-zero value to this field, DDRC exits Self-Refresh mode anytime a transaction is issued, regardless of the reason it entered this mode. However, when writing a non-zero value to this field, DDR_SDRAM_CFG_3[SR_FAST_WK_EN] must be 1.
15-0 —	Reserved

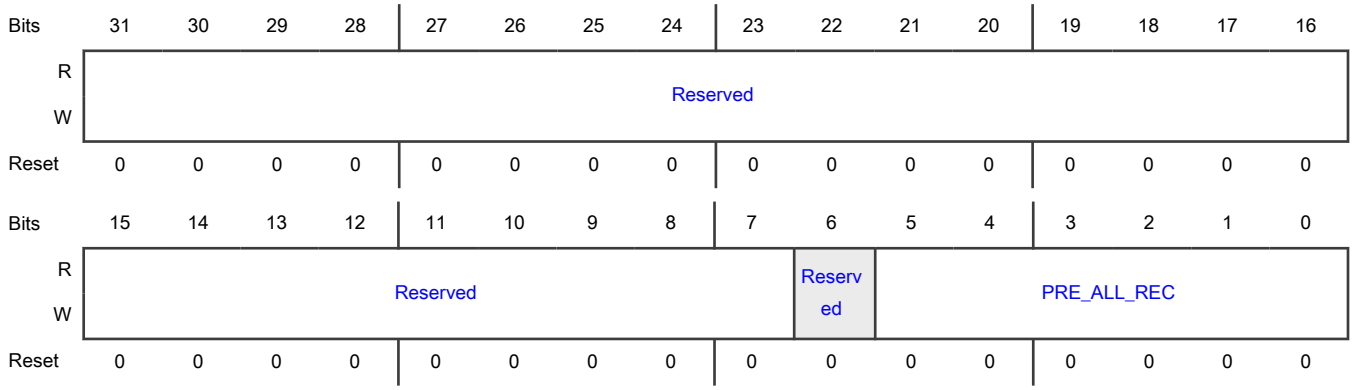
35.6.35 DDR SDRAM Timing Configuration 8 (TIMING_CFG_8)

Provides additional timing fields .

Offset

Register	Offset
TIMING_CFG_8	250h

Diagram



Fields

Field	Description
31-7 —	Reserved
6 —	Reserved
5-0 PRE_ALL_REC	Precharge All-To-Activate Interval Determines the number of clock cycles from a precharge-all command until an activate or refresh command is allowed. When this field is 0, DDRC uses the value of TIMING_CFG_1[PRETOACT] for this interval.

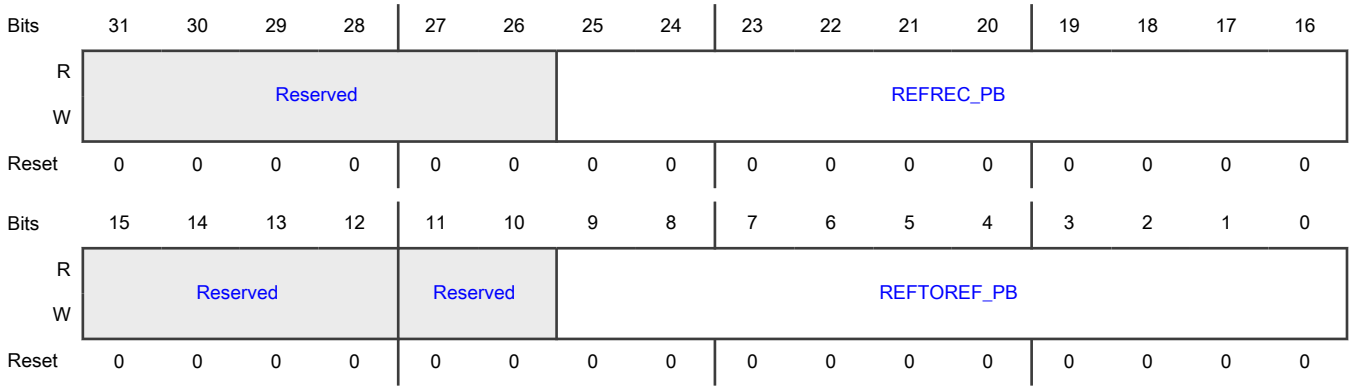
35.6.36 DDR SDRAM timing configuration 9 (TIMING_CFG_9)

The DDR SDRAM timing configuration 9 register provides additional timing fields required to support per-bank refresh.

Offset

Register	Offset
TIMING_CFG_9	254h

Diagram



Fields

Field	Description
31-26 —	Reserved
25-16 REFREC_PB	Refresh Recovery Per-Bank Refresh Refresh recovery time for per-bank refresh. Controls the number of clock cycles from a refresh command to an activate for the same bank. Note that hardware adds an additional 8 clock cycles to the final, 7-bit value of the refresh recovery, such that it (t_{RFCpb}) is calculated as follows: 00_0000_0000 - 8 clocks 00_0000_0001 - 9 clocks 00_0000_0010 - 10 clocks 10_1111_1110 - 774 clocks 10_1111_1111 - 775 clocks
15-12 —	Reserved
11-10 —	Reserved
9-0 REFTOREF_PB	Refresh-to-refresh interval for per-bank refresh. Note that hardware adds an additional 8 clock cycles to the final, 7-bit value of this field, such that it ($t_{pbR2pbR}$) is calculated as follows: 00_0000_0000 - disable PB refresh 00_0000_0001 - 9 clocks 00_0000_0010 - 10 clocks 10_1111_1110 - 774 clocks 10_1111_1111 - 775 clocks

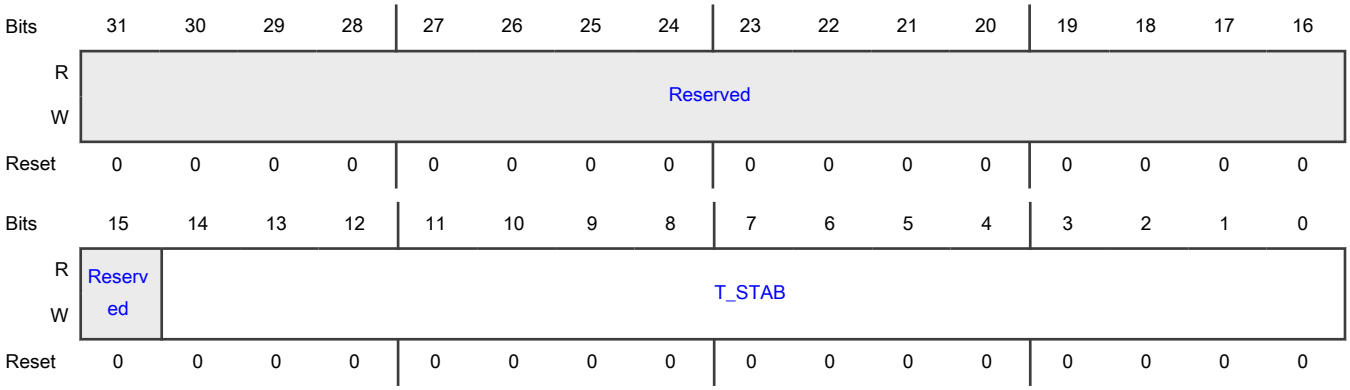
35.6.37 DDR SDRAM Timing Configuration 10 (TIMING_CFG_10)

Provides an additional timing field required to support wake up time for self refresh mode.

Offset

Register	Offset
TIMING_CFG_10	258h

Diagram



Fields

Field	Description
31-15 —	Reserved
14-0 T_STAB	Stabilization Wait Time Defines the stabilization wait time (in MCK clock cycles) when exiting self refresh in slow wakeup mode. If you write 0 to this field, DDRC ignores this constraint. **NOTE** Be aware of a special constraint if you expect the temperature to drift by more than 20 degrees Celsius. In this case, the value you write to this field must be t_{STAB} or 10 µs, whichever is greater, during self-refresh.

35.6.38 DDR SDRAM Timing Configuration 11 (TIMING_CFG_11)

Provides additional timing fields required to support LPDDR4 ,LPDDR4x .

Offset

Register	Offset
TIMING_CFG_11	25Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				Reserved	Reserved										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				PRE_TO_PRE				Reserved				MWWT			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27 —	Reserved
26-12 —	Reserved
11-8 PRE_TO_PRE	Precharge-To-Precharge Time Determines the number of clock cycles that can be issued from a precharge command to another one for the same rank (t_{PPD}). 0000,0100 - 4 0001 - 1 0010 - 2 0011 - 3 0101 - 5 0110 - 6 0111 - 7 1000 - 8 1001 - 9 1010 - 10 1011 - 11 1100 - 12 1101 - 13 1110 - 14

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1111 - 15
7-4 —	Reserved
3-0 MWWT	Masked Write-To-Write Turnaround (tCCDMW) Determines the number of clock cycles from a masked write until another write can be issued, according to the following equation: Number of clock cycles = MWWT + 32

35.6.39 DDR SDRAM Control Configuration 3 (DDR_SDRAM_CFG_3)

Provides additional control configuration for DDRC.

Offset

Register	Offset
DDR_SDRAM_CFG_3	260h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DDRC_RST	Reserved	Reserved		Reserved				Reserved		Reserved		Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	DM_CFG			DRAIN_F...	Reserved	Reserved		DYN_REF...	Reserved	Reserved		NON_PWR...	Reserved	SR_FAST...	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 DDRC_RST	DDRC Reset Specifies whether DDRC operates normally or is undergoing reset. You must write 1 to this field if you want to reset DDRC. This resets all state machines and internal FIFOs. The field does not reset DDRC's configuration registers. Before you write 1 to this field, you must write 0 to DDR_SDRAM_CFG[MEM_EN].

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field returns to 0 by itself. Therefore, you must poll this field and wait until it returns to 0 before re-enabling DDRC. 0 - Operating normally 1 - Undergoing reset
30 —	Reserved
29-28 —	Reserved
27-24 —	Reserved
23-22 —	Reserved
21-20 —	Reserved
19 —	Reserved
18 —	Reserved
17 —	Reserved
16 —	Reserved
15 —	Reserved
14-12 DM_CFG	Data Mask Configuration Determines how DDRC uses the MDM pins. When DM_CFG = 100b, DDRC supports both DBI and data masks. You can use this configuration only with LPDDR4 ,LPDDR4x DDR SDRAM. For more information on value 000b below, see DDR_SDRAM_CFG[SDRAM_TYPE]. 000 - Normal data masks based on the settings defined in DDR_SDRAM_CFG[SDRAM_TYPE] 010 - DBI

Table continues on the next page...

Table continued from the previous page...

Field	Description
	011 - Neither data masks nor DBI 100 - DBI with data masks All other values are reserved.
11 DRAIN_FOR_SR	Drain Queues For Self-Refresh Forces DDRC to drain its main command queue before servicing a self-refresh request. This field is only used for: • Software-initiated, self-refresh requests via DDR_SDRAM_CFG_2[FRC_SR] • Sleep requests • Self-refresh requests enabled in DDR_SR_CNTR[SR_IT] 0 - Do not drain 1 - Drain
10 —	Reserved
9-8 —	Reserved
7 DYN_REF_RATE_EN	Dynamic Refresh Rate Enable Enables only the dynamic refresh rate adjustments for LPDDR4 ,LPDDR4x DDR SDRAM. If you write 1 to this field, DDRC polls the MR4 register every 64 refresh sequences to determine a proper refresh rate. If two ranks are enabled, DDRC alternates between polling rank 0 and rank 1, and uses the lesser of the two refresh intervals. You can find the results of the previous MR4 reads in DDR Refresh Rate (DDR_SDRAM_REF_RATE). This feature cannot be used with rank 2 or rank 3 enabled. 0 - Disables 1 - Enables
6 —	Reserved
5-4 —	Reserved
3 NON_PWR_2	Non Power of 2 Enable This field controls whether a non-power of 2 density DRAM is used. For example, 3Gbit, 6Gbit, and 12Gbit DRAMs are considered non-power of 2 DRAMs. You must program the CSn_BNDS to include the full data available when using a non-power of 2 DRAM. 0 - Disables

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enables
2 —	Reserved
1 SR_FAST_WK_EN	Self Refresh Fast Wakeup Enable Controls the type of wakeup enable (slow or fast) used after a self-refresh. **NOTE** If you will ever use self refresh with this bit cleared after a given reset, then you must clear this bit during the initial configuration of the DDRC. 0 - Slow 1 - Fast
0 —	Reserved

35.6.40 DDR SDRAM Control Configuration 4 (DDR_SDRAM_CFG_4)

Provides more control configuration for DDRC, along with timings for blocking certain accesses.

Offset

Register	Offset
DDR_SDRAM_CFG_4	264h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PHY_WAKEUP				Reserved		PHY_FREQ_FSP		Reserved		PHY_FREQ_HS					PHY_FREQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHY_FREQ				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 PHY_WAKEUP	PHY Wakeup. This field represents the value used for the phy_wakeup pins to the DDR PHY.
27-26 —	Reserved
25-24 PHY_FREQ_FSP	PHY Frequency FSP. This field represents the bits that will be provided on the phy_freq_fsp pins to the DDR PHY.
23-22 —	Reserved
21-17 PHY_FREQ_HS	PHY Frequency This field represents the bits that will be provided on the phy_freq pins to the DDR PHY.
16-12 PHY_FREQ	Frequency Change and Retention Setup This field represents the bits that will be provided on the phy_freq pins to the DDR PHY. This field needs to be setup prior to changing frequency or entering retention mode. 0_0000 – Frequency State 0 0_0001 – Frequency State 1 0_0010 – Frequency State 2 Prior to entering retention mode, it is necessary to program the values as follows: 1_1111 – Retention mode All other values are reserved.
11-0 —	Reserved

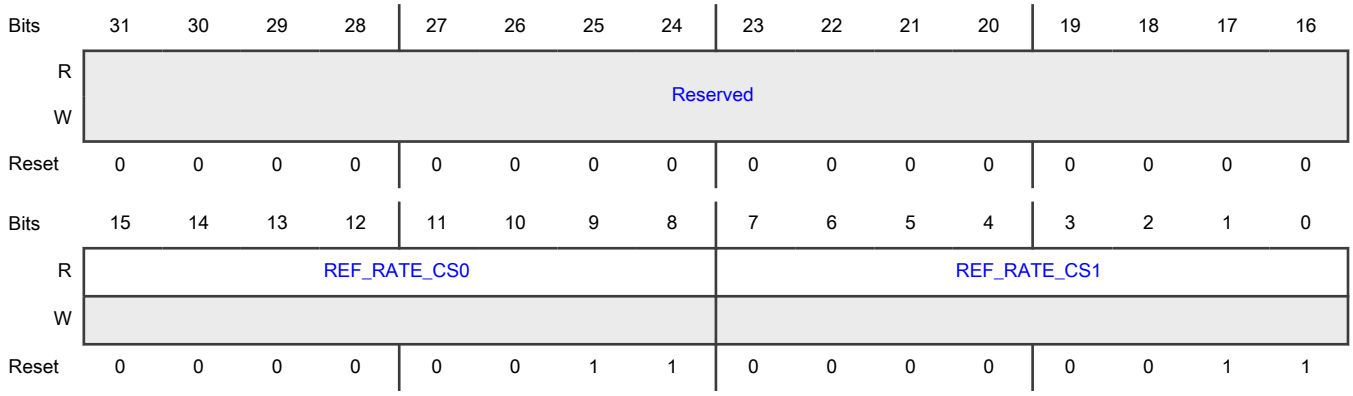
35.6.41 DDR Refresh Rate (DDR_SDRAM_REF_RATE)

Provides the rank-0 and rank-1 status of the last MR4 read to determine the refresh rate.

Offset

Register	Offset
DDR_SDRAM_REF_RATE	2C0h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 REF_RATE_CS0	Refresh Rate Rank 0 Represents the value read for the MR4 register on rank 0 during the last dynamic read. This field provides meaningful information only if DDR_SDRAM_CFG_3[DYN_REF_RATE_EN] = 1 and rank 0 is enabled.
7-0 REF_RATE_CS1	Refresh Rate Rank 1 Represents the value read for the MR4 register on rank 1 during the last dynamic read. This field provides meaningful information only if DDR_SDRAM_CFG_3[DYN_REF_RATE_EN] = 1 and rank 1 is enabled.

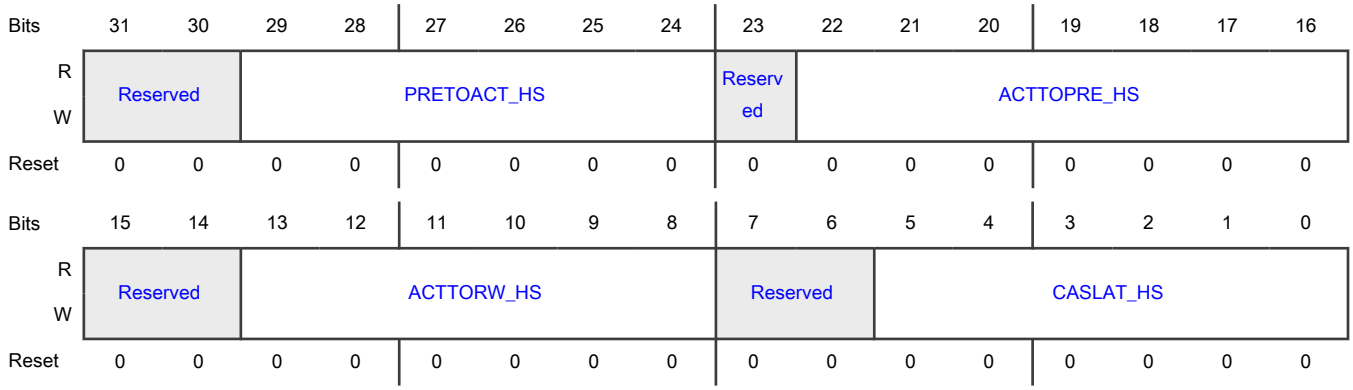
35.6.42 DDR SDRAM Timing Configuration 12 (TIMING_CFG_12)

Provides additional timing fields required to support half-frequency operation when dynamically switching the clock frequency.

Offset

Register	Offset
TIMING_CFG_12	300h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-24 PRETOACT_HS	Precharge-To-Activate Time For Half Speed Defines clock cycles to define t_{RP} when operating at half frequency. 00_0000 - Reserved All other values - Clock cycles as defined in the description
23 —	Reserved
22-16 ACTTOPRE_HS	Activate-To-Precharge Time For Half Speed Defines clock cycles to define t_{RAS} when operating at half frequency.
15-14 —	Reserved
13-8 ACTTORW_HS	Activate To Read Or Write For Half Speed Defines clock cycles that define t_{RCD} when operating at half frequency. 00_0000 - Reserved All other values - Clock cycles as defined in the description
7-6 —	Reserved
5-0 CASLAT_HS	CAS Latency For Half Speed Defines clock cycles that define the CAS latency when operating at half frequency. This field must be programmed to 1 less than the desired CAS latency, similar to {TIMING_CFG_3[EXT_CASLAT], TIMING_CFG_1[CASLAT]}.

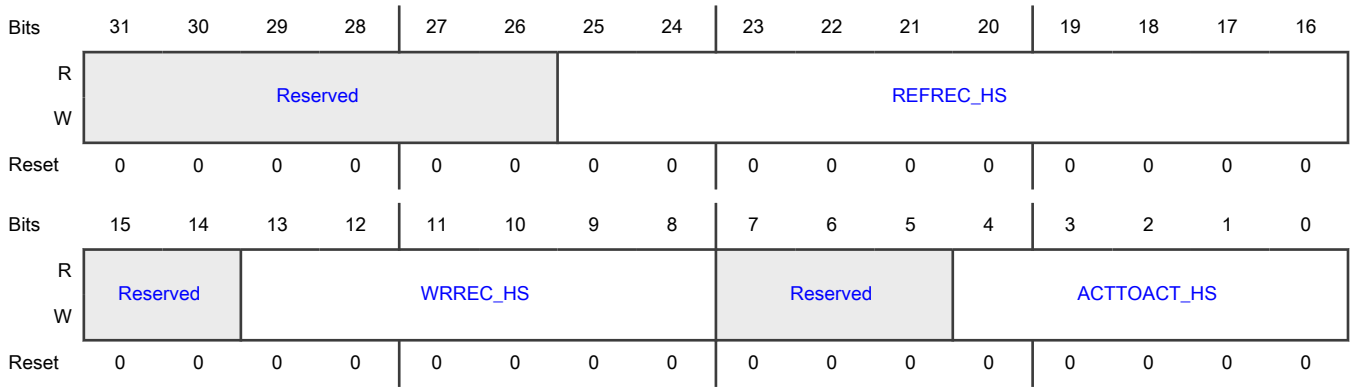
35.6.43 DDR SDRAM Timing Configuration 13 (TIMING_CFG_13)

Provides additional timing fields required to support half-frequency operation when dynamically switching the clock frequency.

Offset

Register	Offset
TIMING_CFG_13	304h

Diagram



Fields

Field	Description
31-26 —	Reserved
25-16 REFREC_HS	Refresh Recovery For Half Speed Defines clock cycles to define t_{RFC} when operating at half frequency. Note that the DDRC adjusts this field by 8 cycles, similar to {TIMING_CFG_3[EXT_REFREC], TIMING_CFG_1[REFREC]}.
15-14 —	Reserved
13-8 WRREC_HS	Write Recovery For Half Speed Defines clock cycles to define n_{WR} when operating at half frequency. The value 0b is reserved. 00_0000 - Reserved All other values - Clock cycles as defined in the description
7-5 —	Reserved
4-0	Activate-To-Activate Interval For Half Speed

Table continues on the next page...

Table continued from the previous page...

Field	Description
ACTTOACT_HS	Specifies the number of clock cycles from an activate command until another activate command is allowed for a different logical bank in the same physical bank when operating at half frequency. This field indicates the activate-to-activate interval (t_{RRD}). 0_0000 - Reserved All other values - Clock cycles as defined in the description

35.6.44 DDR SDRAM Timing Configuration 14 (TIMING_CFG_14)

Provides additional timing fields required to support half-frequency operation when dynamically switching the clock frequency.

Offset

Register	Offset
TIMING_CFG_14	308h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				WRLAT_HS				Reserved	RD_TO_PRE_HS					Reserved	REFINT_...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	REFINT_HS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-30 —	Reserved
29-24 WRLAT_HS	Write Latency For Half Speed Indicates the number of clock cycles used to determine the write latency when operating at half frequency. The maximum write latency supported is 34 clocks.
23 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
22-18 RD_TO_PRE_HS	Read-To-Precharge Time For Half Speed Specifies the number of clock cycles indicating read-to-precharge (n_{RTP}) when operating at half frequency. The value 0b is reserved. 0_0000 - Reserved All other values - Clock cycles as defined in the description
17 —	Reserved
16-0 REFINT_HS	Refresh Interval For Half Speed Represents the number of memory bus clock cycles between refresh cycles when operating at half frequency. This way, the field controls auto-refresh. Writing 0 to this field turns off auto-refresh. Depending on DDR_SDRAM_CFG_2[NUM_PR], some quantity of rows is refreshed in each DDR SDRAM physical bank during each refresh cycle. The value of this field depends on the specific DDR SDRAM used and the interface clock frequency.

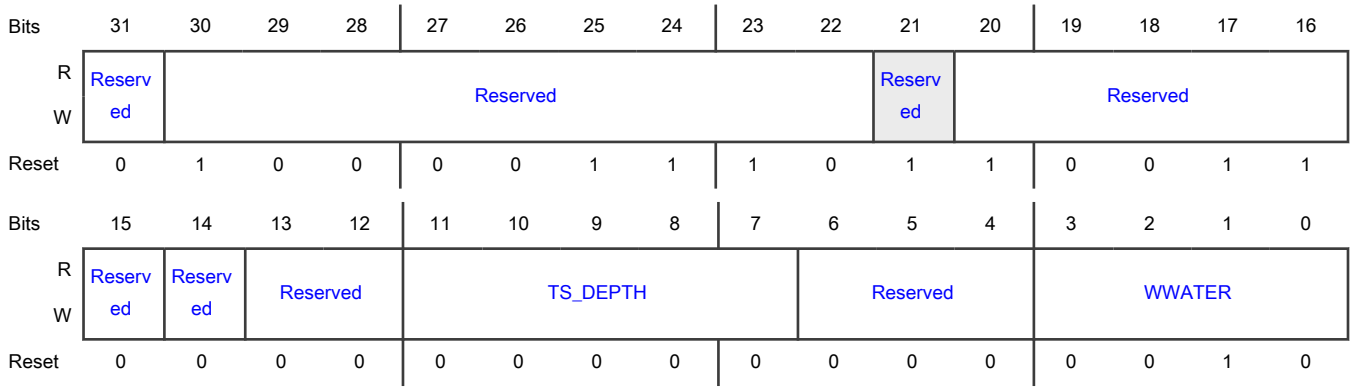
35.6.45 Transaction Configuration Register 1 (TX_CFG_1)

This register contains miscellaneous settings to fine-tune arbitration performance.

Offset

Register	Offset
TX_CFG_1	800h

Diagram



Fields

Field	Description
31 —	Reserved
30-22 —	Reserved
21 —	Reserved
20-15 —	Reserved
14 —	Reserved
13-12 —	Reserved
11-7 TS_DEPTH	Transaction Scheduler Depth This field determines the depth of the Transaction Scheduler. A value of 0b00000 provides the maximum number of entries. You must program the depth to 0b11010 or less if ERR_EN[INLINE_ECC_EN] is set.
6-4 —	Reserved
3-0 WWATER	Write Watermark. Write queue level before writes allowed to arbitrate. This field must be set to 0b0000 if ERR_EN[INLINE_ECC_EN] is set.

35.6.46 DDR SDRAM Debug Status 2 (DDRDSR_2)

Contains the current settings of the driver impedance.

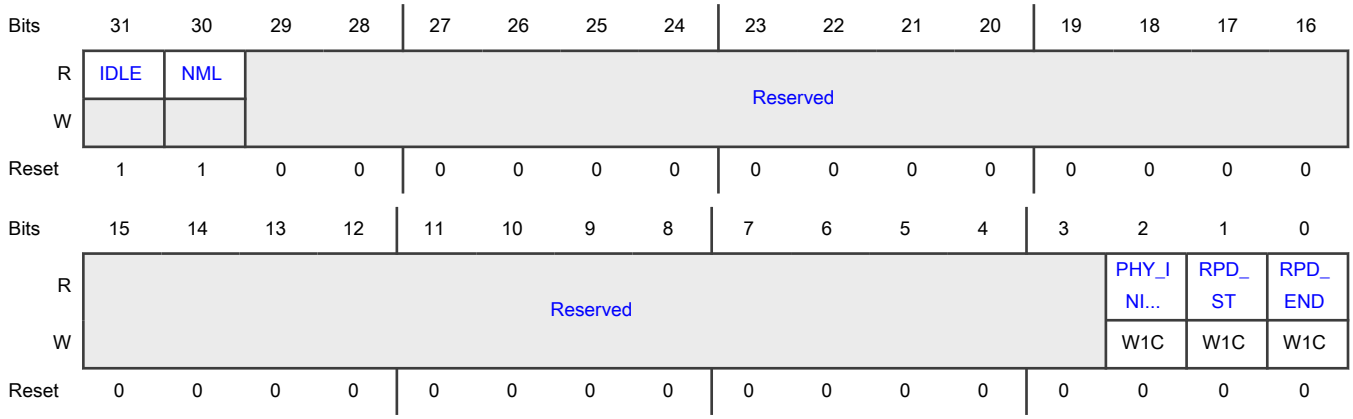
NOTE

Rapid Clear is done only on memory range selected by MTCR register. Address range in MTCR can override rapid clear. Rapid clear is not supported if REMAPing is enabled. Rapid clear will follow similar rules that D_INIT does if REMAP ing is used. Rapid memory status bits must be cleared before accessing memory again.

Offset

Register	Offset
DDRDSR_2	B24h

Diagram



Fields

Field	Description
31 IDLE	Memory controller idle (read only). 0 - Memory controller is busy. 1 - Memory controller is idle.
30 NML	No Modified Lines Indicates whether modified lines exist in the ECC cache. 0 - Exist 1 - Do not exist
29-3 —	Reserved
2 PHY_INIT_CMPLT	DDR PHY Initialization Complete Specifies whether PHY initialization is complete. DDRC changes this field to 1 after the DDR PHY initialization is complete. You must wait until this field becomes 1 before writing 1 to DDR_SDRAM_CFG[MEM_EN]. 0 - Not complete 1 - Complete
1 RPD_ST	Rapid Clear Of Memory Start Specifies whether the rapid clear of memory function has started. If the rapid clear of memory feature is available for DDRC, this field indicates that the sequence has started. After the sequence starts, writes to DDRC's memory-mapped registers do not affect this register's contents. 0 - Not started 1 - Started
0	Rapid Clear Of Memory End

Table continues on the next page...

Table continued from the previous page...

Field	Description
RPD_END	Specifies whether the rapid clear of memory function is complete. If the rapid clear of memory feature is available for DDRC, this field indicates a successful end of the rapid clear sequence. 0 - Not complete 1 - Complete

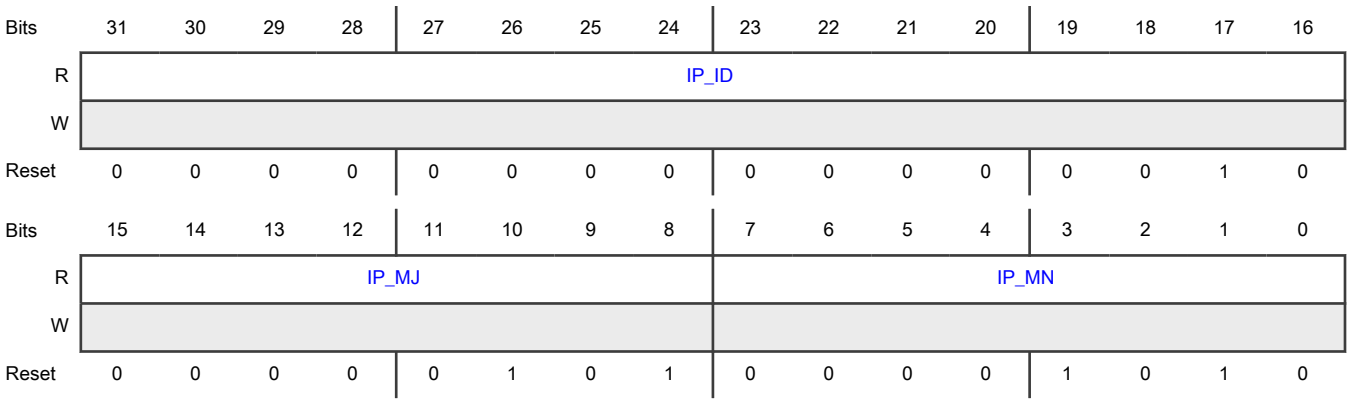
35.6.47 DDRC Revision 1 (DDR_IP_REV1)

Provides the DDRC ID, major revision, and minor revision.

Offset

Register	Offset
DDR_IP_REV1	BF8h

Diagram



Fields

Field	Description
31-16 IP_ID	IP Block ID For DDRC, this value is 2h.
15-8 IP_MJ	Major Revision This is the major revision for DDRC.
7-0 IP_MN	Minor Revision For DDRC, this value is 8'd10h.

35.6.48 DDR SDRAM Memory Test Control (DDR_MTCR)

Enables the controls for an automatic memory test.

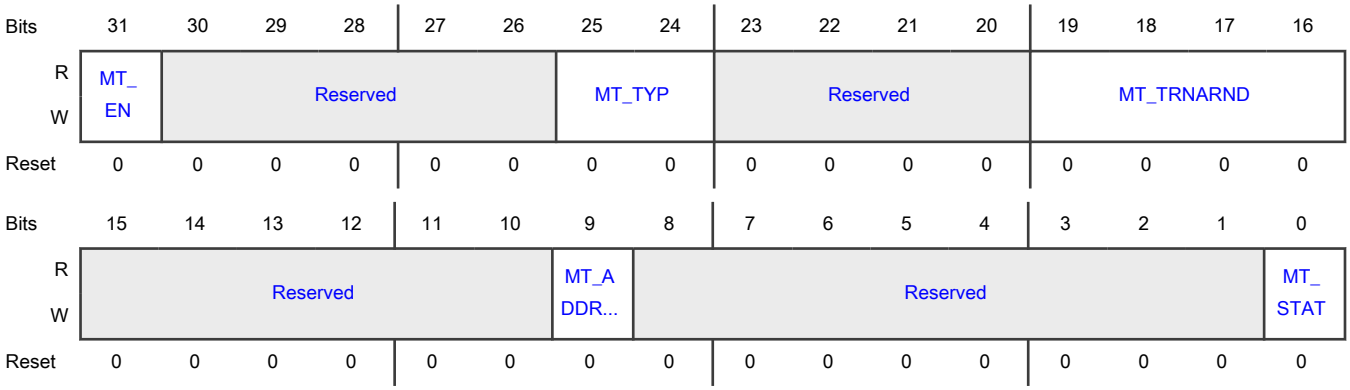
NOTE

You must clear the fields in this register after running a memory test.

Offset

Register	Offset
DDR_MTCR	D00h

Diagram



Fields

Field	Description
31 MT_EN	Memory Test Enable Determines the status of the memory test. You can write 1 to this field to enable the memory test. Depending on the value of DDR_MTCR[MT_TYP], DDRC issues only reads, only writes, or both reads and writes. DDRC writes 0 to this field after the memory test is complete. DDRC: • Issues transactions throughout the memory, as defined by Rank n Configuration (CS0_CONFIG - CS1_CONFIG) and Rank n Memory Bounds (CS0_BNDS - CS1_BNDS). • Checks the data during the memory test. If ECC is enabled, you must read Memory Error Detect (ERR_DETECT) after the memory test to determine whether any ECC errors occurred during the test. Then: • If an ECC error is detected, the error capture registers hold the address, data, and attributes captured for the first fail. • If no ECC errors are detected and the test fails (for a data miscompare), the capture registers hold information for the transaction that caused the first data miscompare.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Transactions that fail with ECC errors are prioritized in capture registers. DDRC writes 0 to this field after the memory test is complete. You can write 1 to this field in either of these conditions: • Before DDR_SDRAM_CFG[MEM_EN] becomes 1 • After the memory controller is enabled and idle You cannot write 1 to this field while the DDRC is processing other transactions. You cannot write 1 to this field when DDR_SDRAM_CFG_2[D_INIT] = 1. 0 - Disables 1 - Enables
30-26 —	Reserved
25-24 MT_TYP	Memory Test Type Determines whether the memory test uses only reads, only writes, or both reads and writes. You must not use the only-read test unless you have already initialized the memory. 00 - Both writes and reads 01 - Only writes 10 - Only reads 11 - Reserved
23-20 —	Reserved
19-16 MT_TRNARND	Memory Test Turnaround Determines the number of writes issued during the memory test before reads to the same addresses are issued. This field allows longer streams of writes and reads, and it can be used to test the write→read and read→write turnarounds. The field is only relevant if DDR_MTCR[MT_TYP] = 0. 0000 - Entire memory is written to before read transactions are issued. 0001 - Total write and read streams are one transaction each. 0010 - Total write and read streams are two transactions each. 0011 - Total write and read streams are four transactions each. All other values are reserved.
15-10 —	Reserved
9	Memory Test Address Range Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
MT_ADDR_EN	Specifies the memory range used for the memory test. • If this field is 0, the full memory range defined by Rank n Memory Bounds (CS0_BNDS - CS1_BNDS) is used. • If this field is 1, the address range defined by DDR SDRAM Memory Test Start Extended Address (DDR_MT_ST_EXT_ADDR), DDR SDRAM Memory Test Start Address (DDR_MT_ST_ADDR), DDR SDRAM Memory Test End Extended Address (DDR_MT_END_EXT_ADDR), and DDR SDRAM Memory Test End Address (DDR_MT_END_ADDR) is used. To use this field to limit the address range to be tested, for any individual test, the value of neither of the following can be less than that used during the previous test: • The starting address (DDR_MT_ST_EXT_ADDR || DDR_MT_ST_ADDR) • The ending address (DDR_MT_END_EXT_ADDR || DDR_MT_END_ADDR) Additionally, you cannot write 1 to this field if DDR_MTCR[MT_TRNARND] = 0. 0 - Memory range that the CSn_BNDS registers define 1 - Memory range that the DDR_MT_ST_EXT_ADDR, DDR_MT_ST_ADDR, DDR_MT_END_EXT_ADDR, and DDR_MT_END_ADDR registers define
8-1 —	Reserved
0 MT_STAT	Memory Test Status Specifies whether a fail or data miscompare is detected during the memory test. After DDRC writes 0 to DDR_MTCR[MT_EN], this field becomes 1 in case of a data mismatch during a memory test read. In case of a failure during the memory test (that is, a data miscompare), this field becomes 1 at the same time DDR_MTCR[MT_EN] becomes 0. You can write 0 to this field after DDRC writes 1 to it to indicate a data mismatch. **NOTE** You must ignore this status bit if ERR_EN[INLINE_ECC_EN] is set. If you enable inline ECC, then the ERR_DETECT register can be monitored for detected errors during the memory test. 0 - No fail detected 1 - Data miscompare detected

35.6.49 DDR SDRAM Memory Test Pattern n (DDR_MTP0 - DDR_MTP9)

Provides the data pattern that is written during the *n*th set of 32 bits of each 40-byte memory test pattern.

This field works with DDR_MTCR[MT_EN] to enable the memory write and read test.

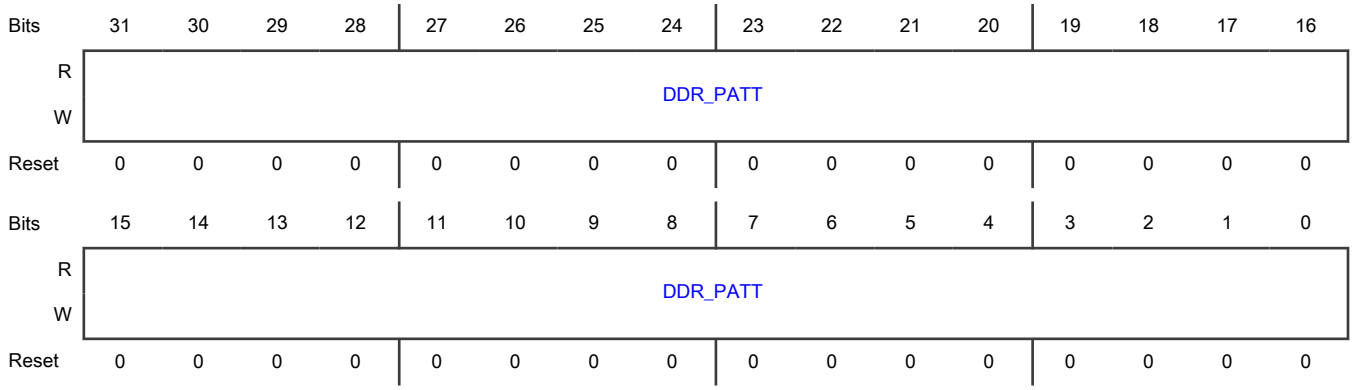
The memory test read compares data that is written to this register.

Offset

For n = 0 to 9:

Register	Offset
DDR_MTPn	D20h + (n × 4h)

Diagram



Fields

Field	Description
31-0 DDR_PATT	DDR SDRAM Pattern Stores the programmable pattern for the memory test.

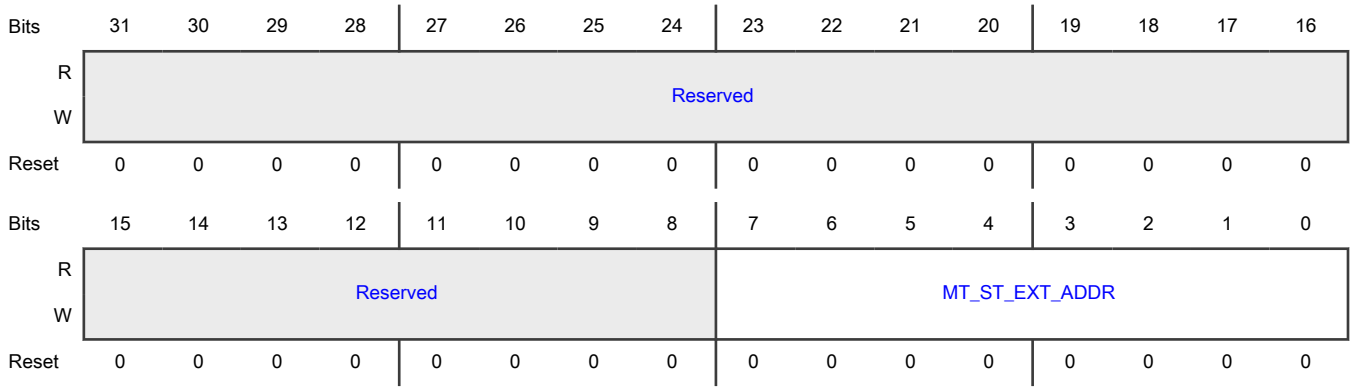
35.6.50 DDR SDRAM Memory Test Start Extended Address (DDR_MT_ST_EXT_ADDR)

Works with DDR_MTCR[MT_ADDR_EN] to specify the extended address for the DDR memory test start.

Offset

Register	Offset
DDR_MT_ST_EXT_ADDR	D60h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 MT_ST_EXT_ADDR	Memory Test Start Extended Address Represents the starting extended address that is used for memory testing when DDR_MTCR[MT_ADDR_EN] = 1. See Memory-test address restrictions for restrictions on what you can specify.

35.6.51 DDR SDRAM Memory Test Start Address (DDR_MT_ST_ADDR)

Works with DDR_MTCR[MT_ADDR_EN] to specify the DDR memory test start address.

Offset

Register	Offset
DDR_MT_ST_ADDR	D64h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MT_ST_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MT_ST_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MT_ST_ADDR	Memory Test Start Address Specifies the starting address that is used for memory testing when DDR_MTCR[MT_ADDR_EN] = 1. See Memory-test address restrictions for restrictions on what you can specify.

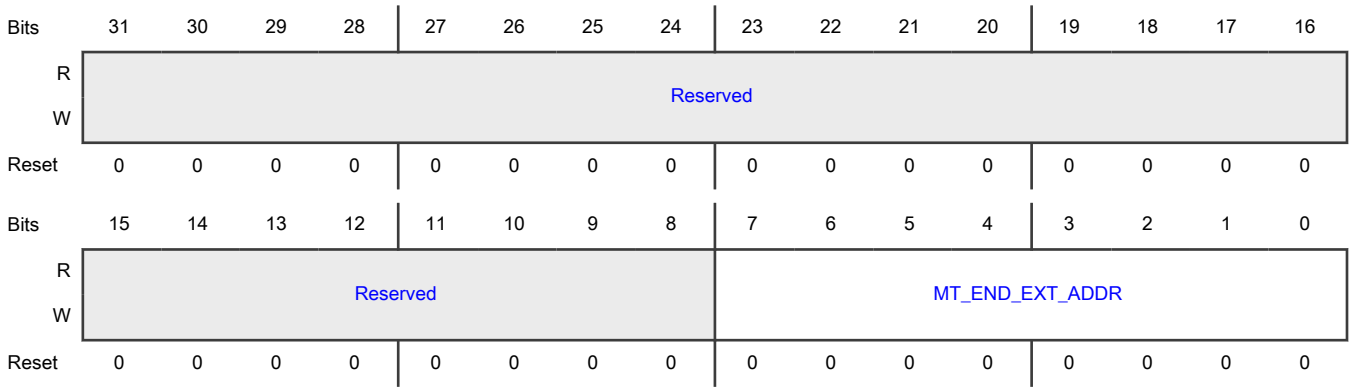
35.6.52 DDR SDRAM Memory Test End Extended Address (DDR_MT_END_EXT_ADDR)

Works with DDR_MTCR[MT_ADDR_EN] to specify the DDR memory test end extended address.

Offset

Register	Offset
DDR_MT_END_EXT_ADDR	D68h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 MT_END_EXT_ADDR	Memory Test End Extended Address Specifies the ending extended address that is used when DDR_MTCR[MT_ADDR_EN] = 1. See Memory-test address restrictions for restrictions on what you can specify.

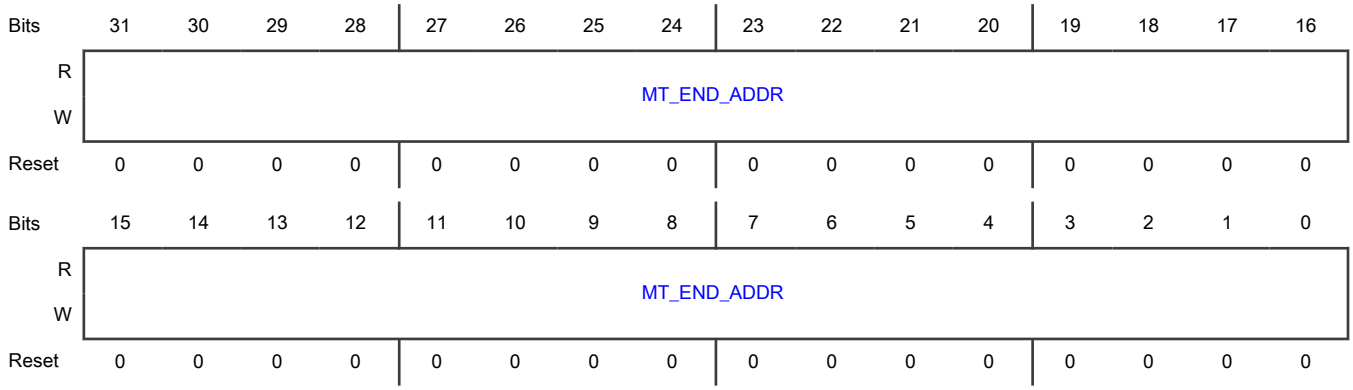
35.6.53 DDR SDRAM Memory Test End Address (DDR_MT_END_ADDR)

Works with DDR_MTCR[MT_ADDR_EN] to specify the DDR memory test end address.

Offset

Register	Offset
DDR_MT_END_ADDR	D6Ch

Diagram



Fields

Field	Description
31-0 MT_END_ADDR	Memory Test End Address Specifies the ending address that is used when DDR_MTCR[MT_ADDR_EN] = 1. See Memory-test address restrictions for restrictions on what you can specify.

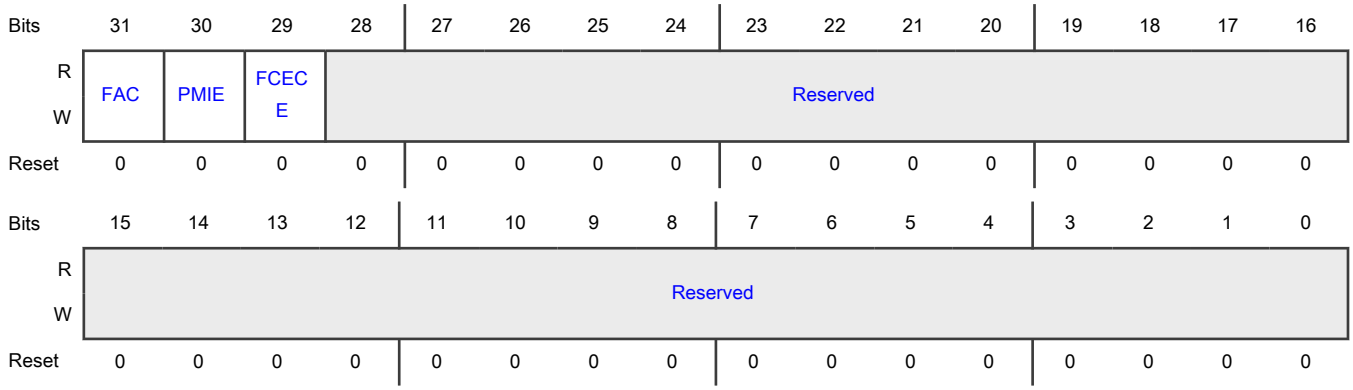
35.6.54 Performance Monitor Global Control (PMGC0)

Controls all PMC's.

Offset

Register	Offset
PMGC0	E00h

Diagram



Fields

Field	Description
31 FAC	Freeze All Counters Specifies whether PMCs are incremented. They are if PMLCA*n*[FC] permits. If PMCs are not incremented, the performance monitor signals an interrupt when PMGC0[FCECE] = 1. 0 - Incremented 1 - Not incremented
30 PMIE	Performance Monitor Interrupt Enable Enables performance monitor interrupts that PMC overflows lead to. Interrupts are enabled and occur when an enabled condition or event occurs. 0 - Disables 1 - Enables
29 FCECE	Freeze Counters On Enabled Condition Or Event Specifies when PMCs can be incremented. These are the conditions: • If PMLCA*n*[CE] is enabled • If the most-significant bit of the counter is 1, that is, when PMC*n* = 1 The use of the trigger and freeze counter conditions depends on the enabled condition. 0 - Enabled if PMLCAn[CE] = 1, until an event or condition occurs. 1 - Enabled if PMLCAn[CE] = 1, until an event or condition occurs. At this point, if PMGC0[FAC] = 1, you must write 0 to it.
28-0 —	Reserved

35.6.55 Performance Monitor Local Control A0 (PMLCA0)

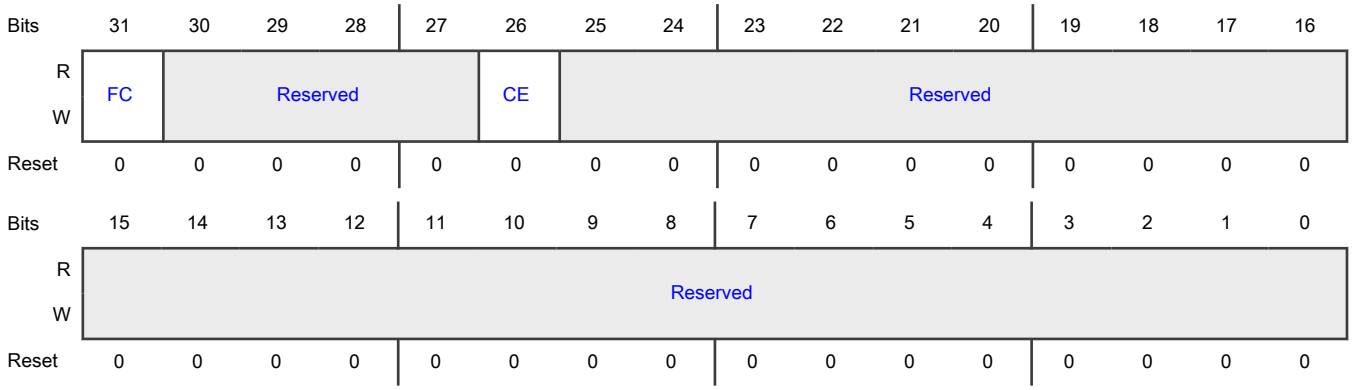
Works with Performance Monitor Local Control B0 (PMLCB0) to associate with an individual counter for specifying the usage of that counter and the event monitored on the counter.

This register controls the operations of the PMC0*n* registers.

Offset

Register	Offset
PMLCA0	E10h

Diagram



Fields

Field	Description
31 FC	Freeze Counter Indicates whether PMCs are enabled and incremented. They are if PMGC0[FAC] permits. This field enables the corresponding PMC*n* register to increment. 0 - Enabled 1 - Disabled
30-27 —	Reserved
26 CE	Condition Enable Controls counter overflow conditions. You must write 0 to this field when PMC0*n* is used as a trigger or is selected for chaining. 0 - Counter overflow conditions for PMC0n cannot occur (PMC0n cannot cause interrupts or freeze counters) 1 - Counter overflow conditions occur when the most-significant bit of PMC0n is 1
25-0 —	Reserved

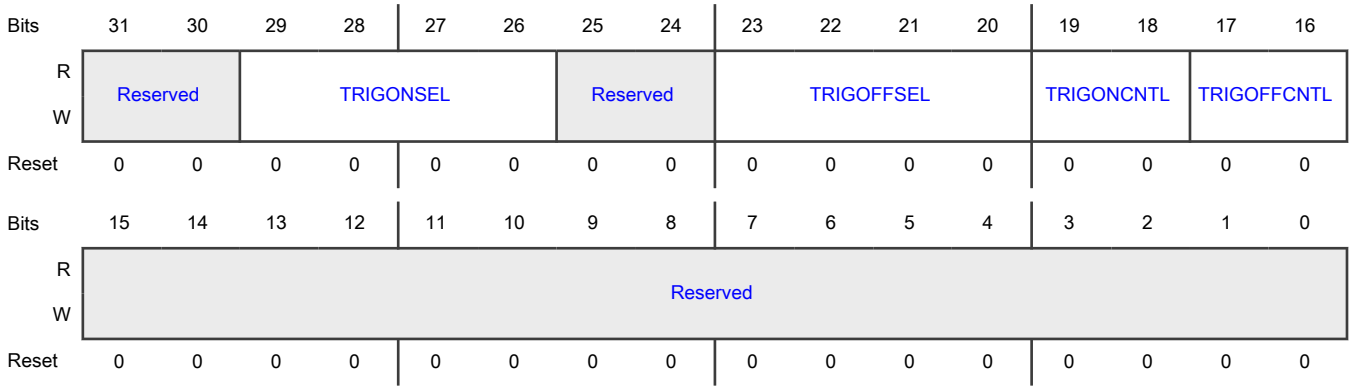
35.6.56 Performance Monitor Local Control B0 (PMLCB0)

Associates with counter 0 to specify how the counter is used and the events monitored on the counter.

Offset

Register	Offset
PMLCB0	E14h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-26 TRIGONSEL	Trigger-On Select Specifies the number of the counter that starts event counting. When the corresponding TRIGONCNTL event overflows, the current counter begins counting. No triggering occurs if the value of this field is self-referential, that is, when the value of this field is set to the current counter number.
25-24 —	Reserved
23-20 TRIGOFFSEL	Trigger-Off Select Specifies the number of the counter that stops event counting. When the corresponding TRIGONCNTL event overflows, the current counter stops counting. No triggering occurs if the value of this field is self-referential, that is, when the value of this field is set to the current counter number.
19-18 TRIGONCNTL	Trigger-On Control Indicates the condition that triggers the counting to start. 00 - Triggering off (no triggering to start) 01 - Trigger on change 10 - Trigger on overflow 11 - Reserved
17-16 TRIGOFFCNTL	Trigger-Off Control Indicates the condition that triggers the counting to stop. 00 - Triggering off (no triggering to start)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	01 - Trigger on change 10 - Trigger on overflow 11 - Reserved
15-0 —	Reserved

35.6.57 PMC 0a (PMC0a)

Holds the lower 32 bits of a 64-bit clock-cycle counter. PMC 0b (PMC0b) holds the upper 32 bits of that counter. This counter counts events that Performance Monitor Local Control A (PMLCA1 - PMLCA10) and Performance Monitor Local Control B (PMLCB1 - PMLCB10) select.

Offset

Register	Offset
PMC0a	E18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PMC0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PMC0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PMC0	Counter 0 Counts only clock cycles. This field specifies a read-write counter, which you can program with an initial value to count from.

35.6.58 PMC 0b (PMC0b)

Holds the upper 32 bits of a 64-bit clock-cycle counter. PMC 0a (PMC0a) holds the lower 32 bits of that counter. This counter counts events that Performance Monitor Local Control A (PMLCA1 - PMLCA10) and Performance Monitor Local Control B (PMLCB1 - PMLCB10) select.

Offset

Register	Offset
PMC0b	E1Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PMC0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PMC0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PMC0	Counter 0 Counts only clock cycles. This field specifies a read-write counter, which you can program with an initial value to count from.

35.6.59 Performance Monitor Local Control A (PMLCA1 - PMLCA10)

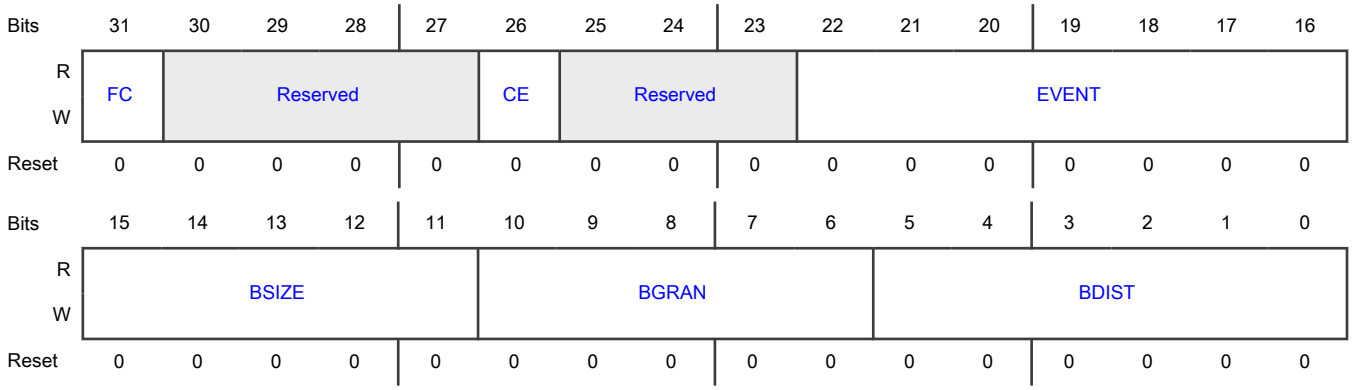
Controls the operation of the PMC*n* registers.

Offset

For a = 1 to 10:

Register	Offset
PMLCAa	E10h + (a × 10h)

Diagram



Fields

Field	Description
31 FC	Freeze Counter Indicates whether PMCs are enabled and incremented. They are if PMGC0[FAC] permits. This field enables the corresponding PMC*n* register to increment. 0 - Enabled 1 - Disabled
30-27 —	Reserved
26 CE	Condition Enable Controls the counter overflow conditions for PMC0. These conditions occur when the most-significant bit of PMC*n* is 1. When the conditions do not occur, they cannot cause interrupts or freeze counters. You must write 0 to this field when PMC0 is used as a trigger or is selected for chaining. 0 - Counter overflow conditions cannot occur 1 - Counter overflow conditions occur
25-23 —	Reserved
22-16 EVENT	Event Selector Selects events to be counted. See Event signals table for a list of events, counters, and the corresponding EVENT field values.
15-11 BSIZE	Burst Size Represents the fewest event occurrences that constitute a burst, which means, a rapid sequence of events followed by a relatively long pause. A value less than two events implies regular event counting. Any non-threshold, regular event may be counted in a bursty fashion.

Table continues on the next page...

Table continued from the previous page...

Field	Description
10-6 BGRAN	Burst Granularity Represents the maximum number of clock cycles between events that are considered part of a single burst.
5-0 BDIST	Burst Distance Specifies the burst distance, which is the number of clock cycles between bursts. This field works with PMLCB*n*[TBMULT] and must have a value greater than PMLCA*n*[BSIZE] for proper burstiness counting behavior. Writing 0 to this field indicates regular counting.

35.6.60 Performance Monitor Local Control B (PMLCB1 - PMLCB10)

Associates with counter 0 to specify how the counter is used and the events monitored on that counter.

Offset

For a = 1 to 10:

Register	Offset
PMLCBa	E14h + (a × 10h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		TRIGONSEL				Reserved		TRIGOFFSEL				TRIGONCNTL		TRIGOFFCNTL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				TBMULT				Reserved		THRESHOLD					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-30 —	Reserved
29-26	Trigger-On Select

Table continues on the next page...

Table continued from the previous page...

Field	Description
TRIGONSEL	Specifies the number of the counter that starts event counting. When the corresponding TRIGONCNTL event overflows, the current counter begins counting. No triggering occurs if the value of this field is self-referential, that is, when it is set to the current counter number.
25-24 —	Reserved
23-20 TRIGOFFSEL	Trigger-Off Select Specifies the number of the counter that stops event counting. When the corresponding TRIGONCNTL event overflows, the current counter stops counting. No triggering occurs if the value of this field is self-referential, that is, when it is set to the current counter number.
19-18 TRIGONCNTL	Trigger-On Control Indicates the condition that triggers the counting to start. 00 - Triggering off (no triggering to start) 01 - Trigger on change 10 - Trigger on overflow 11 - Reserved
17-16 TRIGOFFCNTL	Trigger-Off Control Indicates the condition that triggers the counting to stop. 00 - Trigger off (no triggering to start) 01 - Trigger on change 10 - Trigger on overflow 11 - Reserved
15-11 —	Reserved
10-8 TBMULT	Threshold And Burstiness Multiplier Represents threshold and burstiness multiplier. Threshold events are counted when the event duration exceeds a specified threshold value. The threshold is scaled based on the settings of this field, which is also used to scale the burst distance for burstiness counting. For all events that scale the threshold, the threshold field is multiplied from 1 to 128. This field is not used to scale the threshold value for quantity threshold events. 000 - 1 001 - 2 010 - 4 011 - 8

Table continues on the next page...

Table continued from the previous page...

Field	Description
	100 - 16 101 - 32 110 - 64 111 - 128
7-6 —	Reserved
5-0 THRESHOLD	Threshold Specifies a value that is used as an indicator to start counting events. When used, only those events are counted for which the value exceeds this programmed threshold value. For example, if PMC2 counts events for which the duration exceeds the threshold, you can obtain the distribution of those events for a given program by monitoring the program repeatedly, using a different threshold value each time.

35.6.61 Performance Monitor Counter (PMC1)

Specifies counters that can monitor 64 unique events in addition to the 64 reference events that the full set of PMC*n* registers can count.

Offset

Register	Offset
PMC1	E28h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PMC1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PMC1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0	Event Count

Table continues on the next page...

Field	Description
PMC1	Indicates an overflow when the most-significant bit of this counter is 1. Writing 1 to this field may cause an immediate interrupt.

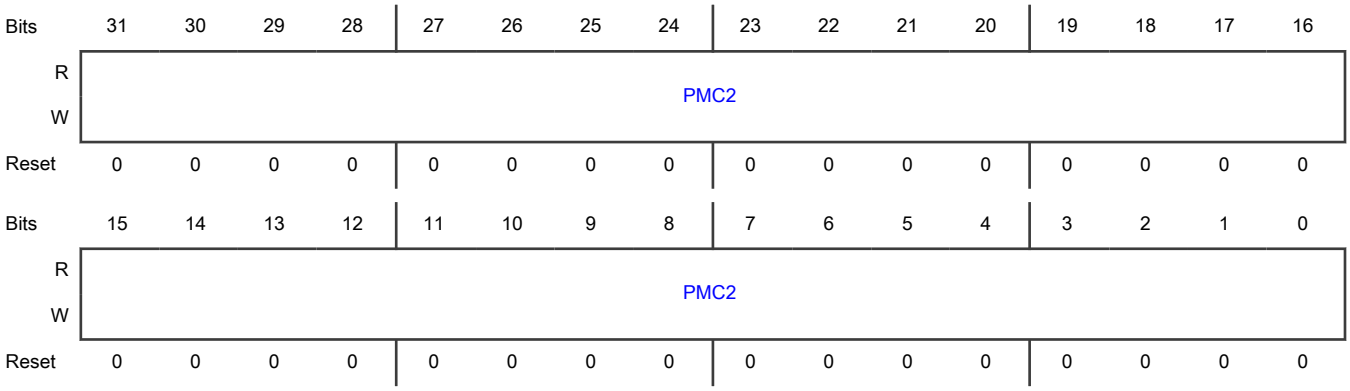
35.6.62 Performance Monitor Counter (PMC2)

Specifies counters that can monitor 64 unique events in addition to the 64 reference events that the full set of PMC*n* registers can count.

Offset

Register	Offset
PMC2	E38h

Diagram



Fields

Field	Description
31-0 PMC2	Event Count Indicates an overflow when the most-significant bit of this counter is 1. Writing 1 to this field may cause an immediate interrupt.

35.6.63 Performance Monitor Counter (PMC3)

Specifies counters that can monitor 64 unique events in addition to the 64 reference events that the full set of PMC*n* registers can count.

Offset

Register	Offset
PMC3	E48h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PMC3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PMC3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PMC3	Event Count Indicates an overflow when the most-significant bit of this counter is 1. Writing 1 to this field may cause an immediate interrupt.

35.6.64 Performance Monitor Counter (PMC4)

Specifies counters that can monitor 64 unique events in addition to the 64 reference events that the full set of PMC*n* registers can count.

Offset

Register	Offset
PMC4	E58h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PMC4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PMC4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PMC4	Event Count Indicates an overflow when the most-significant bit of this counter is 1. Writing 1 to this field may cause an immediate interrupt.

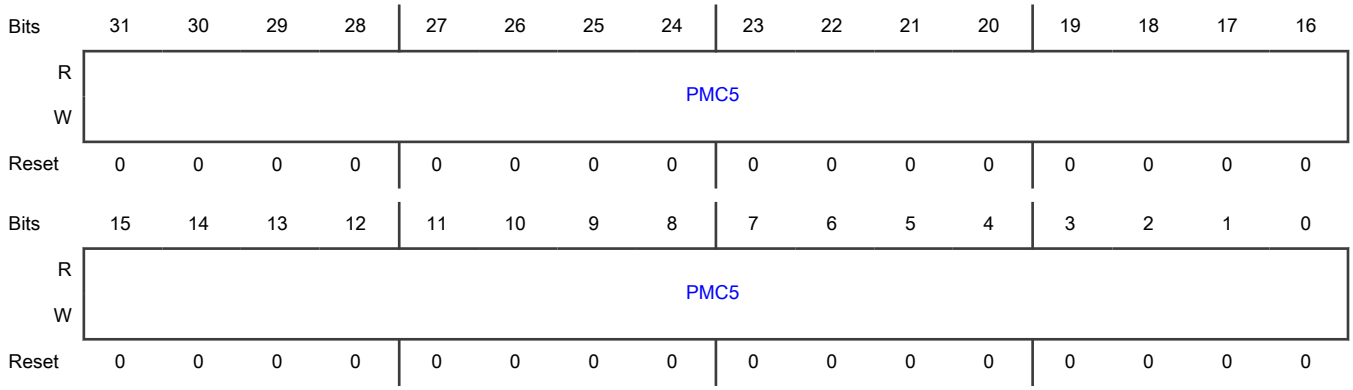
35.6.65 Performance Monitor Counter (PMC5)

Specifies counters that can monitor 64 unique events in addition to the 64 reference events that the full set of PMC*n* registers can count.

Offset

Register	Offset
PMC5	E68h

Diagram



Fields

Field	Description
31-0 PMC5	Event Count Indicates an overflow when the most-significant bit of this counter is 1. Writing 1 to this field may cause an immediate interrupt.

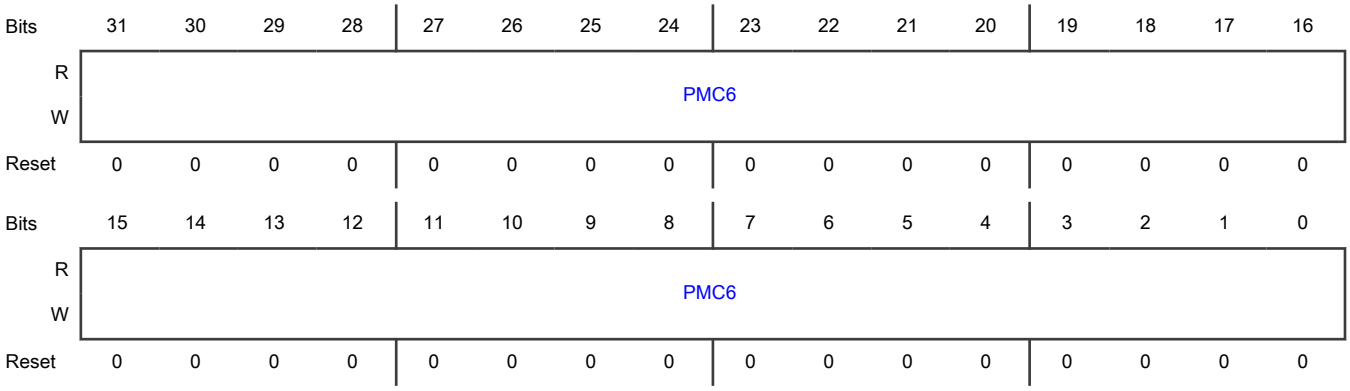
35.6.66 Performance Monitor Counter (PMC6)

Specifies counters that can monitor 64 unique events in addition to the 64 reference events that the full set of PMC*n* registers can count.

Offset

Register	Offset
PMC6	E78h

Diagram



Fields

Field	Description
31-0 PMC6	Event Count Indicates an overflow when the most-significant bit of this counter is 1. Writing 1 to this field may cause an immediate interrupt.

35.6.67 Performance Monitor Counter (PMC7)

Specifies counters that can monitor 64 unique events in addition to the 64 reference events that the full set of PMC*n* registers can count.

Offset

Register	Offset
PMC7	E88h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PMC7															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PMC7															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PMC7	Event Count Indicates an overflow when the most-significant bit of this counter is 1. Writing 1 to this field may cause an immediate interrupt.

35.6.68 Performance Monitor Counter (PMC8)

Specifies counters that can monitor 64 unique events in addition to the 64 reference events that the full set of PMC*n* registers can count.

Offset

Register	Offset
PMC8	E98h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PMC8															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PMC8															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PMC8	Event Count Indicates an overflow when the most-significant bit of this counter is 1. Writing 1 to this field may cause an immediate interrupt.

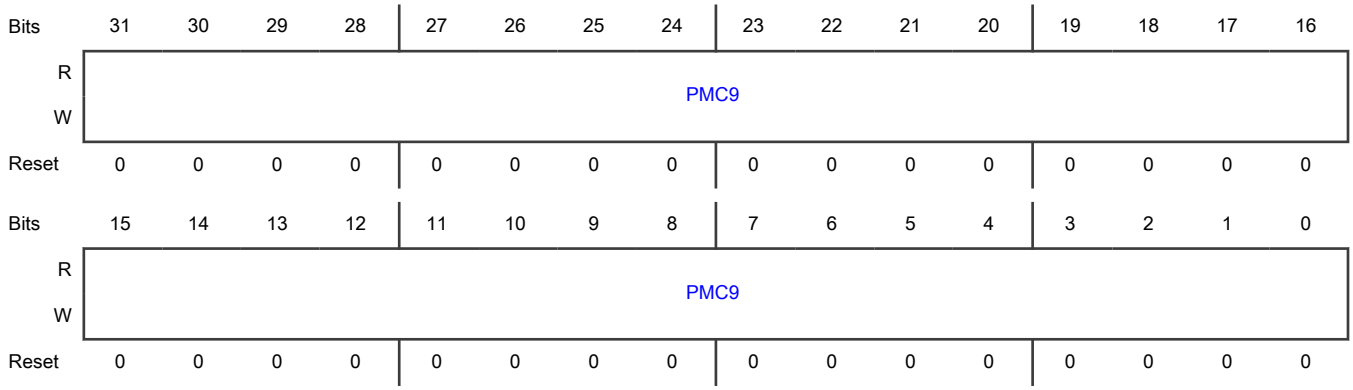
35.6.69 Performance Monitor Counter (PMC9)

Specifies counters that can monitor 64 unique events in addition to the 64 reference events that the full set of PMC*n* registers can count.

Offset

Register	Offset
PMC9	EA8h

Diagram



Fields

Field	Description
31-0 PMC9	Event Count Indicates an overflow when the most-significant bit of this counter is 1. Writing 1 to this field may cause an immediate interrupt.

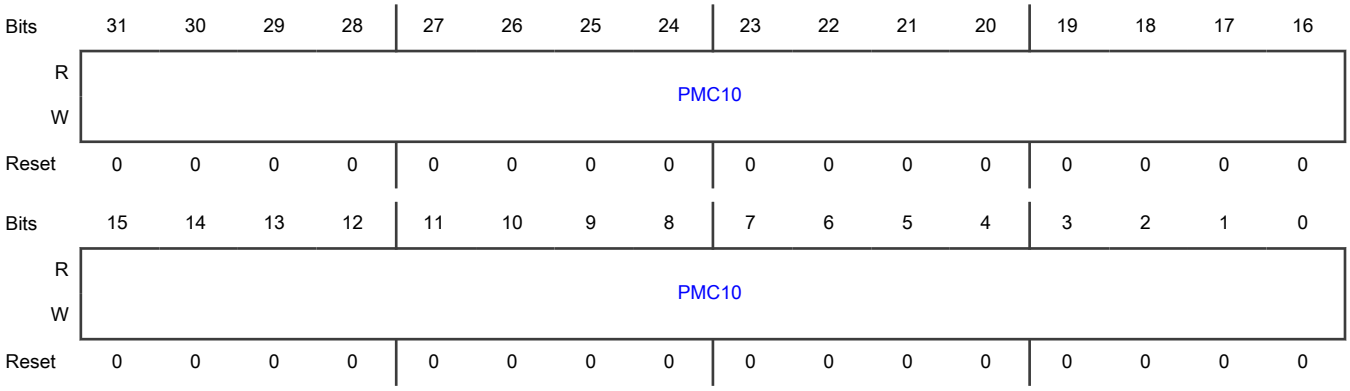
35.6.70 Performance Monitor Counter (PMC10)

Specifies counters that can monitor 64 unique events in addition to the 64 reference events that the full set of PMC*n* registers can count.

Offset

Register	Offset
PMC10	EB8h

Diagram



Fields

Field	Description
31-0 PMC10	Event Count Indicates an overflow when the most-significant bit of this counter is 1. Writing 1 to this field may cause an immediate interrupt.

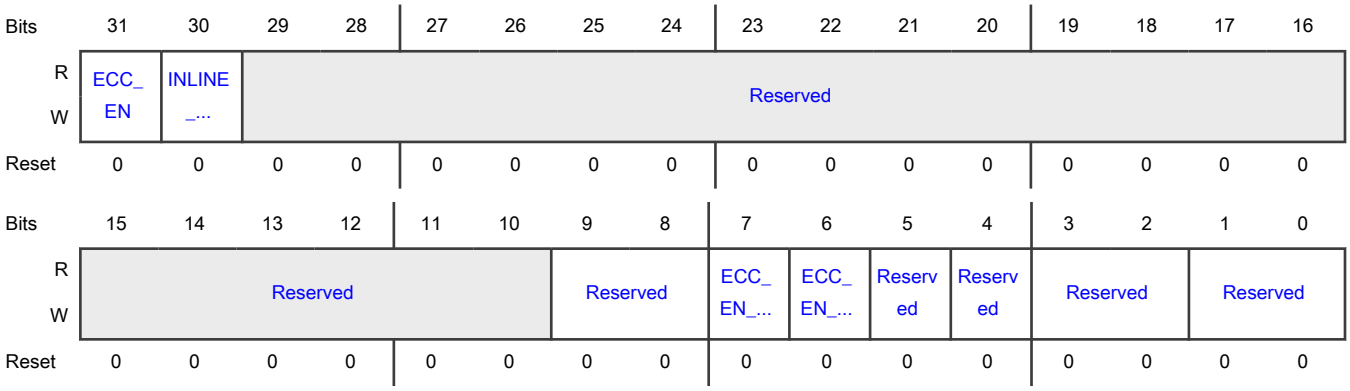
35.6.71 Error Enable (ERR_EN)

Enables controls for many errors that can be reported and injected.

Offset

Register	Offset
ERR_EN	1000h

Diagram



Fields

Field	Description
31 ECC_EN	ECC Enable Enabled global ECC protection for the DDR SDRAM space. This field is the global DDR SDRAM ECC enable. If this field is 0, no ECC errors are reported and no ECC interrupts are generated unless one or more ECC regions are enabled, refer to ECC Region Configuration registers ECC_REG_0 to ECC_REG_7. If this field is 1, the entire DDR SDRAM space is ECC protected. In case protection is enabled only for certain regions, you must program these registers to enable ECC: • ECC Region 0 Configuration (ECC_REG_0) • ECC Region 1 Configuration (ECC_REG_1) • ECC Region 2 Configuration (ECC_REG_2) • ECC Region 3 Configuration (ECC_REG_3) • ECC Region 4 Configuration (ECC_REG_4) • ECC Region 5 Configuration (ECC_REG_5) • ECC Region 6 Configuration (ECC_REG_6) • ECC Region 7 Configuration (ECC_REG_7) Uncorrectable read errors may cause an interrupt. 0 - Disables 1 - Enables
30 INLINE_ECC_EN	Inline ECC Enable Enables inline ECC. You can write 1 to this field only if all of the following conditions are met: • The memory has at least 13 row bits, 10 column bits, and 3 bank bits. • You have enabled data masks (DDR_SDRAM_CFG_3[DM_CFG]) • Either of these conditions is true: — ERR_EN[ECC_EN] = 1. — Certain ECC regions are enabled. When using two ranks of memory, each rank must use the same amount of memory with the same number of row, column, and bank address fields if inline ECC is enabled. When using a 17x10x3 configuration for the DRAM, after DRAM has valid data and ECC, you must read the initial 32k of DRAM, starting with the first defined address. This will load the internal ECC cache with valid data. This is only required for this maximum row/col/bank configuration. **NOTE** Use of inline ECC will reduce overall memory bandwidth. 0 - Disables 1 - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Description
29-10 —	Reserved
9-8 —	Reserved
7 ECC_EN_RAM_1	ECC Enable For On-Chip RAM 1 Enables ECC for on-chip RAM 1. 0 - Disables 1 - Enables
6 ECC_EN_RAM_2	ECC Enable For On-Chip RAM 2 Enables ECC for on-chip RAM 2. 0 - Disables 1 - Enables
5 —	Reserved
4 —	Reserved
3-2 —	Reserved
1-0 —	Reserved

35.6.72 Memory Data Path Error Injection Mask High (DATA_ERR_INJECT_HI)

Tests ECC by forcing errors on the high word of the data path.

Offset

Register	Offset
DATA_ERR_INJECT_HI	1100h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EIMH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EIMH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 EIMH	Error Injection Mask High Data Path Tests ECC by forcing errors on the high word of the data path. Writing 1 to a bit in this field causes the corresponding data-path bit to be inverted on memory bus writes.

35.6.73 Memory Data Path Error Injection Mask Low (DATA_ERR_INJECT_LO)

Tests ECC by forcing errors on the low word of the data path.

Offset

Register	Offset
DATA_ERR_INJECT_LO	1104h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EIML															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EIML															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 EIML	Error Injection Mask Low Data Bit Used to test ECC by forcing errors on the low word of the data bit. Writing 1 to a bit in this field causes the corresponding data-path bit to be inverted on memory bus writes.

35.6.74 Memory Data Path Error Injection Mask ECC (ERR_INJECT)

Enables you to:

- Set the ECC mask.
- Enable errors to write to the ECC memory.
- Enable the ECC byte to mirror the most-significant data byte.
- Inject a single address-parity error.

Offset

Register	Offset
ERR_INJECT	1108h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR_TEN	Reserved					Reserved		FRC2B	ECC_INJ_SRC			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_ECC_INJ				Reserved		Reserved	EIEN	EEIM							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 ADDR_TEN	Address Trigger Enable Enables address triggering for error injection. Address Error Inject (ADDR_ERR_INJ) defines the address to be used as a trigger for the error injection. 0 - Disables 1 - Enables

Table continues on the next page...

Table continued from the previous page...

Field	Description
30-26 —	Reserved
25-24 —	Reserved
23 FRC2B	Force 2-Bit Error Forces an SBE or 2-bit error to DRAM or SRAMs when injecting ECC errors. If this field is 0, only SBEs are forced to DRAM or SRAM. If this field is 1, a 2-bit error is forced when ECC error injection is enabled. 0 - SBE 1 - 2-bit error
22-21 ECC_INJ_SRC	ECC Injection Source Indicates the ECC source used for error injection when EIEN is set. 00 - DDR SDRAM ECC using programmed data and ECC injection masks 01 - On-chip RAM ECC 1 10 - On-chip RAM ECC 2 11 - DDR SDRAM ECC. This setting forces a 1 or 2-bit ECC syndrome error based on the value of FRC2B
20-16 —	Reserved
15-12 NUM_ECC_INJ	Number Of ECC Errors Injected Indicates the number of ECC errors injected when error injection is enabled. After the field reaches its threshold, you must disable error injection and then re-enable it before more ECC errors are injected. 0000 - ECC errors are injected until the error injection is disabled 0001 - 4 0010 - 8 0011 - 16 0100 - 20 0101 - 24 0110 - 28 0111 - 32 1000 - 36 1001 - 40

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1010 - 44 1011 - 48 1100 - 52 1101 - 56 1110 - 60 1111 - 64
11-10 —	Reserved
9 —	Reserved
8 EIEN	Error Injection Enable Enables error injection. If you write 1 to this field, error injection applies to the data mask bits, ECC mask bits, and ECC mirror bits. You must not enable error injection until you enable DDRC via DDR_SDRAM_CFG[MEM_EN]. 0 - Disables 1 - Enables
7-0 EEIM	ECC Error Injection Mask Sets a mask bit that inverts the corresponding ECC bit on memory bus writes.

35.6.75 Address Error Inject (ADDR_ERR_INJ)

Works with ERR_INJECT[ADDR_TEN] to define the address that you can use as a trigger when injecting ECC errors.

Offset

Register	Offset
ADDR_ERR_INJ	110Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ADDR	Address If ERR_INJECT[ADDR_TEN] = 1, provides the address that serves as a trigger when injecting ECC errors.

35.6.76 Memory Extended Data Path Read Capture High (CAPTURE_EXT_DATA_HI)

Stores the high word of the read extended data path during error capture for SRAM ECC errors.

Offset

Register	Offset
CAPTURE_EXT_DATA_HI	1118h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECEHD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECEHD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ECEHD	Error Capture Extended High Data Path Captures the high word of the extended data path when errors are detected.

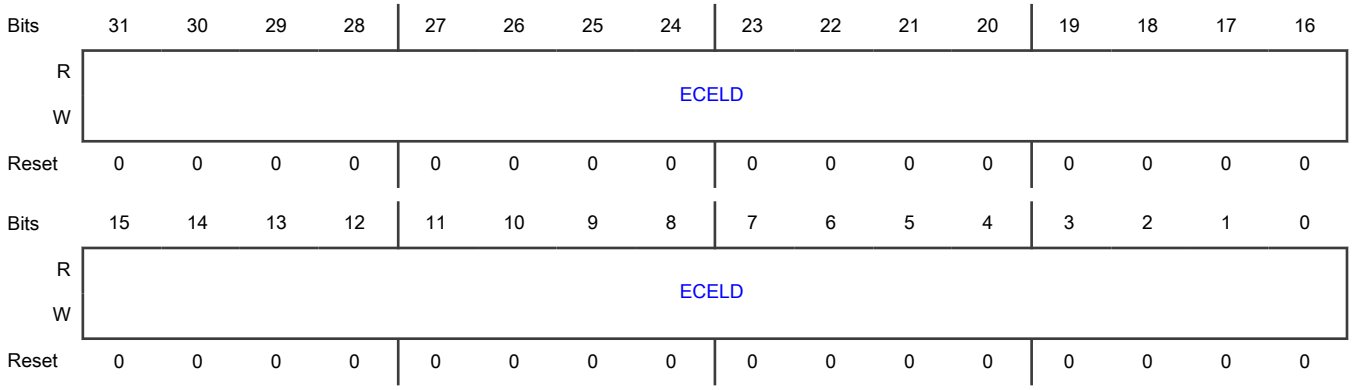
35.6.77 Memory Extended Data Path Read Capture Low (CAPTURE_EXT_DATA_LO)

Stores the low word of the extended read data path during error capture.

Offset

Register	Offset
CAPTURE_EXT_DATA_LO	111Ch

Diagram



Fields

Field	Description
31-0 ECELD	Error Capture Extended Low Data Path Captures the low word of the extended data path when errors are detected.

35.6.78 Memory Data Path Read Capture High (CAPTURE_DATA_HI)

Stores the high word of the read data path during error capture.

Offset

Register	Offset
CAPTURE_DATA_HI	1120h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECHD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECHD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ECHD	Error Capture High Data Path Captures the high word of the data path when errors are detected.

35.6.79 Memory Data Path Read Capture Low (CAPTURE_DATA_LO)

Stores the low word of the read data path during error capture.

Offset

Register	Offset
CAPTURE_DATA_LO	1124h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ECLD	Error Capture Low Data Path Captures the low word of the data path when errors are detected.

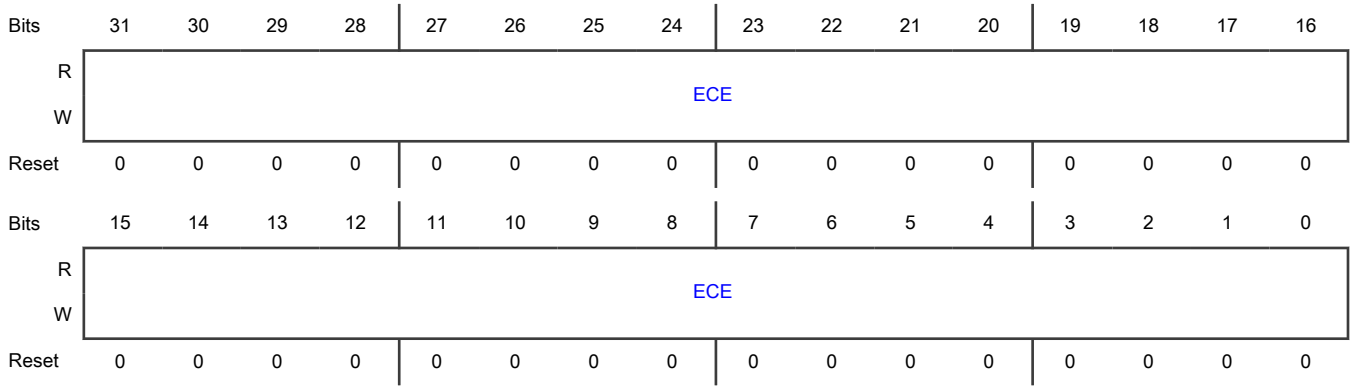
35.6.80 Memory Data Path Read Capture ECC (CAPTURE_ECC)

Stores the ECC syndrome bits that are on the data bus when an error is detected.

Offset

Register	Offset
CAPTURE_ECC	1128h

Diagram



Fields

Field	Description
31-0 ECE	Error Capture ECC Captures the ECC bits on the data path whenever errors are detected. ECC is always generated or checked on 64-bit data chunks. The bits of this field mean the following:

ECE bit numbers	Meaning
0–7	Show the 8-bit ECC for the fourth 64 bits of a 32-byte granule
8–15	Show the 8-bit ECC for the third 64 bits of a 32-byte granule
16–23	Show the 8-bit ECC for the second 64 bits of a 32-byte granule
24–31	Show the 8-bit ECC for the first 64 bits of a 32-byte granule

35.6.81 Memory Error Detect (ERR_DETECT)

Stores the detection fields for multiple-memory errors, single- and multiple-bit ECC errors, and memory-select errors.

You can determine the type of memory error by examining the contents of this register. If an error is disabled using Memory Error Disable (ERR_DISABLE), the corresponding error is never detected or captured in Memory Error Detect (ERR_DETECT).

NOTE

SSBE2, SMBE2 bits are not applicable for LPFLASH.

Offset

Register	Offset
ERR_DETECT	1140h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MME	Reserved	Reserved		Reserved	Reserved	SSBE1	SSBE2	SMBE1	SMBE2	Reserved	UPDTMTE	Reserved	Reserved	Reserved	PHYE
W	W1C	W1C			W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C			W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		Reserved	Reserved	Reserved	Reserved	Reserved		REFRATEE	Reserved	Reserved	Reserved	MBE	SBE	Reserved	MSE
W									W1C				W1C	W1C		W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 MME	Multiple Memory Errors Indicates whether multiple memory errors of the same type are detected. Only DRAM ECC errors and memory select errors cause this bit to get set. 0 - Not detected 1 - Detected
30 —	Reserved
29-28 —	Reserved
27 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
26 —	Reserved
25 SSBE1	SRAM SBE 1 Indicates whether an SRAM 1 SBE occurred. The receiving multiple SRAM errors do not cause ERR_DETECT[MME] to become 1. 0 - Did not occur 1 - Occurred
24 SSBE2	SRAM SBE 2 Indicates whether an SRAM 2 SBE occurred. The receiving multiple SRAM errors do not cause ERR_DETECT[MME] to become 1. 0 - Did not occur 1 - Occurred
23 SMBE1	SRAM Multi-Bit Error 1 Indicates whether an SRAM 1 multi-bit error occurred. The receiving multiple SRAM errors do not cause ERR_DETECT[MME] to become 1. 0 - Did not occur 1 - Occurred
22 SMBE2	SRAM Multi-Bit Error 2 Indicates whether an SRAM 2 multi-bit error occurred. The receiving multiple SRAM errors do not cause ERR_DETECT[MME] to become 1. 0 - Did not occur 1 - Occurred
21 —	Reserved
20 UPDTMTE	Update Timeout Error Indicates whether a training update timeout is detected. The receiving multiple write configuration errors do not cause ERR_DETECT[MME] to become 1. 0 - Not detected 1 - Detected
19 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
18 —	Reserved
17 —	Reserved
16 PHYE	PHY error. Cleared by software writing a 1. Note that receiving multiple PHY errors will not cause the MME bit to get set. 0 - A DDR PHY error has not been detected. 1 - An error has been detected by the DDR PHY.
15-14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9-8 —	Reserved
7 REFRATEE	Refresh rate error. Cleared by software writing a 1. 0 - A refresh rate error has not been detected. 1 - A refresh rate error has been detected.
6 —	Reserved
5 —	Reserved
4	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
3 MBE	Multiple-Bit Error Indicates whether DDRC detected a multiple-bit error. 0 - Not detected 1 - Detected
2 SBE	Single-Bit ECC Errors Indicates whether the number of detected single-bit ECC errors crossed the threshold defined in ERR_SBE[SBET]. 0 - Did not cross 1 - Crossed
1 —	Reserved
0 MSE	Memory-Select Error Indicates whether a memory-select error is detected. This error occurs if a transaction arriving at DDRC does not match any valid rank windows. 0 - Not detected 1 - Detected

35.6.82 Memory Error Disable (ERR_DISABLE)

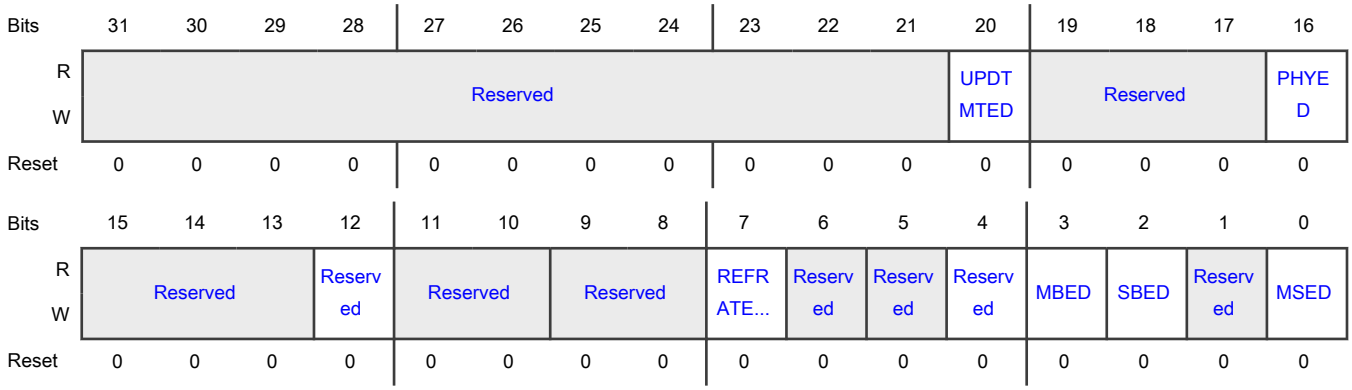
Allows selective disabling of DDRC's error-detection circuitry.

Disabled errors are not detected or reported.

Offset

Register	Offset
ERR_DISABLE	1144h

Diagram



Fields

Field	Description
31-21 —	Reserved
20 UPDTMTED	Update Timeout Error Disable Enables training timeout error checking. 0 - Enables 1 - Disables
19-17 —	Reserved
16 PHYED	PHY Error Disable Enables PHY error checking. 0 - Enables 1 - Disables
15-13 —	Reserved
12 —	Reserved
11-10 —	Reserved
9-8 —	Reserved
7	Refresh Rate Error Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
REFRATEED	Enables refresh rate error checking. 0 - Enables 1 - Disables
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 MBED	Multiple-Bit ECC Error Disable Indicates if multiple-bit ECC errors are detected or reported. Multiple-bit ECC errors are detected if ERR_EN[ECC_EN] = 1. They are reported if ERR_INT_EN[MBEE] = 1. See Error management for more information. To ensure that an interrupt is generated: • This field must be 0. • ERR_INT_EN[MBEE] and ERR_EN[ECC_EN] must be 1. 0 - Detected 1 - Not detected or reported
2 SBED	Single-Bit ECC Error Disable Enables single-bit ECC errors. 0 - Enables 1 - Disables
1 —	Reserved
0 MSED	Memory-Select Error Disable Enables memory-select errors. 0 - Enables 1 - Disables

35.6.83 Memory Error Interrupt Enable (ERR_INT_EN)

Enables ECC interrupts or memory-select error interrupts.

When an enabled interrupt condition occurs, DDRC generates an interrupt.

Offset

Register	Offset
ERR_INT_EN	1148h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											UPDTMTEE	Reserved			PHYEE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			Reserved	Reserved		Reserved		REFRATE...	Reserved	Reserved	SSBE12E	MBEE	SBEE	Reserved	MSEE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-21 —	Reserved
20 UPDTMTEE	Update Timeout Interrupt Enable Specifies whether a training update timeout causes an interrupt. 0 - No 1 - Yes
19-17 —	Reserved
16 PHYEE	PHY error interrupt enable. 0 - PHY errors cannot generate interrupts. 1 - PHY errors generate interrupts.
15-13 —	Reserved
12 —	Reserved
11-10 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
9-8 —	Reserved
7 REFRATEEE	Refresh Rate Interrupt Enable Specifies whether a reported refresh rate that is outside the defined temperature range causes an interrupt. You should only set this bit if also setting DDR_SDRAM_CFG_3[DYN_REF_RATE_EN] 0 - No 1 - Yes
6 —	Reserved
5 —	Reserved
4 SSBE12E	SRAM Single-Bit Error Interrupt Enable Indicates whether SRAM SBE1 or SRAM SBE2 errors generate interrupts. 0 - No 1 - Yes
3 MBEE	Multiple-Bit ECC Error Interrupt Enable Indicates whether multiple-bit ECC errors generate interrupts. Uncorrectable read errors may lead to interrupts. To ensure that an interrupt is generated, these conditions must be true: • ERR_DISABLE[MBED] must be 0. • ERR_INT_EN[MBEE] and ERR_EN[ECC_EN] must be 1 if one or more ECC regions are enabled. See Error management for more information. 0 - No 1 - Yes
2 SBEE	Single-Bit ECC Error Interrupt Enable Indicates whether single-bit ECC errors generate interrupts. 0 - No 1 - Yes
1 —	Reserved
0	Memory-Select Error Interrupt Enable Indicates whether memory-select errors generate interrupts.

Table continues on the next page...

Table continued from the previous page...

Field	Description
MSEE	0 - No 1 - Yes

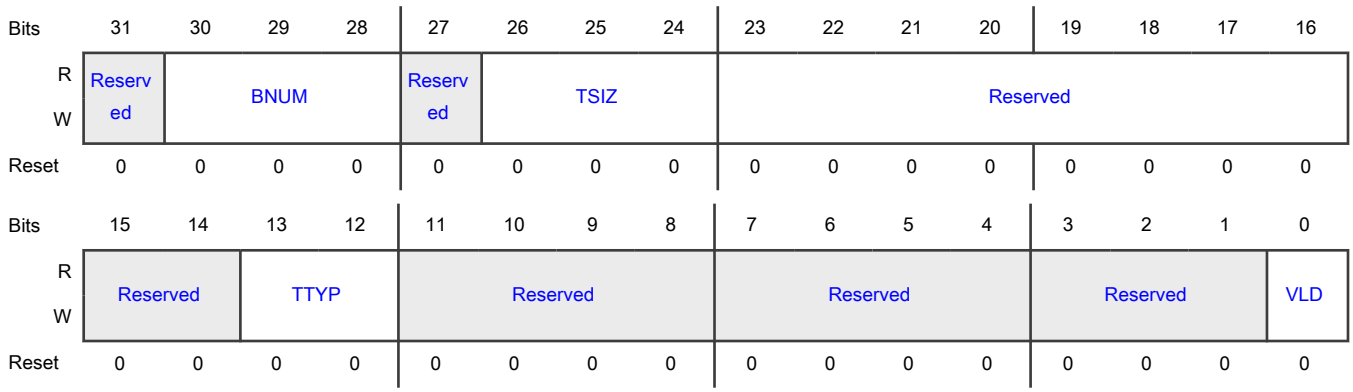
35.6.84 Memory Error Attributes Capture (CAPTURE_ATTRIBUTES)

Defines attributes of the originating read and write transactions that cause errors related to type, size, source, and others.

Offset

Register	Offset
CAPTURE_ATTRIBUTES	114Ch

Diagram



Fields

Field	Description
31 —	Reserved
30-28 BNUM	Data Beat Number Captures the doubleword number for the detected error. This field is relevant only for ECC errors.
27 —	Reserved
26-24	Error Transaction Size Captures the transaction size in double words.

Table continues on the next page...

Table continued from the previous page...

Field	Description
TSIZ	000 - 8 001 - 1 010 - 2 011 - 3 100 - 4 101 - 5 110 - 6 111 - 7
23-16 —	Reserved
15-14 —	Reserved
13-12 TTYP	Error Transaction Type 00 - Reserved 01 - Write 10 - Read 11 - Reserved
11-8 —	Reserved
7-4 —	Reserved
3-1 —	Reserved
0 VLD	Valid Becomes 1 as soon as valid information is captured in the error-capture registers. This other attribute fields in this register are not valid for failures when DDR_MTCR[MT_EN] = 1.

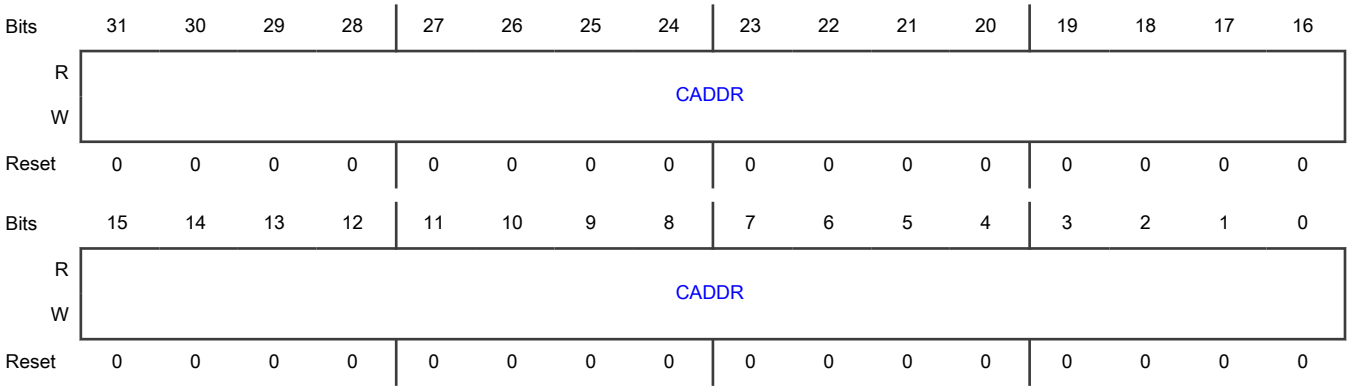
35.6.85 Memory Error Address Capture (CAPTURE_ADDRESS)

Holds the 32 least-significant bits of a transaction when a DDR SDRAM ECC error is detected.

Offset

Register	Offset
CAPTURE_ADDRESS	1150h

Diagram



Fields

Field	Description
31-0 CADDR	Captured Address Captures the 32 least-significant bits of the transaction address when an error is detected.

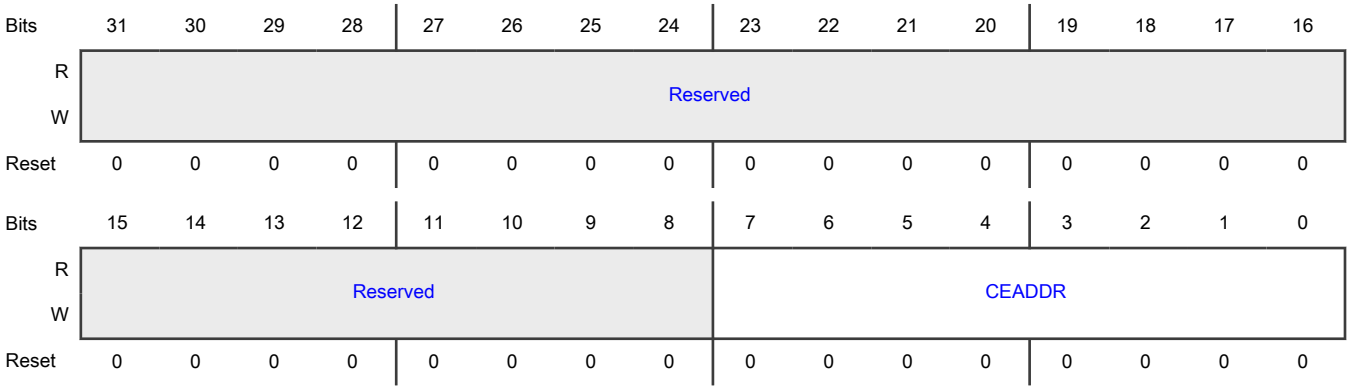
35.6.86 Memory Error Extended Address Capture (CAPTURE_EXT_ADDRESS)

Holds the 4 most-significant transaction bits when an error is detected.

Offset

Register	Offset
CAPTURE_EXT_ADDRESS	1154h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 CEADDR	Captured Extended Address Captures the 8 most-significant bits of the transaction address when an error is detected.

35.6.87 Single-Bit ECC Memory Error Management (ERR_SBE)

Stores:

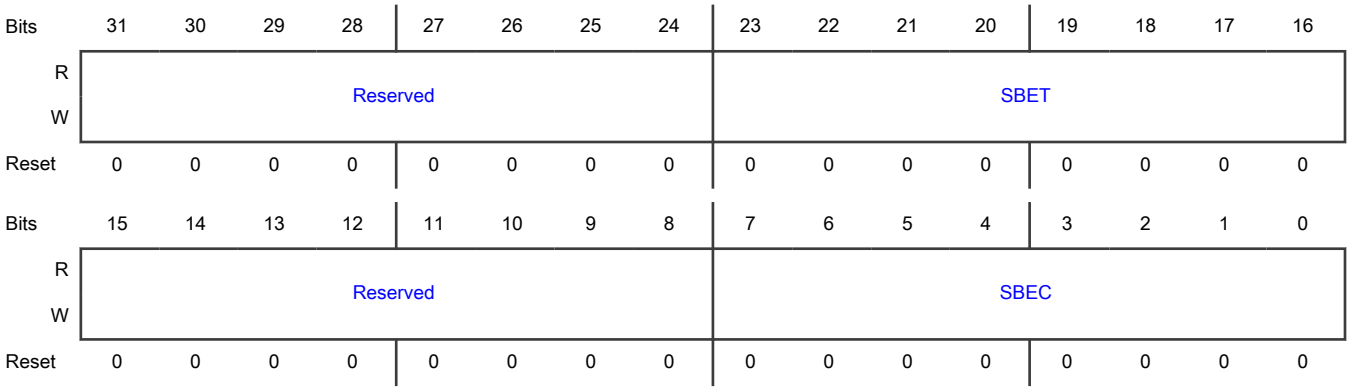
- The threshold value for reporting SBEs.
- The number of SBEs counted since the last error report.

ERR_SBE[SBEC] resets when it reaches the threshold. If necessary, you must reset the counter after it manages the error.

Offset

Register	Offset
ERR_SBE	1158h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 SBET	SBE Threshold Establishes the number of SBEs that must be detected before an error condition is reported.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-8 —	Reserved
7-0 SBEC	SBE Counter Indicates the number of SBEs detected and corrected since the last error report. If you enable SBE reporting, an error is reported and an interrupt is generated when the value of this field equals that of ERR_SBE[SBET]. This field becomes 0 when it reaches its threshold value.

35.6.88 ECC Region 0 Configuration (ECC_REG_0)

Provides bits 31:20 for the starting and ending addresses of a region to be enabled for ECC. The region must be defined only if:

- You enable specific ECC regions.
- ERR_EN[ECC_EN] = 0.

NOTE

ECC region registers should not be programmed without first programming this register. If per region ECC enablement is used for any region, then you must have region 0 enabled.

Offset

Register	Offset
ECC_REG_0	1240h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REG_0_EN	Reserved			REG_0_SA											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				REG_0_EA											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 REG_0_EN	Region 0 Enable Indicates whether DDRC applies ECC protection to addresses from region 0 or does not use region 0 for ECC enablement. Note that if per region ECC enablement is used for any region, then you must have region 0 enabled by setting REG_0_EN. This field enables region 0 to be used for ECC. 0 - Does not use region 0 for ECC enablement 1 - Protects addresses from region 0 with ECC
30-28 —	Reserved
27-16 REG_0_SA	Region 0 Start Address Provides the value to compare with the upper 12 bits of a 32-bit address when determining if the address is within region 0.
15-12 —	Reserved
11-0 REG_0_EA	Region 0 End Address Provides the value to compare with the upper 12 bits of a 32-bit address when determining if the address is within region 0.

35.6.89 ECC Region 1 Configuration (ECC_REG_1)

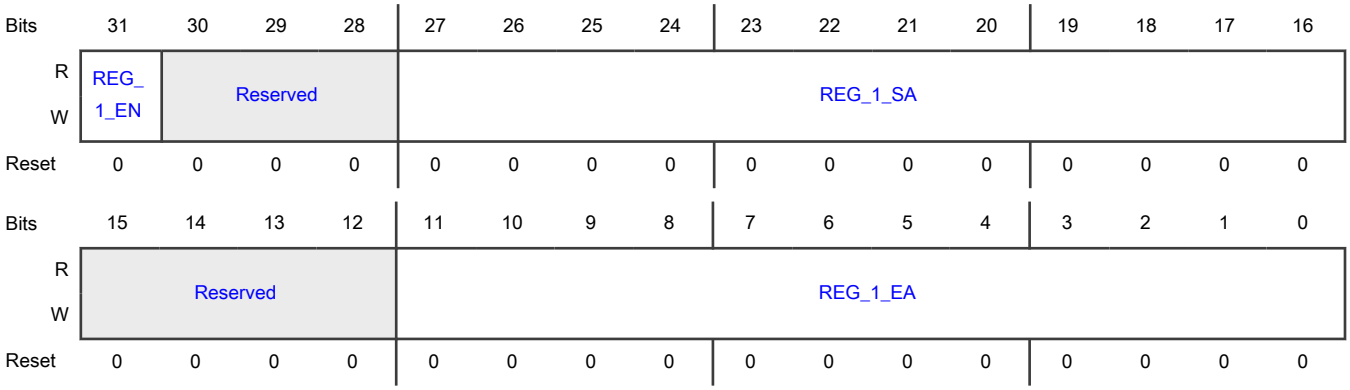
Provides bits 31:20 for the starting and ending addresses of a region to be enabled for ECC. The region must be used only if:

- You enable specific ECC regions.
- ERR_EN[ECC_EN] = 0.

Offset

Register	Offset
ECC_REG_1	1244h

Diagram



Fields

Field	Description
31 REG_1_EN	Region 1 Enable Indicates whether DDRC applies ECC protection to addresses from region 1 or does not use region 1 for ECC enablement. **NOTE** If per region ECC enablement is used for any region, then you must have region 0 enabled by setting ECC_REG_0[REG_0_EN]. This field enables region 1 to be used for ECC. 0 - Does not use region 1 for ECC enablement 1 - Protects addresses from region 1 with ECC
30-28 —	Reserved
27-16 REG_1_SA	Region 1 Start Address Provides the value to compare with the upper 12 bits of a 32-bit address when determining if the address is within region 1.
15-12 —	Reserved
11-0 REG_1_EA	Region 1 End Address Provides the value to compare with the upper 12 bits of a 32-bit address when determining if the address is within region 1.

35.6.90 ECC Region 2 Configuration (ECC_REG_2)

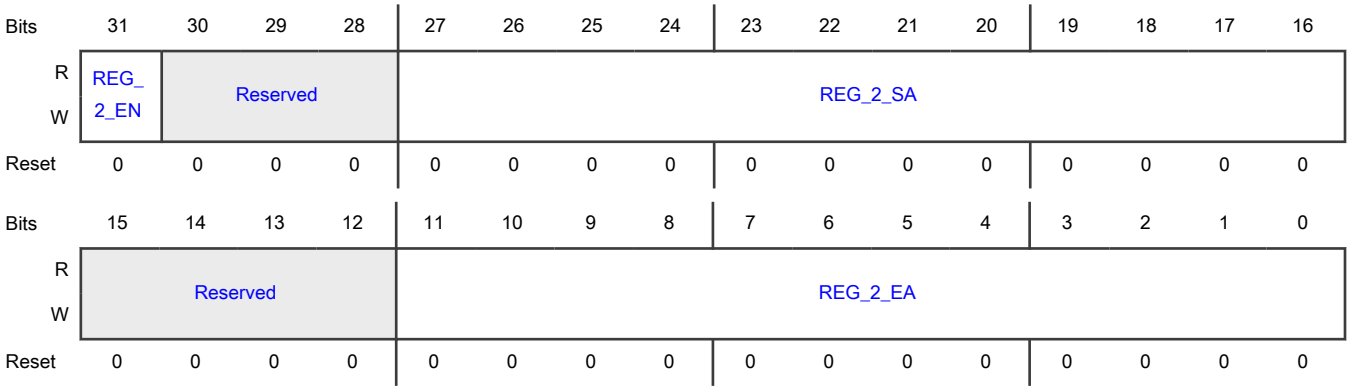
Provides bits 31:20 for the starting and ending addresses of a region to be enabled for ECC. The region must be used only if:

- You enable specific ECC regions.
- ERR_EN[ECC_EN] = 0.

Offset

Register	Offset
ECC_REG_2	1248h

Diagram



Fields

Field	Description
31 REG_2_EN	Region 2 Enable Indicates whether DDRC applies ECC protection to addresses from region 2 or does not use region 2 for ECC enablement. This field enables region 2 to be used for ECC. **NOTE** If per region ECC enablement is used for any region, then you must have region 0 enabled by setting ECC_REG_0[REG_0_EN]. 0 - Does not use region 2 for ECC enablement 1 - Protects addresses from region 2 with ECC
30-28 —	Reserved
27-16 REG_2_SA	Region 2 Start Address Provides the value to compare with the upper 12 bits of a 32-bit address when determining if the address is within region 2.
15-12 —	Reserved
11-0 REG_2_EA	Region 2 End Address Provides the value to compare with the upper 12 bits of a 32-bit address when determining if the address is within region 2.

35.6.91 ECC Region 3 Configuration (ECC_REG_3)

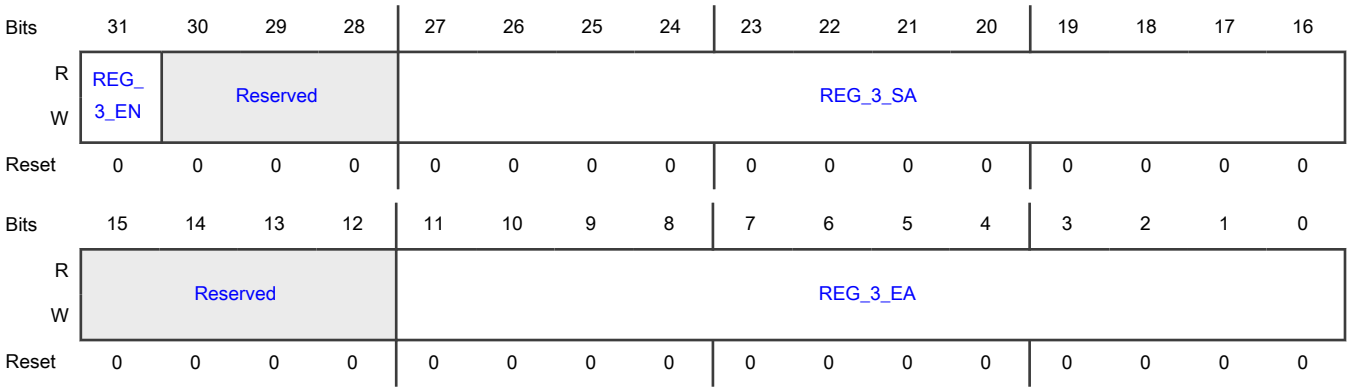
Provides bits 31:20 for the starting and ending addresses of a region to be enabled for ECC. The region must be used only if:

- You enable specific ECC regions.
- ERR_EN[ECC_EN] = 0.

Offset

Register	Offset
ECC_REG_3	124Ch

Diagram



Fields

Field	Description
31 REG_3_EN	Region 3 Enable Indicates whether DDRC applies ECC protection to addresses from region 3 or does not use region 3 for ECC enablement. This field enables region 3 to be used for ECC. **NOTE** If per region ECC enablement is used for any region, then you must have region 0 enabled by setting ECC_REG_0[REG_0_EN]. 0 - Does not use region 3 for ECC enablement 1 - Protects addresses from region 3 with ECC
30-28 —	Reserved
27-16 REG_3_SA	Region 3 Start Address Provides the value to compare with the upper 12 bits of a 32-bit address when determining if the address is within region 3.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-12 —	Reserved
11-0 REG_3_EA	Region 3 End Address Provides the value to compare with the upper 12 bits of a 32-bit address when determining if the address is within region 3.

35.6.92 ECC Region 4 Configuration (ECC_REG_4)

Provides bits 31:20 for the starting and ending addresses of a region to be enabled for ECC. The region must be used only if:

- You enable specific ECC regions.
- ERR_EN[ECC_EN] = 0.

Offset

Register	Offset
ECC_REG_4	1250h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REG_4_EN	Reserved			REG_4_SA											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				REG_4_EA											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 REG_4_EN	Region 4 Enable Indicates whether DDRC applies ECC protection to addresses from region 4 or does not use region 4 for ECC enablement. This field enables region 4 to be used for ECC.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE If per region ECC enablement is used for any region, then you must have region 0 enabled by setting ECC_REG_0[REG_0_EN]. 0 - Does not use region 4 for ECC enablement 1 - Protects addresses from region 4 with ECC
30-28 —	Reserved
27-16 REG_4_SA	Region 4 Start Address Provides the value to compare with the upper 12 bits of a 32-bit address when determining if the address is within region 4.
15-12 —	Reserved
11-0 REG_4_EA	Region 4 End Address Provides the value to compare with the upper 12 bits of a 32-bit address when determining if the address is within region 4.

35.6.93 ECC Region 5 Configuration (ECC_REG_5)

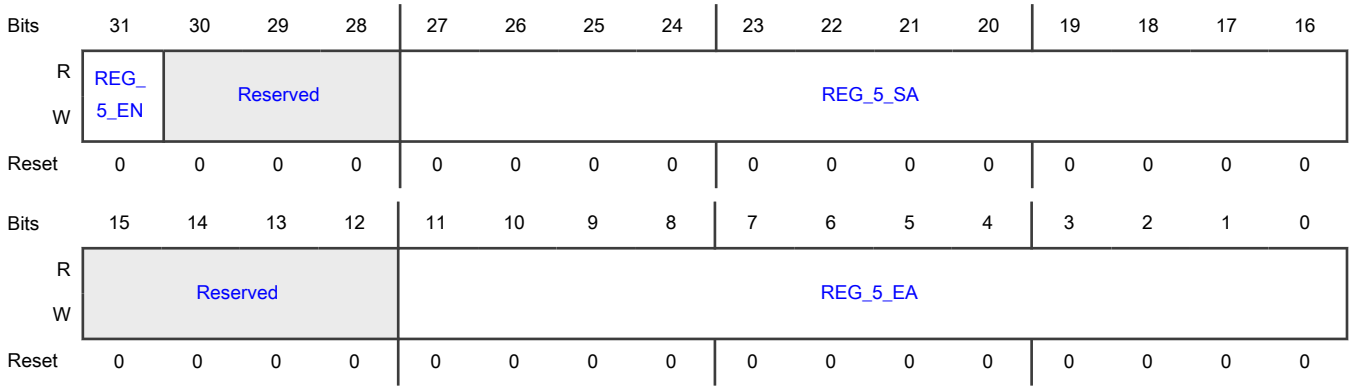
Provides bits 31:20 for the starting and ending addresses of a region to be for ECC. The region must be used only if:

- You enable specific ECC regions.
- ERR_EN[ECC_EN] = 0.

Offset

Register	Offset
ECC_REG_5	1254h

Diagram



Fields

Field	Description
31 REG_5_EN	Region 5 Enable Indicates whether DDRC applies ECC protection to addresses from region 5 or does not use region 5 for ECC enablement. This field enables region 5 to be used for ECC. **NOTE** If per region ECC enablement is used for any region, then you must have region 0 enabled by setting ECC_REG_0[REG_0_EN]. 0 - Does not use region 5 for ECC enablement 1 - Protects addresses from region 5 with ECC
30-28 —	Reserved
27-16 REG_5_SA	Region 5 Start Address Provides the value to be compared with the upper 12 bits of a 32-bit address when determining if the address is within region 5.
15-12 —	Reserved
11-0 REG_5_EA	Region 5 End Address Provides the value to be compared with the upper 12 bits of a 32-bit address when determining if the address is within region 5.

35.6.94 ECC Region 6 Configuration (ECC_REG_6)

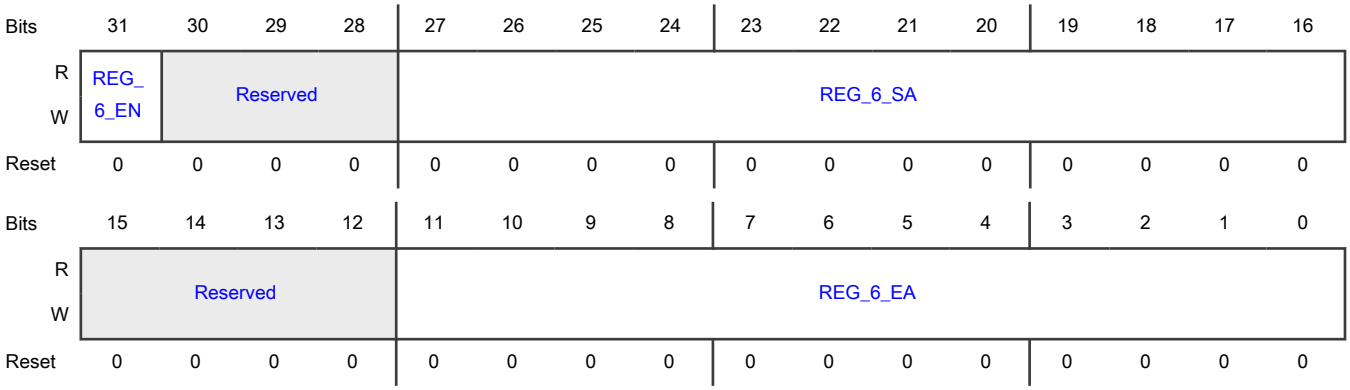
Provides bits 31:20 for the starting and ending addresses of a region to be enabled for ECC. The region must be used only if:

- You enable specific ECC regions.
- ERR_EN[ECC_EN] = 0.

Offset

Register	Offset
ECC_REG_6	1258h

Diagram



Fields

Field	Description
31 REG_6_EN	Region 6 Enable Indicates whether DDRC applies ECC protection to addresses from region 6 or does not use region 6 for ECC enablement. This field enables region 6 to be used for ECC. **NOTE** If per region ECC enablement is used for any region, then you must have region 0 enabled by setting ECC_REG_0[REG_0_EN]. 0 - Does not use region 6 for ECC enablement 1 - Protects addresses from region 6 with ECC
30-28 —	Reserved
27-16 REG_6_SA	Region 6 Start Address Provides the value to compare with the upper 12 bits of a 32-bit address when determining if the address is within region 6.
15-12 —	Reserved
11-0 REG_6_EA	Region 6 End Address Provides the value to compare with the upper 12 bits of a 32-bit address when determining if the address is within region 6.

35.6.95 ECC Region 7 Configuration (ECC_REG_7)

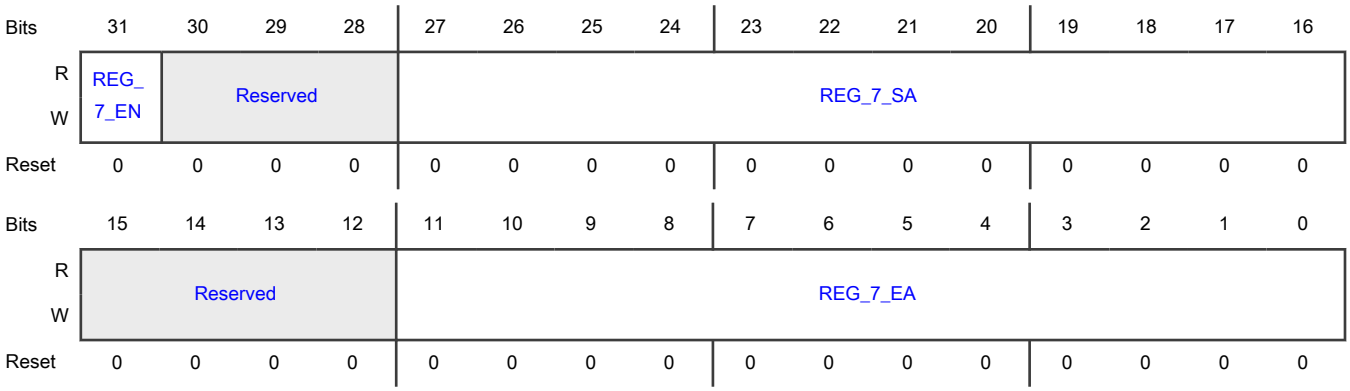
Provides bits 31:20 for the starting and ending addresses of a region to be enabled for ECC. These regions must be used only if:

- You enable specific ECC regions.
- ERR_EN[ECC_EN] = 0.

Offset

Register	Offset
ECC_REG_7	125Ch

Diagram



Fields

Field	Description
31 REG_7_EN	Region 7 Enable Indicates whether DDRC applies ECC protection to addresses from region 7 or does not use region 7 for ECC enablement. This field enables region 7 to be used for ECC. **NOTE** If per region ECC enablement is used for any region, then you must have region 0 enabled by setting ECC_REG_0[REG_0_EN]. 0 - Does not use region 7 for ECC enablement 1 - Protects addresses from region 7 with ECC
30-28 —	Reserved
27-16 REG_7_SA	Region 7 Start Address Provides the value to be compared with the upper 12 bits of a 32-bit address when determining if the address is within region 7.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-12 —	Reserved
11-0 REG_7_EA	Region 7 End Address Provides the value to be compared with the upper 12 bits of a 32-bit address when determining if the address is within region 7.

35.7 Glossary

Bank	A collection of memory-cell rows and columns
CS	Chip select
DBI	Data bus inversion
DLL	Delay-locked loop
ECC	Error checking and correction
FFI	Freedom from interference
MRW	Mode register write
ODT	On-die termination
Page	A synonym for row, usually used in the context of Open-Page mode
PIC	Programmable interrupt controller
PMC	Performance monitor counter
POR	Power-on reset
Posted refresh	In a posted refresh, more than one refresh command can be performed at a time. For example if you set the posted refresh to 5, then at each refresh interval 5 consecutive refreshes are issued, keeping the DDR interface blocked until all 5 refreshes are completed. this allows for a longer period of uninterrupted operation in DDR.
SBE	Single-bit error
SECDED	Single-error correcting, double-error detecting

Chapter 36
Flexible Serial Peripheral Interface (FlexSPI)

36.1 Chip-specific FLEXSPI information

Table 182. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
Full description	FLEXSPI	FLEXSPI
System memory map		System memory map
Signal multiplexing	Port control	Signal multiplexing

The chip includes one instance of FLEXSPI with support for one channel.

NOTE

Low power mode entry-exit for FLEXIO is done without CLK gate and without QCHANNEL handshake.

36.1.1 Module instances

This device has one instance of the FLEXSPI module.

36.1.2 Master ID Assignments

To enable FlexSPI to view different IDs from different masters, follow the steps shown below:

- Set AHBCR[HMSTRIDREMAP] =1 to enable master ID mapping
- Configure HMSTRxCR[MSTRID] using the table below (The “n” should be masked by the mask value. Other bits can be selectively masked to group some masters into one AHB buffer region):

Table 183. Master IDs

Module	4-bit Master ID
Cortex-A55 cores	000n
Reserved	0010 - 0111
Cortex-M33	1000
Reserved	1001
Edgelock	1010
eDMA3_1(AHB)	1011
eDMA4_2(AXI)	1100
Reserved	1101 - 1110
All other bus masters	1111

NOTE

After reset, the FlexSPI sees all masters IDs as zero.

36.1.2.1 AHB Memory Map Address Range

The table below shows the address range calculations for the FlexSPI:

Table 184. Memory Map Address Ranges

Description	FlexSPI1
Address range mapped for AHB read access to FLEXSPI RX FIFO	0x4742_0000 – 0x4742_01FF
Address range mapped for AHB write access to FLEXSPI TX FIFO	0x4743_0000 – 0x4743_03FF
AHB Memory for Serial Flash memory access	0x2800_0000 - 0x2FFF_FFFF
AHB RX Buffer size	2 KByte
AHB TX Buffer size	64 Bytes
FLEXSPI RX FIFO	512 KByte
FLEXSPI TX FIFO	1 KByte

NOTE

- FLEXSPI command should be able to access Flash address space which is greater than 128MB.
- AHB command cannot access Flash address space which is greater than 128MB, it can only access within 128MB.
- But for IP command, AHB/IPS BUS both can access IP RX/TX FIFO(different reg configuration will choose IPS or AHB), which means IPS/AHB bus both can access greater than 128MB flash address space indirectly by IP RX/TX FIFO in IP command mode only.

36.2 Overview

FlexSPI supports one SPI channel and up to two external devices. Each channel supports Single/Dual/Quad/Octal mode data transfer (1/2/4/8 bidirectional data lines).

The FlexSPI configuration depends on the chip configuration. See the system-level section for boot information and pinmux for chip-specific information regarding the modes and number of devices supported.

FlexSPI supports communication with both serial flash memory and serial RAM devices. While many of the descriptions, registers, and fields specifically reference flash memory, almost all information can also be applied to serial RAM. Flash memory is used as an example in the tables and figures in this chapter.

NOTE

Terminology in this chapter has been updated to align with JEDEC standard *Expanded Serial Peripheral Interface (xSPI) for Non Volatile Memory Devices, Version 1.0*.

Table 185. Updated terms

Updated term	Deprecated term
Controller	Master
Target	Slave

36.2.1 Block diagram

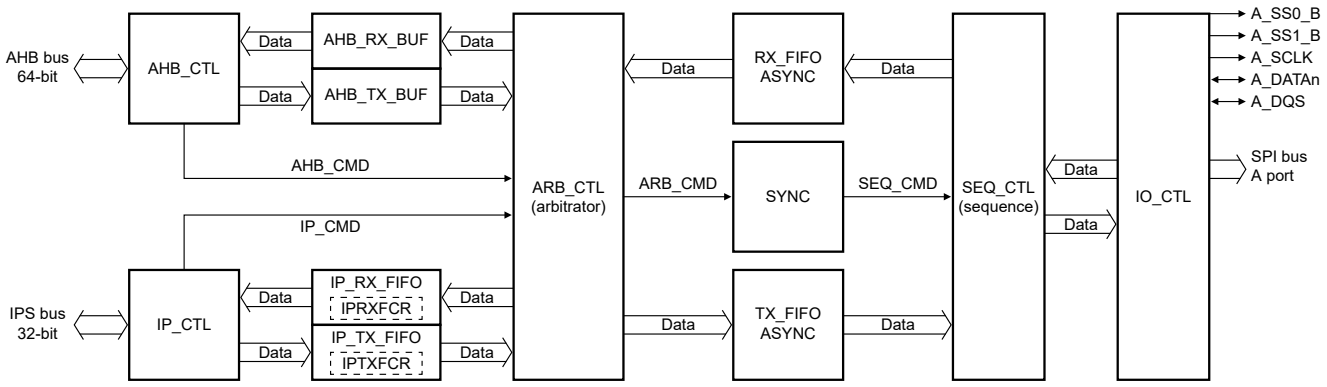


Figure 84. Block diagram

36.2.2 Features

FlexSPI supports:

- Flexible sequence engine (lookup table) to support various vendor devices
  - Serial NOR flash memory and other devices with SPI protocol similar to serial NOR flash memory
  - Serial NAND flash memory
  - HyperBus devices (HyperFlash/HyperRAM)
  - XCCELA devices
  - FPGA devices
- Flash memory access modes
  - Single, Dual, Quad, Octal mode
  - Single Data Transfer Rate (SDR) and Double Data Transfer Rate (DDR) mode
  - Individual mode
- Sampling clock mode
  - Internal dummy read strobe looped back internally
  - Internal dummy read strobe looped back from pad
  - Flash-memory-provided read strobe
- Automatic data learning to select the correct sample clock phase
- Memory mapped read and write access by AHB bus
  - AHB receive buffer implemented to reduce read latency. See the chip-specific section for AHB buffer size.
  - 16 AHB controllers with programmable priority for read access from each
  - 8 flexible and configurable buffers in AHB receive buffer
  - AHB transmit buffer implemented to buffer all write data from one AHB burst. See the chip-specific section for AHB buffer size.

NOTE

All AHB controllers share this AHB transmit buffer. There is no AHB controller number limitation for write access.

- Software-triggered flash memory read and write access by IP bus
  - IP receive FIFO implemented to buffer all read data from external devices. FIFO size: 512 bytes
  - IP transmit FIFO implemented to buffer all write data to external devices. FIFO size: 1024 bytes
  - DMA support to read IP receive FIFO
  - DMA support to fill IP transmit FIFO
  - SCLK stops when IP receive FIFO is full during reading flash memory data
  - SCLK stops when IP transmit FIFO is empty during writing flash memory data

36.3 Glossary for FlexSPI module

Table 186. Glossary

Term	Definition
AHB Command	AHB Command is one or several Serial Flash Memory access command sequences triggered by AHB read/write access to AHB address ranged mapped for Serial Flash Memory.
ASFM_BASE	Base address of AHB address space mapped to Serial Flash Memory.
ARDF_BASE	Base address of AHB address space mapped to IP RX FIFO.
ATDF_BASE	Base address of AHB address space mapped to IP TX FIFO.
Set	Write 1 to register bit to establish logic level one on the bit.
Clear	Write 0 to register bit to establish logic level zero on the bit.
Command Sequence	A command sequence is 4*32 bits sequence code consisting of up to 8 instructions. Command sequence should be programmed in LUT according to Flash command. When a command sequence executed, the chip selection signal becomes asserted. When a command sequence execution finished, the chip selection signal becomes negated.
DDR	Dual data transfer rate for flash access mode Flash receive data on both SCLK rise and fall edge and transmit data on both SCLK rise and fall edge.
Endianness	Byte Ordering scheme.
Field	Two or more register bits grouped together.
Fill	To add entries to a FIFO by software or hardware.
Instruction Code	16 bits code defining the type of command to be executed on FlexSPI interface.
IP Command	IP Command is one or several Serial Flash Memory access command sequences triggered by set register bit IPCMD.TRG.
Negated	A signal that is negated is in its inactive state. An active low signal changes from logic level 0 to logic level 1 when negated, and an active high signal changes from logic level 1 to logic level 0.
SDR	Single data transfer rate for flash access mode

Table continues on the next page...

Table 186. Glossary (continued)

Term	Definition
	Flash receive data on SCLK rise edge and transmit data on SCLK fall edge.
Set	To set a bit or bits means to establish logic level one on the bit or bits.
SFM	Serial Flash Memory
Individual Mode	Access to a single, individual serial flash device at a time.
Memory Command	Serial access Command for reading/programming/configuring. A memory command may consist one or several command sequences, for each command sequence the Chip selection signal will be asserted and deasserted once.
LUT	Look-up table to preserve command sequence. LUT is programmed by software with command sequences which is used to issue memory commands.

36.4 Functional description

36.4.1 Flash connection

There is one FlexSPI interface port (port A) that supports up to two flash devices by providing two chip select outputs.

NOTE

FlexSPI configuration depends on the chip configuration. See the chip-specific FlexSPI information regarding the number of devices supported.

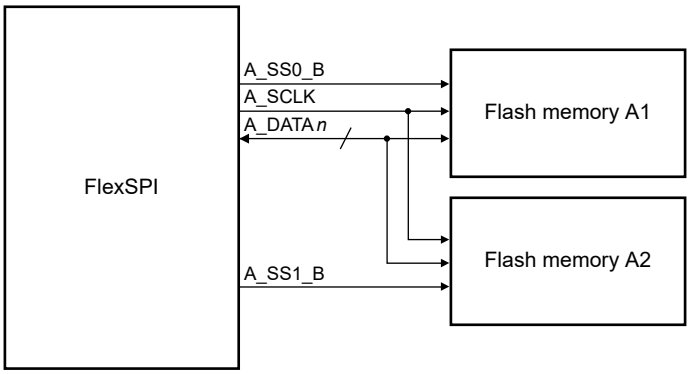


Figure 85. FlexSPI connection diagram with two devices.

NOTE

- Flash A1 and Flash A2 can be two flash chip packages or two flash dies in the same package.
- In individual mode, flash devices cannot be accessed at the same time. But these two devices can be accessed separately.

36.4.2 Flash memory access mode

36.4.2.1 SPI clock mode

FlexSPI only supports SPI clock mode 0: clock polarity (CPOL) = 0 and clock phase (CPHA) = 0. When the SPI bus is idle, SCLK stays at a logic low state.

36.4.2.2 Individual mode

In individual mode, read and write data are received and transmitted on port A or port B.

NOTE

FlexSPI does not support 16-bit read and write for the SPI interface.

36.4.2.3 SDR mode and DDR mode

In SDR mode, flash memory receives data on the rising edge of SCLK and transmits data on the falling edge of SCLK.

In DDR mode, flash memory receives and transmits data on both the rising and falling edges of SCLK.

The instruction (opcode) in the LUT sequence dynamically determines SDR and DDR modes. There is no static configuration register field setting for SDR and DDR modes. See FlexSPI input timing and FlexSPI output timing for details about input and output timing.

36.4.2.4 Single, Dual, Quad, and Octal mode

In Single mode, FlexSPI transmits and receives data on one data pin (DATA0 for transmitting, DATA1 for receiving).

In Dual mode, FlexSPI transmits and receives data on two data pins (DATA0–DATA1 for both transmitting and receiving).

In Quad mode, FlexSPI transmits and receives data on four data pins (DATA0–DATA3 for both transmitting and receiving).

In Octal mode, FlexSPI transmits and receives data on eight data pins (DATA0–DATA7 for both transmitting and receiving).

Instruction (num_pads) in the LUT sequence dynamically determine Single, Dual, Quad, and Octal modes. There is no static configuration register field setting for Single, Dual, Quad, and Octal modes.

36.4.3 Modes of operation

FlexSPI operates with these modes.

Table 187. Operating modes

Mode	Description
Module Disable	This mode is a low-power mode for FlexSPI. In Module Disable mode, the AHB clock and serial clock domains are gated off internally, but the IPS bus clock domain is not gated off. Read and write access to the control and status register are available, but the LUT, IP receive FIFO, and IP transmit FIFO cannot be accessed. Serial memory access is also not available. Write 1 to MCR0[MDIS] to enter this mode. Write 0 to MCR0[MDIS] to exit this mode.
Doze	This mode is a low-power mode for the chip. When the chip requires FlexSPI to enter Doze mode and MCR0[DOZEEN] = 1, FlexSPI enters Doze mode after all transactions are completed (STS0[ARBIDLE] = 1). In Doze mode, the AHB clock and serial clock domains are gated off internally, but the IPS bus clock is not gated off. Read and write access to the control and status register are available, but the LUT, IP receive FIFO, the IP transmit FIFO cannot be accessed. Serial memory access is also not available.

Table continues on the next page...

Table 187. Operating modes (continued)

Mode	Description
	This mode is entered via system request, and exited by deasserting this system request.
Stop	This mode is a low-power mode for the chip. When the chip requires the FlexSPI to enter Stop mode, FlexSPI waits for all transactions to complete (STS0[ARBIDLE] = 1) and return ACK handshake to the system. After the ACK handshake is returned, FlexSPI gates off the AHB clock and serial clock domains internally. The system can gate off the AHB bus clock, IPS bus clock, and serial clock at the system level. This mode is entered via system request. This mode is exited by deasserting this system request, and the ACK handshake message is also deasserted immediately.
Normal	No clock is gated off internally. Normal register access and serial memory access are available.

36.4.4 AHB access memory map

FlexSPI allocates AHB memory space for each memory device starting from the FlexSPI region base address of the system memory map. The FLSHxCR0[FLSHSZ] field determines the amount of memory allocated for each memory device in KB, Fx_SIZE. The memory is then allocated sequentially and contiguously.

NOTE

See the chip-specific section for the maximum flash size supported.

When MCR2[SAMEDEVICEEN] = 1:

- FA1_SIZE = FLSHA1CR0[FLSHSZ] × 1 KB
- FA2_SIZE = FLSHA1CR0[FLSHSZ] × 1 KB
- FB1_SIZE = FLSHA1CR0[FLSHSZ] × 1 KB
- FB2_SIZE = FLSHA1CR0[FLSHSZ] × 1 KB

When MCR2[SAMEDEVICEEN] = 0:

- FA1_SIZE = FLSHSZ × 1 KB
- FA2_SIZE = FLSHA2CR0[FLSHSZ] × 1 KB
- FB1_SIZE = FLSHB1CR0[FLSHSZ] × 1 KB
- FB2_SIZE = FLSHB2CR0[FLSHSZ] × 1 KB

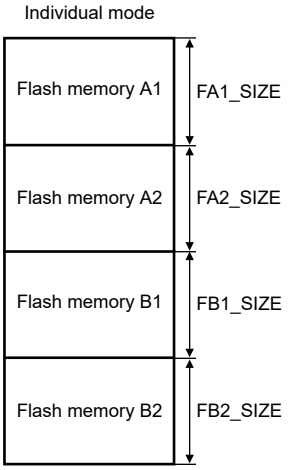


Figure 86. AHB access memory map in individual mode

AHB access memory map in individual mode:

- Flash A1 address range: 0000_0000h to FA1_SIZE
- Flash A2 address range: FA1_SIZE to (FA1_SIZE + FA2_SIZE)
- Flash B1 address range: (FA1_SIZE + FA2_SIZE) to (FA1_SIZE + FA2_SIZE + FB1_SIZE)
- Flash B2 address range: (FA1_SIZE + FA2_SIZE + FB1_SIZE) to (FA1_SIZE + FA2_SIZE + FB1_SIZE + FB2_SIZE)

NOTE

The address used in equations in this topic is the address presented to the memory. The base address for FlexSPI in the system memory map has already been removed, so it not used or shown here.

36.4.4.1 Flash address sent to flash memory devices

The AHB address (AHB command) or IPCR0[SFAR] (IP command) determines the flash memory access start address. See Flash memory access via AHB command and Flash memory access via IP command for more details.

For AHB commands, the FlexSPI controller removes the flash memory base address automatically when sending flash addresses to flash devices. The flash address is sent to devices in two parts: row address and column address. For flash devices that do not support the column address, set FLSHxCR1[CAS] to 0. This setting causes all flash address bits to be sent to flash devices as row addresses.

For word-addressable flash devices, the last bit of the address is not needed, because flash memory is read and programmed in terms of two bytes.

36.4.5 Lookup table (LUT)

The LUT is an internal memory that preserves a number of preprogrammed sequences. Each sequence consists of up to eight instructions which are executed sequentially. When an IP command or an AHB command triggers a flash memory access, the FlexSPI controller:

1. Fetches the sequence from LUT (sequence index or number).
2. Executes the flash memory access to generate a valid flash transaction on the SPI interface.

Figure 87 shows the LUT structures, sequences, and instructions.

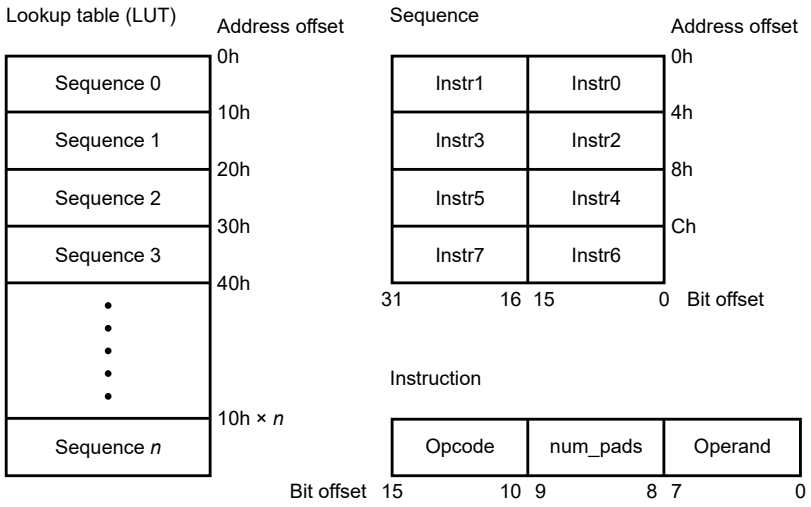


Figure 87. LUT structure, sequences, and instructions

NOTE

If the number of instructions needed for a flash transaction is less than eight, the STOP instruction (code 0000_0000h) must be programmed for unneeded instructions.

For IP and AHB write commands, the FlexSPI controller always executes from instruction pointer 0. For AHB read commands, the FlexSPI controller executes from a saved instruction start pointer. The FlexSPI controller saves the instruction start pointers separately for each flash device. All these saved instruction pointers are zero before the JMP_ON_CS instruction is executed. When the JMP_ON_CS instruction is executed, the operand in the JMP_ON_CS instruction is saved as instruction start pointer. See Execute-In-Place (XIP) enhanced mode.

The reset value of the LUT is unknown because it is implemented as internal memory. The LUT must be programmed according to the connected device. To protect its contents during a code runover, the LUT can be locked or unlocked to prevent unwanted changes after the LUT has been configured. The key to lock or unlock the LUT is **5AF05AF0h**.

To lock the LUT:

1. Write the key (5AF05AF0h) to LUT Key (LUTKEY).
2. Write 1 to LUTCR[LOCK] and write 0 to LUTCR[UNLOCK]. When there is another register write access to FlexSPI between these two write accesses, this LUT is not successfully locked.

To unlock the LUT:

1. Write the key (5AF05AF0h) to the LUT Key Register LUT Key (LUTKEY).
2. Write 0 to LUTCR[LOCK] and write 1 to LUTCR[UNLOCK]. When there is another register write access to FlexSPI between these two write accesses, this LUT is not successfully locked.

The lock status of the LUT can be read from LUTCR[LOCK] and LUTCR[UNLOCK].

36.4.6 Programmable sequence engine

The FlexSPI controller implements a programmable sequence engine that executes the sequence from the LUT. The FlexSPI controller executes the instructions sequentially, and generates flash transactions on the SPI interface accordingly. Table 188 is a complete list of the supported instructions.

Table 188. Instruction set

Name	Opcode	num_pads	Action on SPI interface	Transmit data	Bits/bytes/cycle number
CMD_SDR	00_0001h	00h - one pad (Single mode) 01h - two pads (Dual mode) 02h - four pads (Quad mode) 03h - eight pads (Octal mode)	Transmit command code to flash memory	Command code: Operand[7:0]	Bit number: 8
CMD_DDR	00_0021h				
RADDR_SDR	00_0002h		Transmit row address to flash memory See Flash address sent to flash memory devices	Row_Address[31:0] Row_Address comes from AHB bus (AHB command) or IPCR0[SFAR] (IP command).	Bit number: operand[7:0] Value of operand determines number of bits sent in Row_Address.
RADDR_DDR	00_0022h				
CADDR_SDR	00_0003h		Transmit column address to flash memory. See Flash address sent to flash memory devices.	Column_Address[31:0] Column_Address comes from AHB bus (AHB command) or IPCR0[SFAR] (IP command).	Bit number: operand[7:0] Value of operand determines number of bits sent in Column_Address.
CADDR_DDR	00_0023h				
MODE1_SDR	00_0004h		Transmit mode bits to flash memory	Mode bits: Operand[0]	Bit number: 1
MODE1_DDR	00_0024h				
MODE2_SDR	00_0005h			Mode bits: Operand[1:0]	Bit number: 2
MODE2_DDR	00_0025h				
MODE4_SDR	00_0006h			Mode bits: Operand[3:0]	Bit number: 4
MODE4_DDR	00_0026h				
MODE8_SDR	00_0007h			Mode bits: Operand[7:0]	Bit number: 8
MODE8_DDR	00_0027h				
WRITE_SDR	00_0008h		Transmit program data to flash device	Program data in IP_TX_FIFO or AHB_TX_BUF	AHB burst size and burst type (AHB command) or IPCR1[IDATSZ] (IP command) determines byte number (data size). For details about flash read or program data size, see Flash memory access via AHB command and Flash memory access via IP command.
WRITE_DDR	00_0028h				
READ_SDR	00_0009h		Receive read data from flash device Read data is put into AHB_RX_BUF or IP_RX_FIFO.	-	
READ_DDR	00_0029h				
DATSZ_SDR	00_000Bh		Transmit read or program data size	Internal logic	Bit number: operand[7:0]
DATSZ_DDR	00_002Bh				

Table continues on the next page...

Table 188. Instruction set (continued)

Name	Opcode	num_pads	Action on SPI interface	Transmit data	Bits/bytes/cycle number
			(byte number) to flash device	Read or program data size for current command sequence	Never set operand to zero or greater than 64 for DATSZ instruction.
DUMMY_SDR	00_000Ch		Leave data lines undriven by FlexSPI controller. Turnaround cycles are provided from host driving to device driving. num_pads determines number of pads in input mode.	-	Dummy cycle number (in serial root clock): Operand[7:0] Dummy cycle (N), described in data sheet of flash device, is in number of SCLK cycles. This number may be configurable. In SDR mode, SCLK cycle is same as serial root clock. Operand value must be set to N. In DDR mode, SCLK cycle is double serial root clock cycle. Operand value must be set to 2N, 2N-1 or 2N+1 depending on definition of dummy in data sheet of flash device. See Flash memory access sequence examples and dummy cycle definition in data sheet of external memory.
DUMMY_DDR	00_002Ch				
DUMMY_RWDS_SDR	00_000Dh		Similar to DUMMY_SDR/ DUMMY_DDR instruction. Difference is in dummy cycle number. DQS pin is called RWDS in HyperBus specification. See Dummy instruction for details. **Set operand to "Latency count" for HyperBus devices.**	-	For read command, dummy cycle number (in serial root clock): (operand[7:0] × 4 - 1) if RWDS (DQS pin) is high; (operand[7:0] × 2 - 1) if RWDS (DQS pin) is low; For write command, dummy cycle number (in serial root clock): (operand[7:0] × 4 - 2) if RWDS (DQS pin) is high; (operand[7:0] × 2 - 2) if RWDS (DQS pin) is low;
DUMMY_RWDS_DDR	00_002Dh				

Table continues on the next page...

Table 188. Instruction set (continued)

Name	Opcode	num_pads	Action on SPI interface	Transmit data	Bits/bytes/cycle number
LEARN_SDR	00_000Ah	00h - one pad (Single mode) 01h - two pads (Dual mode) 02h - four pads (Quad mode) 03h - eight pads (Octal mode)	Receive read data or preamble bit from flash device FlexSPI controller compares data line bits with the DLPR register to determine a correct sampling clock phase.	-	Bit number: operand[7:0] Never set operand to zero for LEARN instruction. Value of operand indicates number of bits to receive and compare to DLPR value. For example, 8-bit pattern 5Ah on each data line must set operand to 8.
LEARN_DDR	00_002Ah				
JMP_ON_CS	00_001Fh	Num_pads setting ignored.	Stop execution, deassert CS and save operand[7:0] as instruction start pointer for next sequence. Normally this instruction is used to support Execute-In-Place enhanced mode. See Execute-In-Place (XIP) enhanced mode. This instruction is only allowed for AHB read commands. When using this instruction in IP command or AHB write command, interrupt status flag set (INTR[IPCMDERR] or INTR[AHBCMDERR]).	-	No transaction on SPI interface.
STOP	00_0000h		Stop execution and deassert CS. Next command sequence (to same flash device) starts from instruction pointer 0.	-	

The programmable sequence engine allows configuration of the LUT according to the connected external serial device. The flexible LUT structure easily adapts to new command or protocol changes from different vendors.

The DDR sequence is a flash memory access sequence that contains DDR instructions other than DUMMY_DDR, and may contain SDR instructions. The output and input timing on FlexSPI differs for SDR and DDR sequences. When included as part of a DDR sequence, SDR instructions execute differently from SDR instructions in an SDR sequence. See FlexSPI input timing and FlexSPI output timing.

36.4.6.1 Executing instructions on SPI interface

This section describes the execution of instructions on the SPI interface. For all instructions that receive bits from or transmit bits to flash devices:

- The bit order in one byte is higher on DATA7 than DATA0.
- The bit order is higher on port B than port A.

This order is shown in Figure 88.

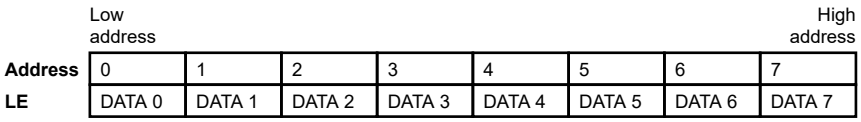


Figure 88. Bus bit order

Figure 89. Bit order precedence in combination mode

Table 189. SPI instructions

Instructions	Description
Command (CMD_SDR, CMD_DDR)	Normally used to transmit a command code to the external device. The command code is the 8-bit operand in the instructions. See Flash memory access sequence examples.
Address (RADDR_SDR, RADDR_DDR, CADDR_SDR, CADDR_DDR)	Normally used to send the flash memory access start address (row address or column address) to an external device. FlexSPI determines the bits of the row address or column address according to the AHB access address or IP command address. See Flash address sent to flash memory devices. The operand value in the instruction code represents the number of address bits to send. For example, when NUM_PADSn = 8, a memory reading the flash memory address on four SCK edges requires the sum of bits in CADDR and RADDR instructions to be 32. Otherwise, FlexSPI does not generate four SCK edges of address phase. See Flash memory access sequence examples.
Mode (MODEx_SDR, MODEx_DDR)	Normally used to send mode bits to external devices. Mode bits are the lower bits of an operand. For example, the bit number is 1 for MODE1_SDR and MODE1_DDR, 2 for MODE2_SDR and MODE2_DDR, 4 for MODE4_SDR and MODE4_DDR, and 8 for MODE8_SDR and MODE8_DDR. The pad number should not be greater than the mode bit number. For example, you cannot set NUM_PADSn to 11b (Octal mode) for MODE4_* instructions. See Flash memory access sequence examples for more details.
Data Size (DATSZ_SDR, DATSZ_DDR)	Used to send the size of program data or read data (byte number) to external devices. This instruction is normally used in FPGA applications where the memory space in external device acts like a FIFO. The external device requires data size information to determine how much

Table continues on the next page...

Table 189. SPI instructions (continued)

Instructions	Description
	data are popped from or pushed into the internal FIFO. The operation value in the data size instructions is the bit number. See Flash memory access sequence examples.
Write (WRITE_SDR, WRITE_DDR)	Normally used to send program data to external device. Programming data is fetched from IP_TX_FIFO (IP Command) or AHB_TX_Buffer (AHB command). For details about flash program data size, see Flash memory access via AHB command and Flash memory access via IP command. The byte order for program data is always from low to high. See Flash memory access sequence examples for more details.
Read (READ_SDR, READ_DDR)	Normally used to receive flash memory data from external devices. Received data is put into IP_RX_FIFO (IP Command) or AHB_RX_Buffer (AHB command). For information about flash memory read data size, see Flash memory access via AHB command and Flash memory access via IP command. The byte order for reading data is always from low to high. See Flash memory access sequence examples.
Dummy (DUMMY_SDR, DUMMY_DDR, DUMMY_RWDS_SDR, DUMMY_RWDS_DDR)	Used to provide turnaround cycles on the SPI interface. During dummy instruction, the FlexSPI controller and external devices do not drive the SPI interface. See Programmable sequence engine for details about dummy cycle number. DUMMY_RWDS_DDR can be used for a HyperBus device that uses RWDS pin to indicate whether extra latency is needed. DUMMY_RWDS_SDR is reserved. The FlexSPI controller checks the DQS pin input level at the fourth cycle after SCLK output toggling is enabled. The DQS pin is called RWDS in HyperBus specification. See Flash memory access sequence examples. **NOTE** FlexSPI releases the bus after at least one cycle. To avoid data contention, the NUM_PADSn value for the dummy commands should be configured to match the number of data lines used for the external memory.
Learn (LEARN_SDR, LEARN_DDR)	Used to determine the correct sample clock phase for flash memory read data sampling. External device drives read data (or data learning pattern) bits on the FlexSPI interface. The FlexSPI controller compares the data line bits to the DLPR register to determine a correct sampling clock phase. Clock phase selection is automatically updated after executing learn instructions. See Data learning for more details. The operand value in learn instructions is the byte number. FlexSPI checks the same data pattern on each data line. The byte order is byte 0, byte 1, byte 2, byte 3, byte0, byte 1, and so on. • Byte 0 is DLPR register bits 7–0. • Byte 1 is DLPR register bits 15–8. • Byte 2 is DLPR register bits 23–16. • Byte 3 is DLPR register bit 31–24. The bit order is from high to low in each byte. See Flash memory access sequence examples.

36.4.6.2 Flash memory access sequence examples

Diagrams below all assume MCR0[SERCLKDIV] = 0.

Figure 90 shows an example SDR single I/O read sequence (Cypress Serial Nor Flash S25FS512S) in individual mode.

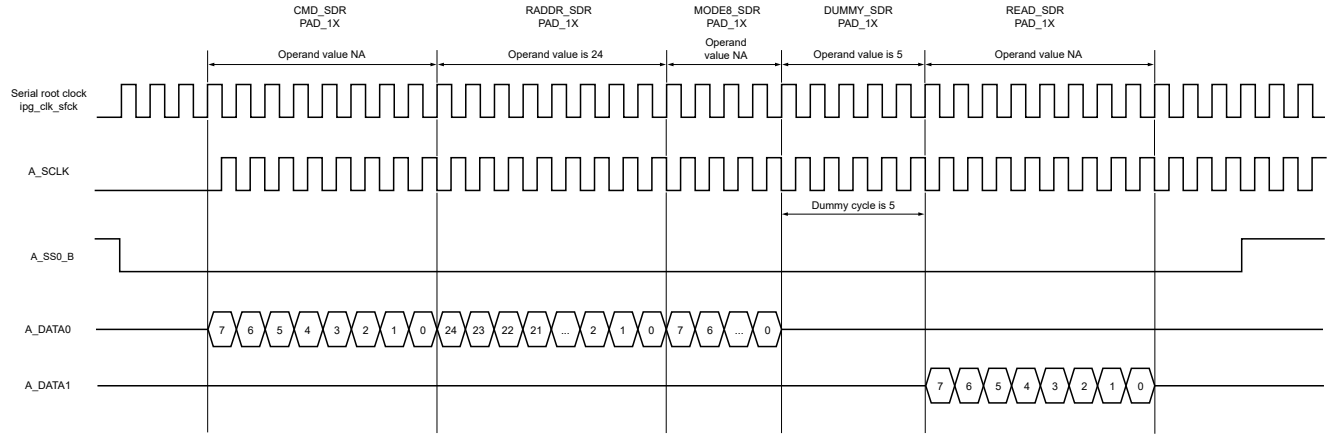


Figure 90. Flash memory access sequence example (SDR single I/O read sequence)

NOTE

- FlexSPI dummy instruction starts and ends at the rising edge of serial root clock.
- Device dummy cycle starts and ends at the falling edge of SCLK (on Cypress S25FS512S data sheet).

Figure 91 shows an example SDR quad I/O read sequence (Cypress Serial Nor Flash S25FS512S) in individual mode.

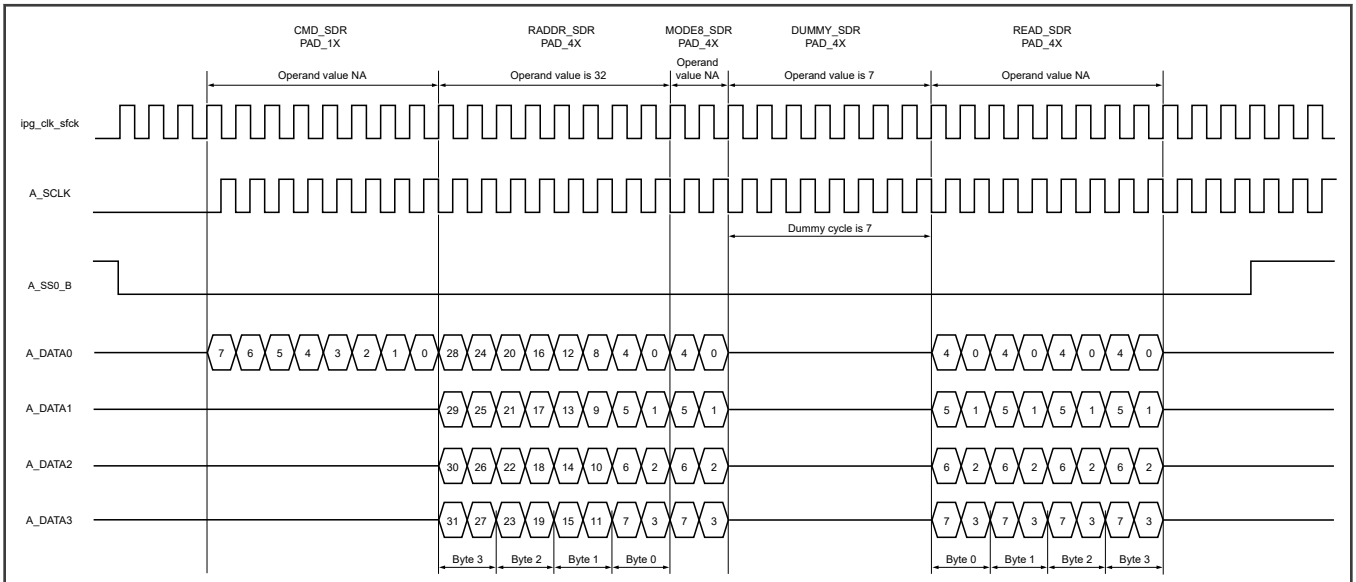


Figure 91. Flash memory access sequence example (SDR quad I/O read sequence)

NOTE

- FlexSPI dummy instruction starts and ends at the rising edge of serial root clock.
- Device dummy cycle starts and ends at the falling edge of SCLK (on Cypress S25FS512S data sheet).

Figure 92 shows an example learn instruction (not for a specified flash device) in individual mode.

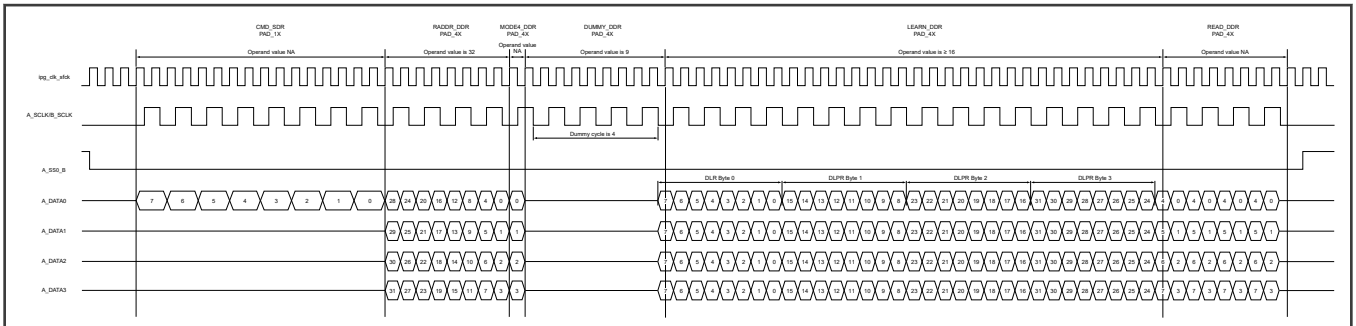


Figure 92. Flash memory access sequence example (data learning)

NOTE

- FlexSPI dummy instruction starts and ends at the rising edge of serial root clock.
- Device dummy cycle starts and ends at the falling edge of SCLK.
- The operand value of DUMMY_DDR instruction is odd because the total cycle number before DUMMY_DDR cycle is odd.

Figure 93 shows an example HyperBus device read transaction (single latency count) in individual mode.

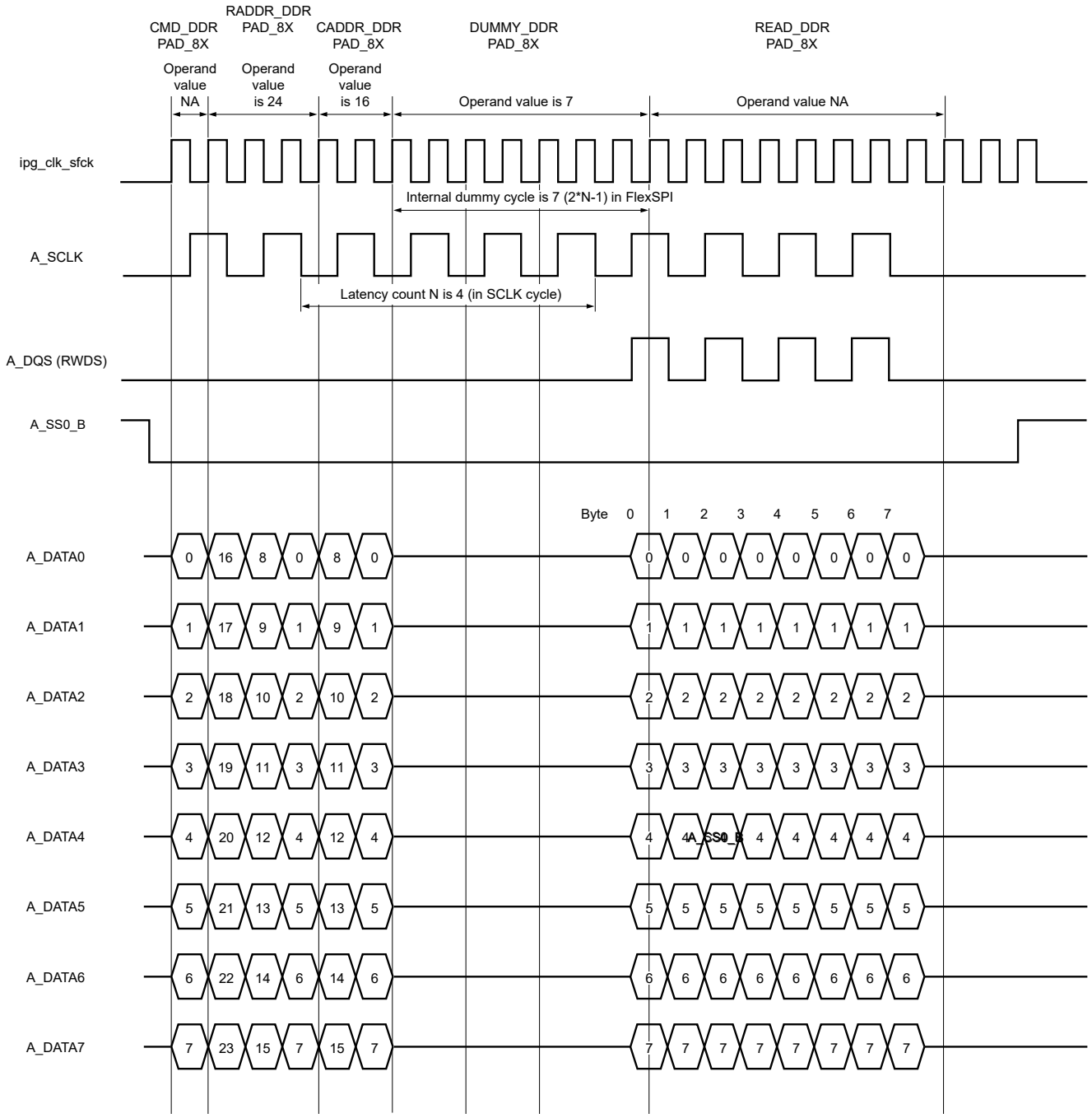


Figure 93. HyperBus device read transaction with single latency count

NOTE

FlexSPI continues to drive the clock until it has latched all of the read data it is expecting. On-chip delay for the data to reach FlexSPI can increase the number of SCK clock pulses.

Figure 94 shows an example HyperBus device read transaction (additional latency count) in individual mode.

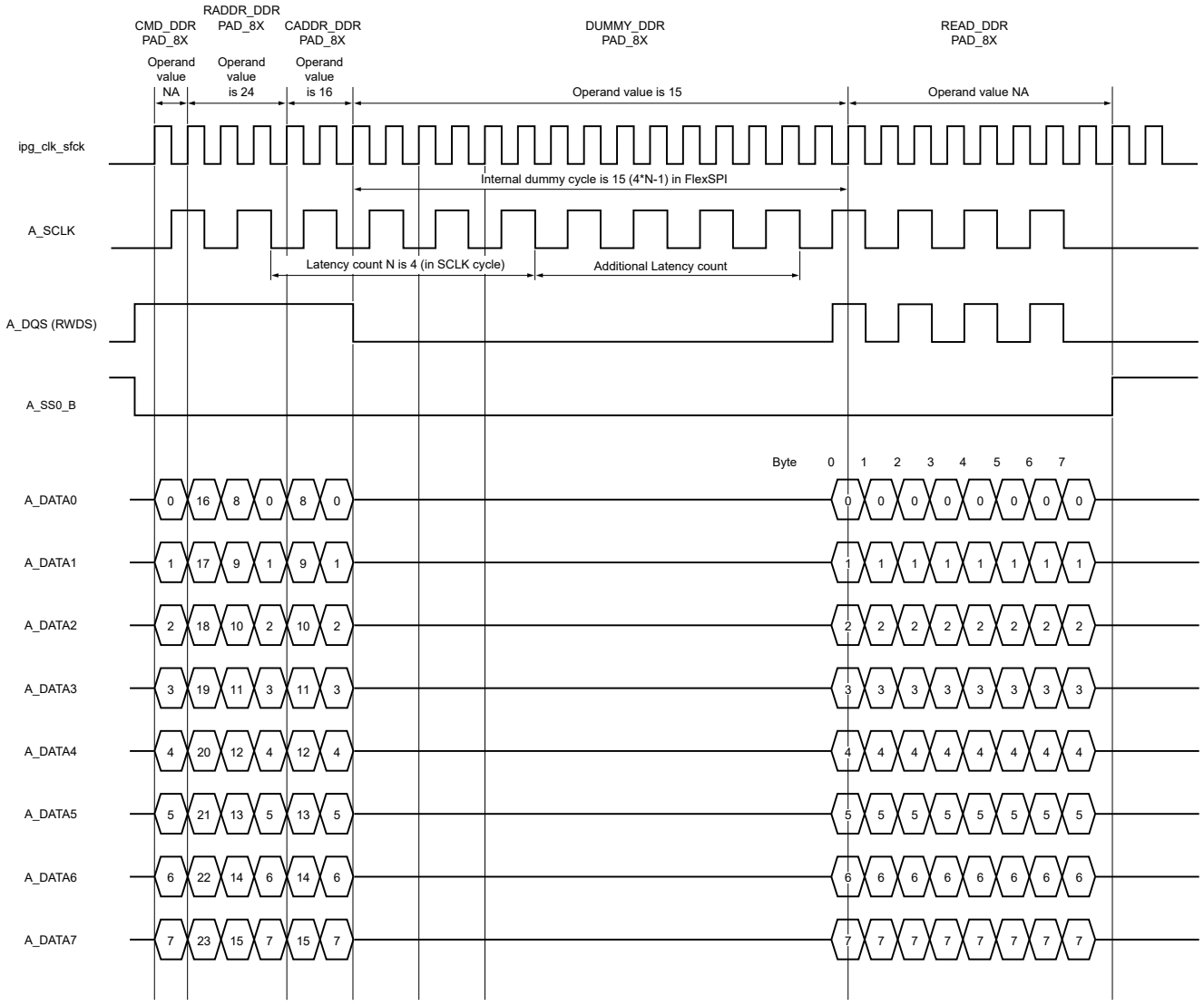


Figure 94. HyperBus device read transaction with additional latency count

Figure 95 shows an example HyperBus device write transaction (single latency count) in individual mode.

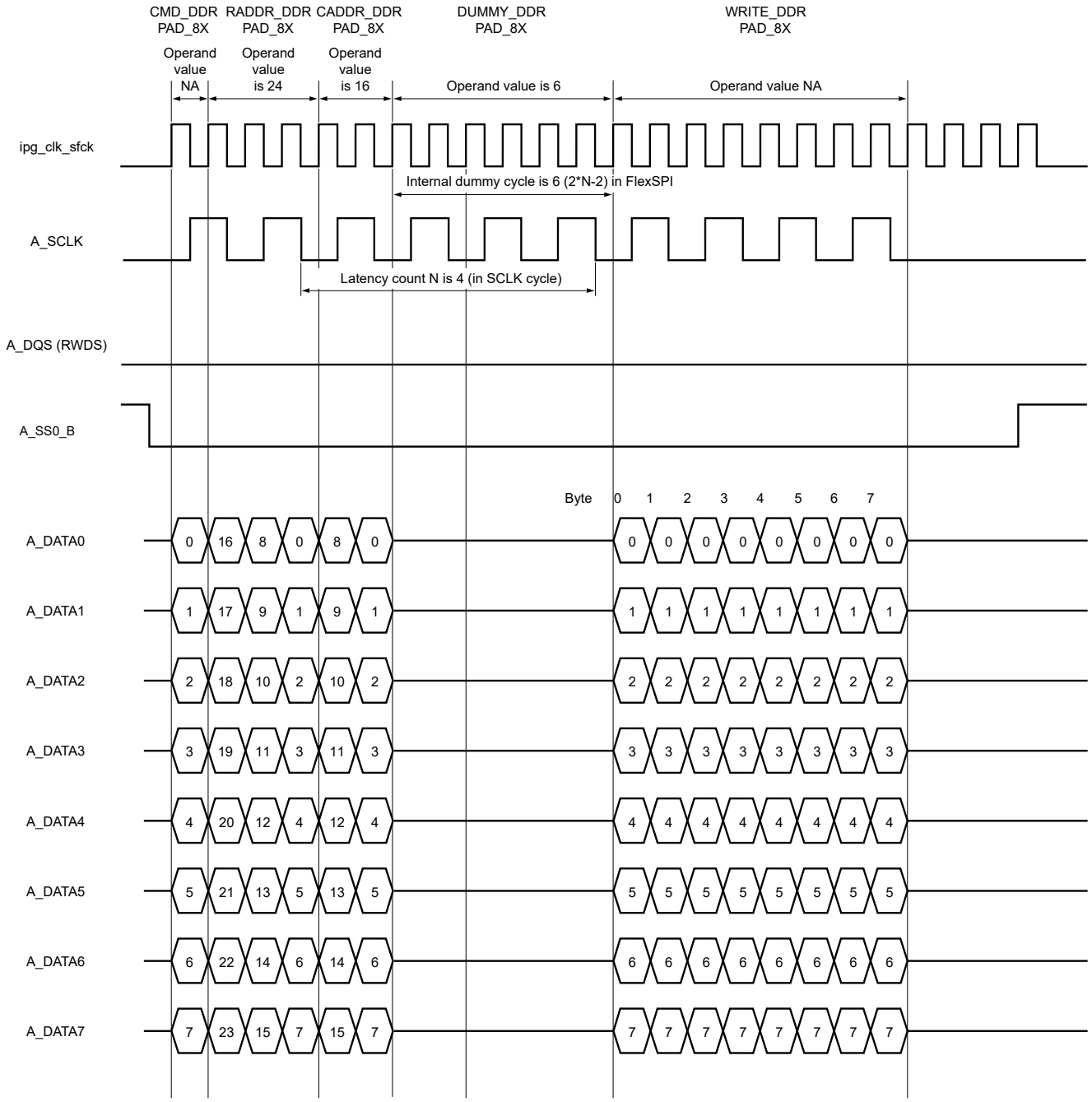


Figure 95. HyperBus device write transaction with single latency count

Figure 96 shows an example HyperBus device write transaction (additional latency count) in individual mode.

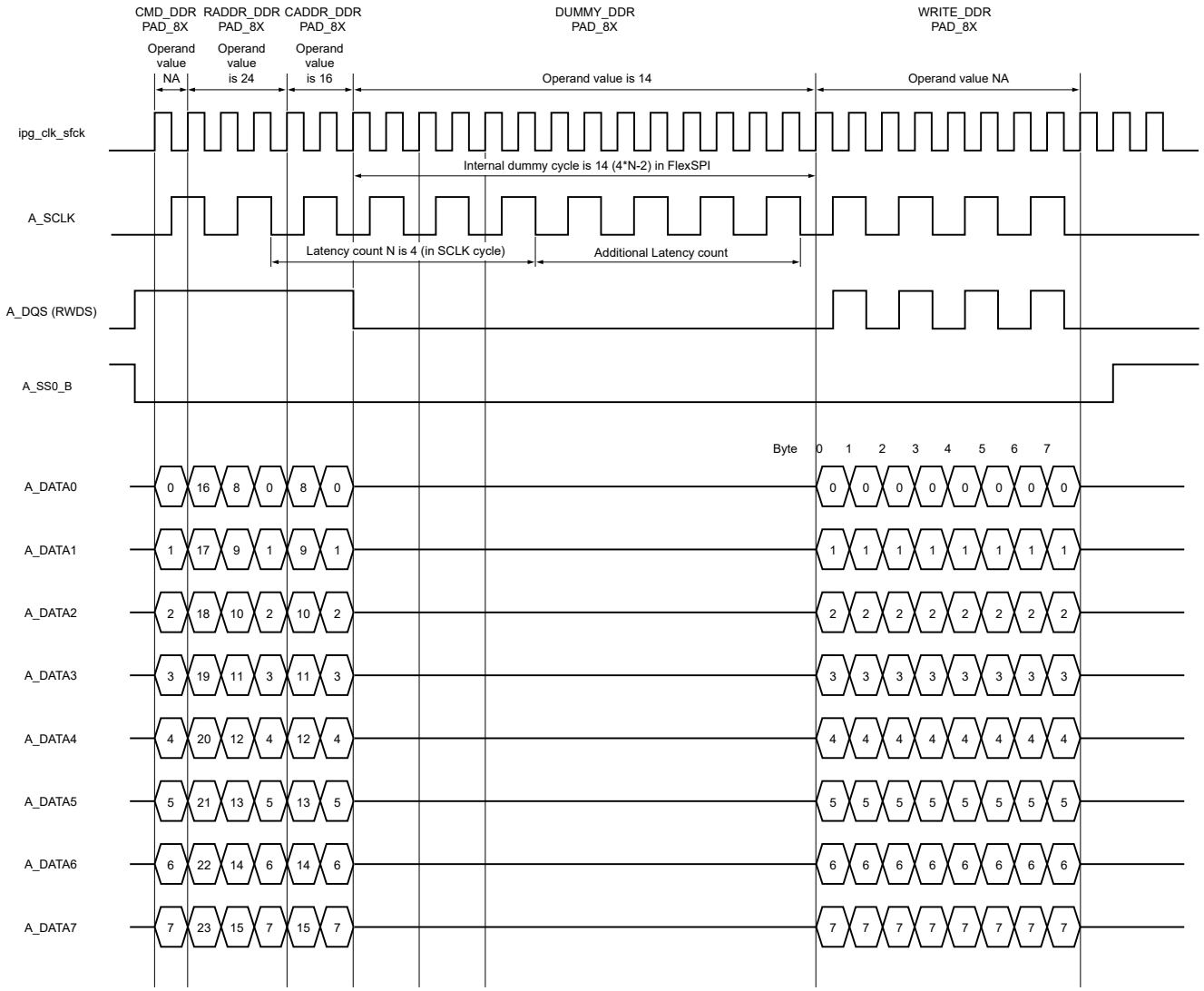


Figure 96. HyperBus device write transaction with additional latency count

36.4.7 Clocking

This section describes FlexSPI clocks and special clocking requirements.

Table 190. Clock usage

Clock name	Description	Comment
serial clock root (ipg_clk_sfck)	Root clock for Serial domain	FlexSPI core clock, used to generate the serial clock output. The clock is optionally divided using MCR0[SERCLKDIV] to obtain the SCLK.

Table continues on the next page...

Table 190. Clock usage (continued)

Clock name	Description	Comment
		Its frequency is same as SPI SCK frequency in SDR mode if MCR0[SERCLKDIV] = 0.
ahb clock (hclk)	AHB bus clock	hclk frequency is higher than 1/4 of serial root clock, so internal async FIFO is never full. This clock is used for internal logic in FlexSPI.
ipg clock (ipg_clk)	IPS bus clock	Register access clock. ipg_clk must be equal to hclk or an integer division of hclk. For example, ipg_clk = hclk/2.
SCLK	FlexSPI serial clock. Output clock on SCLK pin	Half clock frequency of serial clock root in DDR mode, and same frequency as serial clock root in SDR mode. Clock output toggles during the entire flash memory access sequence.
DQS_OUT	Dummy Read Strobe output	Same frequency as SCLK. Clock output toggles during READ and LEARN instructions. The DQS ensures that the data can be strobed correctly. It is only used for data strobe, instead of command or address strobe. The data includes useful read data and data learning patterns.
DQS_IN	Sample clock for receive data	Same frequency as SCLK. Sample clock comes from looped-back dummy read strobe, looped-back SCLK, or flash-memory-provided read strobe.

36.4.8 Interrupts

Table 191 describes all interrupts that FlexSPI generates.

Table 191. Flash memory address, row address, and column address

Interrupt	Enable	Condition
IP command done	INTEN[IPCMDDONEEN]	An IP command is finished.
IP command grant error	INTEN[IPCMDGEEN]	An IP command grant timeout occurs (bus not granted after MCR0[IPGRANTWAIT] × 1024 AHB clock cycles). See Overview of error categories, flags, and triggered sources.
AHB command grant error	INTEN[AHBCMDGEEN]	An AHB command grant timeout occurs (bus not granted after MCR0[AHBGRANTWAIT] × 1024 AHB

Table continues on the next page...

Table 191. Flash memory address, row address, and column address (continued)

		clock cycles). See Overview of error categories, flags, and triggered sources.
IP command error	INTEN[IPCMDERREN]	A command check error or a command execution error for an IP command occurs. See Overview of error categories, flags, and triggered sources.
AHB command error	INTEN[AHBCMDERREN]	A command check error or a command execution error for AHB command occurs. See Overview of error categories, flags, and triggered sources.
IP_RX_FIFO watermark available	INTEN[IPRXWAEN]	The fill level of IP_RX_FIFO reaches the watermark level (IPRXFCR[RXWMRK]).
IP_TX_FIFO watermark empty	INTEN[IPTXWEEN]	The empty level of IP_TX_FIFO reaches the watermark level (IPTXFCR[TXWMRK]).
Data learning failed	INTEN[DATALEARNFAILEN]	No valid sample clock phase found after LEARN instruction is executed. See Overview of error categories, flags, and triggered sources.
Sequence execution timeout	INTEN[SEQTIMEOUTEN]	A sequence execution time exceeds the timeout wait time (MCR1[SEQWAIT]). For example, the following flash memory read command sequence lasts about 800_0000h cycles (ipg_clk_sfck). If SEQWAIT is set to FFFFh, a sequence timeout interrupt is generated. • IP command triggers timeout. • Flash memory read data size is 100_0000h bytes. • Flash memory accesses in single mode and SDR mode.
AHB bus error	INTEN[AHBBUSERROREN]	AHB bus response timeout occurs, or AHB read access occurs while OTFAD is enabled and FlexSPI is still fetching the key blob. For example, AHB read sequence is not configured properly in LUT (such as without a READ instruction). No data is read from the external device. As a result, FlexSPI cannot reach the read data in AHB_RX_Buffers for the AHB read command.
Write command stops SCLK	INTEN[SCKSTOPBYWREN]	IP_TX_FIFO is empty while executing a write command sequence. FlexSPI

Table continues on the next page...

Table 191. Flash memory address, row address, and column address (continued)

		stops SCLK output clock toggling and waits for write data filling.
Read command stops SCLK	INTEN[SCKSTOPBYRDEN]	IP_RX_FIFO is full while executing a read command sequence. FlexSPI stops SCLK output clock toggling and waits to read the data from IP_RX_FIFO.
Key done	INTEN[KEYDONEEN]	OTFAD is enabled and key blob processing is enabled, key blob processing is done, and no error is detected.
Key error	INTEN[KEYERROREN]	OTFAD is enabled and key blob processing is enabled, and an error occurs during key blob processing. Once this interrupt is set, only reset can clear it.

36.4.9 On-The-Fly AES Decryption (OTFAD) support

After power-up, if OTFAD and key fetches are enabled in the system, FlexSPI automatically starts fetching the key blobs from location 0 of flash A1. The key fetches continue until OTFAD receives and processes all key blobs. During the fetch operation, any AHB access causes an error response, and INTR[AHBBUSERROR] is set. No IPS access is allowed during the fetching of key blobs. INTR[KEYDONE] or INTR[KEYERROR] represents the key blob processing result.

After OTFAD processes all key blobs, the OTFAD engine decrypts any reads from FlexSPI and returns them on the AHB bus. When OTFAD is enabled, FlexSPI always enables prefetch; only AHBCR[READSZALIGN] can disable the prefetch. The Read Resume function is also disabled. All read access addresses are eight-byte-aligned automatically, regardless the setting of AHBCR[READADDROPT]. If OTFAD is enabled, the prefetch never crosses the 1 KB boundary.

36.4.10 Flash memory access via IP command

Follow these steps to trigger a flash memory access via IP command.

1. If this command is a programing command, fill the IP transmit FIFO with programming data (for instance, programming flash data and flash memory status registers.)
2. Write the flash memory access start address to IPCR0[SFAR].
3. Write the read or program data size to IPCR1[IDATSZ]. Write the sequence index to IPCR1[ISEQID]. Write the sequence number to IPCR1[ISEQNUM].
4. Write 1 to IPCMD[TRG] to trigger a flash memory access command.
5. Wait for the INTR[IPCMDDONE] flag to set or for the IP command done interrupt to fire, indicating the command has completed on the FlexSPI interface.

NOTE

- The IP transmit FIFO can be filled before or after writing the IPCR0, IPCR1, and IPCMD registers. If the SFM command starts with the IP transmit FIFO empty, FlexSPI stops SCLK toggling to wait for transmit data ready automatically.
- The IPCMD register must be written after writing the IPCR0 and IPCR1 registers.
- One IP command can issue up to eight command sequences.
- You cannot issue another IP command before the previous IP command is finished. Performing this action results in unknown behavior.

If this command is a read command, all read data from flash memory is put into the IP receive FIFO. Software must read data from the IP receive FIFO via AHB bus or IP bus. When the IP receive FIFO is full and the flash device still contains data to read, FlexSPI stops SCLK toggling until the FIFO has free space. See SCLK stop.

A triggered serial flash command contains:

- A flash memory access start address. IPCR0[SFAR] determines this address.
- A flash chip select. The flash memory access address and flash memory size setting (FLSHxCR0[FLSHSZ]) determine the chip select.
- A flash command sequence index and sequence number. FlexSPI sequentially executes the sequences indexed from IPCR1[ISEQID] to (IPCR1[ISEQID] + IPCR1[ISEQNUM]) in LUT.
- Flash individual memory access mode:
- The flash memory read or program data size, in bytes.
  - If the value of IPCR1[IDATSZ] is nonzero, the data size is IPCR1[IDATSZ].
  - If the value of IPCR1[IDATSZ] is zero, the operand value in the read or write instruction determines the data size.

NOTE

- Software must ensure that the last sequence index never exceeds the LUT sequence number. That is, IPCR1[ISEQID] + IPCR1[ISEQNUM] must be less than 32.
- For sequence numbers greater than one, the data size is applied to every command sequence.
- If there is no write or read instruction in the command sequence, the data size is ignored.

The IP command request is sent to the arbitrator after software triggers it. It is not executed on the FlexSPI interface until the arbitrator grants it. See Command arbitration.

36.4.10.1 Reading data from IP receive FIFO

FlexSPI puts read data from the external device into the IP receive FIFO for IP commands.

The data stored in the IP receive FIFO can be read using IPS and register or system and AHB bus transactions:

- Via IP bus
- Via AHB bus

If MCR0[ARDFEN] = 1, only the AHB bus can read the read data in the IP receive FIFO. Read access to the IP receive FIFO using IP bus always returns with data zero, and no bus error occurs.

If MCR0[ARDFEN] = 0, only the IPS bus can read the read data in the IP receive FIFO. Read access to the IP receive FIFO using the AHB bus triggers a bus error.

FlexSPI pushes read data into the IP receive FIFO in sets of 64 bits each time it receives 64 bits of data from an external device. When the number of read data bits is not 64-bit-aligned, FlexSPI pushes additional zero bits into the IP receive FIFO for the last push.

The processor or DMA can read the IP receive FIFO.

36.4.10.1.1 Reading via processor

To read data from the IP receive FIFO via processor, configure the items below.

- Write 0 to IPRXFCR[RXDMAEN].
- Write the watermark level to IPRXFCR[RXWMRK]. The watermark level is (IPRXFCR[RXWMRK] + 1) × 8 bytes.
- Write 1 to INTEN[IPRXWAEN] to enable the IP receive FIFO watermark available interrupt (optional).

The processor must poll INTR[IPRXWA] or wait for the IP receive FIFO watermark available interrupt before reading data from the FIFO. Waiting ensures that a watermark level amount of data is filled in the IP receive FIFO before reading.

After reading a watermark level amount of data from the FIFO, software must set the INTR[IPRXWA] flag to pop that data from the FIFO.

Figure 97 shows the reading flow from the IP receive FIFO via processor.

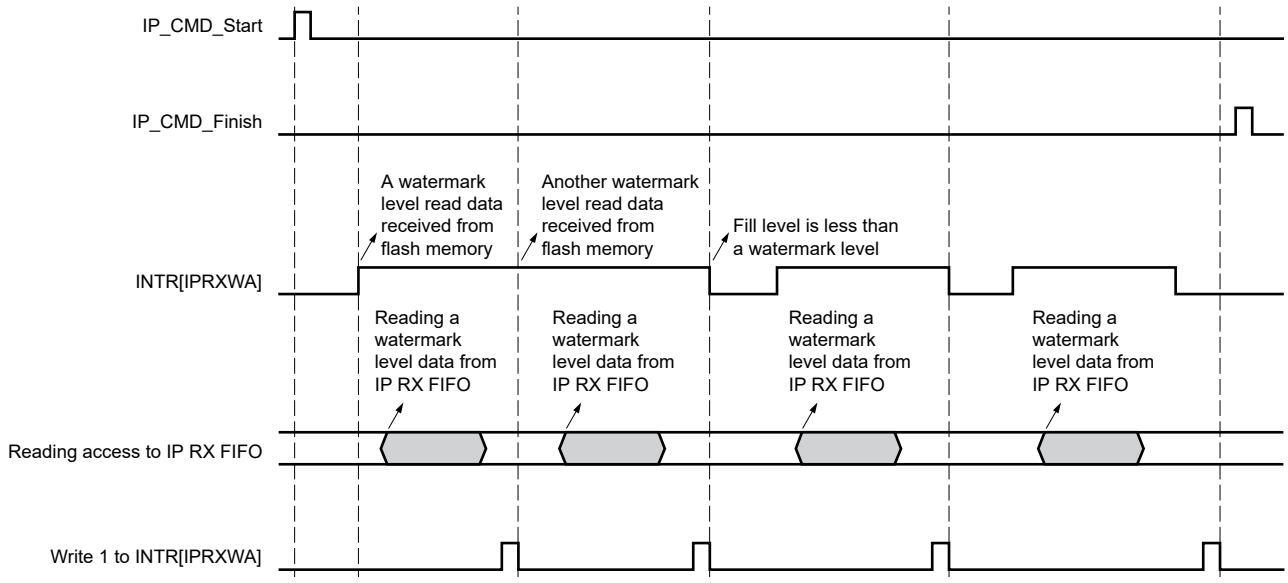


Figure 97. Reading IP receive FIFO via processor

NOTE

- The processor must read a watermark level of data from the IP receive FIFO before setting the INTR[IPRXWA] flag.
- The total flash read or program data size is not required to be a multiple of the watermark level. When the reading data size from the FIFO is less than a watermark level for the last time, software polls IPRXFSTS[FILL], not INTR[IPRXWA]. After all data from the FIFO are copied and all command sequences to flash memory are finished (INTR[IPCMDDONE] = 1), software sets IPRXFCR[CLRIPRXF] to clear the FIFO. If the FIFO is not cleared, the reading data will be incorrect for the next reading command to flash memory.
- Setting the INTR[IPRXWA] flag pops the IP receive FIFO data. Each read access to the FIFO does not pop the data.

36.4.10.1.2 Reading via DMA

To read data from the IP receive FIFO via DMA, configure the items below.

- Write 1 to IPRXFCR[RXDMAEN].
- Write the watermark level to IPRXFCR[RXWMRK]. The watermark level is (IPRXFCR[RXWMRK] + 1) × 8 bytes.

- Configure the DMA transfer minor loop size to match the watermark level.

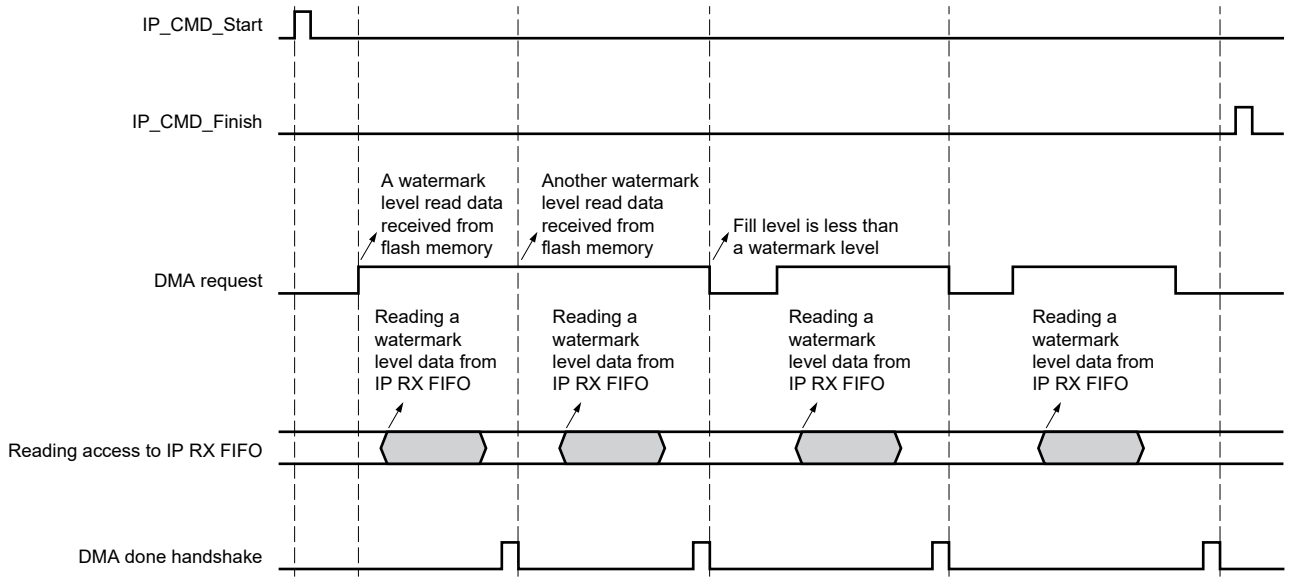


Figure 98. Reading from IP receive FIFO via DMA

NOTE

- When the fill level of the FIFO meets or exceeds the watermark level, a DMA request is generated. This request is level-valid, not pulse-valid.
- DMA reads a watermark level amount of data from the IP receive FIFO each time. (Configure the minor loop size to match the watermark level).
- DMA must return a done handshake (pulse valid) to FlexSPI each time it finishes reading a watermark level amount of data.
- The DMA done handshake (not each read access) pops the IP receive FIFO data.
- The total read and write data size (major loop size) must be a multiple of the watermark level. The DMA does not know when the data is ready for the last reading. When the data size is not a multiple of the watermark level, it is too complex for the DMA driver to poll IPRXFSTS[FILL].

36.4.10.2 Writing data to IP transmit FIFO

The programming data is put into the IP transmit FIFO, then FlexSPI transmits it to the flash memory. IPS or AHB access can write it:

- Via IP bus
- Via AHB bus

To write the programming data into the IP transmit FIFO via AHB bus, write 1 to MCR0[ATDFEN]. The IP bus write access to the FIFO is ignored, but no bus error occurs.

To write the programming data into the IP transmit FIFO via IP bus, write 0 to MCR0[ATDFEN]. The AHB bus write access to the FIFO returns a bus error.

When FlexSPI fetches data to transmit, 64 bits of data from the IP transmit FIFO are popped. If the programming data size is not a multiple of 64 bits, the number of last popped valid bits is less than 64 bits. No error occurs; these invalid bits are not transmitted to the flash memory.

The processor or DMA can fill the IP transmit FIFO with programming data.

36.4.10.2.1 Writing via processor

To write data into the IP transmit FIFO via processor, configure the items below.

- Write 0 to IPTXFCR[TXDMAEN].
- Write the watermark level to IPTXFCR[TXWMRK]. The watermark level is (IPTXFCR[TXWMRK] + 1) × 8 bytes.
- Write 1 to INTEN[IPTXWEEN] to enable the IP transmit FIFO watermark available interrupt (optional).

The processor must poll INTR[IPTXWE] or wait for the IP transmit FIFO watermark empty interrupt before writing data to the FIFO. Waiting ensures that a watermark level amount of data is available in the IP transmit FIFO before writing.

After writing a watermark level amount of data to the FIFO, software must set the INTR[IPTXWE] flag to push that data into the FIFO. (The write pointer is incremented.)

NOTE

Setting the INTR[IPTXWE] flag pushes the IP transmit FIFO data. Each write access to the FIFO does not push the data.

Figure 99 shows the writing flow to the IP transmit FIFO via processor.

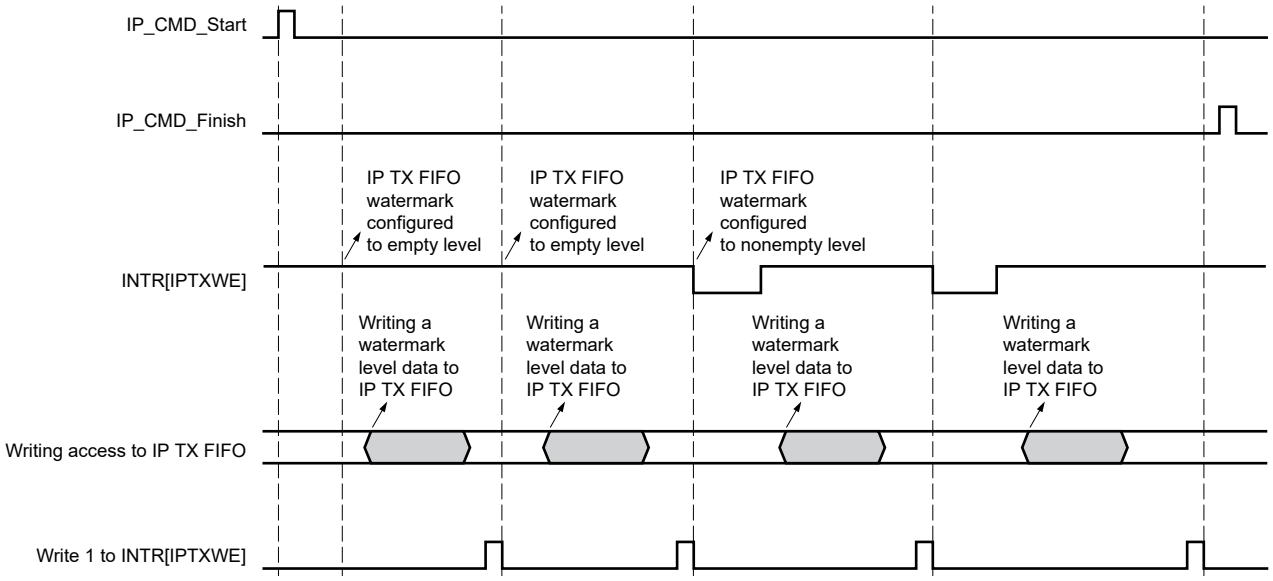


Figure 99. Writing to IP transmit FIFO via processor

NOTE

- The processor must write a watermark level amount of data into the IP transmit FIFO each time.
- The total flash write data size is not required to be a multiple of the watermark level. In this case, the writing data size to the FIFO is less than a watermark level for the last time. After all data to the FIFO are written and all command sequences to flash memory are finished (INTR[IPCMDDONE] = 1), the processor sets IPTXFCR[CLRIPTXF] to clear the FIFO. If the FIFO is not cleared, the programming data will be incorrect for the next programming command to flash memory.

36.4.10.2.2 Writing via DMA

To write data into the IP transmit FIFO via DMA, configure the items below.

- Write 1 to IPTXFCR[TXDMAEN].
- Write the watermark level to IPTXFCR[TXWMRK]. The watermark level is (IPTXFCR[TXWMRK] + 1) × 8 bytes.

- Configure the DMA transfer minor loop size to match the watermark level.

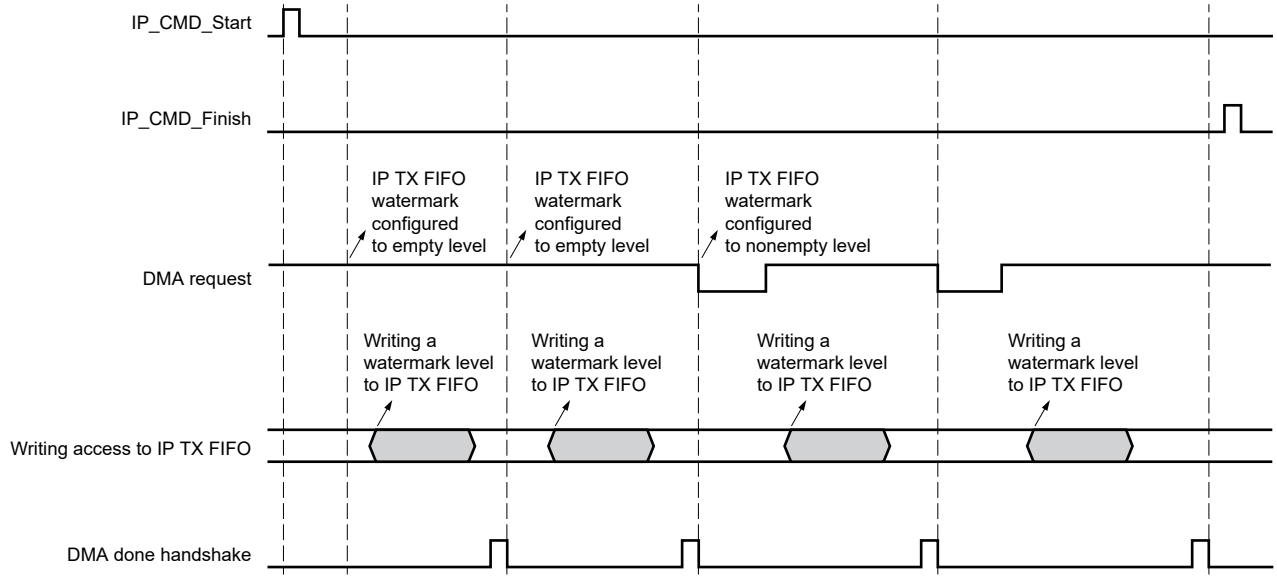


Figure 100. Writing into IP transfer FIFO via DMA

NOTE

- When there is empty space in the IP transmit FIFO greater than the watermark level, a DMA request is generated. This request is level-valid, not pulse-valid.
- DMA writes a watermark level amount of data into the FIFO each time. (Configure the minor loop size to match the watermark level.
- DMA returns a done handshake (pulse valid) to FlexSPI each time it finishes writing a watermark level amount of data.
- The DMA done handshake (not each write access) pushes the IP transmit FIFO data.
- The total program data size (major loop size) is not required to be a multiple of the watermark level. After writing all data into the IP transmit FIFO and all command sequences to flash memory are finished (INTR[IPCMDDONE] = 1), write 1 to IPTXFCR[CLRIPTXF] to clear the FIFO. Failure to clear the FIFO results in incorrect programming data for the next programming command to flash memory.

36.4.11 Flash memory access via AHB command

Flash memory can be accessed via AHB bus directly on the AHB address space. This address space is mapped to serial flash memory in FlexSPI. AHB bus accesses to this address space trigger flash memory access command sequences as needed.

For AHB read access to serial flash memory, FlexSPI fetches data from flash memory into the AHB receive buffer, then returns the data on the AHB bus. For AHB write access to serial flash memory, FlexSPI buffers AHB bus write data into the AHB transmit buffer, then transmits the data to serial flash memory.

No software configuration or polling is needed for AHB commands, except for FlexSPI initialization. As with a normal AHB target, the AHB controller accesses the external flash device transparently.

AHB commands are normally used to access serial flash memory space. IP commands must be used to access the control and status registers or other spaces, such as a one-time programmable (OTP) space in an external flash device.

The AHB commands for read and write access are discussed in detail below.

36.4.11.1 AHB write access to flash memory

For AHB write access to flash memory, FlexSPI buffers the write data from the AHB bus into the internal AHB transmit buffer. FlexSPI then transmits the data to flash memory. FlexSPI only buffers write data for one AHB burst. Figure 101 shows the hardware operation in response to an AHB write access to flash memory.

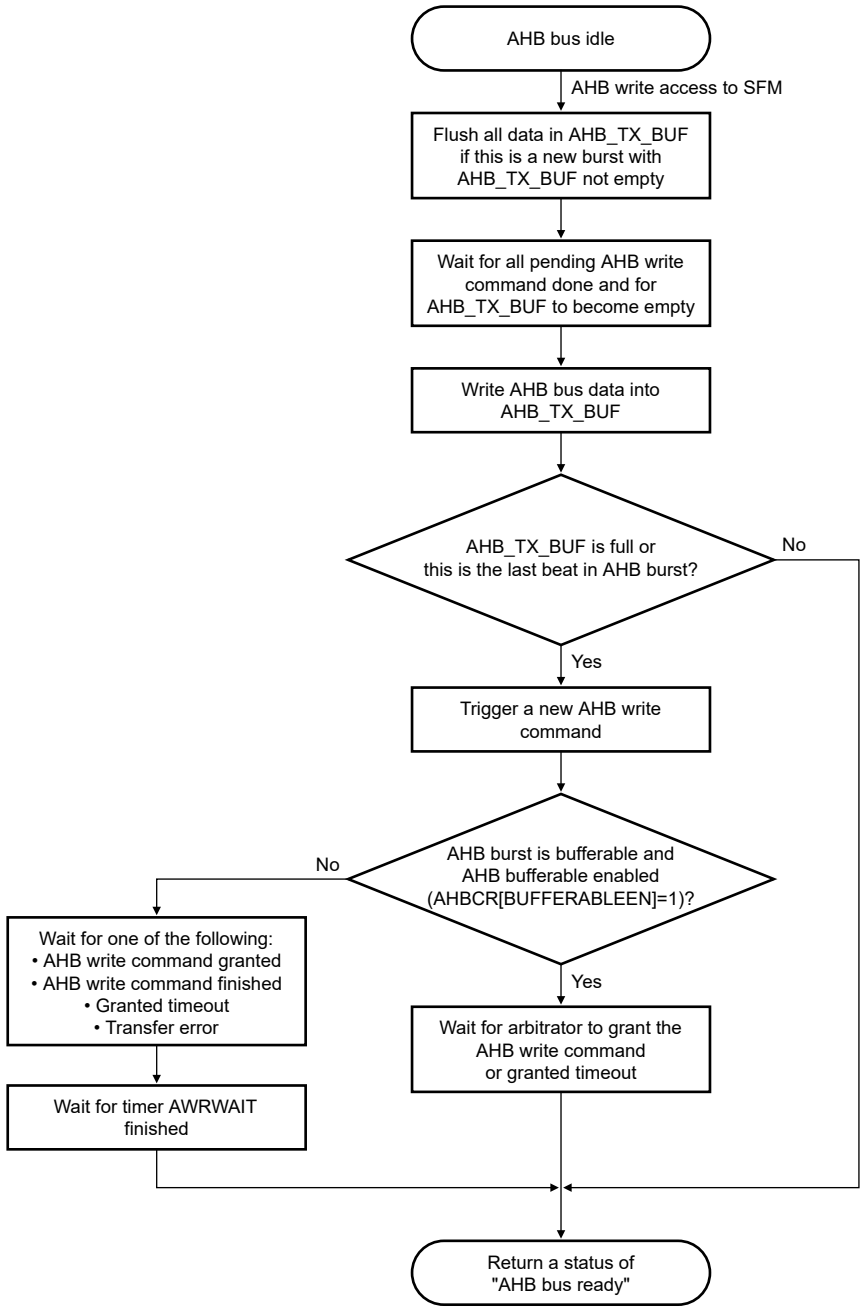


Figure 101. Hardware operation in response to AHB write access to flash memory

NOTE

1. If previous AHB burst is INCR write burst, AHB transmit buffer may not be empty when AHB bus is idle or new burst arrives. A new AHB write command is triggered to flush all data in AHB transmit buffer to external flash device. The new burst is held until this AHB write command finishes.
2. If the current access is bufferable, when the AHB write command is triggered, AHB ready is returned after the write command is granted. If the current access is non-bufferable, AHB ready is returned after the write command finishes.

FlexSPI triggers a new AHB write command in these cases:

- This beat is the last one in the current AHB burst, for any burst type except INCR.
- AHB transmit buffer is full after the current beat data is buffered.
- AHB bus becomes idle or a new burst arrives with the AHB transmit buffer not empty.

Table 192 describes the details of a triggered AHB write command.

Table 192. Triggered AHB write command

Item	Description
Flash memory access start address	Determined by AHB burst address. FlexSPI records the start address for the data in AHB transmit buffer, which is used as flash memory access start address.
Flash memory chip select	FlexSPI uses the settings of the flash memory access start address and flash memory size to determine the chip selection.
Flash memory command sequence index	Determined by FLSHxCR2[AWRSEQID].
Flash memory command sequence number	Determined by FLSHxCR2[AWRSEQNUM]. If AWRSEQNUM is not zero, multiple flash memory access command sequences are triggered every time for an AHB write command. The sequences indexed from AWRSEQID to (AWRSEQID + AWRSEQNUM) in LUT are executed sequentially.
Flash memory access mode (Individual)	
Flash memory data size	Determined by byte number of buffered data in AHB transmit buffer.

The following examples indicate internal logic for AHB write access to flash memory, where AHB_TX_BUF is 64 bytes :

- AHB INCR 64-bit bufferable burst with address sequence 8h, 10h, 18h, 20h, ..., 50h (10 beats total).

  Two AHB write commands are triggered: the first command with flash memory start address 8h and data size 40h; the second command with flash memory start address 48h and data size 10h. See Figure 102.

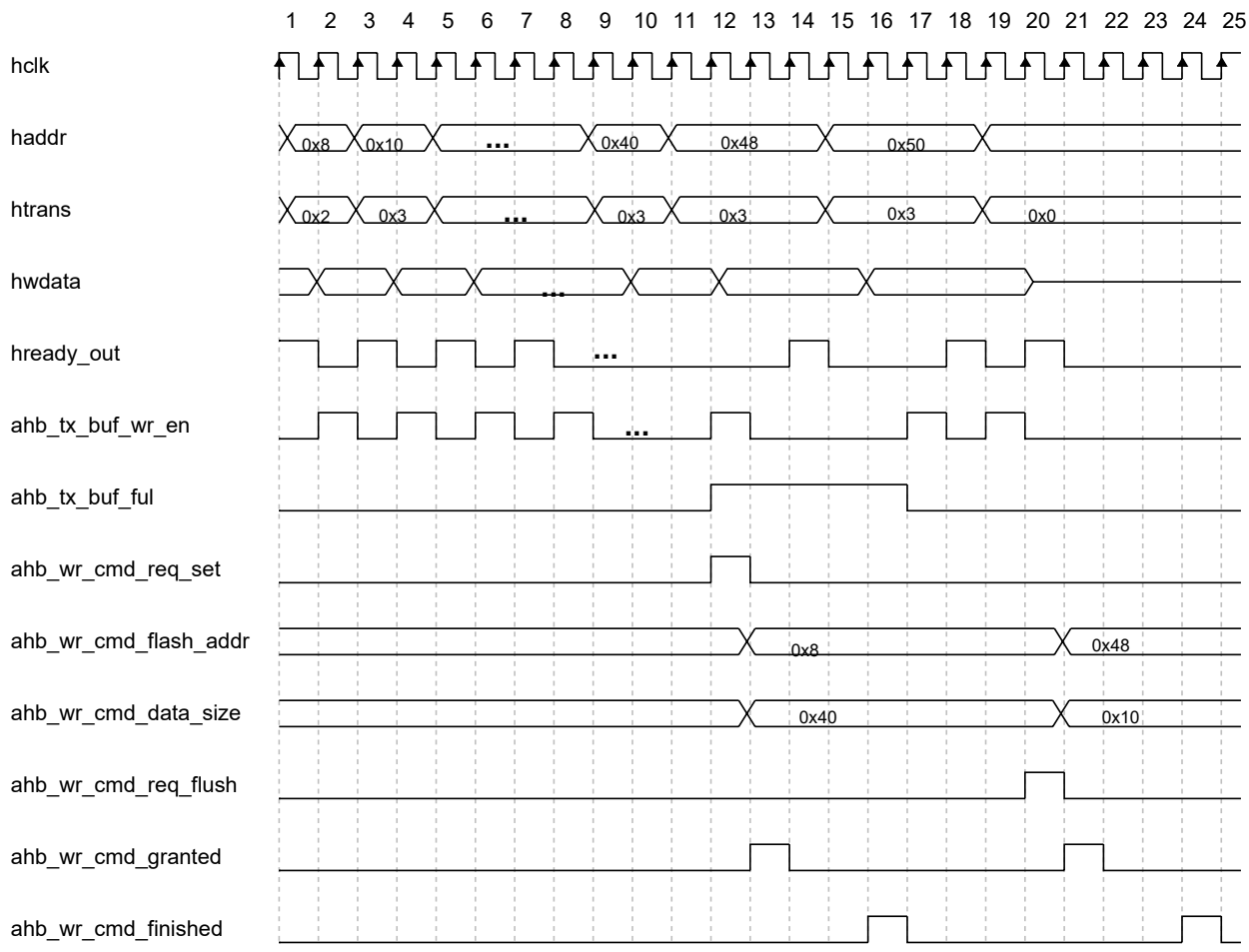


Figure 102. AHB write access (INCR, 64-bit, bufferable)

- AHB WRAP8 64-bit bufferable burst with address sequence 28h, 30h, 38h, 0h, 8h, 10h, 18h, 20h.

  One AHB write command is triggered with flash memory start address 0h and data size 40h. See Figure 103.

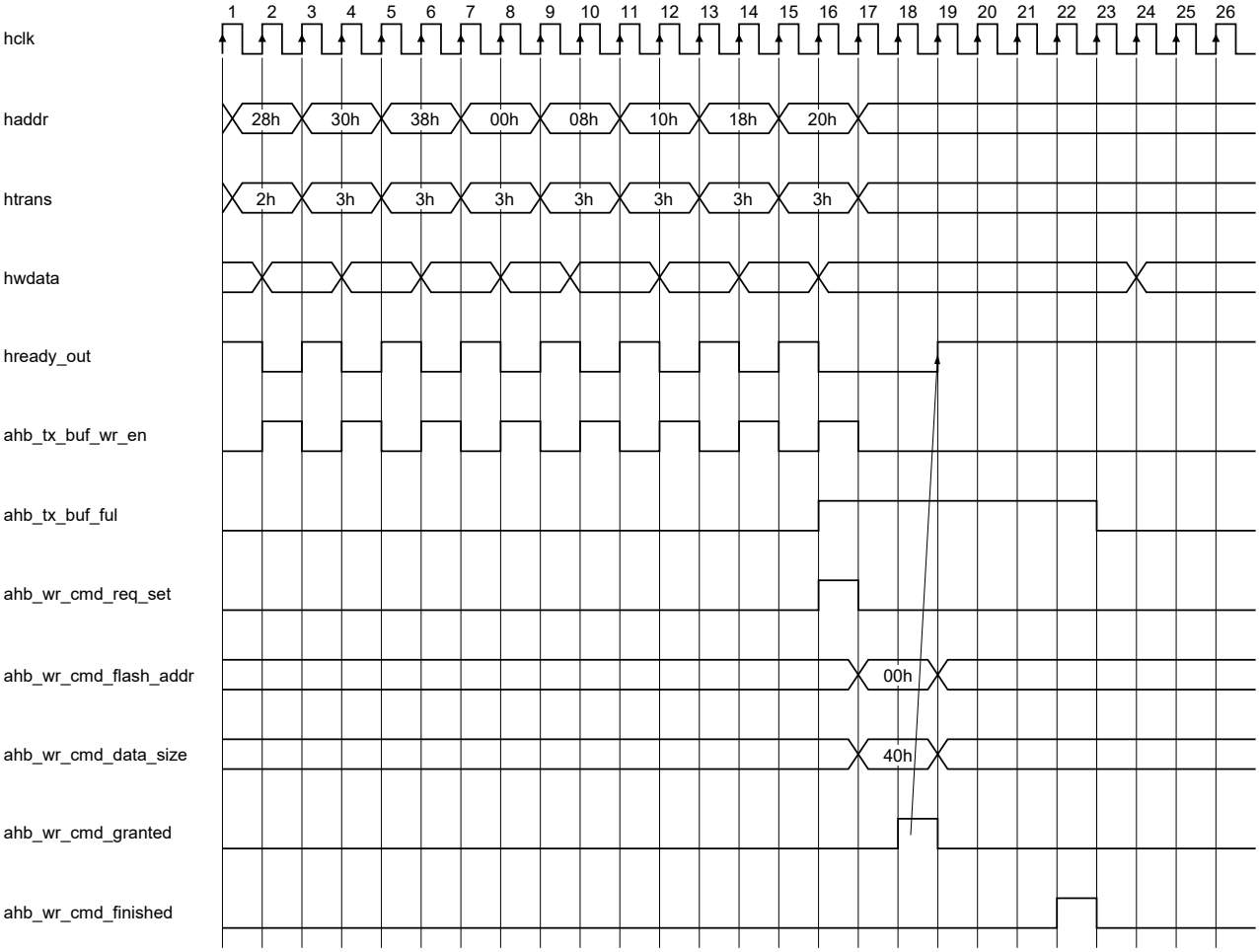


Figure 103. AHB write access (WRAP8, 64-bit, bufferable)

- AHB WRAP8 64-bit non-bufferable burst with address sequence 28h, 30h, 38h, 0h, 8h, 10h, 18h, 20h.

  One AHB write command is triggered with flash memory start address 0h and data size 40h.

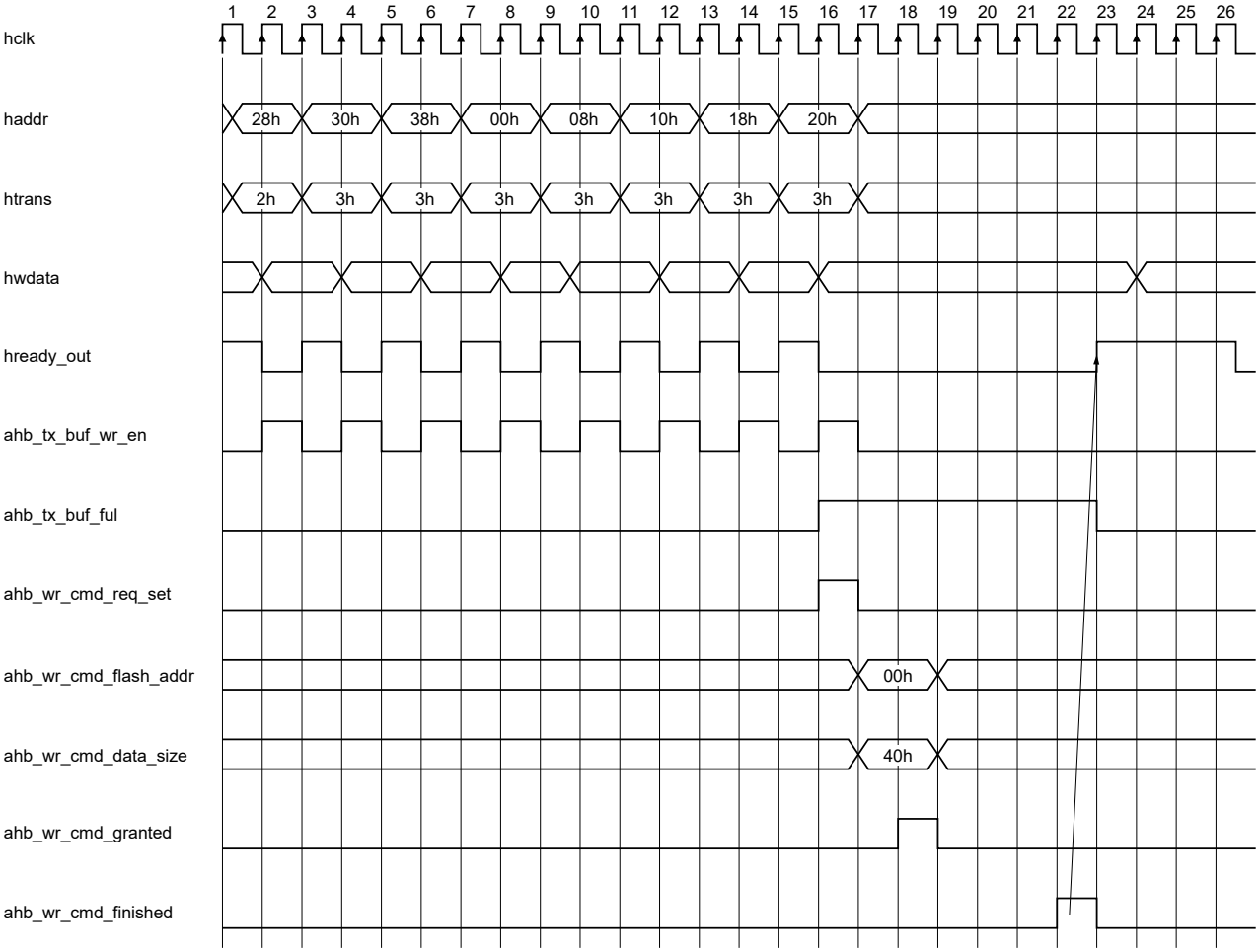


Figure 104. AHB write access (WRAP8, 64-bit, non-bufferable)

NOTE

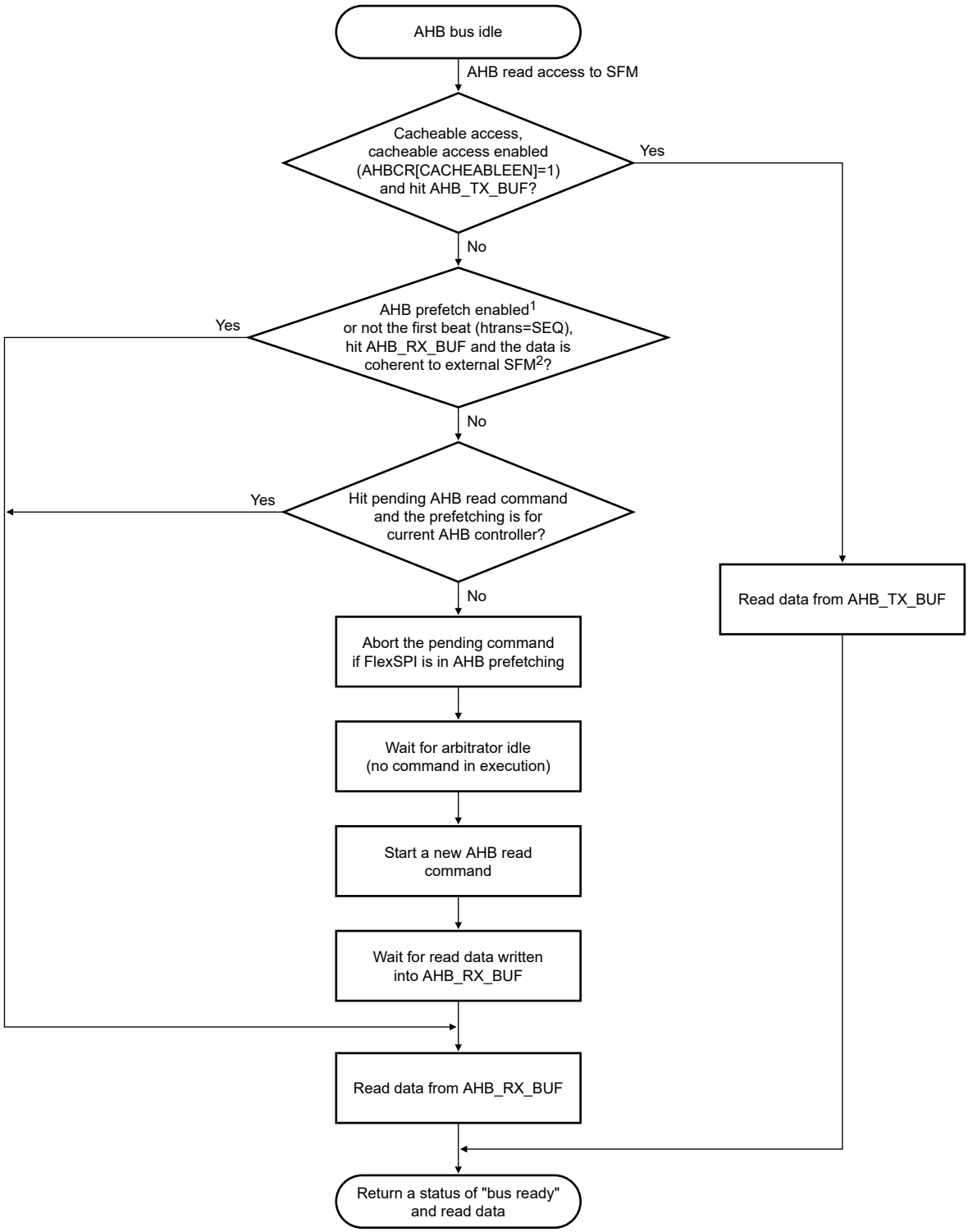
If the burst data size (in bytes) is greater than the AHB transmit buffer size, wrapper burst is not supported. For example, if AHB_TX_BUF is 64 bytes, AHB WRAP16 (16 × 64 bits = 128 bytes) write burst access is not supported.

36.4.11.2 AHB read access to flash memory

For AHB read access to flash memory, FlexSPI checks the burst access type and register setting to determine whether to access these locations:

- The AHB transmit buffer
- The AHB receive buffer
- A pending AHB read command

If all of these options are missed, FlexSPI triggers a new AHB read command to fetch data from flash memory. Figure 105 shows the hardware operation in response to AHB read access to flash memory.



Notes:
1) AHB prefetch is enabled when both AHBCR[PREFETCHEN]=1 and AHBRXBUFxCR0[PREFETCHEN]=1 (x is the AHB RX buffer ID for the current AHB controller) or OTFAD is enabled.
2) The AHB RX buffer holds the data read from flash memory. This data may become incoherent if AHB writes to the same flash-memory address.

Figure 105. Hardware operation in response to AHB read access to flash memory

Table 193 describes the details of a triggered AHB read command.

Table 193. Triggered AHB read command

Item	Description
Flash memory access start address and data size	Determined by AHB address, burst type, and burst size. FlexSPI fetches read data from the start address for the current burst or beat. See Table 194 for details.
Flash memory chip select	FlexSPI uses the settings of the flash memory access start address and flash memory size to determine the chip select.
Flash memory command sequence index	Determined by FLSHxCR2[AWRSEQID].
Flash memory command sequence number	Determined by FLSHxCR2[AWRSEQNUM]. If AWRSEQNUM is not zero, multiple flash memory access command sequences are triggered every time for an AHB read command. The sequences indexed from AWRSEQID to (AWRSEQID + AWRSEQNUM) in LUT are executed sequentially.
Flash memory access mode (Individual)	

NOTE

- FlexSPI automatically determines which ARDSEQNUM and ARDSEQID fields that the flash device uses as sequence ID for chip selection. See MCR2[SAMEDEVICEEN].
- FlexSPI determines which AHB receive buffer to use for the current AHB read access by controller ID. For details about the AHB receive buffer ID and AHB controller ID mapping, see AHB receive buffer management.
- You cannot allocate the size of the AHB receive buffer to be less than the AHB Burst size. This allocation results in unknown behavior.

Table 194. AHB read command flash memory start address and data size

Prefetch enable	Cross flash boundary	Burst type	Flash memory start address [26:0]	Data size
0	Never cross flash boundary, because AHB burst never crosses 1 KB boundary.	SINGLE, INCR4, INCR8, INCR16	hbeat_start_address	(hburst_end_address-hbeat_start_address)
		INCR	hbeat_start_address	byte size of current beat For INCR burst with prefetch disabled, each beat is handled same as SINGLE burst.
		WRAP4, WRAP8, WRAP16	hburst_start_address	(hburst_end_address-hburst_start_address)
1	No	INCR, SINGLE, INCR4, INCR8, INCR16	hbeat_start_address	ahb_rx_buf_sz
	No	WRAP4, WRAP8, WRAP16	hburst_start_address	ahb_rx_buf_sz

Table continues on the next page...

Table 194. AHB read command flash memory start address and data size (continued)

Prefetch enable	Cross flash boundary	Burst type	Flash memory start address [26:0]	Data size
	Yes	INCR, SINGLE, INCR4, INCR8, INCR16	hbeat_start_address	(flash_top_address-hbeat_start_address)
	Yes	WRAP4, WRAP8, WRAP16	hburst_start_address	(flash_top_address-hburst_start_address)

NOTE

- hbeat_start_address is HADDR input from the AHB controller for current beat.
- hburst_start_address (for example 00h) is the lowest address for current burst. hburst_end_address (for example 10h) is the highest address in current burst plus 1. For example, WRAP4 burst with HADDR 8h, Ch, 0h, 4h.
- ahb_rx_buf_sz is the buffer size in bytes of AHB receive buffer that the current AHB controller uses.
- flash_top_address is the top address of currently accessed flash memory.

36.4.11.3 AHB receive buffer management

There are 8 buffers (Buffer 0–7) in the AHB receive buffer. These buffers are transparent to AHB controllers. FlexSPI fetches flash memory data and returns it on the AHB bus automatically. No status register polling is needed for AHB read access to Serial Flash Memory (SFM).

See the chip-specific section for AHB buffer size.Use AHBRXBUF0CR0[BUFSZ]–AHBRXBUFCR7[BUFSZ] to configure the buffer size of each AHB receive buffer. The buffer size for Buffer 0 to Buffer 7 could be set to 0. If the buffer size is set to 0, FlexSPI ignores the related setting of AHBRXBUFxCR0[MSTRID]. Buffer 7 is used for all AHB controllers not assigned to Buffer 0–6. FlexSPI ignores the Buffer 7 size setting field (AHBRXBUF7CR0[BUFSZ]). Its buffer size is: (AHB receive buffer total size) - (the sum of the sizes of Buffer 0–6).

When an AHB read access to serial flash memory occurs, FlexSPI determines which buffer to use via this method:

1. If the controller ID equals AHBRXBUF0CR0[MSTRID] and AHBRXBUF0CR0[BUFSZ] is not zero, Buffer 0 is used.
2. If the controller ID equals AHBRXBUF1CR0[MSTRID] and AHBRXBUF1CR0[BUFSZ] is not zero, Buffer 1 is used.
3. If the controller ID equals AHBRXBUF2CR0[MSTRID] and AHBRXBUF2CR0[BUFSZ] is not zero, Buffer 2 is used.
4. If the controller ID equals AHBRXBUF3CR0[MSTRID] and AHBRXBUF3CR0[BUFSZ] is not zero, Buffer 3 is used.
5. If the controller ID equals AHBRXBUF4CR0[MSTRID] and AHBRXBUF4CR0[BUFSZ] is not zero, Buffer 4 is used.
6. If the controller ID equals AHBRXBUF5CR0[MSTRID] and AHBRXBUF5CR0[BUFSZ] is not zero, Buffer 5 is used.
7. If the controller ID equals AHBRXBUF6CR0[MSTRID] and AHBRXBUF6CR0[BUFSZ] is not zero, Buffer 6 is used.
8. If all the above cases are not met, Buffer 7 is used.

NOTE

- Software must ensure that the size of each buffer is not less than the maximum burst size of an AHB read access from the AHB controller using this buffer. Otherwise, the behavior is undefined. The minimum buffer size request depends on the chip implementation. For example, if the capability of the chip to send out a maximum burst is 64 bytes, the minimum AHB receive buffer is 64 bytes.
- Assigning multiple buffers to a single AHB controller is not supported, unless the prefetch buffer enhancement feature is being used.
- When AHB read prefetch is enabled (AHBCR[PREFETCHEN] = 1), the size of the buffer determines the prefetch data size. If it does not cross the flash boundary, FlexSPI fetches data from the external flash device with that buffer size.
- The priority setting of AHB controller (AHBRXBUFxCR0[PRIORITY] = 1) is used only for suspending control of AHB prefetching. See Command abort and suspend.

36.4.12 Command arbitration

There are four flash memory access command sources:

1. AHB command, that AHB write access to serial flash memory space triggers
2. AHB command, that AHB read access to serial flash memory space triggers
3. IP command, that writing IPCMD[TRG] triggers
4. Suspended command, an AHB read prefetch sequence which is suspended

An AHB bus access never triggers a write command and a read command at the same time.

The AHB prefetch sequence is an AHB command sequence triggered via AHB read access when AHB prefetch is enabled. After all read data is fetched for the current AHB read burst, FlexSPI prefetches more data to reduce read latency for the next AHB read access. An AHB command for a read operation is never aborted while fetching read data for the current AHB read burst. However, a new IP command or AHB command request while prefetching data (not for the current read burst) can abort an AHB read command.

When the arbitrator is idle (STS0[ARBIDLE] = 1), the granted priority of these command sources is:

1. AHB command (read or write)
2. IP command
3. Suspended command

NOTE

When the arbitrator is idle and there is no AHB or IP command request, a suspended command is not granted immediately. The arbitrator waits MCR2[RESUMEWAIT] AHB clock cycle idle states before resuming the suspended command to avoid AHB prefetch sequence being resumed and suspended frequently.

If the arbitrator is busy executing AHB or IP commands (not suspended commands), no new command requests are granted. If the grant times out, an AHB or IP command granted error occurs.

If a new AHB or IP command request comes while arbitrator is executing AHB read prefetch sequence, AHB read prefetch sequence is aborted (but not immediately). Arbitrator grants the AHB or IP command request after AHB read prefetch sequence is aborted on the FlexSPI interface and saved all internal data pointers.

36.4.12.1 Command abort and suspend

Table 195 describes the command abort and suspend mechanism.

Table 195. Command abort and suspend

Action	Description
Aborting a command	As mentioned above, if a new AHB or IP command request arrives, an AHB read prefetch sequence can be aborted.
Suspending a command	When an AHB read prefetch sequence is aborted on the FlexSPI interface, FlexSPI saves this suspended sequence in the cases listed below. FlexSPI resumes this sequence after the arbitrator has been idle for enough time. • There is no valid suspended command (AHBSPNDSTS[ACTIVE] = 0). This condition can occur when no command is suspended or when the suspended command is resumed. • The priority of aborted AHB read prefetch sequence is higher than the active suspend sequence. **NOTE** FlexSPI ignores and never resumes the original suspended sequence.
Activating a suspended command	The suspended command (internal status) becomes active when an AHB prefetch command is aborted and suspended. It becomes inactive in the following cases: • The arbitrator resumes the suspended command. • The AHB controller triggers a new AHB read command request using the same AHB receive buffer (buffer ID).

Figure 106 shows an example of the flow of command abort, suspend, or resume.

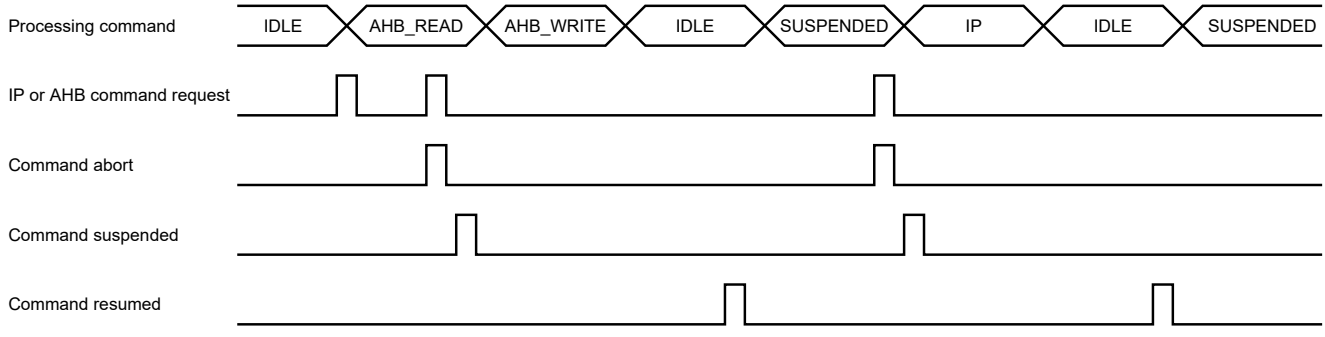


Figure 106. Command instruction execution on FlexSPI interface

NOTE

- The AHB prefetch sequence is aborted, suspended, or resumed twice in Figure 106. A new AHB write command request aborts the sequence the first time, and a new IP command request aborts the sequence the second time.
- The suspended sequence is resumed after a wait time of idle state. If there are frequent IP or AHB command requests and the wait time is not set reasonably, the suspend and resume activities may be triggered frequently.

36.4.13 SCLK stop

FlexSPI stops SCLK output toggling:

- When programming data is not ready for a programming command sequence.
- When an internal FIFO has no space to receive data for reading command sequence.

Certain devices may not support the stopping of SCLK during a command sequence (chip select is valid). SCLK stopping can be avoided via the methods described below.

- For flash memory reading that an IP command triggers:
  - Never trigger a read command with data size larger than IP receive FIFO size.
  - Internal asynchronous FIFO for flash memory reading should never be full.

    FlexSPI pops 64 bits of data per AHB clock cycle from this FIFO, and receives data from FlexSPI interface on the serial root clock. The flash memory access mode (Single, Dual, Quad, or Octal mode and Individual mode) determines the receiving speed. For example, in Octal mode, FlexSPI receives 16 bits per serial root clock cycle. If the AHB clock frequency is faster than 1/4 of the serial root clock, this asynchronous FIFO is never full.
- For flash memory programming that an IP command triggers:
  - Never trigger a program command with data size larger than the IP transmit FIFO size. Write all programming data into the IP transmit FIFO before triggering the IP command.
  - The internal asynchronous FIFO for flash memory programming must never be empty.

    FlexSPI fetches 64 bits of programming data per AHB clock cycle into this FIFO, and transmits data to the FlexSPI interface on the serial root clock. The flash memory access mode (Single, Dual, Quad, or Octal mode and Individual mode) determines the transmitting speed. For example, in Octal mode, FlexSPI transmits 16 bits per serial root clock cycle. If the AHB clock frequency is faster than 1/4 of the serial root clock, this asynchronous FIFO is never empty.
- For flash memory reading or programming that an AHB command triggers:
  - The internal asynchronous FIFO for flash memory reading and programming must never be full or empty. The frequency ratio limitation is same as it is for flash memory reading or flash memory programming that an IP command triggers.

NOTE

- FlexSPI never triggers an AHB read command with data size larger than the internal AHB receive buffer size.
- FlexSPI never triggers an AHB program command with data size larger than the internal AHB transmit buffer size.
- All programming data is buffered into the AHB transmit buffer before triggering an AHB program command in FlexSPI.

36.4.14 FlexSPI output timing

36.4.14.1 Output timing between data and SCLK

This section describes the output timing relationship of data (on A_DATA and B_DATA) and SCLK. There are three cases.

- **SDR instruction in SDR sequence**

  An SDR sequence contains only SDR instructions. In this case, all data bits last one serial root clock cycle on the FlexSPI interface. Figure 107 shows the relationship between the serial root clock, data, and SCLK:

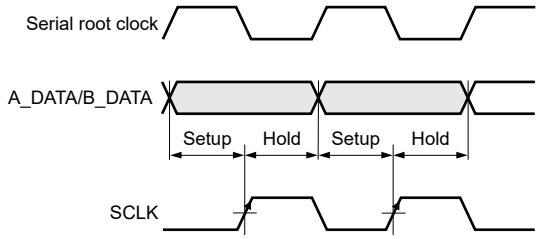


Figure 107. SDR instruction in SDR sequence

- **SDR instructions in a DDR sequence**

  A DDR sequence is a flash memory access command sequence that contains DDR instructions other than DUMMY_DDR. It may contain SDR instructions. If there are SDR instructions in a DDR sequence, all data bits last for two serial root clock cycles on the FlexSPI interface. Figure 108 shows the relationships between the serial root clock, data, and SCLK.

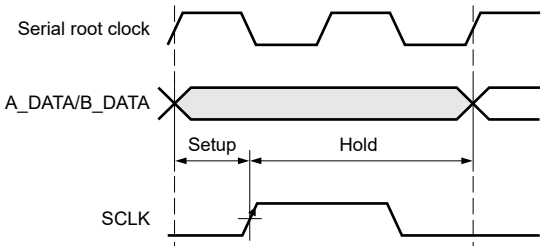


Figure 108. SDR instruction in DDR sequence

- **DDR instructions in a DDR sequence**

  For DDR instructions in a DDR sequence, all data bits last one serial root clock cycle on the FlexSPI interface. Figure 109 shows the relationships between the serial root clock, data, and SCLK.

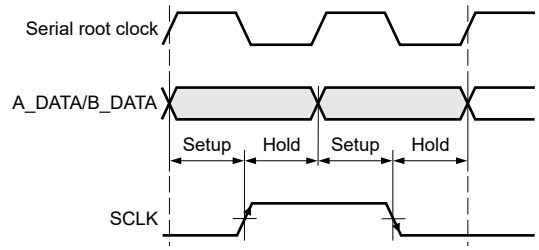


Figure 109. DDR instruction in DDR sequence

36.4.14.2 Output timing between chip select and SCLK

This section describes the output timing relationship between chip select (on A_SS0_B , A_SS1_B, B_SS0_B, and B_SS1_B) and SCLK. The timing relationship for the SDR sequence differs a little from the timing relationship for the DDR sequence.

- **Chip select timing in SDR sequence**

  For an SDR sequence, the delay from chip select assertion to the SCLK rising edge is (FLSHxCR1[TCSS] + 0.5) cycles of the serial root clock. The delay from SCLK falling edge to chip select deassertion is FLSHxCR1[TCSH] cycles of serial root clock. Figure 110 shows the timing relationship between the chip select and SCLK.

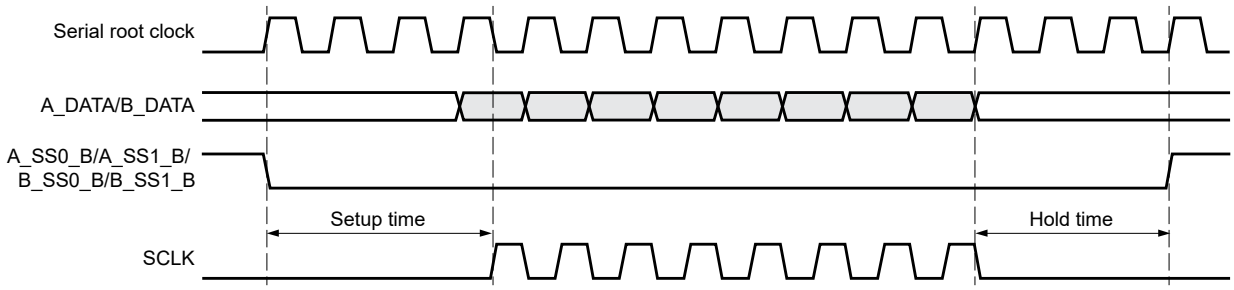


Figure 110. Chip select output timing for SDR sequence

- **Chip select timing in DDR sequence**

  For a DDR sequence, the delay from chip select assertion to the SCLK rising edge is **(FLSHxCR1[TCSS] + 0.5)** cycles of the serial root clock. The delay from the SCLK falling edge to chip select deassertion is **(FLSHxCR1[TCSH] + 0.5)** cycles of serial root clock.

  NOTE

  When AHB receive prefetch is enabled, and prefetch can be aborted, configure TCSH to 1 or greater. Configuring TCSH in this way guarantees a positive chip select hold time after the SCLK falling edge.

  Figure 111 shows the timing relationship between chip select and SCLK.

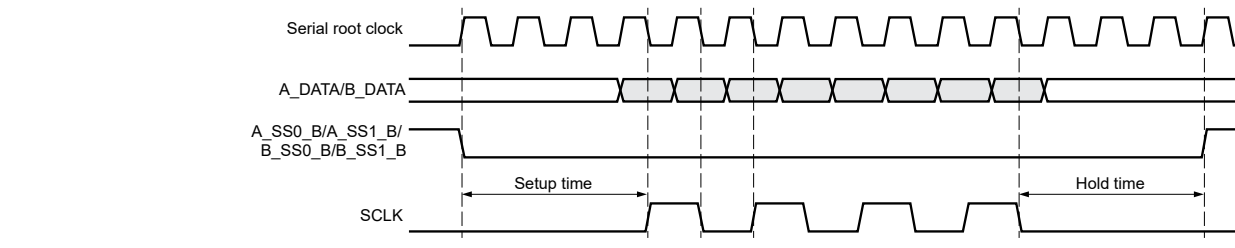


Figure 111. Chip select output timing for DDR sequence

For devices such as an FPGA device, a minimum chip select negation time (time between two chip select assertions) may be required. If the value of FLSHxCR1[CSINTERVAL] is nonzero, FlexSPI ensures a delay between chip select valid assertions. The delay time is CSINTERVAL × 256 cycles of serial root clock for an SDR or DDR sequence. If the external device has no limitation, configure the value of this field to zero. Figure 112 shows the timing of the chip select interval.

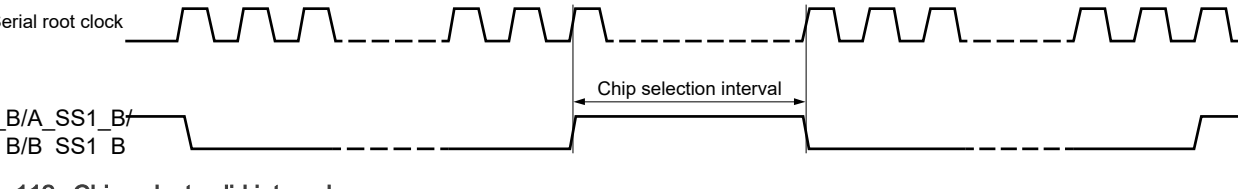


Figure 112. Chip select valid interval

36.4.15 FlexSPI input timing

This section describes the input timing of FlexSPI.

36.4.15.1 Receiving block

This section describes the receiving block in FlexSPI.

FlexSPI decodes instruction code to determine SDR mode or DDR mode. It also determines whether both the rising and falling edges are used for flash data sampling. FlexSPI samples the data line for learn instructions and read instructions.

MCR0[RXCLKSRC] selects the clock source from these four sources for the sampling clock:

- Internal dummy read strobe and looped back internally
- Internal dummy read strobe and looped back from DQS pad
- Flash-memory-provided read strobe

For details about DLL configuration, see DLL configuration for sampling.

Figure 113 shows the sampling block in FlexSPI.

FlexSPI uses both rising and falling edges to sample data from flash_A and flash_B. Two muxes with four clock source inputs generate the signals ipp_ind_io_fa, ipp_ind_io_fb. See Receive clock source features. ipp_ind_io_fa_int and ipp_ind_io_fb_int are reversed from ipp_ind_io_fa and ipp_ind_io_fb. Both sampling flip-flops are used when FlexSPI is in DDR mode. Only the falling edge sampling flip-flop is used in SDR mode.

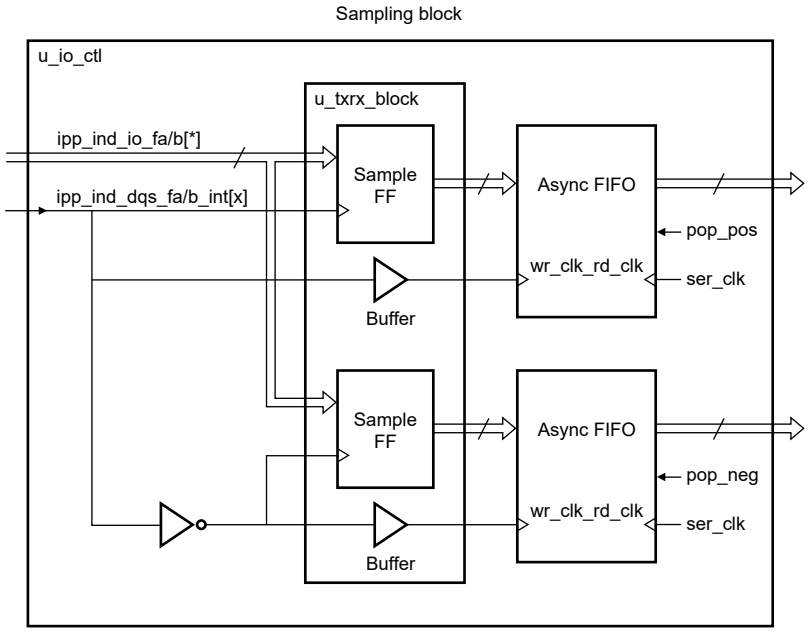


Figure 113. Sampling block in FlexSPI

Figure 114 is the receiving block diagram for data learning.

There are 16 clock phases for sampling data. An internal delay chain delays the clock phases.

See Data learning.

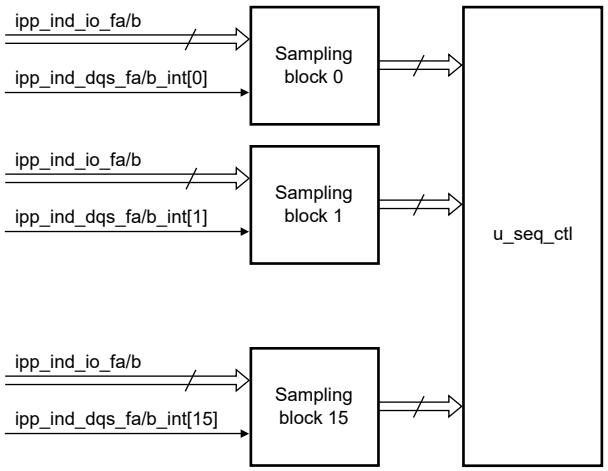


Figure 114. Sampling blocks for data learning

FlexSPI does not support read operations with a zero-turnaround cycle for direction switching on I/O pins. This read operation often occurs in QuadSPI configuration register reads. In Figure 115 and Figure 116, SIO[3:0] is an output from FlexSPI in the CMD phase, and is an input to FlexSPI in the DATA phase. No time remains for FlexSPI to switch the direction. This kind of read is not supported. FlexSPI and flash memory drive the I/O pins at the same time, which may cause a problem. To avoid this behavior on I/O pins, using SPI mode or reading with dummy cycles is required.

NOTE

The cycle numbers are not accurate in the examples below. See the device spec for accurate cycle numbers.

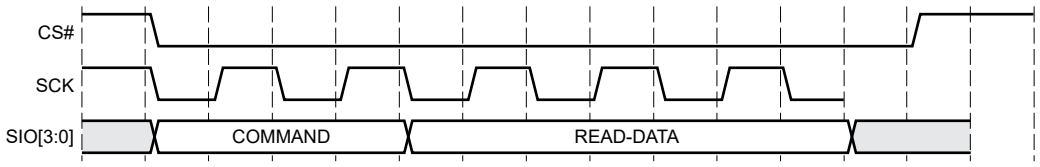


Figure 115. Unsupported read behavior

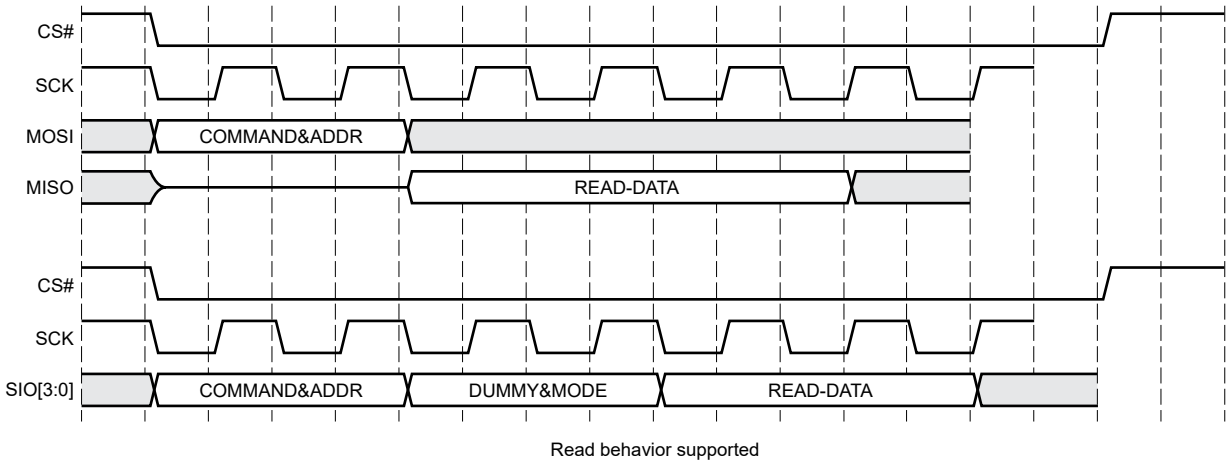


Figure 116. Supported read behavior

36.4.15.2 Receive clock source features

FlexSPI uses a read strobe to sample incoming data. There are several options for generating the read clock strobe that MCR0[RXCLKSRC] selects. Using the receive clock source method has implications for the board connections and can affect the maximum frequency that the interface supports. See the device data sheet for details on the maximum frequency supported for each RXCLKSRC option.

Table 196. RXCLKSRC options

MCR0[RXCLKSRC]	Description	Board connection	Max Frequency
0	Internal dummy read strobe and internal loopback	DQS pin is not used. The DQS pin can be configured for an alternate signal function.	Lowest
1	Internal dummy read strobe and loopback from DQS pad	FlexSPI uses the DQS pin, and it must be configured for its FlexSPI function. The internally generated read strobe is sent to the DQS pin and is sampled at the pin to match more closely the data pin timings. The DQS pin is typically left floating. External capacitance can be added to adjust timing.	Medium
3	Flash-memory-provided read strobe	DQS pin is connected to the DQS or RWDS signal that the memory provides.	Highest

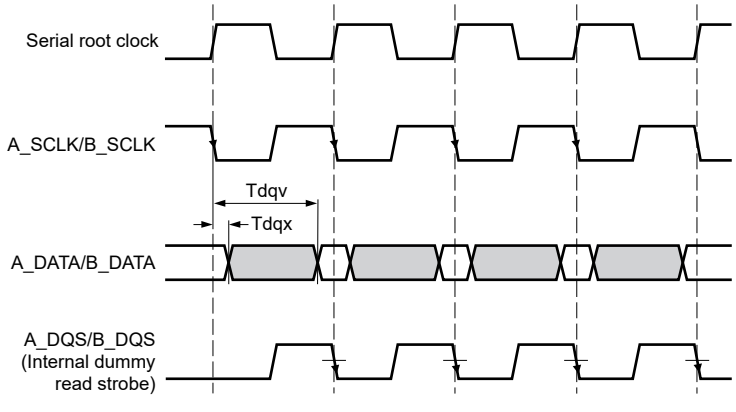
36.4.15.3 Input timing for sampling with dummy read strobe

This section describes the input timing when sampling with the internal dummy read strobe (MCR0[RXCLKSRC] = 0 or 1). The timing for sampling with an internal dummy read strobe loopback is very similar to the timing for loopback from pad. Sampling with a dummy read strobe loopback from the DQS pad can achieve a higher read frequency. It compensates for the delay of the SCLK output path and data pin input path.

The input timing is different for SDR mode and DDR mode.

- **Input timing for sampling with dummy read strobe in SDR mode**

  For SDR read or learn instructions, FlexSPI samples input data pins with the falling edge of the dummy read strobe. Figure 117 shows the input timing for sampling with dummy read strobe in SDR mode.



Tdqv is the setup time from SCLK to the next valid data.
Tdqx is the hold time from SCLK to the current valid data.

Figure 117. Input timing for sampling with dummy read strobe in SDR mode

- **Input timing for sampling with dummy read strobe in DDR mode**

  For DDR read or learn instructions, FlexSPI samples input data pins on rising and falling edges of the dummy read strobe. Figure 118 shows the input timing for sampling with dummy read strobe in DDR mode.

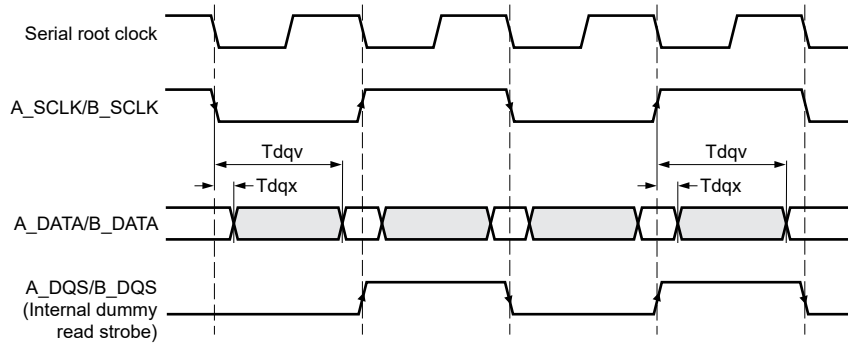


Figure 118. Input timing for sampling with dummy read strobe in DDR mode

36.4.15.4 Input timing for sampling with flash-memory-provided read strobe

This section describes the input timing when sampling with flash-memory-provided read strobe (MCR0[RXCLKSRC] = 3). The input timing is different for SDR mode and DDR mode.

NOTE

There are no known devices that provide a read strobe and support SDR mode operation.

- **Flash-memory-provided read strobe with SCLK**

  Certain flash devices provide both read data and read strobes with SCLK. Then the read strobe edge is aligned with the read data change. The FlexSPI controller delays the read strobe for one half cycle of the serial root clock (with DLL), then samples read data with the delayed strobe. See DLL configuration for sampling. Figure 119 and Figure 120 show the input timing for sampling with flash memory read strobe in SDR mode and DDR mode.

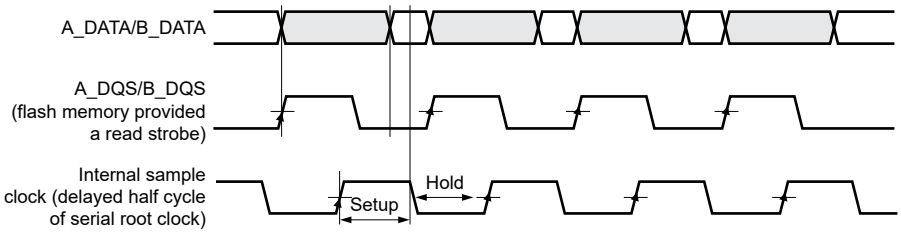


Figure 119. Input timing 2 for flash-memory-provided read strobe in SDR mode

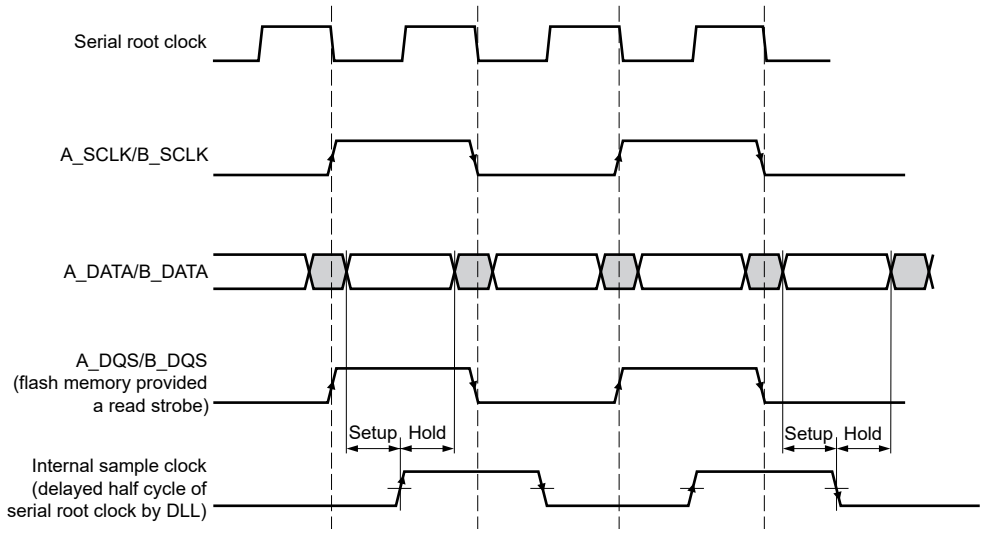


Figure 120. Input timing 2 for flash-memory-provided read strobe in DDR mode

36.4.15.5 DLL configuration for sampling

For the four sampling clock sources described previously, the input timing differs depending on the DLLxCR register configuration, according to the sampling clock source mode. The DLL is a delay-line chain that can be set to a fixed number of delay cells. It can also be auto-adjusted to lock on a certain phase delay to the reference clock.

- DLLxCR must be set to 0000_0100h (one fixed delay cell in DLL delay chain) in the following cases:
  - Sampling data with dummy read strobe looped back internally (MCR0[RXCLKSRC] = 0h)
  - Sampling data with dummy read strobe looped back from DQS pad (MCR0[RXCLKSRC] = 1h)
- When data is sampled with a flash-memory-provided read strobe (MCR0[RXCLKSRC] = 3h) and flash memory provides the read strobe with SCLK, DLL must be configured as follows (when serial root clock is over 100 Mhz. These settings ensure a lock of the reference clock (serial root clock).
  - DLLxCR[SLVDLYTARGET] is up to flash parameters used.
  - DLLxCR[REFPHASEGAP] = 2h
  - DLLxCR[DLLEN] = 1h
  - DLLxCR[OVRDEN] = 0h
  - Other fields in DLLxCR must be kept at their reset values (all zeroes).

    After the register settings above are completed, set the DLL with following steps.

    1. Reset the DLL(DLLxCR[DLLRESET] or MCR0[SWRESET]).
    2. Wait for STS2[xREFLOCK] and STS2[xSLVLOCK] being asserted.
    3. Wait for equal or more than 512 serial root clock cycles.
    4. Start read or write operation.

NOTE

If the serial root clock is slower than 100 MHz, DLL cannot lock on a half cycle of serial root clock. The delay cell number is limited in the delay chain. DLL must be configured as follows instead:

- DLLxCR[OVRDEN] = 1
- DLLxCR[OVRDVAL] = N

  Each delay cell in DLL is about 75 ps to 225 ps. The delay of DLL delay chain is (N × Delay_cell_delay). N is set based on the maximum DDR frequency (N ≤ equation). This value is calculated based on the MAX_DDR_FREQ parameter. N = 21. This value is a recommended value. If a failure occurs, the value might require adjustment in a real application.

  You can calculate the delay of one delay cell and obtain the value of N. If the serial root clock is lower than 100 MHz, the delay chain can't lock with DLLxCR[SLVDLYTARGET] = 0xF (half cycle of root clock delay). You need to set DLLxCR[SLVDLYTARGET] as 0xE (15/32 cycle delay) or less. When DLL is locked (read STS2[AREFLOCK] and STS2[ASLVLOCK]), the delay cell number = STS2[ASLVSEL] +1 and the delay value (one delay cell) = (total delay)/(delay cell number). Based on delay value, the N can be easily calculated.
- Other fields in DLLxCR must be kept at their reset values (all zeroes).

36.4.16 Data learning

The FlexSPI controller generates 16 clock phases (Phase 0–15) via delay cell line (DCL) with the selected sample clock. Clock phase 0 is the selected sample clock (DQS_IN) with no delay cell. 16 sampling blocks are implemented for Port A.

During a learn instruction, FlexSPI compares the sampled data bits with the internal data learn pattern in the Data Learning Pattern (DLPR) register and determines the correct sampling clock phase. The phase selection is automatically updated after a learn instruction and applied to subsequent read instructions or sequences.

When data learning is disabled (MCR0[LEARNEN] = 0), FlexSPI always uses clock phase 0 to sample FlexSPI data lines. Attempting to execute a learn instruction results in an error flag (INTR[IPCMDERR] or INTR[AHBCMDERR]).

FlexSPI supports data learning in both SDR mode and DDR mode. Data learning is not supported if the sampling clock source is a flash-memory-provided read strobe (MCR0[RXCLKSRC] = 3h).

Data learning can achieve high read frequency, but the flash memory input timing still limits the highest frequency. Flash memory must receive the command code and flash address bits.

When data learning is enabled, FlexSPI uses clock phase 0 after reset and before executing a learn instruction. The clock phase selection is updated after FlexSPI executes the learn instruction. The selected sample clock phase can be polled via STS0[DATALEARNPHASEA]. When data learning fails, the INTR[DATALEARNFAIL] interrupt flag is set, and the previous clock phase selection is kept. You can write 1 to MCR2[CLRLEARNPHASE] to reset the internal clock phase selection to clock phase 0.

36.4.16.1 Data learning with flash-memory-provided preamble bit

Certain flash devices support driving with preamble bits before driving read data in each read command sequence. Preamble bits are also called the data-learning pattern (DLP). The DLP is programmable via configuration register in the flash device.

For the specified read command sequence, flash memory returns preamble bits after dummy cycles and before returning read data.

For these flash devices, the operation flow with data learning is:

1. Configure the data learning pattern in Data Learning Pattern (DLPR) register.
2. Configure the same data learning pattern in the flash device by triggering an IP command.
3. Enable data learning in the flash device by triggering an IP command.
4. Configure LUT sequence with valid read command sequence containing the learn instruction.
5. Trigger a flash memory read command via AHB or IP command as normal.

NOTE

The first four steps are executed only once. They are not needed for every flash memory read command.

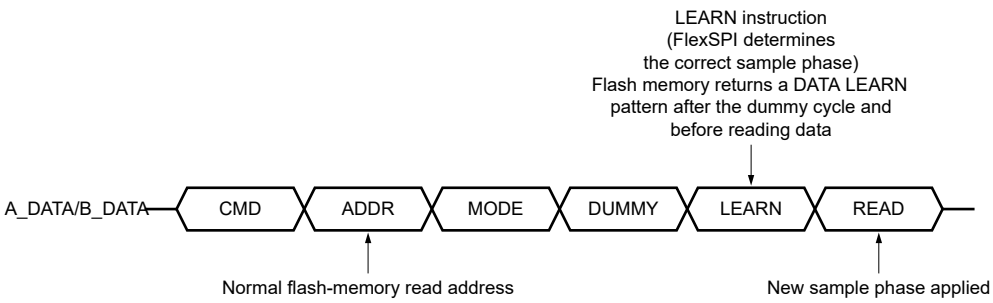


Figure 121. Data learning flow with flash-memory-provided preamble bit

36.4.16.2 Data learning without flash-memory-provided preamble bit

For flash devices that do not provide a preamble bit, the DLP cannot be returned automatically for each read command sequence. For these flash devices, the data learning sequence is:

1. Set data learning pattern in Data Learning Pattern (DLPR) register.
2. Reserve a certain flash memory area and program data (according to data learning pattern and flash memory access mode) to this area via IP command.
3. Configure LUT sequence with read sequence using a learn instruction instead of a read instruction. Trigger the read sequence to the reserved flash memory area via IP command.
4. Trigger the flash memory read command via AHB or IP command as normal. No learn instruction is needed in this read sequence.

NOTE

- The first three steps are executed only once. They are not needed for every flash memory read command.
- Step 4 should be executed with a certain interval to ensure that the internal sampling clock phase is adjusted in time.

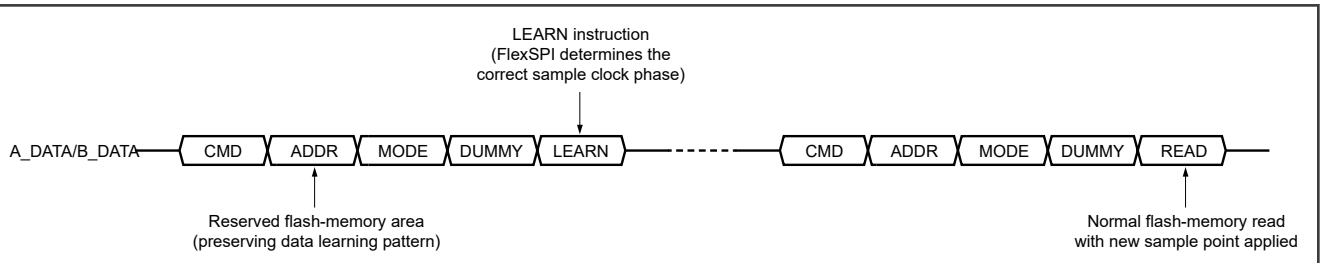


Figure 122. Data learning flow without flash-memory-provided preamble bit

NOTE

- For flash devices that do not provide a preamble bit, software overhead is required to read the reserved flash memory area at the interval.

These items determine the preprogramming data:

- Data Learning Pattern (DLPR register setting)
- DLP bit length

- Individual mode for read command
- Octal, Quad, Dual, or Single mode for read command

Figure 123 shows a preprogramming data example where the DLPR value is 43h, eight bits, and flash memory is read in Quad mode.

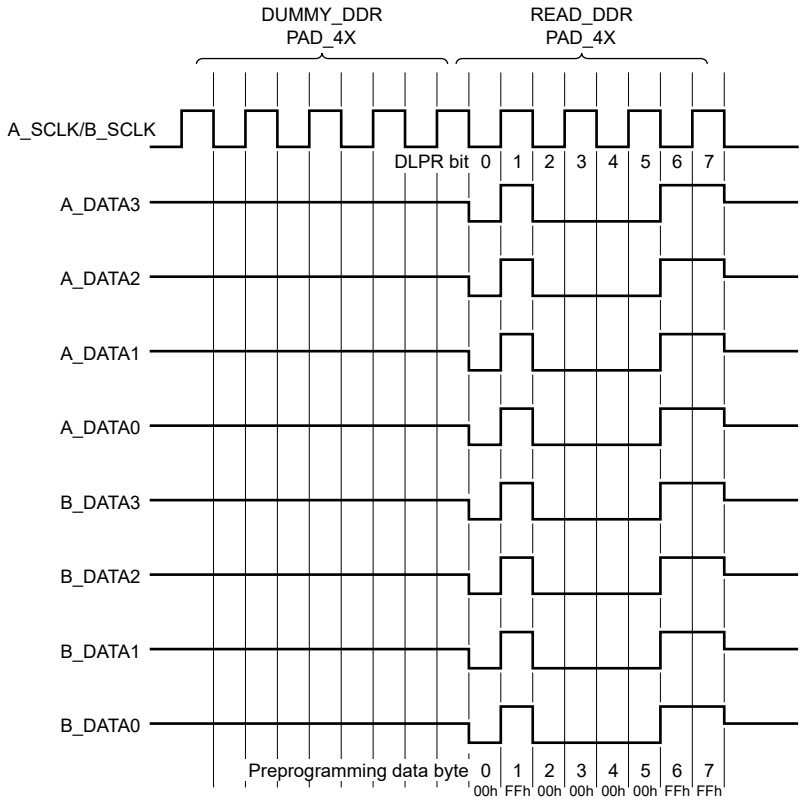


Figure 123. Preprogramming data for data learning

36.4.17 Execute-In-Place (XIP) enhanced mode

FlexSPI always supports XIP, regardless of whether external devices provide XIP enhanced mode. To support XIP, put program codes on an external device, then read or execute these codes directly via AHB read access to serial flash memory space. There is no configuration or status polling needed during AHB read access to the memory of the external device. The AHB receive buffer is fully transparent to software.

Certain devices provide XIP enhanced mode to improve code execution. In this mode, the command code is only sent in the first read instruction. It saves many cycles for command instructions and improves code execution. Devices enter or exit XIP enhanced mode via a special sequence that external devices specify. See the external device data sheet for more details.

Normally, a device enters XIP enhanced mode via this sequence:

1. Enable XIP enhanced mode in external flash device via IP command.
2. Send the first read sequence to external flash device with correct mode bits. Command code is included in this read sequence.
3. Send the subsequent read sequences to external flash device with correct mode bits. Command code is not included in these read sequences. Mode bits must be sent according to the flash specified; otherwise, the flash exits XIP enhanced mode.

The instruction JMP_ON_CS in FlexSPI must be used to support XIP enhanced mode in external devices. This instruction is only allowed in the AHB read command; otherwise, the instruction generates an IPCMDERR or AHBCDMERR interrupt (when the

interrupt is enabled). When external devices do not support XIP enhanced mode, JMP_ON_CS must never be used. To support XIP enhanced mode:

- The first instruction in the read sequence must be the command instruction.
- The last valid instruction must be JMP_ON_CS (with operand 1h).

For the first AHB read command triggered, FlexSPI executes the instructions from the instruction pointer 0 in the sequence (which is the command instruction). After this sequence is executed, FlexSPI saves the operand in the JMP_ON_CS instruction as the start pointer for the next command to current device internally. For the next AHB read command triggered, FlexSPI executes from the instruction pointer 1h, bypassing the command instruction.

After XIP enhanced mode is started, it is possible to access flash via other commands like program commands. You must, however, use a special sequence to exit enhanced XIP mode before you can run a different command. Figure 124 indicates XIP operation with Flash XIP enhanced mode.

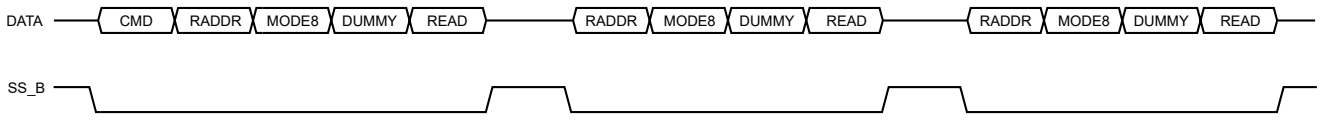


Figure 124. XIP Enhanced Mode Operation

36.4.18 AHB bus blocking

FlexSPI can temporarily block some or all AHB transactions. This feature is useful when IP commands are in progress that could cause a potential conflict with an AHB request. Examples of situations where IP bus blocking would be used are:

- Avoiding RWW (read while write). While programming or erasing flash, reads are not allowed. AHB bus blocking can prevent the AHB bus from requesting a read while IP write commands are in progress.

To block all AHB transactions:

1. Configure blocking for all AHB transactions by writing 1 to IPCR2[IPBLKALLAHB].
2. Request AHB bus blocking by writing 1 to IPCR2[IPBLKAHBREQ].
3. Wait for the AHB bus blocking request to be acknowledged (IPCR2[IPBLKAHBACK] = 1).
4. Perform IP write commands as needed.
5. Request normal operation by writing 0 to IPCR2[IPBLKAHBREQ].
6. Wait for AHB blocking request clear to be acknowledged (IPCR2[IPBLKAHBACK] = 0).

NOTE

AHB bus blocking can only be used to prevent conflicts between AHB and IP transactions. Software must handle conflicts between multiple AHB transactions.

36.5 External signals

This section provides the external signal information for the FlexSPI module.

Table 197. External Signal List

Signal name	Function	Direction	Description
A_SS0_B	Peripheral Chip Select Flash A1	O	Chip select for the serial flash device A1

Table continues on the next page...

Table 197. External Signal List (continued)

Signal name	Function	Direction	Description
A_SCLK	Serial Clock Flash A	O	Serial clock output to the serial flash device A. This clock runs at half the frequency of serial clock root in DDR mode, and at the same frequency as serial clock root in SDR mode. Clock output toggles during the entire flash memory access sequence.
A_DATAn	Serial I/O Flash A	I/O	Data I/O lines to and from the serial flash device A
A_DQS	Data Strobe Signal Flash A	I/O	Data strobe signal for port A. This signal has three functions: • Driven with read strobe by external device. Some flash devices provide the read strobe signal together with read data. In this case, if the external device drives this pad only when reading flash memory data, this pad may require a **pull-down** resistor. • Driven with latency information by external device. Some devices use this pin to indicate the dummy cycles needed (before program or read data transfer) such as HyperRAM or HyperFlash. • Loopback dummy read strobe. The FlexSPI controller provides internal dummy read strobe for flash memory read data. Higher read frequency can be achieved by looping back this dummy read strobe from the pad. This pin can be floated or have some capacitive load added at the board level to compensate for load on DATA or SCLK pins.

36.6 Initialization

The FlexSPI controller initialization sequence is:

1. Enable controller clocks (AHB clock, IP bus clock, or serial root clock) at system level.
2. To enter module stop mode, write 1 to MCR0[MDIS].
3. Configure module control registers: MCR0, MCR1, and MCR2. Do not change MCR0[MDIS].
4. When AHB commands are used, configure AHB Bus Control (AHBCR) and AHB Receive Buffer Control (AHBRXBUFxCR0) registers.
5. Configure flash control registers (FLSHxCR0, FLSHxCR1, or FLSHxCR2) according to external device type.
6. Configure DLL control register (DLLxCR) according to sample clock source selection.
7. To exit module stop mode, write 0 to MCR0[MDIS].
8. Configure LUT as needed for AHB command or IP command.

9. Optionally, to reset controller, write 1 to MCR0[SWRESET].

External devices must be configured via IP command normally after initialization. For example, the WRITE STATUS command performs device configuration for most serial NOR flash memory.

36.7 Application information

This section describes applications that FlexSPI supports.

36.7.1 Application on serial NOR flash device

This section provides the example LUT instruction sequences for serial NOR flash device (Cypress Flash S25FS128S).

36.7.1.1 Write enable command

Table 198 shows the WRITE ENABLE command sequence.

Table 198. WRITE ENABLE command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h	06h	Command name: WREN
1-7	STOP (00h)	0h	00h	

36.7.1.2 Write registers command

Table 199 shows the Write Registers command sequence.

Table 199. Write Registers command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	01h	Command name: WRR
1	WRITE_SDR	0h (Single)	1h or 2h	Data size is 1 byte or 2 bytes. Byte 0 is write data for Status Register 1; Byte 1 is write data for Configuration Register 1. IPCR1[IDATSZ] can override this value.
2-7	STOP (00h)	0h	00h	

36.7.1.3 Page Program command

Table 200 shows Page Program command sequence.

Table 200. PAGE PROGRAM command (Cypress serial NOR flash)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	02h or 12h	Command name: PP or 4PP PP is 3-byte address mode. 4PP is 4-byte address mode.

Table continues on the next page...

Table 200. PAGE PROGRAM command (Cypress serial NOR flash) (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
1	ADDR_SDR	0h (Single)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	WRITE_SDR	0h (Single)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default programming data size. This value is ignored for AHB command.
3-7	STOP (00h)	0h	00h	

Table 201 shows Page Program command sequence (QPI mode).

Table 201. PAGE PROGRAM (QPI mode) command (Cypress serial NOR flash)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	2h (Quad)	02h or 12h	Command name: PP or 4PP PP is 3-byte address mode. 4PP is 4-byte address mode.
1	ADDR_SDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	WRITE_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default programming data size. This value is ignored for AHB command.
3-7	STOP (00h)	0h	00h	

36.7.1.4 Read Status 1 command

Table 202 shows READ STATUS 1 command sequence.

Table 202. READ STATUS 1 command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	05h	Command name: RDSR1
1	READ_SDR	0h (Single)	1h	1 byte for Status register 1
2-7	STOP (00h)	0h	00h	

36.7.1.5 Read command

Table 203 shows READ command sequence.

Table 203. READ command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	03h or 13h	Command name: READ or 4READ READ is 3-byte address mode. 4READ is 4-byte address mode.
1	ADDR_SDR	0h (Single)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	READ_SDR	0h (Single)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
3-7	STOP (00h)	0h	00h	

36.7.1.6 Fast Read command

Table 204 shows Fast Read command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[3:0] = 8h.

Table 204. FAST_READ command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	0Bh or 0Ch	Command name: FAST_READ or 4FAST_READ FAST_READ is 3-byte address mode. 4FAST_READ is 4-byte address mode.
1	ADDR_SDR	0h (Single)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	DUMMY_SDR	0h (Single)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
3	READ_SDR	0h (Single)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
4-7	STOP (00h)	0h	00h	

36.7.1.7 Dual IO Fast Read command

Table 205 shows Dual IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[3:0] = 8h, Continuous Read mode.

Table 205. Dual IO FAST_READ command (Continuous Read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	BBh or BCh	Command name: DIOR or 4DIOR DIOR is 3-byte address mode. 4DIOR is 4-byte address mode.
1	ADDR_SDR	1h (Dual)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	1h (Dual)	Axh	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_SDR	1h (Dual)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	1h (Dual)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
5	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
6-7	STOP (00h)	0h	00h	

Table 206 shows Dual IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[3:0] = 8h, Noncontinuous Read mode.

Table 206. Dual IO FAST_READ command (Noncontinuous Read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	BBh or BCh	Command name: DIOR or 4DIOR DIOR is 3-byte address mode. 4DIOR is 4-byte address mode.
1	ADDR_SDR	1h (Dual)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	1h (Dual)	Any value other than Axh	Exit continuous read mode or remain in noncontinuous read mode.
3	DUMMY_SDR	1h (Dual)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	1h (Dual)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
5-7	STOP (00h)	0h	00h	

36.7.1.8 Quad IO Fast Read command

Table 207 shows Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 0, CR2V[3:0] = 8h, Non-QPI mode, Noncontinuous read mode.

Table 207. Quad IO FAST_READ command (Non-QPI mode, Noncontinuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	EBh or ECh	Command name: QIOR or 4QIOR QIOR is 3-byte address mode. 4QIOR is 4-byte address mode.
1	ADDR_SDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	2h (Quad)	Any value other than Axh	Exit continuous read mode or remain in noncontinuous read mode.
3	DUMMY_SDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
5-7	STOP (00h)	0h	00h	

Table 208 shows Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 0, CR2V[3:0] = 8h, Non-QPI mode, Continuous read mode.

Table 208. Quad IO FAST_READ command (Non-QPI mode, Continuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	EBh or ECh	Command name: QIOR or 4QIOR QIOR is 3-byte address mode. 4QIOR is 4-byte address mode.
1	ADDR_SDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	2h (Quad)	A0h	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_SDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.

Table continues on the next page...

Table 208. Quad IO FAST_READ command (Non-QPI mode, Continuous read mode) (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
5	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
6-7	STOP (00h)	0h	00h	

Table 209 shows Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 1, CR2V[3:0] = 8h, QPI mode, Continuous read mode.

Table 209. Quad IO FAST_READ command (QPI mode, Continuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	2h (Quad)	EBh or ECh	Command name: QIOR or 4QIOR QIOR is 3-byte address mode. 4QIOR is 4-byte address mode.
1	ADDR_SDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	2h (Quad)	A0h	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_SDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
5	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
6-7	STOP (00h)	0h	00h	

36.7.1.9 DDR Quad IO Fast Read command

Table 210 shows DDR Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 0, CR2V[3:0] = 8h, Non-QPI mode, Non-continuous read mode.

Table 210. DDR Quad IO FAST_READ command (Non-QPI mode, Noncontinuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	EDh or EEh	Command name: QIOR_DDR or 4QIOR_DDR QIOR_DDR is 3-byte address mode. 4QIOR_DDR is 4-byte address mode.

Table continues on the next page...

Table 210. DDR Quad IO FAST_READ command (Non-QPI mode, Noncontinuous read mode) (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
1	ADDR_DDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_DDR	2h (Quad)	Any value other than Axh	Exit continuous read mode or keep in noncontinuous read mode.
3	DUMMY_DDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	LEARN_DDR	2h (Quad)	1h	DLP is 8 bits.
5	READ_DDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
6-7	STOP (00h)	0h	00h	

Table 211 shows DDR Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration register bits CR2V[7] = 0, CR2V[6] = 0, CR2V[3:0] = 8h, Non-QPI mode, Continuous read mode.

Table 211. DDR Quad IO FAST_READ command (Non-QPI mode, Continuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	EDh or EEh	Command name: QIOR_DDR or 4QIOR_DDR QIOR_DDR is 3-byte address mode. 4QIOR_DDR is 4-byte address mode.
1	ADDR_DDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_DDR	2h (Quad)	A0h	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_DDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	LEARN_DDR	2h (Quad)	1h	DLP is 8 bits.
5	READ_DDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
6	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
7	STOP (00h)	0h	00h	

Table 212 shows DDR Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 1, CR2V[3:0] = 8h, QPI mode, Continuous read mode.

Table 212. DDR Quad IO FAST_READ command (QPI mode, Continuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	2h (Quad)	EDh or EEh	Command name: QIOR_DDR or 4QIOR_DDR QIOR_DDR is 3-byte address mode. 4QIOR_DDR is 4-byte address mode.
1	ADDR_DDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_DDR	2h (Quad)	A0h	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_DDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	LEARN_DDR	2h (Quad)	1h	DLP is 8 bits.
5	READ_DDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
6	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
7	STOP (00h)	0h	00h	

36.7.2 Application on HyperBus device

This section provides the example LUT instruction sequences for HyperBus device (Cypress RPC flash, HyperRam, or HyperFlash).

36.7.2.1 HyperFlash

This section provides example sequences for HyperFlash devices (Cypress S26KS series).

Table 213 shows the read status command sequence.

Table 213. Read status command

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0 (Write - Addr = 555h, Data = 70h)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W# = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 0000AAh (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	AAh	

Table continues on the next page...

Table 213. Read status command (continued)

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
	4	CMD_DDR	3h (Octal)	00h	Column Address: 05h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	05h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 0070h
	7	CMD_DDR	3h (Octal)	70h	
1 (Read - Addr = xxx, Data = Status register data)	0	CMD_DDR	3h (Octal)	A0h	CA bit 47: (R/W#) = 1h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: 24 bits
	2	CADDR_DDR	3h (Octal)	10h	Column Address: (13 zero bits + 3 valid bits)
	3	DUMMY_RWDS_DDR	3h (Octal)	0Bh	When latency count = 11
	4	READ_DDR	0h (Octal)	4h	4-Byte read
	5-7	STOP (00h)	0h	00h	

Table 214 shows the read (memory) command sequence.

Table 214. Read (memory) command

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0 (Read - Addr = xxx, Data = memory data)	0	CMD_DDR	3h (Octal)	A0h	CA bit 47: (R/W#) = 1h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: 24 bits
	2	CADDR_DDR	3h (Octal)	10h	Column Address: (13 zero bits + 3 valid bits)
	3	DUMMY_RWDS_DDR	3h (Octal)	0Bh	When latency count = 11
	4	READ_DDR	3h (Octal)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
	5-7	STOP (00h)	0h	00h	

Table 215 shows the word program command sequence.

Table 215. Word program command

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0 (Write - Addr = 555h, Data = AAh)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 0000AAh (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	AAh	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 05h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	05h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 00AAh
	7	CMD_DDR	3h (Octal)	AAh	
1 (Write - Addr = 2AAh, Data = 55h)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 000055h (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	55h	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 02h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	02h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 0055h
	7	CMD_DDR	3h (Octal)	55h	
2 (Write - Addr = 555h, Data = A0h)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 0000AAh (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	AAh	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 05h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	55h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 00A0h

Table continues on the next page...

Table 215. Word program command (continued)

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
	7	CMD_DDR	3h (Octal)	A0h	
3 (Word Program)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: 24 bits
	2	CADDR_DDR	3h (Octal)	10h	Column Address: (13 zero bits + 3 valid bits)
	3	WRITE_DDR	3h (Octal)	02h	2-Byte written data
	4-7	STOP (0h)	0h	00h	

Table 216 shows Write-to-Buffer and Program-Buffer-to-Flash command sequence.

Table 216. Write-to-Buffer and Program-Buffer-to-Flash command

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0 (Write - Addr = 555h, Data = AAh)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 0000AAh (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	AAh	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 05h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	05h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 00AAh
	7	CMD_DDR	3h (Octal)	AAh	
1 (Write - Addr = 2AAh, Data = 55h)	0	CMD_DDR	0h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 000055h (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	55h	

Table continues on the next page...

Table 216. Write-to-Buffer and Program-Buffer-to-Flash command (continued)

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
	4	CMD_DDR	3h (Octal)	00h	Column Address: 02h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	02h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 0055h
	7	CMD_DDR	3h (Octal)	55h	
2 (Write - Addr = SA, Data = 25h)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: SA (24 bit) SA is sector address. Write SA to IPCR0[SFAR].
	2	CADDR_DDR	3h (Octal)	10h	Column Address: 13 zero bits + 3 valid bits
	3	CMD_DDR	3h (Octal)	00h	Write Data: 0025h
	4	CMD_DDR	3h (Octal)	25h	
2 (Write - Addr = SA, Data = WC)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: SA (24 bit) SA is sector address. Write SA to IPCR0[SFAR].
	2	CADDR_DDR	3h (Octal)	10h	Column Address: 13 zero bits + 3 valid bits
	3	CMD_DDR	3h (Octal)	WC	Write Data: WC WC is word count.
	4	CMD_DDR	3h (Octal)		
3 - N (Write - Addr = WBL, Data = PD) N is the word count + 2	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: WBL (24 bit) WBL is write buffer location. Write WBL to IPCR0[SFAR].

Table continues on the next page...

Table 216. Write-to-Buffer and Program-Buffer-to-Flash command (continued)

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
	2	CADDR_DDR	3h (Octal)	10h	Column Address: 13 zero bits + 3 valid bits
	3	WRITE_DDR	3h (Octal)	02h	2-Byte write data
	4-7	STOP (0h)	0h	00h	
N + 1 (Write - Addr = SA, Data = 29) Program Buffer to Flash	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: SA (24 bit) SA is sector address. Write SA to IPCR0[SFAR].
	2	CADDR_DDR	3h (Octal)	10h	Column Address: 13 zero bits + 3 valid bits
	3	CMD_DDR	3h (Octal)	00h	Write Data: 29h
	4	CMD_DDR	3h (Octal)	29h	
	5-7	STOP (0h)	0h	00h	

36.7.2.2 HyperRAM

This section provides example sequences for HyperRAM (Cypress S27KL series).

The Read (memory) command sequence is same as HyperFlash. Table 217 shows the Write (memory) command sequence.

Table 217. Write (memory) command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
1	RADDR_DDR	3h (Octal)	18h	Row Address: 24 bits
2	CADDR_DDR	3h (Octal)	10h	Column Address: (13 zero bits + 3 valid bits)
3	DUMMY_RWDS_DDR	3h (Octal)	0Bh	When latency count = 11
4	WRITE_DDR	3h (Octal)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default writing

Table continues on the next page...

Table 217. Write (memory) command (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
				data size. This value is ignored for AHB command.
5-7	STOP (00h)	0h	00h	

36.7.3 Application on Serial NAND flash device

This section provides example LUT instruction sequences for serial NAND flash device (Micron Flash MT29 series). The operation of serial NAND flash is similar to serial NOR flash.

The read operation sequence is:

- Page read. Transfer the data from the NAND flash array to the cache register.
- Get feature to read the status.
- Random data read

Table 218 shows the page read command sequence.

Table 218. Page Read command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	1h (Single)	13h	Command code: 13h
1	RADDR_SDR	1h (Single)	18h	Row address: 24 bit
2-7	STOP (0h)	0h	00h	

Table 219 shows Get Feature command sequence.

Table 219. Get Feature command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	0Fh	Command code: 0Fh
1	CMD_SDR	1h (Single)	C0h	Status register address (C0h)
2	READ_SDR	0x1 (Single)	02h	2 bytes read data
3-7	STOP (0h)	0h	00h	

Table 220 shows Random Data Read command sequence.

Table 220. Read from Cache x4 command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	1h (Single)	6Bh	Command code: 6Bh
1	MODE4_SDR	1h (Single)	0h or 1h	If plane selection is one, software should decode the flash address and write 1h to mode bits. If plane selection

Table continues on the next page...

Table 220. Read from Cache x4 command (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
				is zero, software should write 0h to mode bits. Plane selection bit is 18th bit of flash address. If NAND flash size is less than 4 Gbit, plane selection is always zero.
2	CADDR_SDR	1h (Single)	0Ch	Column address: 12 bit
3	DUMMY_SDR	2h (Quad)	08h	Dummy cycle number: 8 (serial root clock)
4	READ_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size.
5-7	STOP (0x0)	0x0	0x00	

The program operation sequence is:

- Write enable
- Program Load. Transfer the write data to the cache register.
- Program Execute
- Get Feature to read the status.

Table 221 shows Program Load command sequence.

Table 221. Program Load command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	1h (Single)	02h	Command code: 02h
1	MODE4_SDR	1h (Single)	0h or 1h	If plane selection is one, software should decode the flash address and write 1h to mode bits. If plane selection is zero, software should write 0h to mode bits. Plane selection bit is 18th bit of flash address. If NAND flash size is less than 4 Gbit, plane selection is always zero.
2	CADDR_SDR	1h (Single)	0Ch	Column address: 12 bit
3	WRITE_SDR	1h (Single)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default writing data size.
4-7	STOP (0h)	0h	00h	

Table 222 shows Program Execute command sequence.

Table 222. Program Execute command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	1h (Single)	10h	Command code: 10h
1	RADDR_SDR	1h (Single)	18h	Row Address: 24 bit
2-7	STOP (0h)	0h	00h	

36.7.4 Application on FPGA device

An FPGA device can be accessed via both AHB and IP commands. All AHB accesses to FPGA are transparent to the software driver (no software intervention). An FPGA device may have some special requirements.

Table 223. Special requirements for FPGA devices

Condition	How to manage the condition
Device type may be different on A1, A2, B1, or B2.	Write 0 to MCR2[SAMEDEVICEEN] and configure the FLSHxCR0 and FLSHxCR1 registers separately for up to four external devices.
Device requires different wait cycle for programming.	The AHB write wait cycle number can be set separately for these four external devices (via FLSHxCR2[AWRWAIT]). Software can configure the sequences in LUT with different dummy instructions (operand determines dummy cycle). FlexSPI holds the AHB bus ready for this wait time; AHB bus performance may decrease when this wait time is very long.
Device requires different wait cycle for reading.	The AHB read sequence index and sequence number can be set separately for these four external devices (via FLSHxCR2[ARDSEQID] and FLSHxCR2[ARDSEQNUM]). Software can configure the sequences in LUT with different dummy instructions (operand determines dummy cycle).
Device may be sensitive to read instruction clock cycle number.	If its internal memory is implemented similar to a FIFO, the device is sensitive to the read instruction clock cycle number. Software can send the data size information to the external device via DATSZ instruction. The FPGA device decodes the data size information and determines how many data bytes must be popped.
Device may require interval time between chip select valid.	This interval can be managed via FLSHxCR1[CSINTERVAL].
Device may use SCLK as reference clock for its internal PLL.	SCLK must be free-running and the clock frequency must be stable. To achieve this configuration, write 1 to MCR0[SCKFREERUNEN] and only use SDR sequences.

36.7.5 Overview of error categories, flags, and triggered sources

Table 224 provides an overview of the categories, flags, and triggered sources of errors in FlexSPI.

Table 224. Error category, triggered sources, and flags

Error Category	Triggered Sources	Description	Error Flags
Command grant error	AHB write command	Command grant timeout	INTR[AHBCMDGE] is set AHB bus error response

Table continues on the next page...

Table 224. Error category, triggered sources, and flags (continued)

Error Category	Triggered Sources	Description	Error Flags
	AHB read command		INTR[AHBCMDGE] is set AHB bus error response
	IP command		INTR[IPCMDGE] is set
Command check error	AHB write command	• AHB write command with JMP_ON_CS instruction used in the sequence • Unknown instruction opcode in the sequence • Instruction DUMMY_SDR or DUMMY_RWDS_SDR used in DDR sequence • Instruction DUMMY_DDR or DUMMY_RWDS_DDR used in SDR sequence	INTR[AHBCMDERR] is set Command is not executed when an error is detected in command check
	AHB read command	• Unknown instruction opcode in the sequence • Instruction DUMMY_SDR or DUMMY_RWDS_SDR used in DDR sequence • Instruction DUMMY_DDR or DUMMY_RWDS_DDR used in SDR sequence	INTR[AHBCMDERR] is set Command is not executed when an error is detected in command check
	IP command	• IP command with JMP_ON_CS instruction used in the sequence • Unknown instruction opcode in the sequence • Instruction DUMMY_SDR or DUMMY_RWDS_SDR used in DDR sequence • Instruction DUMMY_DDR or DUMMY_RWDS_DDR used in SDR sequence • Flash boundary across	INTR[IPCMDERR] is set Command is not executed when an error is detected in command check
Command execution error	AHB write command	Command timeout during execution	INTR[AHBCMDERR] is set INTR[SEQTIMEOUT] is set An AHB bus error response

Table continues on the next page...

Table 224. Error category, triggered sources, and flags (continued)

Error Category	Triggered Sources	Description	Error Flags
			occurs, except in following cases: • Flush triggers AHB write command (INCR burst ended with AHB_TX_BUF not empty) • AHB bufferable write access and bufferable enabled (AHBCR[BUFFERABLEEN] = 1)
	AHB read command		INTR[AHBCMDERR] is set INTR[SEQTIMEOUT] is set AHB bus error response
	IP command		INTR[IPCMDERR] is set INTR[SEQTIMEOUT] is set
AHB bus timeout	AHB write command	AHB bus timeout (no bus ready return)	AHB bus error response
	AHB read command		
Data learning failed	Any command	No valid sample clock phase found after learn instruction is executed	INTR[DATALEARNFAIL] is set

36.8 Memory map and register definition

This section includes the FlexSPI module memory map and detailed descriptions of all registers.

36.8.1 Register access

All registers can be accessed with 8-bit, 16-bit, and 32-bit width operations. Never change the values of reserved fields in control registers. Changing the values of reserved fields may impact the normal functioning of the controller.

NOTE

When FlexSPI can access sensitive memory contents, the FlexSPI controller should protect those contents using resource isolation such as XRDC or TrustZone. Ensure that only trusted software is allowed to access the FlexSPI controller register interface.

36.8.2 FLEXSPI register descriptions

36.8.2.1 FlexSPI memory map

FLEXSPI base address: 425E_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Module Control 0 (MCR0)	32	RW	FFFF_80C2
4	Module Control 1 (MCR1)	32	RW	FFFF_FFFF
8	Module Control 2 (MCR2)	32	RW	2000_81F7
C	AHB Bus Control (AHBCR)	32	RW	0000_0018
10	Interrupt Enable (INTEN)	32	RW	0000_0000
14	Interrupt (INTR)	32	RW	0000_0000
18	LUT Key (LUTKEY)	32	RW	5AF0_5AF0
1C	LUT Control (LUTCR)	32	RW	0000_0002
20	AHB Receive Buffer 0 Control 0 (AHBRXBUF0CR0)	32	RW	8000_0020
24	AHB Receive Buffer 1 Control 0 (AHBRXBUF1CR0)	32	RW	8001_0020
28	AHB Receive Buffer 2 Control 0 (AHBRXBUF2CR0)	32	RW	8002_0020
2C	AHB Receive Buffer 3 Control 0 (AHBRXBUF3CR0)	32	RW	8003_0020
30	AHB Receive Buffer 4 Control 0 (AHBRXBUF4CR0)	32	RW	8004_0020
34	AHB Receive Buffer 5 Control 0 (AHBRXBUF5CR0)	32	RW	8005_0020
38	AHB Receive Buffer 6 Control 0 (AHBRXBUF6CR0)	32	RW	8006_0020
3C	AHB Receive Buffer 7 Control 0 (AHBRXBUF7CR0)	32	RW	8007_0020
60	Flash Control 0 (FLSHA1CR0)	32	RW	0000_0000
64	Flash Control 0 (FLSHA2CR0)	32	RW	0000_0000
68	Flash Control 0 (FLSHB1CR0)	32	RW	0000_0000
6C	Flash Control 0 (FLSHB2CR0)	32	RW	0000_0000
70	Flash Control 1 (FLSHA1CR1)	32	RW	0000_0063
74	Flash Control 1 (FLSHA2CR1)	32	RW	0000_0063
78	Flash Control 1 (FLSHB1CR1)	32	RW	0000_0063
7C	Flash Control 1 (FLSHB2CR1)	32	RW	0000_0063
80	Flash Control 2 (FLSHA1CR2)	32	RW	0000_0000
84	Flash Control 2 (FLSHA2CR2)	32	RW	0000_0000
88	Flash Control 2 (FLSHB1CR2)	32	RW	0000_0000
8C	Flash Control 2 (FLSHB2CR2)	32	RW	0000_0000
94	Flash Control 4 (FLSHCR4)	32	RW	0000_00C3
A0	IP Control 0 (IPCR0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
A4	IP Control 1 (IPCR1)	32	RW	0000_0000
A8	IP Control 2 (IPCR2)	32	RW	0000_0000
B0	IP Command (IPCMD)	32	RW	0000_0000
B4	Data Learning Pattern (DLPR)	32	RW	0000_0000
B8	IP Receive FIFO Control (IPRXFCR)	32	RW	0000_0000
BC	IP Transmit FIFO Control (IPTXFCR)	32	RW	0000_0000
C0	DLL Control 0 (DLLACR)	32	RW	0000_0100
C4	DLL Control 0 (DLLBCR)	32	RW	0000_0100
E0	Status 0 (STS0)	32	R	0000_0002
E4	Status 1 (STS1)	32	R	0000_0000
E8	Status 2 (STS2)	32	R	0100_0100
EC	AHB Suspend Status (AHBSPNDSTS)	32	R	0000_0000
F0	IP Receive FIFO Status (IPRXFSTS)	32	R	0000_0000
F4	IP Transmit FIFO Status (IPTXFSTS)	32	R	0000_0000
100 - 17C	IP Receive FIFO Data a (RFDR0 - RFDR31)	32	R	0000_0000
180 - 1FC	IP TX FIFO Data a (TFDR0 - TFDR31)	32	W	0000_0000
200 - 3FC	Lookup Table a (LUT0 - LUT127)	32	RW	See section
440	Receive Buffer Start Address of Region 0 (AHBBUFREGIONSTART0)	32	RW	0000_0000
444	Receive Buffer Region 0 End Address (AHBBUFREGIONEND0)	32	RW	0000_0000
448	Receive Buffer Start Address of Region 1 (AHBBUFREGIONSTART1)	32	RW	0000_0000
44C	Receive Buffer Region 1 End Address (AHBBUFREGIONEND1)	32	RW	0000_0000
450	Receive Buffer Start Address of Region 2 (AHBBUFREGIONSTART2)	32	RW	0000_0000
454	Receive Buffer Region 2 End Address (AHBBUFREGIONEND2)	32	RW	0000_0000
458	Receive Buffer Start Address of Region 3 (AHBBUFREGIONSTART3)	32	RW	0000_0000
45C	Receive Buffer Region 3 End Address (AHBBUFREGIONEND3)	32	RW	0000_0000

36.8.2.2 Module Control 0 (MCR0)

Controls basic functions of FlexSPI module

Offset

Register	Offset
MCR0	0h

Diagram

Fields

Field	Description
31-24 AHBGRANTWAIT	Timeouts Wait Cycle for AHB command Grant Sets duration of timeout wait cycle for AHB commands. If the arbitrator does not grant the AHB-triggered command, it times out after (AHBGRANTTIMEOUT × 1024) AHB clock cycles. When the pending command sequence is IP-triggered and the read or write data size is too large, this grant timeout may occur. When an AHB command grant timeout occurs, an INTR[AHBCMDGE] interrupt is generated. When INTEN[AHBCMDGEEN] = 1, the arbitrator ignores AHB commands. **NOTE** This field is for debugging only. Do not change the value from its default.
23-16 IPGRANTWAIT	Timeout Wait Cycle for IP Command Grant Sets duration of timeout wait cycle for IP commands. If the arbitrator does not grant the IP-triggered command, it times out after (IPGRANTTIMEOUT × 1024) AHB clock cycles. When the pending command sequence is AHB-triggered and the read or write data size is too large, this grant timeout may occur. When IP command grant timeout occurs, an INTR[IPCMDGE] interrupt is generated. When INTEN[IPCMDGEEN] = 1, the arbitrator ignores IP commands. **NOTE** This field is for debugging only. Do not change the value from its default.
15 LEARNEN	Data Learning Enable Enables data learning feature. When data learning is disabled, the sampling clock phase 0 is always used for receive data sampling, even when a learn instruction is correctly executed. See Data learning. 0 - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable
14 SCKFREERUNEN	SCLK Free-running Enable Enables free-running SCLK output. For FPGA applications, the external device may use SCLK as a reference clock to its internal PLL. 0 - Disable 1 - Enable
13 —	Reserved
12 DOZEEN	Doze Mode Enable Enables Doze mode. When enabled, AHB clock and serial clock are gated off when there is a Doze-mode request from the system. 0 - Disable 1 - Enable
11 HSEN	Half Speed Serial Flash Memory Access Enable Enables the clock divider to provide a half-speed clock to external serial flash devices (A_SCLK/B_SCLK) for all commands in SDR and DDR mode. Write 1 to MCR0[MDIS] before changing the value of this field. Failure to do so may cause issues in the internal logic or state machine. 0 - Disable 1 - Enable
10-8 SERCLKDIV	Serial Root Clock Divider Sets divider value for serial root clock. The serial root clock can be divided inside FlexSPI . See Clocking. **NOTE** Do not modify this field while FlexSPI is active (MCR0[MDIS] = 0). 000 - Divided by 1 001 - Divided by 2 010 - Divided by 3 011 - Divided by 4 100 - Divided by 5 101 - Divided by 6 110 - Divided by 7 111 - Divided by 8
7	AHB Write Access to IP Transmit FIFO Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
ATDFEN	Enables AHB write access to IP transmit FIFO. See Writing data to IP transmit FIFO. 0 - AHB write access disabled. IP bus writes to IP transmit FIFO. AHB bus write access to IP transmit FIFO memory space produces bus error. 1 - AHB write access enabled. AHB bus writes to IP transmit FIFO. IP Bus write access to IP transmit FIFO memory space is ignored and causes no bus error.
6 ARDFEN	AHB Read Access to IP Receive FIFO Enable Enables AHB read access to IP receive FIFO. See Reading data from IP receive FIFO. 0 - AHB read access disabled. IP bus reads IP receive FIFO. AHB Bus read access to IP receive FIFO memory space produces bus error. 1 - AHB read access enabled. AHB bus reads IP receive FIFO. IP Bus read access to IP receive FIFO memory space returns data zero and causes no bus error.
5-4 RXCLKSRC	Sample Clock Source for Flash Reading Selects sampling clock source for flash memory reading. See Receive clock source features. 00 - Dummy Read strobe that FlexSPI generates, looped back internally 01 - Dummy Read strobe that FlexSPI generates, looped back from DQS pad 10 - Reserved 11 - Flash-memory-provided read strobe and input from DQS pad
3-2 —	Reserved
1 MDIS	Module Disable Disables FlexSPI module. When the module is disabled, AHB and serial clock are gated off internally to save power. Only register access is allowed, except for the LUT, IP receive FIFO, and IP transmit FIFO. 0 - No impact 1 - Module disable
0 SWRESET	Software Reset Resets the internal FlexSPI state machine and aborts any transactions in progress. Hardware automatically writes 0 to this field after software reset is done. Configuration registers are not reset. 0 - No impact 1 - Software reset

36.8.2.3 Module Control 1 (MCR1)

Controls basic functions of FlexSPI module

Offset

Register	Offset
MCR1	4h

Diagram

Fields

Field	Description
31-16 SEQWAIT	Command Sequence Wait Sets wait time for command sequence. Command sequence execution times out and aborts after (SEQWAIT × 1024) serial root clock cycles. When this timeout occurs, if the interrupt is enabled (INTEN[SEQTIMEOUTEN] = 1), an INTR[SEQTIMEOUT] interrupt is generated. Also, the arbitrator ignores AHB commands. **NOTE** You cannot write 0 to this field.
15-0 AHBBUSWAIT	AHB Bus Wait Sets wait time for AHB bus. When data is not received from or transmitted to serial flash memory space after (AHBBUSWAIT × 1024) AHB clock cycles, the read or write access times out. When this timeout occurs, the AHB bus receives an error response, and an interrupt is generated (INTEN[AHBBUSERROREN]). When INTR[AHBBUSERROREN] = 1, the arbitrator ignores AHB commands. **NOTE** You cannot write 0 to this field.

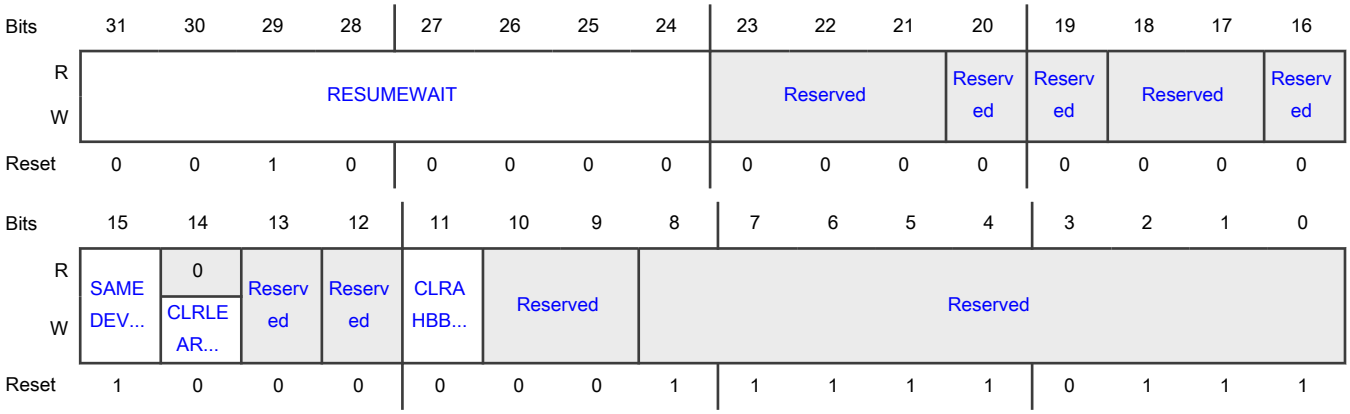
36.8.2.4 Module Control 2 (MCR2)

Controls basic functions of FlexSPI module.

Offset

Register	Offset
MCR2	8h

Diagram



Fields

Field	Description
31-24 RESUMEWAIT	Resume Wait Duration Determines duration (in AHB clock cycles) to remain in idle state before suspended command sequence is resumed. See Command abort and suspend.
23-21 —	Reserved
20 —	Reserved
19 —	Reserved
18-17 —	Reserved
16 —	Reserved
15 SAMEDEVICEEN	Same Device Enable Sets all external devices (A1, A2, B1, and B2) to be the same devices (in type and in size). 0 - In Individual mode, FLSHA1CRx and FLSHA2CRx, FLSHB1CRx and FLSHB2CRx settings are applied to Flash A1, A2, B1, B2 separately.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - FLSHA1CR0, FLSHA1CR1, and FLSHA1CR2 register settings are applied to Flash A1, A2, B1, B2. FLSHA2CRx, FLSHB1CRx, and FLSHB2CRx settings are ignored.
14 CLRLEARNPHASE	Clear Learn Phase Selection Resets the sampling clock phase selection to 0. When 1 is written to this field, it becomes 0 again immediately. 0 - No impact 1 - Reset sample clock phase selection to 0
13 —	Reserved
12 —	Reserved
11 CLRAHBBUFOPT	Clear AHB Buffer Determines whether AHB receive and transmit buffers are cleared automatically when FlexSPI returns Stop mode ACK. If an AHB receive buffer or transmit buffer will be powered off in Stop mode, software should write 1 to this field. Otherwise, an AHB read access after exiting Stop mode may hit either buffer, but their data entries are invalid. 0 - Not cleared automatically 1 - Cleared automatically
10-9 —	Reserved
8-0 —	Reserved

36.8.2.5 AHB Bus Control (AHBCR)

Controls AHB bus interface functions.

Offset

Register	Offset
AHBCR	Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AFLASHBASE				Reserved						ALIGNMENT		Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved		Reserved	Reserved	READ SZA...	Reserved	Reserved	RESU MED...	READ ADD...	PREF ETC...	BUFF ERA...	CACH ABL...	CLRA HBT...	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Fields

Field	Description
31-27 AFLASHBASE	AHB Memory-Mapped Flash Base Address
26-22 —	Reserved
21-20 ALIGNMENT	AHB Boundary Alignment Configures the size of AHB read and write boundary. All accesses that cross the boundary are divided into smaller sub accesses. 00 - No limit 01 - 1 KB 10 - 512 bytes 11 - 256 bytes
19 —	Reserved
18 —	Reserved
17 —	Reserved
16 —	Reserved
15 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
14-13 —	Reserved
12 —	Reserved
11 —	Reserved
10 READSZALIGN	AHB Read Size Alignment Configures how AHB read size is determined. See Table 194. **NOTE** When 1, data prefetching is disabled regardless of other field settings such as PREFETCH_EN and OTFAD_EN. Read size is rounded up to the nearest eight-byte multiple. For example, if current AHB read size is 12 bytes, it is aligned to 16 bytes. When the fetch size must be eight-byte aligned but no speculative fetching is wanted, write 1 to this field. 0 - Register settings such as PREFETCH_EN and OTFAD_EN determine AHB read size. 1 - AHB read size to up size to 8 bytes aligned, no prefetching
9 —	Reserved
8 —	Reserved
7 RESUMEDISABLE	AHB Read Resume Disable Disables the resumption of AHB read prefetch. **NOTE** When OTFAD is enabled, this field does not affect the resume function; AHB read resume is always disabled. When OTFAD is disabled, suspended AHB read prefetch resumes based on the values of this field and MCR2[RESUMEWAIT]. 0 - Suspended AHB read prefetch resumes when AHB is IDLE. 1 - Suspended AHB read prefetch does not resume once aborted.
6 READADDROPT	AHB Read Address Option Removes AHB burst start address alignment limitation. When the FlexSPI controller is used for FPGA application, this field may be required for FlexSPI to fetch the exact byte number as an AHB burst. In this case, FPGA device should be non-word-addressable and this field must be 0.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE When OTFAD is enabled, FlexSPI functions as it does when READADDROPT = 1 regardless the value of this field. 0 - AHB read burst start address alignment is limited when flash memory is accessed in flash is word-addressable. 1 - AHB read burst start address alignment is not limited. FlexSPI fetches more data than the AHB burst requires for address alignment.
5 PREFETCHEN	AHB Read Prefetch Enable Enables AHB read prefetch. When enabled, FlexSPI fetches more flash read data than the current AHB burst requires. This action reduces the read latency for next AHB read access. **NOTE** When OTFAD is enabled, the AHB prefetch function is enabled regardless of the value of this field. **NOTE** AHB read prefetch is enabled only when both this field and AHBRXBUFnCR0[PREFETCHEN] are 1. 0 - Disable 1 - Enable
4 BUFFERABLEEN	Bufferable Write Access Enable Enables AHB bus bufferable write access. This field affects the last beat of an AHB write access. See AHB write access to flash memory. 0 - Disabled. For all AHB write accesses (bufferable or nonbufferable), FlexSPI returns AHB Bus Ready after transmitting all data and finishing command. 1 - Enabled. For AHB bufferable write access, FlexSPI returns AHB Bus Ready when the arbitrator grants the AHB command. FlexSPI does not wait for the AHB command to finish.
3 CACHABLEEN	Cacheable Read Access Enable Enables AHB bus cacheable read access. 0 - Disabled. When an AHB bus cacheable read access occurs, FlexSPI does not check whether it hit the AHB transmit buffer. 1 - Enabled. When an AHB bus cacheable read access occurs, FlexSPI first checks whether the access hit the AHB transmit buffer.
2 CLRAHBTXBUF	Clear AHB Transmit Buffer Clears status and pointers of AHB transmit buffer. When the clear operation completes, this field clears automatically, so it always reads as 0. 0 - No impact.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable clear operation.
1 —	Reserved
0 —	Reserved

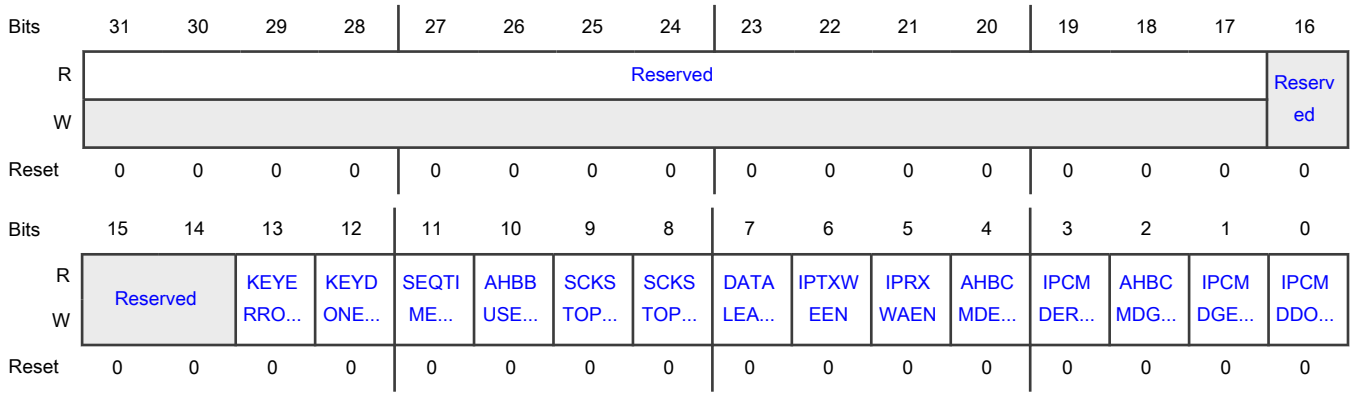
36.8.2.6 Interrupt Enable (INTEN)

Includes AHB error response, IPS error response, and ECC error interrupt enables. See Interrupts.

Offset

Register	Offset
INTEN	10h

Diagram



Fields

Field	Description
31-17 —	Reserved
16 —	Reserved
15-14 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
13 KEYERROREN	OTFAD Key Blob Processing Error Interrupt Enable 0 - Disable interrupt or no impact 1 - Enable interrupt
12 KEYDONEEN	OTFAD Key Blob Processing Done Interrupt Enable 0 - Disable interrupt or no impact 1 - Enable interrupt
11 SEQTIMEOUTEN	Sequence execution Timeout Interrupt Enable 0 - Disable interrupt or no impact 1 - Enable interrupt
10 AHBBUSERROREN	AHB Bus Error Interrupt Enable 0 - Disable interrupt or no impact 1 - Enable interrupt
9 SCKSTOPBYWREN	SCLK Stopped By Write Interrupt Enable Enables interrupt that indicates SCLK is stopped during command sequence because asynchronous transmit FIFO is empty. 0 - Disable interrupt or no impact 1 - Enable interrupt
8 SCKSTOPBYRDEN	SCLK Stopped By Read Interrupt Enable Enables interrupt that indicates SCLK is stopped during command sequence because asynchronous receive FIFO is full. 0 - Disable interrupt or no impact 1 - Enable interrupt
7 DATALEARNFAILEN	Data Learning Failed Interrupt Enable 0 - Disable interrupt or no impact 1 - Enable interrupt
6 IPTXWEEN	IP Transmit FIFO Watermark Empty Interrupt Enable Enables interrupt that indicates IP transmit FIFO contains more empty space than watermark level. 0 - Disable interrupt or no impact 1 - Enable interrupt
5 IPRXWAEN	IP Receive FIFO Watermark Available Interrupt Enable Enables interrupt that indicates IP receive FIFO contains more valid data than the watermark level. 0 - Disable interrupt or no impact

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable interrupt
4 AHBCMDERREN	AHB-Triggered Command Sequences Error Detected Interrupt Enable 0 - Disable interrupt or no impact 1 - Enable interrupt
3 IPCMDERREN	IP-Triggered Command Sequences Error Detected Interrupt Enable 0 - Disable interrupt or no impact 1 - Enable interrupt
2 AHBCMDGEEN	AHB-Triggered Command Sequences Grant Timeout Interrupt Enable. 0 - Disable interrupt or no impact 1 - Enable interrupt
1 IPCMDGEEN	IP-Triggered Command Sequences Grant Timeout Interrupt Enable 0 - Disable interrupt or no impact 1 - Enable interrupt
0 IPCMDDONEEN	IP-Triggered Command Sequences Execution Finished Interrupt Enable 0 - Disable interrupt or no impact 1 - Enable interrupt

36.8.2.7 Interrupt (INTR)

Includes AHB error response, IPS error response, and ECC error interrupts. See Interrupts.

Offset

Register	Offset
INTR	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		KEYERROR	KEYDONE	SEQTIME...	AHBBUSE...	SCKSTOP...	SCKSTOP...	DATALEA...	IPTXWE	IPRXWA	AHBCMDE...	IPCMDERR	AHBCMDGE	IPCMDGE	IPCMDDO...
W				W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-17 —	Reserved
16 —	Reserved
15-14 —	Reserved
13 KEYERROR	OTFAD Key Blob Processing Error **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred When writing 0 - No effect 1 - Clear the flag
12 KEYDONE	OTFAD key blob processing done interrupt.
11 SEQTIMEOUT	Sequence Execution Timeout **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred When writing 0 - No effect 1 - Clear the flag
10 AHBBUSERROR	AHB Bus Error Generated upon an AHB bus timeout or an AHB bus illegal access to flash during OTFAD key blob processing. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred When writing 0 - No effect 1 - Clear the flag
9 SCKSTOPBYWR	SCLK Stopped Due To Empty Transmit FIFO Generated when SCLK is stopped during command sequence because the asynchronous transmit FIFO is empty. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred When writing 0 - No effect 1 - Clear the flag
8 SCKSTOPBYRD	SCLK Stopped Due To Full Receive FIFO Generated when SCLK is stopped during command sequence because the asynchronous receive FIFO is full. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred When writing 0 - No effect 1 - Clear the flag
7 DATALEARNFAIL	Data Learning Failed **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When writing 0 - No effect 1 - Clear the flag
6 IPTXWE	IP Transmit FIFO Watermark Empty Generated when the IP transmit FIFO contains more empty space than the watermark level. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred When writing 0 - No effect 1 - Clear the flag
5 IPRXWA	IP Receive FIFO Watermark Available Generated when the IP receive FIFO contains more valid data than the watermark level. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred When writing 0 - No effect 1 - Clear the flag
4 AHBCMDERR	AHB-Triggered Command Sequences Error Generated upon an AHB-triggered command sequence error. When an error is detected for an AHB command, the command is ignored and not executed. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred When writing

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - No effect 1 - Clear the flag
3 IPCMDERR	IP-Triggered Command Sequences Error Generated upon an IP-triggered command sequence error. When an error is detected for an IP command, the command is ignored and not executed. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred When writing 0 - No effect 1 - Clear the flag
2 AHBCMDGE	AHB-Triggered Command Sequences Grant Timeout **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred When writing 0 - No effect 1 - Clear the flag
1 IPCMDGE	IP-Triggered Command Sequences Grant Timeout **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred When writing 0 - No effect 1 - Clear the flag
0	IP-Triggered Command Sequences Execution Finished

Table continues on the next page...

Table continued from the previous page...

Field	Description
IPCMDDONE	Generated upon completion of an IP-triggered command sequence. Also generated when IPCMDGE or IPCMDERR interrupt is generated **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt condition has not occurred 1 - Interrupt condition has occurred When writing 0 - No effect 1 - Clear the flag

36.8.2.8 LUT Key (LUTKEY)

Contains the key to lock and unlock LUT. See Lookup table (LUT).

Offset

Register	Offset
LUTKEY	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	KEY															
Reset	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KEY															
Reset	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0

Fields

Field	Description
31-0 KEY	LUT Key Contains key to lock or unlock LUT. The key is 5AF05AF0h. Read value is always 5AF05AF0h.

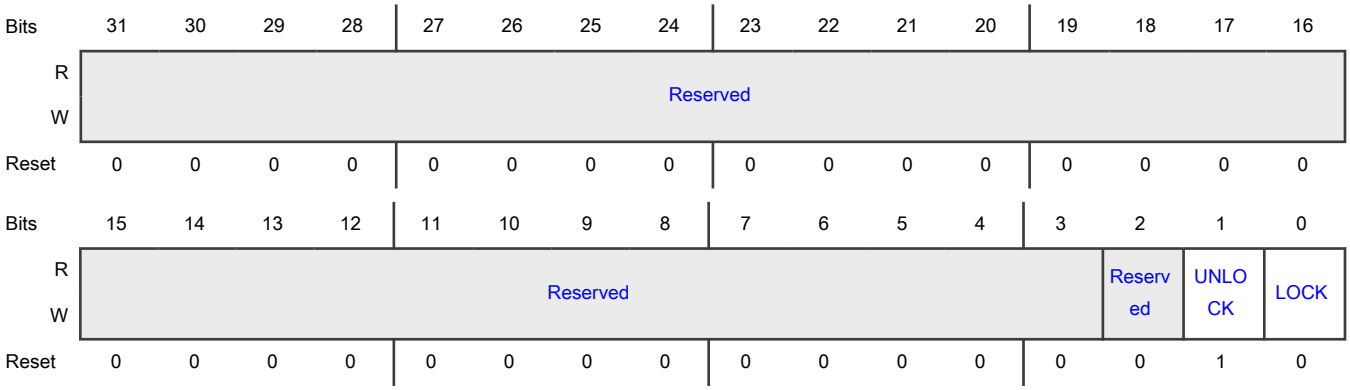
36.8.2.9 LUT Control (LUTCR)

Used with LUTKEY register to lock or unlock LUT. For the lock or unlock operation to be successful, this register must be written immediately after writing 5AF05AF0h to the LUTKEY register. See Lookup table (LUT) for details on locking and unlocking LUT. You cannot write 00 or 11 to the LOCK and UNLOCK fields.

Offset

Register	Offset
LUTCR	1Ch

Diagram



Fields

Field	Description
31-3 —	Reserved
2 —	Reserved
1 UNLOCK	Unlock LUT 0 - LUT is locked (LUTCR[LOCK] must be 1) 1 - LUT is unlocked and can be written
0 LOCK	Lock LUT 0 - LUT is unlocked (LUTCR[UNLOCK] must be 1) 1 - LUT is locked and cannot be written

36.8.2.10 AHB Receive Buffer n Control 0 (AHBRXBUF0CR0 - AHBRXBUF7CR0)

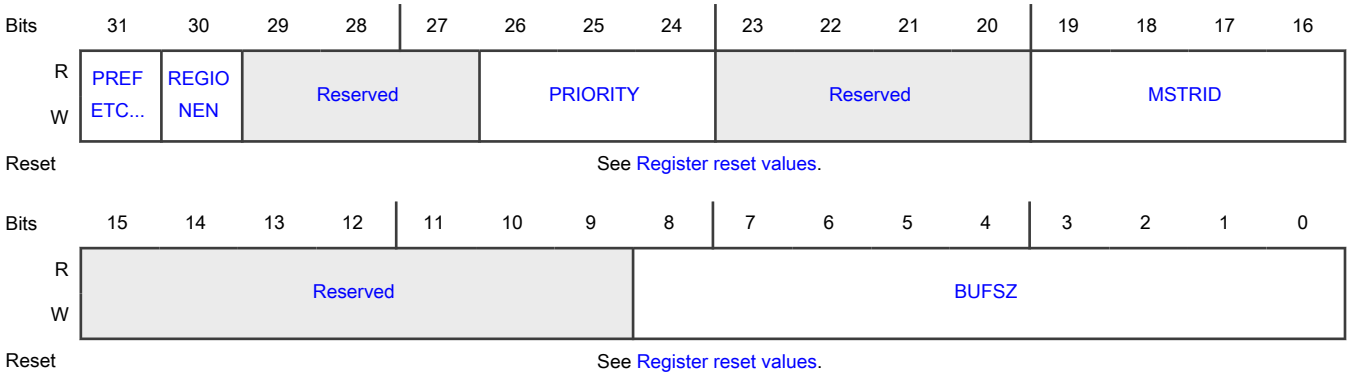
Stores the read data from the SPI interface.

Offset

For n = 0 to 7:

Register	Offset
AHBRXBUFnCR0	20h + (n × 4h)

Diagram



Register reset values

Register	Reset value
AHBRXBUF0CR0	8000_0020h
AHBRXBUF1CR0	8001_0020h
AHBRXBUF2CR0	8002_0020h
AHBRXBUF3CR0	8003_0020h
AHBRXBUF4CR0	8004_0020h
AHBRXBUF5CR0	8005_0020h
AHBRXBUF6CR0	8006_0020h
AHBRXBUF7CR0	8007_0020h

Fields

Field	Description
31 PREFETCHEN	AHB Read Prefetch Enable Enables AHB read prefetch for the controller corresponding to the current AHB receive buffer. The prefetch feature is disabled when AHBCR[PREFETCHEN] is 0. You can use this field to enable or disable prefetch separately for each controller. 0 - Disabled 1 - Enabled when AHBCR[PREFETCHEN] is enabled.
30 REGIONEN	AHB Receive Buffer Address Region Enable Enable prefetch buffer enhancement.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE Only AHBRXBUF 0-3 have this function. 0 - Disabled. The buffer hit is based on the value of MSTRID only. 1 - Enabled. The buffer hit is based on the value of MSTRID and the address within AHBBUFREGIONSTARTn and AHBREGIONENDn.
29-27 —	Reserved
26-24 PRIORITY	AHB Controller Read Priority Configure the priority for the AHB Controller Read to which this AHB receive buffer is assigned. 7 is the highest priority, 0 the lowest. See Command abort and suspend.
23-20 —	Reserved
19-16 MSTRID	AHB Controller ID Configures the ID of the AHB controller to which this AHB receive buffer is assigned. See AHB receive buffer management.
15-9 —	Reserved
8-0 BUFSZ	AHB Receive Buffer Size Configures the size of the AHB receive buffer in multiples of 64 bits. See AHB receive buffer management.

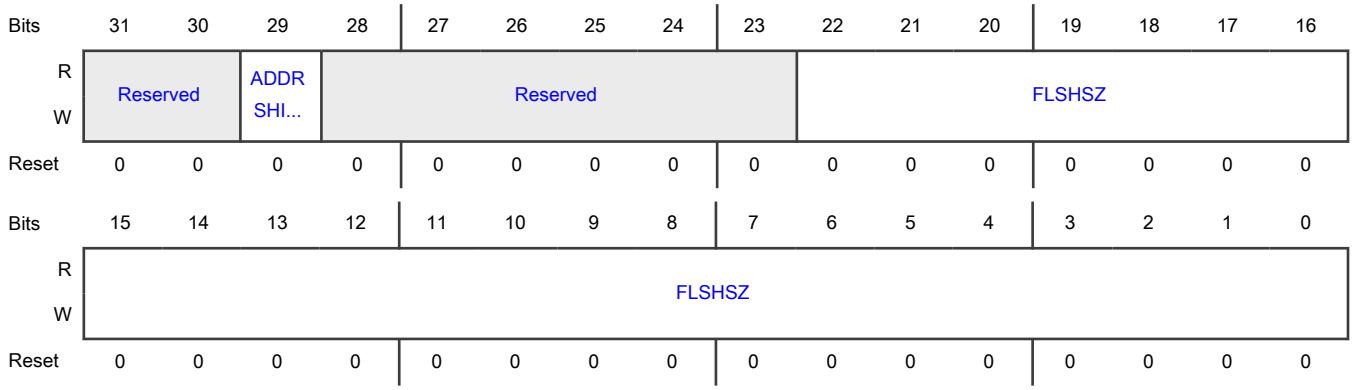
36.8.2.11 Flash Control 0 (FLSHA1CR0 - FLSHB2CR0)

Contains flash memory size setting. FlexSPI determines which device is accessed (Chip Select) via this register.

Offset

Register	Offset
FLSHA1CR0	60h
FLSHA2CR0	64h
FLSHB1CR0	68h
FLSHB2CR0	6Ch

Diagram



Fields

Field	Description
31-30 —	Reserved
29 ADDRSHIFT	AHB Address Shift Function control Used to left-shift five bits for the flash address to support ISSI quad PSRAM. 0 - Disabled 1 - Enabled
28-23 —	Reserved
22-0 FLSHSZ	Flash Size in KB Configures the maximum flash memory size. See the chip-specific section for the maximum flash size supported. When the value of this field is greater than the maximum size, the device flash size is interpreted as the maximum. The maximum flash size is the largest flash size supported for a single device. It is also the maximum total flash size supported for all devices (up to four). If the total flash size is larger than the maximum size, only the maximum flash size address space is accessible. For IPS access, The maximum flash size supported for each device is 4 GB. When the value of this field is greater than 400000h, the device flash size is taken as 4 GB. The max total flash size supported (for all 4 devices) is also 4 GB. If the total flash size is larger than 4 GB, only 4 GB of address space is accessible.

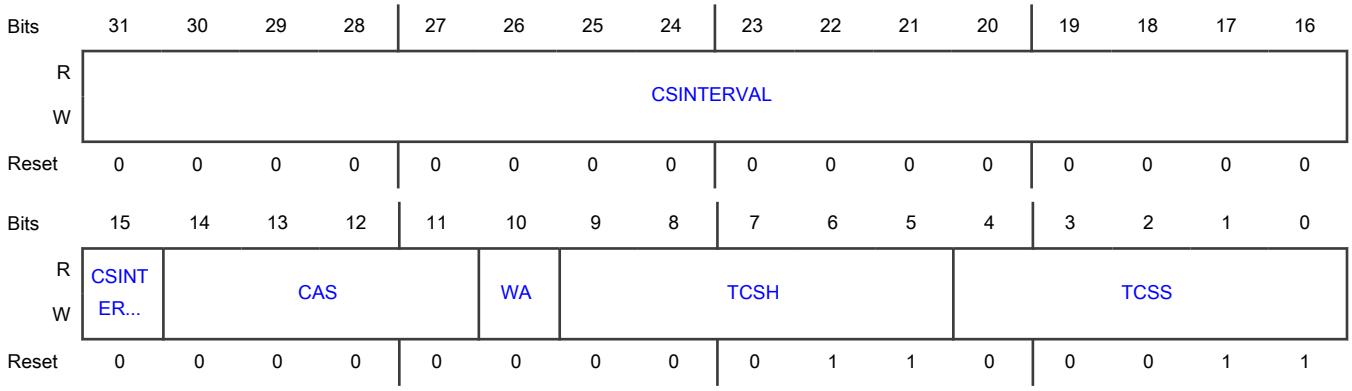
36.8.2.12 Flash Control 1 (FLSHA1CR1 - FLSHB2CR1)

Contains settings for flash device-specific timings and flash internal address space.

Offset

Register	Offset
FLSHA1CR1	70h
FLSHA2CR1	74h
FLSHB1CR1	78h
FLSHB2CR1	7Ch

Diagram



Fields

Field	Description
31-16 CSINTERVAL	Chip Select Interval Configures the minimum interval between flash device chip select deassertion and chip select assertion. If the external flash device has a limitation on the interval between command sequences, configure this field accordingly. If there is no limitation, write 0h to this field. When CSINTERVALUNIT = 0, the chip select invalid interval is: CSINTERVAL × 1 serial clock cycle; When CSINTERVALUNIT = 1, the chip select invalid interval is: CSINTERVAL × 256 serial clock cycles. **NOTE** The minimum chip select interval is 2 cycles, even when the value of CSINTERVAL is less than 2.
15 CSINTERVALUNIT	Chip Select Interval Unit Configures the interval unit for chip select. 0 - 1 serial clock cycle 1 - 256 serial clock cycles
14-11 CAS	Column Address Size When external flash memory has a separate address field for rows and columns, this field configures the flash column address bit width. FlexSPI automatically splits a flash-mapped address into row address and column address according to the values of this field and the WA field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When the external flash memory does not support column address, write 0 to this field. FlexSPI transmits all flash address bits as Row address. For flash address mapping, see Flash address sent to flash memory devices.
10 WA	Word-Addressable Configures whether external flash memory is word-addressable or byte-addressable. If flash memory is word-addressable, it should be accessed in multiples of 16 bits. Currently, FlexSPI does not transmit flash address bit 0 to external flash memory. For flash address mapping, see Flash address sent to flash memory devices. 0 - Byte-addressable 1 - Word-addressable
9-5 TCSH	Serial Flash CS Hold Time Used to meet flash TCSH timing requirement. Serial flash CS Hold time that FlexSPI promises is: (TCSH + 1/2) serial clock cycles for DDR mode, and TCSH serial clock cycles for SDR mode. See Output timing between chip select and SCLK.
4-0 TCSS	Serial Flash CS Setup Time Used to meet flash TCSS timing requirement. Serial flash CS Setup time that FlexSPI promises is: (TCSS + 1/2) serial root clock cycles for SDR and DDR mode. See Output timing between chip select and SCLK.

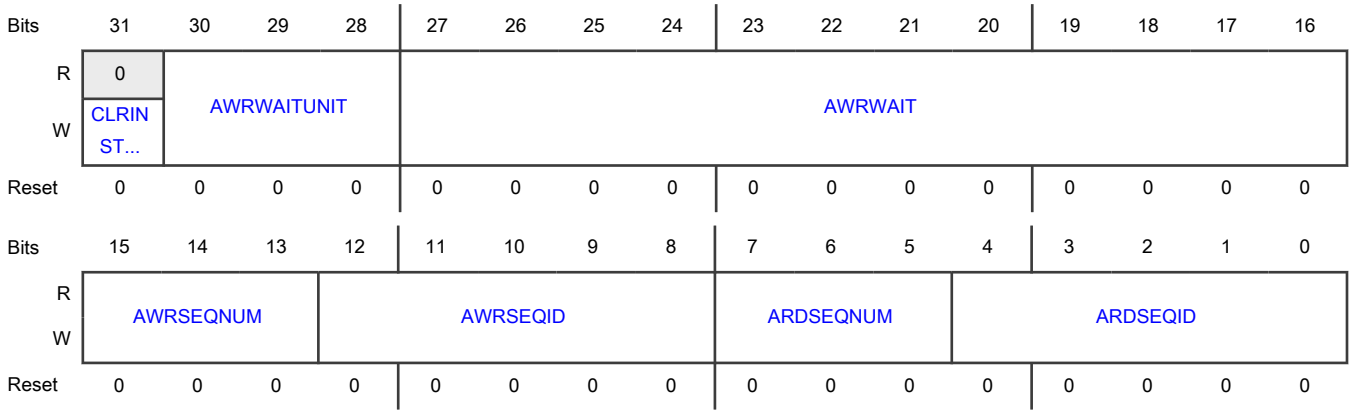
36.8.2.13 Flash Control 2 (FLSHA1CR2 - FLSHB2CR2)

Contains fields to configure AHB bus access. If the four external devices are different types, AHB read and write commands may use different command sequences; AHB bus ready wait time may also differ.

Offset

Register	Offset
FLSHA1CR2	80h
FLSHA2CR2	84h
FLSHB1CR2	88h
FLSHB2CR2	8Ch

Diagram



Fields

Field	Description
31 CLRINSTRPTR	Clear Instruction Pointer Clears the instruction pointer, which is the pointer that JMP_ON_CS saves internally. See Programmable sequence engine. This field is used for AHB read access to external flash memory supporting Execute-In-Place (XIP) mode.
30-28 AWRWAITUNIT	AWRWAIT Unit Configures the unit of AHB write wait time, as the value of AWRWAIT determines, in terms of AHB clock cycles. 000 - 2 001 - 8 010 - 32 011 - 128 100 - 512 101 - 2048 110 - 8192 111 - 32768
27-16 AWRWAIT	AHB Write Wait Configures AHB write wait time, with the AWRWAITUNIT field. Certain devices (such as FPGA) require time to write data into internal memory after the command sequences finished on the FlexSPI interface. If another read command sequence arrives before the current programming finishes internally, the read data may be wrong. This field is used to hold the AHB bus ready for AHB write access, waiting until the programming is finished in the external device. This hold ensures that no AHB read command is triggered before the programming finishes in the external device. The wait cycle between an AHB-triggered command sequence finishing on FlexSPI and the AHB return bus being ready is: AWRWAIT × AWRWAITUNIT.
15-13	Sequence Number for AHB Write-Triggered Command

Table continues on the next page...

Table continued from the previous page...

Field	Description
AWRSEQNUM	Configures the sequence number of an AHB read-triggered command. For certain flash devices (for example, HyperFlash, HyperRam, and Serial NAND flash), a flash programming access is done via several command sequences. An AHB write command triggers (AWRSEQNUM + 1) command sequences to external flash memory each time. FlexSPI executes the sequences in LUT incrementally. **NOTE** • Software should ensure that the last sequence index never exceeds LUT sequence numbers: AWRSEQID + AWRSEQNUM < 32 • Software must ensure that the AWRSEQNUM and LUT fields are configured correctly according to the external device specification. FlexSPI does not check the sequence; it executes the sequences one by one.
12-8 AWRSEQID	Sequence Index for AHB Write-Triggered Command
7-5 ARDSEQNUM	Sequence Number for AHB Read-Triggered Command Configures the sequence number of an AHB read-triggered command in the lookup table. For certain flash devices (for example, HyperFlash, HyperRam, and Serial NAND flash), a flash read access is done via several command sequences. An AHB read command triggers (ARDSEQNUM + 1) command sequences to external flash memory each time. FlexSPI executes the sequences in LUT incrementally. **NOTE** • Software should ensure that the last sequence index never exceeds LUT sequence numbers: ARDSEQID + ARDSEQNUM ≤ 32. • Software must ensure that the ARDSEQNUM and LUT fields are configured correctly according to the external device specification. FlexSPI does not check the sequence; it executes the sequences one by one.
4-0 ARDSEQID	Sequence Index for AHB Read-Triggered Command in LUT

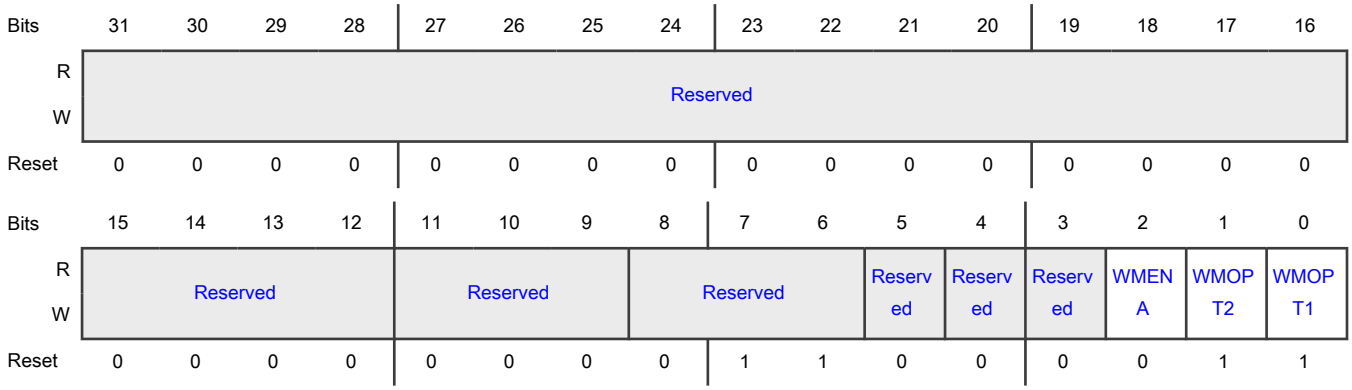
36.8.2.14 Flash Control 4 (FLSHCR4)

Provides configuration for all external devices.

Offset

Register	Offset
FLSHCR4	94h

Diagram



Fields

Field	Description
31-12 —	Reserved
11-9 —	Reserved
8-6 —	Reserved
5 —	Reserved
4 —	Reserved
3 —	Reserved
2 WMENA	Write Mask Enable for Port A Enables write mask for flash device on port A. 0 - Disabled. When writing to external device, DQS(RWDS) pin is not driven. 1 - Enabled. When writing to external device, FlexSPI drives DQS(RWDS) pin as write mask output.
1 WMOPT2	Write Mask Option 2 When using AP memory, used to remove AHB or IP write burst minimum length limitation. When using this field, FLSHCR4[WMOPT1] should be 1. 0 - When writing to an external device, DQS pin is used as write mask. When flash memory is accessed in individual mode, AHB or IP write burst length is not limited.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - When writing to an external device, DQS pin is not used as write mask. When flash memory is accessed in individual mode, AHB or IP write burst length is limited. The minimum write burst length should be 4.
0 WMOPT1	Write Mask Option 1 Used to remove AHB and IP write burst start address alignment limitation. 0 - When writing to an external device, DQS pin is used as write mask. When flash memory is accessed in individual mode, AHB or IP write burst start address alignment is not limited. 1 - When writing to an external device, DQS pin is not used as write mask. When flash memory is accessed in individual mode, AHB or IP write burst start address alignment is limited.

36.8.2.15 IP Control 0 (IPCR0)

Provides all configuration required for IP commands. Provides the start address for the flash device, instead of the chip address, to be accessed for IP command. FlexSPI determines the chip select automatically according to this start address.

NOTE

- You cannot issue an IP command that crosses flash device boundaries. If you do so, it generates an IPCMDERR interrupt.
- Configure this register before an IP command is triggered.
- Do not change the values in this register while an IP command is in progress.

Offset

Register	Offset
IPCR0	A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SFAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SFAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SFAR	Serial Flash Address Configures the serial flash address for IP commands. The address should be the address of the flash device without the base address.

36.8.2.16 IP Control 1 (IPCR1)

Provides all configuration required for IP commands. Provides the flash read and program data size, sequence index in LUT, sequence number and individual mode settings for IP commands.

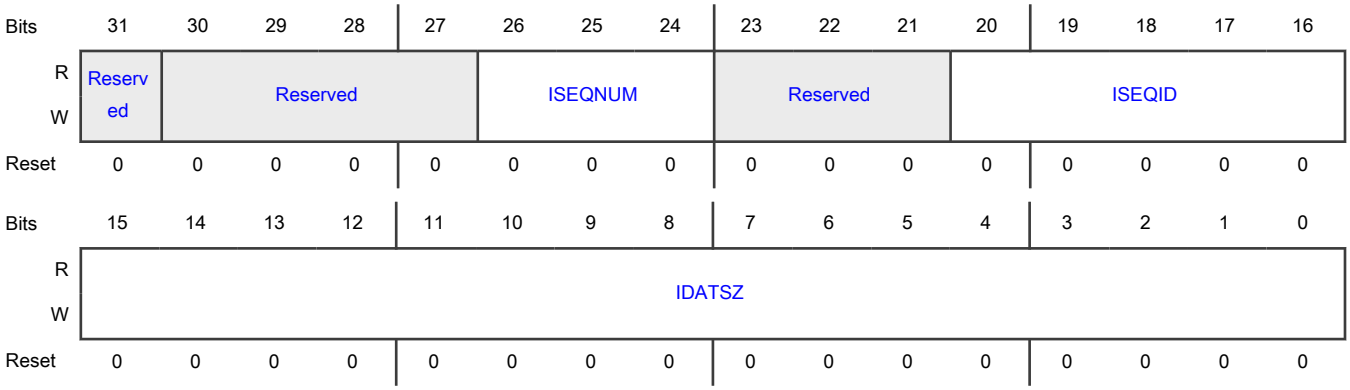
NOTE

- Configure this register before an IP command is triggered.
- Do not change the values in this register while an IP command is in progress.

Offset

Register	Offset
IPCR1	A4h

Diagram



Fields

Field	Description
31 —	Reserved
30-27 —	Reserved
26-24	Sequence Number for IP command: ISEQNUM+1.

Table continues on the next page...

Table continued from the previous page...

Field	Description
ISEQNUM	
23-21 —	Reserved
20-16 ISEQID	Sequence Index in LUT for IP command.
15-0 IDATSZ	Flash Read/Program Data Size (in bytes) for IP command.

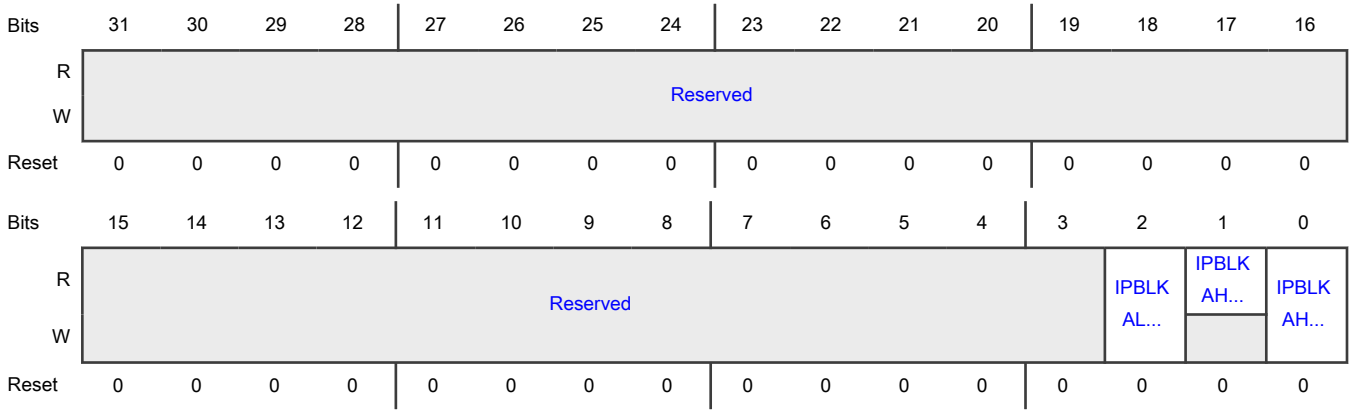
36.8.2.17 IP Control 2 (IPCR2)

Configures whether IP commands can block AHB commands.

Offset

Register	Offset
IPCR2	A8h

Diagram



Fields

Field	Description
31-3 —	Reserved
2	IP Command Blocking All AHB Command Enable 0 - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
IPBLKALLAHB	1 - IP commands block all AHB commands.
1 IPBLKAHBACK	IP Command Blocking AHB Command Acknowledgment Enable 0 - IP commands do not block AHB command acknowledgment. 1 - IP commands block AHB command acknowledgment.
0 IPBLKAHBREQ	IP Command Blocking AHB Command Request Enable 0 - IP commands do not block AHB command requests. 1 - IP commands block AHB command requests.

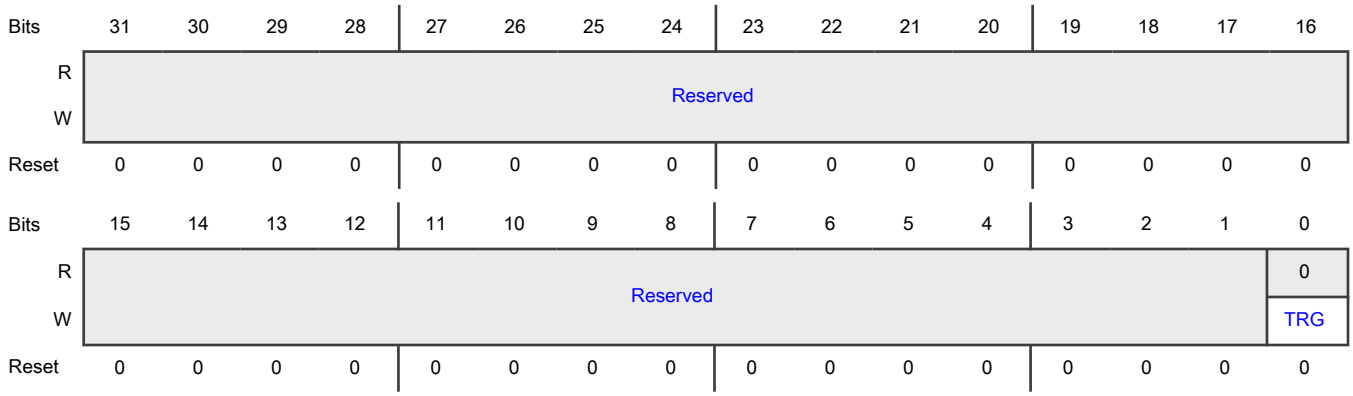
36.8.2.18 IP Command (IPCMD)

Used to trigger an IP command to access external flash device. When the arbitrator grants the IP command, the command is executed on the FlexSPI interface.

Offset

Register	Offset
IPCMD	B0h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 TRG	Command Trigger Triggers an IP command. This field clears automatically after it has been written and always reads as 0.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE You cannot trigger another IP command before the previous IP command finishes on the FlexSPI interface. Software must poll INTR[IPCMDDONE] or wait for this interrupt. 0 - No action 1 - Start the IP command that the IPCR0 and IPCR1 registers define.

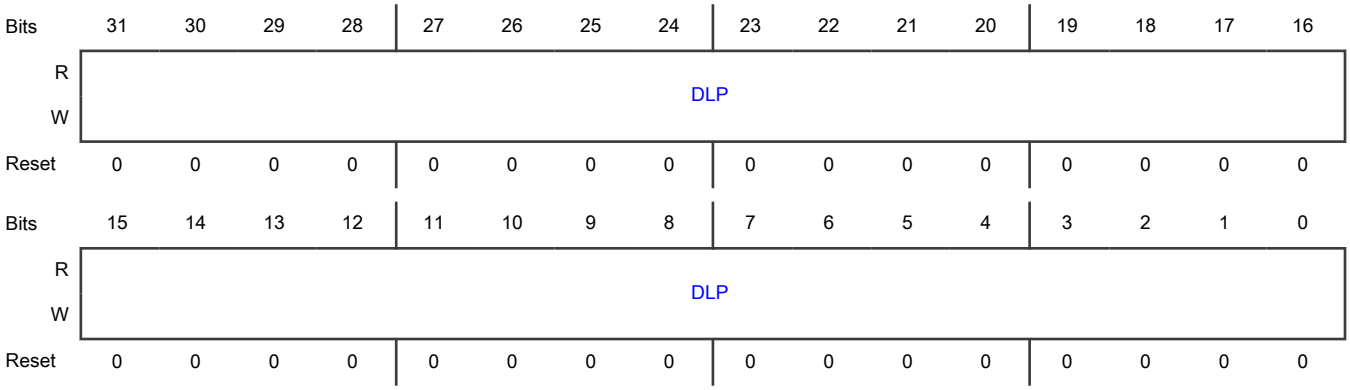
36.8.2.19 Data Learning Pattern (DLPR)

Provides the pattern to be used during Data Learning in FlexSPI.

Offset

Register	Offset
DLPR	B4h

Diagram



Fields

Field	Description
31-0 DLP	Data Learning Pattern The operand in the LEARN_SDR or LEARN_DDR instruction code determines the data learning pattern bit number. This number never exceeds 32 bits. If the operand in the instruction code is greater than 32, a 32-bit pattern is used. See Data learning.

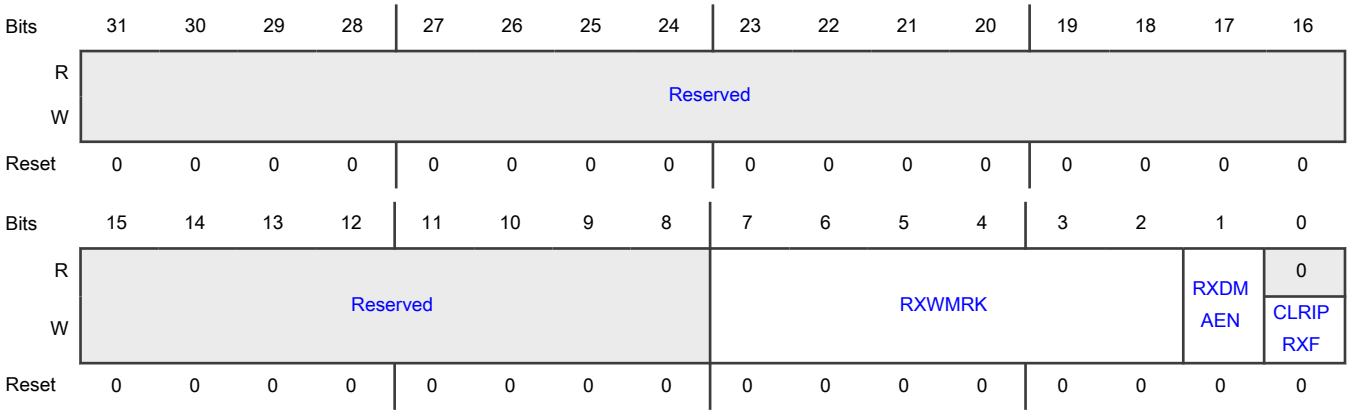
36.8.2.20 IP Receive FIFO Control (IPRXFCR)

Provides the configuration fields for IP receive FIFO management.

Offset

Register	Offset
IPRXFCR	B8h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-2 RXWMRK	IP Receive FIFO Watermark Level Configures the watermark level for the IP receive FIFO. The watermark level is (RXWMRK + 1) × 64 bits. The INTR[IPRXWA] interrupt is set when FlexSPI fills the IP receive FIFO ≥ the watermark level. A DMA request occurs when the fill level is ≥ the watermark level and IPRXFCR[RXDMAEN] = 1. When the fill level is ≥ the watermark level and INTEN[IPRXWAEN] = 1, an INTR[IPRXWA] interrupt is generated. **NOTE** After writing 1 to INTR[IPRXWA] to clear it, the read address should be rolled back to the start address (memory mapped). If the IP bus reads the IP receive FIFO, the read address should roll back to RFDR0. If the AHB bus reads the IP receive FIFO, the read address should roll back to ARDF_BASE.
1 RXDMAEN	IP Receive FIFO Reading by DMA Enable 0 - Disabled. The processor reads the FIFO. 1 - Enabled. DMA reads the FIFO.
0 CLRIPRXF	Clear IP Receive FIFO Clears all valid data entries in IP receive FIFO. Resets the read and write pointers in IP receive FIFO. 0 - No function 1 - A clock cycle pulse clears all valid data entries in IP receive FIFO.

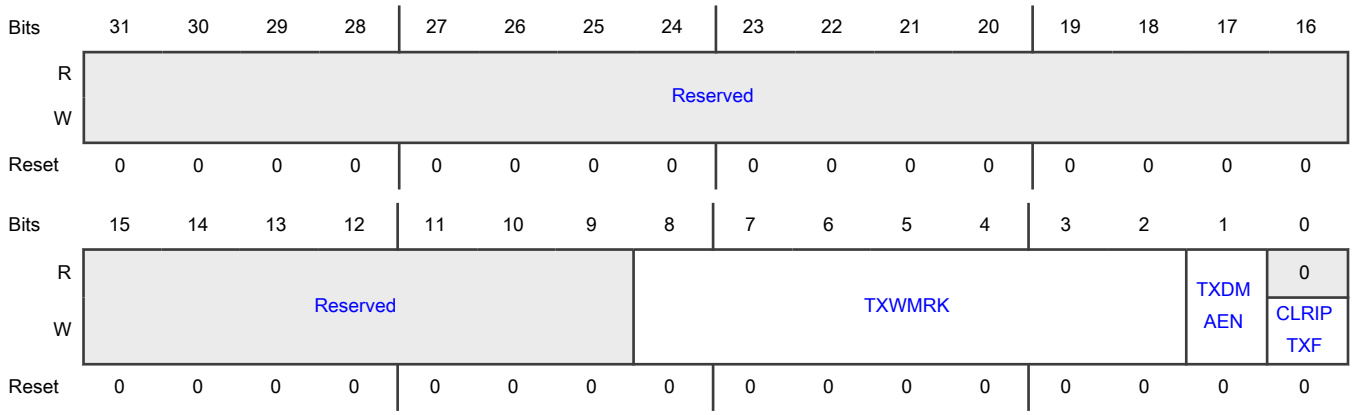
36.8.2.21 IP Transmit FIFO Control (IPTXFCR)

Provides the configuration fields for IP transmit FIFO management.

Offset

Register	Offset
IPTXFCR	BCh

Diagram



Fields

Field	Description
31-9 —	Reserved
8-2 TXWMRK	Transmit Watermark Level Sets the transmit watermark level. The watermark level is (TXWMRK + 1) × 64 bits. INTR[IPTXWE] is set when FlexSPI empties the IP transmit FIFO ≥ the watermark level. When the empty level ≥ the watermark level and IPTXFCR[TXDMAEN] = 1, a DMA request occurs. When the empty level ≥ the watermark level and INTEN[IPTXWEEN] = 1, an INTR[IPTXWE] interrupt is generated. **NOTE** • The watermark level should not be larger than the write window. • The watermark level should not be larger than the IP transmit FIFO size. • The write address to the IP receive FIFO should roll back to the start address of the write window. After pushing the IP transmit fifo, you should write 1 to INTR[IPTXWE] to clear it, which will perform the rollback.
1 TXDMAEN	Transmit FIFO DMA Enable Selects whether DMA or processor fills IP transmit FIFO. 0 - Processor

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - DMA
0 CLRIPTXF	Clear IP Transmit FIFO Clears all valid data entries in IP transmit FIFO. The read and write pointers in IP transmit FIFO are reset. 0 - No function 1 - A clock cycle pulse clears all valid data entries in the IP transmit FIFO.

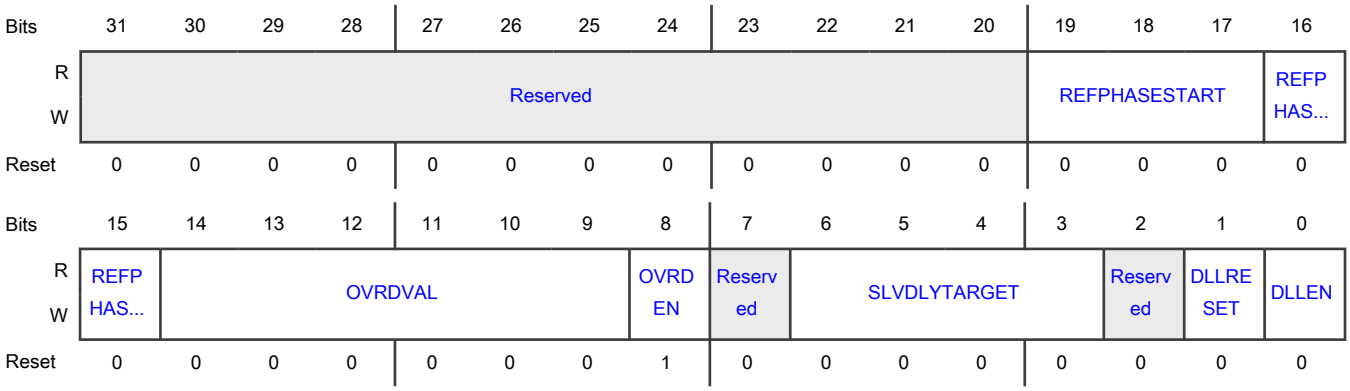
36.8.2.22 DLL Control 0 (DLLACR - DLLBCR)

Configures Flash A and B sample clock DLL.

Offset

Register	Offset
DLLACR	C0h
DLLBCR	C4h

Diagram



Fields

Field	Description
31-20 —	Reserved
19-17 REFPHASESTART	Reference Clock Delay Line Start Phase If the DLL does not lock well when DLLxCR[DLLEN] is enabled, update the register value of REFPHASESTART. 7h is recommended value. Note that DLLxCR[DLLRESET] is required to be triggered after REFPHASESTART is updated.

Table continues on the next page...

Table continued from the previous page...

Field	Description
16-15 REFPHASEGAP	Reference Clock Delay Line Phase Adjust Gap. REFPHASEGAP setting of 2h is recommended if DLLEN is set.
14-9 OVRDVAL	Target Clock Delay Line Override Value Configures the override value for the target clock line delay cell number. When OVRDEN = 1, the delay cell number in DLL is OVRDVAL + 1. See DLL configuration for sampling.
8 OVRDEN	Target Clock Delay Line Override Value Enable Enables the override value for the target clock line delay cell number. 0 - Disable 1 - Enable
7 —	Reserved
6-3 SLVDLYTARGET	Target Delay Line Configures the target delay line for the target. The delay target for target delay line is: ((SLVDLYTARGET+1) × 1/32 × clock cycle of reference clock (serial root clock). If serial root clock is ≥ 100 MHz, DLLEN is set to 1, OVRDEN is set to 0, then SLVDLYTARGET setting is up to flash parameters used.
2 —	Reserved
1 DLLRESET	DLL reset Forces a DLL reset. The forced reset causes the DLL to lose lock and recalibrate to detect a ref_clock half-period phase shift. The reset action is edge-triggered, so software must write 0 to this field after writing 1 to it (no delay limitation). 0 - No function 1 - Force DLL reset.
0 DLLEN	DLL Calibration Enable Enables DLL calibration. When DLL calibration is disabled, the delay cell number in the target delay line is always 1. When DLLxCR[OVRDEN] = 1, the target delay line is overridden and this field is ignored. 0 - Disable 1 - Enable

36.8.2.23 Status 0 (STS0)

Indicates the status of the internal state machine.

Offset

Register	Offset
STS0	E0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				Reserved				DATALEARNPHASEA				ARBCMDSRC		ARBIDLE	SEQIDLE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fields

Field	Description
31-12 —	Reserved
11-8 —	Reserved
7-4 DATALEARNPHASEA	Data Learning Phase Selection on Port A Indicates the sampling clock phase selection on Port A after data learning. There are 16 clock phases for the sampling clock that the delay cell line generates. When data learning is not enabled, the default clock phase 0 is used to sample flash read data. When data learning is enabled and a LEARN_SDR or LEARN_DDR instruction has executed correctly, FlexSPI determines the correct clock phase to sample read data. See Data learning.
3-2 ARBCMDSRC	ARB Command Source Indicates the trigger source of current command sequence that the arbitrator has granted. When ARB_CTL is not busy (STS0[ARBIDLE] = 1), the value of this field does not matter. 00 - Trigger source is AHB read command. 01 - Trigger source is AHB write command. 10 - Trigger source is IP command (by writing 1 to IPCMD[TRG]). 11 - Trigger source is a suspended command that has resumed.
1	ARB_CTL State Machine Idle

Table continues on the next page...

Table continued from the previous page...

Field	Description
ARBIDLE	Indicates whether the state machine in ARB_CTL is idle, or a command sequence that the arbitrator granted has not finished on the FlexSPI interface. When idle, no transaction is occurring on the FlexSPI interface (STS0[SEQIDLE] = 1). When waiting for the FlexSPI controller to become idle, poll this field instead of STS0[SEQIDLE]. 0 - Not idle 1 - Idle
0 SEQIDLE	SEQ_CTL State Machine Idle Indicates whether the state machine in SEQ_CTL is idle, or a command sequence is executing on the FlexSPI interface. 0 - Not idle 1 - Idle

36.8.2.24 Status 1 (STS1)

Indicates the error code of the AHB or IPS interface error response.

Offset

Register	Offset
STS1	E4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				IPCMDERRCODE				Reserved				IPCMDERRID			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				AHBCMDERRCODE				Reserved				AHBCMDERRID			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
27-24 IPCMDERRCODE	IP Command Error Code When an IP command Error is detected, indicates the error code. This field returns to zero when INTR[IPCMDERR] is cleared. 0000 - No error 0010 - IP command with JMP_ON_CS instruction used in the sequence 0011 - Unknown instruction opcode in the sequence 0100 - DUMMY_SDR or DUMMY_RWDS_SDR instruction used in DDR sequence 0101 - DUMMY_DDR or DUMMY_RWDS_DDR instruction used in SDR sequence 0110 - Flash memory access start address exceeds entire flash address range (A1, A2, B1, and B2) 1110 - Sequence execution timeout 1111 - Flash boundary crossed
23-21 —	Reserved
20-16 IPCMDERRID	IP Command Error ID When an IP command error is detected, indicates the sequence index. This field returns to zero when INTR[IPCMDERR] is cleared.
15-12 —	Reserved
11-8 AHBCMDERRCODE	AHB Command Error Code Contains the error code when an AHB command error is detected. This field returns to zero when INTR[AHBCMDERR] is cleared. 0000 - No error 0010 - AHB Write command with JMP_ON_CS instruction used in the sequence 0011 - Unknown instruction opcode in the sequence 0100 - DUMMY_SDR or DUMMY_RWDS_SDR instruction used in DDR sequence 0101 - DUMMY_DDR or DUMMY_RWDS_DDR instruction used in SDR sequence 1110 - Sequence execution timeout
7-5 —	Reserved
4-0 AHBCMDERRID	AHB Command Error ID When an AHB command error is detected, indicates the sequence index. This field returns to zero when INTR[AHBCMDERR] is cleared.

36.8.2.25 Status 2 (STS2)

Indicates the status of Flash A and B sample clock DLLs.

Offset

Register	Offset
STS2	E8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		BREFSEL						BSLVSEL						BREFLOCK	BSLVLOCK
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		AREFSEL						ASLVSEL						AREFLOCK	ASLVLOCK
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Fields

Field	Description
31-30 —	Reserved
29-24 BREFSEL	Flash B Sample Clock Reference Delay Line Delay Cell Number Indicates the sample clock reference delay line delay cell number for Flash B. There are 0-63 phases. • 000001b - Indicates lock phase 0. • 100000b - Indicates lock phase 63.
23-18 BSLVSEL	Flash B Sample Clock Target Delay Line Delay Cell Number Indicates the sample clock target delay line delay cell number for Flash B. There are 0-63 phases. • 000001b - Indicates lock phase 0. • 100000b - Indicates lock phase 63.
17 BREFLOCK	Flash B Sample Clock Reference Delay Line Locked 0 - Not locked 1 - Locked
16	Flash B Sample Target Reference Delay Line Locked

Table continues on the next page...

Table continued from the previous page...

Field	Description
BSLVLOCK	0 - Not locked 1 - Locked
15-14 —	Reserved
13-8 AREFSEL	Flash A Sample Clock Reference Delay Line Delay Cell Number Indicates the sample clock reference delay line delay cell number for Flash A. There are 0-63 phases. • 000001b - Indicates lock phase 0. • 100000b - Indicates lock phase 63.
7-2 ASLVSEL	Flash A Sample Clock Target Delay Line Delay Cell Number Indicates the sample clock target delay line delay cell number for Flash A. There are 0-63 phases. • 000001b - Indicates lock phase 0. • 100000b - Indicates lock phase 63.
1 AREFLOCK	Flash A Sample Clock Reference Delay Line Locked 0 - Not locked 1 - Locked
0 ASLVLOCK	Flash A Sample Target Delay Line Locked 0 - Not locked 1 - Locked

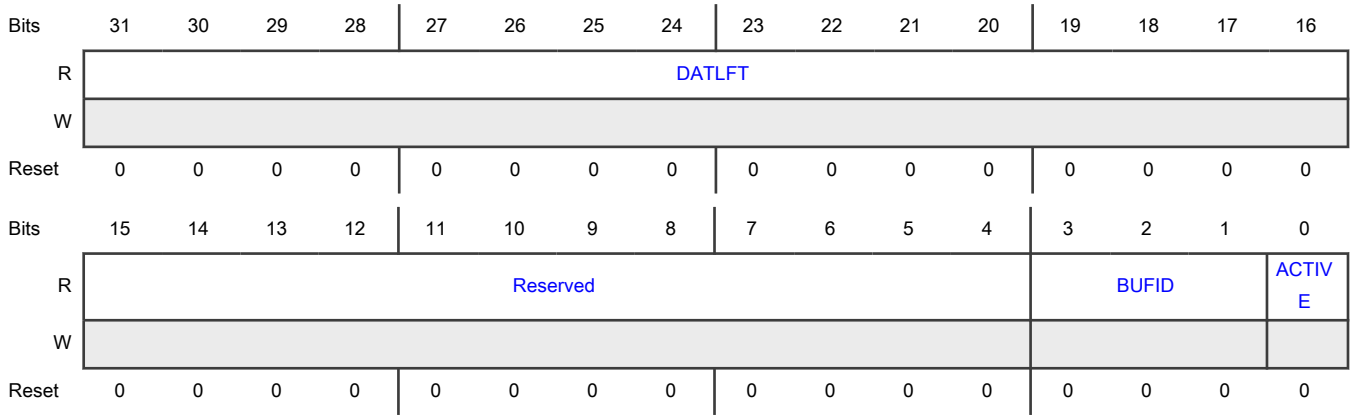
36.8.2.26 AHB Suspend Status (AHBSPNDSTS)

Indicates the status of suspended AHB read prefetch command sequence. When an IP or AHB command is triggered while the arbitrator processes an AHB read sequence (prefetching additional data not for current AHB burst), the prefetch sequence is suspended. When there are no longer transactions on FlexSPI, the prefetch sequence may be resumed. FlexSPI saves only one AHB prefetch sequence. When a new prefetch sequence is suspended with an active sequence already suspended, the previous suspended sequence is removed and never resumed. See Command abort and suspend.

Offset

Register	Offset
AHBSPNDSTS	ECh

Diagram



Fields

Field	Description
31-16 DATLFT	Data Left Contains the data size remaining (in bytes) for a suspended command sequence.
15-4 —	Reserved
3-1 BUFID	AHB Receive Buffer ID for Suspended Command Sequence
0 ACTIVE	Active AHB Read Prefetch Suspended Indicates whether an AHB read prefetch command sequence has been suspended. 0 - No suspended AHB read prefetch command. 1 - An AHB read prefetch command sequence has been suspended.

36.8.2.27 IP Receive FIFO Status (IPRXFSTS)

Indicates the status of IP receive FIFO.

Offset

Register	Offset
IPRXFSTS	F0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDCNTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								FILL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 RDCNTR	Read Data Counter Contains counter for total data read. The data read is the value of this field × 64 bits.
15-8 —	Reserved
7-0 FILL	Fill Level of IP Receive FIFO Indicates how full the IP receive FIFO is. The fill level is the value of this field × 64 bits.

36.8.2.28 IP Transmit FIFO Status (IPTXFSTS)

Indicates the status of IP transmit FIFO.

Offset

Register	Offset
IPTXFSTS	F4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WRCNTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								FILL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 WRCNTR	Write Data Counter Contains counter for total data written. The data written is the value of this field × 64 bits.
15-8 —	Reserved
7-0 FILL	Fill Level of IP Transmit FIFO Indicates how full the IP transmit FIFO is. The fill level is the value of this field × 64 bits.

36.8.2.29 IP Receive FIFO Data a (RFDR0 - RFDR31)

Provides read access to IP receive FIFO via IPS bus. The read value is unknown for read access to invalid entries in the IP receive FIFO.

Offset

For a = 0 to 31:

Register	Offset
RFDRa	100h + (a × 4h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXDATA	Receive Data Contains receive data. See Reading data from IP receive FIFO.

36.8.2.30 IP TX FIFO Data a (TFDR0 - TFDR31)

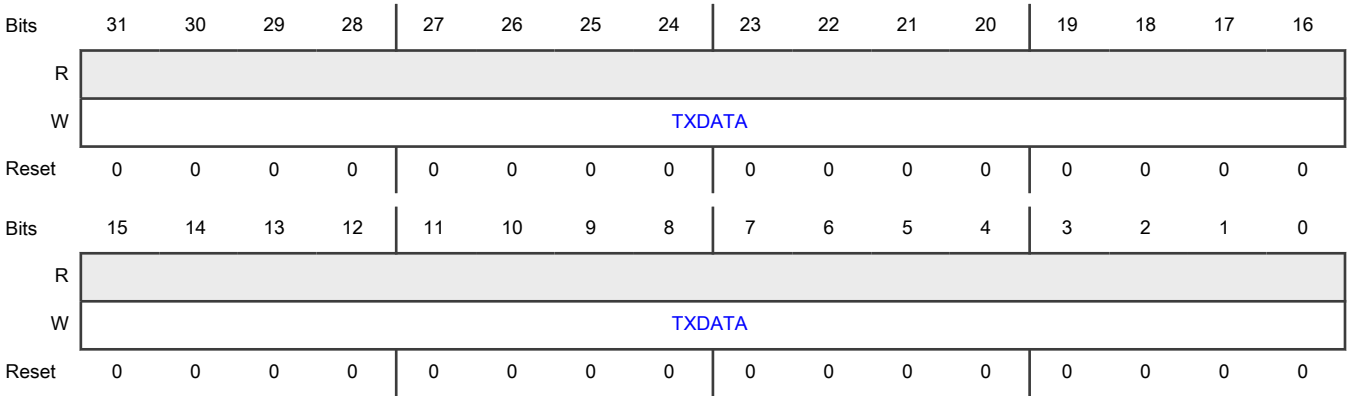
Provides write access to IP transmit FIFO via IPS bus.

Offset

For a = 0 to 31:

Register	Offset
TFDRa	180h + (a × 4h)

Diagram



Fields

Field	Description
31-0 TXDATA	Transmit Data Contains transmit data. See Writing data to IP transmit FIFO.

36.8.2.31 Lookup Table a (LUT0 - LUT127)

Contains a lookup table for command sequences. Software should set the sequence index before triggering an IP command or AHB command. FlexSPI fetches the command sequence from LUT when an IP or AHB command is triggered. There are 32 command sequences in LUT. See Lookup table (LUT) and Programmable sequence engine.

NOTE

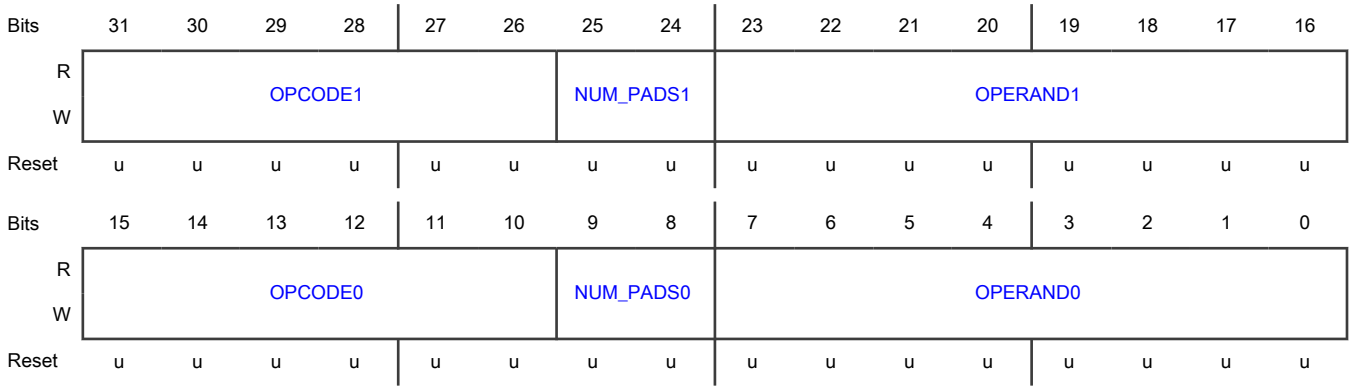
LUT is implemented as memory, so the reset value is unknown.

Offset

For a = 0 to 127:

Register	Offset
LUTa	200h + (a × 4h)

Diagram



Fields

Field	Description
31-26 OPCODE1	OPCODE1
25-24 NUM_PADS1	NUM_PADS1
23-16 OPERAND1	OPERAND1
15-10 OPCODE0	OPCODE
9-8 NUM_PADS0	NUM_PADS0
7-0 OPERAND0	OPERAND0

36.8.2.32 Receive Buffer Start Address of Region a (AHBBUFREGIONSTART0 - AHBBUFREGIONSTART3)

Indicates the start address of AHB receive buffer address region.

Offset

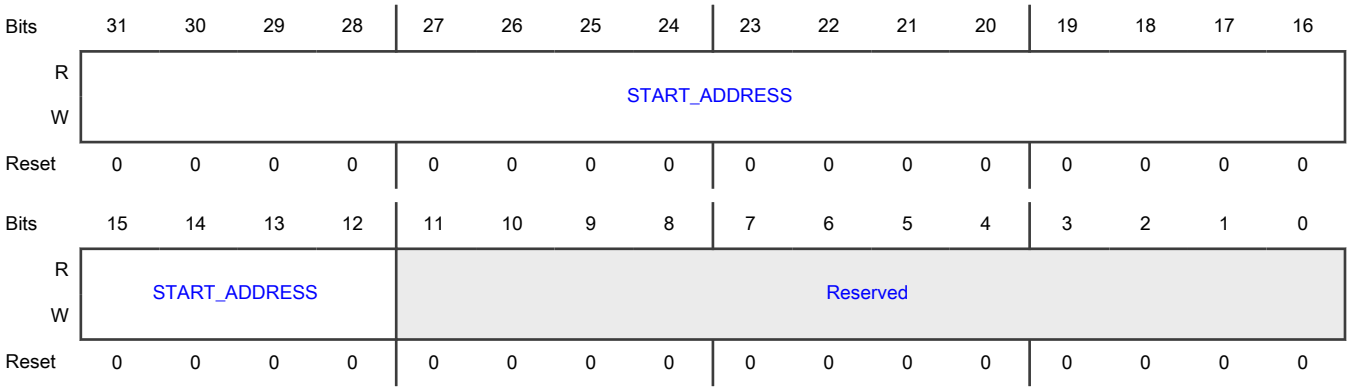
Register	Offset
AHBBUFREGIONSTART0	440h
AHBBUFREGIONSTART1	448h

Table continues on the next page...

Table continued from the previous page...

Register	Offset
AHBBUFREGIONSTART2	450h
AHBBUFREGIONSTART3	458h

Diagram



Fields

Field	Description
31-12 START_ADDRESS	Start Address of Prefetch Sub-Buffer Region Indicates start address. The start address must be at least four-KB aligned. The address used here is the AHB address that is compared with the system bus address to determine if an AHB read hits in the region.
11-0 —	Reserved

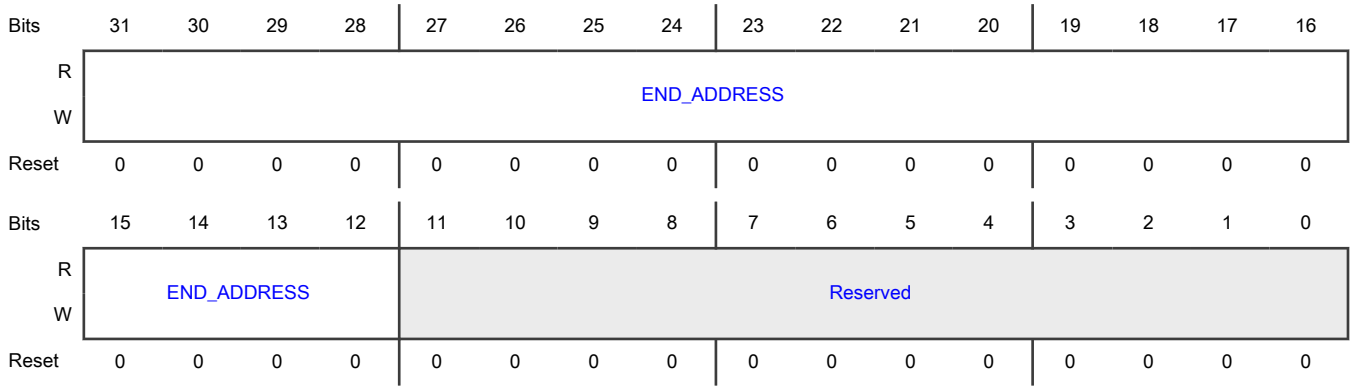
36.8.2.33 Receive Buffer Region a End Address (AHBBUFREGIONEND0 - AHBBUFREGIONEND3)

Indicates the end address of AHB receive buffer address region.

Offset

Register	Offset
AHBBUFREGIONEND0	444h
AHBBUFREGIONEND1	44Ch
AHBBUFREGIONEND2	454h
AHBBUFREGIONEND3	45Ch

Diagram



Fields

Field	Description
31-12 END_ADDRESS	End Address of Prefetch Sub-Buffer Region Indicates end address. The end address must be at least four-KB aligned. The address used here is the AHB address that is compared with the system bus address to determine if an AHB read hits in the region.
11-0 —	Reserved

36.9 AHB memory map definition

This section describes FlexSPI module AHB memory map in detail.

36.9.1 AHB memory map for serial flash memory and RAM access

AHB read and write access for serial flash and RAM memory are mapped to a specific address range. See the system memory map for specific address ranges supported.

AHB bus features supported for serial flash and RAM memory reading:

- Cacheable and non-cacheable access
- Prefetch enable and disable
- Burst size: 8, 16, 32, 64 bits
- All burst types: types: SINGLE, INCR, WRAP4, INCR4, WRAP8, INCR8, WRAP16, INCR16

AHB bus features for Serial RAM writing:

- Bufferable and Non-Bufferable access
- Burst size: 8, 16, 32, 64 bits
- All burst types: SINGLE, INCR, WRAP4, INCR4, WRAP8, INCR8, WRAP16, INCR16

See Flash memory access via AHB command for details about AHB access to serial flash memory and RAM.

36.9.2 AHB Memory Map for IP RX FIFO read access

See chip-specific section for the address range mapped for AHB read access.

AHB Bus feature supported for IP RX FIFO reading:

- Burst size: 8/16/32/64 bits
- All burst type: SINGLE/INCR/WRAP4/INCR4/WRAP8/INCR8/WRAP16/INCR16

Refer Reading data from IP receive FIFO for more details about IP RX FIFO reading.

36.9.3 AHB Memory Map for IP TX FIFO write access

See chip-specific section for the address range mapped for AHB write access.

AHB Bus feature supported for IP TX FIFO writing:

- Burst size: 8/16/32/64 bits
- All burst type: SINGLE/INCR/WRAP4/INCR4/WRAP8/INCR8/WRAP16/INCR16

Refer Writing data to IP transmit FIFO for more details about IP TX FIFO filling.

Chapter 37
FlexSPI Address Remap (AHB_ADDR_REMAP)

37.1 Overview

This module performs a simple address remapping or translation function for system AHB accesses targeted for external SPI memories for Cortex-M33 and EdgeLock secure enclave (ELE). This functionality is provided to simplify the software (SW) creation and build process for systems that support "over the air" (OTA) code updates.

37.2 Functional description

37.2.1 Partition and Connectivity

The AHB_ADDR_REMAP module is located as a system crossbar switch slave module and upstream of the TRDC memory region checker which in turn is upstream from the actual FlexSPI external memory controller. So, the memory access bus connections are:

```
CPU -> Crossbar Slave -> AHB_ADDR_REMAP -> *RDC MRCn -> FlexSPI
```

The AHB_ADDR_REMAP module supports a Control register including the address remap constant. The AHB_ADDR_REMAP slave peripheral space is shared with the FlexSPI's programming model with the remap module controlling and managing the shared IPS address space and bus transactions.

The offset addresses {0x000-0x7FF, 0xC00-0xFFF} are mapped to the FlexSPI while addresses 0x800 - 0xBFF are allocated to the AHB_ADDR_REMAP module.

37.3 Application Information

In the following use case, application code is stored in an external SPI memory. The code typically assumes it is based at address zero so the SW linked memory image reflects this assumption in the resulting binary object file loaded into the memory. For devices that support OTA updates, a significant complication occurs when a new code image is downloaded to the device. Typically, the new code image will be linked into a formerly-unused external memory address space (as the original image is also needed for rollback considerations) and the complication occurs for the application code and how it references the new memory image - the application code must be rewritten as the original "based at address zero" assumption is no longer valid.

The preferred solution with the "based at address zero" assumption can be retained if all addresses are remapped/translated to the new memory image by adding a fixed constant to all access addresses.

Consider the following memory layout example. In the i.MX93, there is a FlexSPI0 external memory controller with the address space mapped to a base address of 0x1400_0000. This is "address zero" for the software application code. At reset, the address remap function is disabled so the addresses generated by the CPU are used directly. Further, assume that the second application code image is mapped to FlexSPI0 address offset of 68 KB, that is, the second "base address" is defined as FlexSPI0 base address + 68KB, 0x1400_0000 + 0x0001_1000 (68K) = 0x1401_1000. When this code image is active, the translated address constant is defined as 0x0001_1000 so the entire code image is remapped by this constant. The result is the application code retains the "based at address zero" concept but the physical addresses sent to the FlexSPI0 control are remapped to the correct physical address:

```
CPU "Virtual" + Remap Address = FlexSPI0 "Physical" // Application SW View
 0x1400_0000     0x0001_1000        0x1401_1000     // offset = 0x0000_0000
 0x1400_1000     0x0001_1000        0x1401_2000     // offset = 0x0000_1000
      ...
 0x1401_0000     0x0001_1000        0x1402_1000     // offset = 0x0001_0000
```

```
      ...
0x1401_0FFF     0x0001_1000        0x1402_1FFF     // offset = 0x0001_0FFF
```

37.4 Memory map and register definition

The AHB_ADDR_REMAP module supports a Control register that contains the remap address constant and control bits. The register can only be accessed by Secure Privileged mode accesses on the IPS slave peripheral bus. Attempted accesses in any other mode are error terminated.

37.4.1 AHB_ADDR_REMAP register descriptions

37.4.1.1 AHB_ADDR_REMAP memory map

AHBRM1 base address: 5E_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
800	Control (CR)	32	RW	0000_0000

37.4.1.2 Control (CR)

Reads of CR depend on the write values; writes of CR update the implemented fields.

NOTE

The CR can only be written while the AHB master bus inputs - Cortex-M33, EdgeLock secure enclave (ELE) are quiescent. Stated differently, writes to the CR register while AHB master bus is processing bus transactions could lead to incoherent and incorrect processing. It is expected the CR register is written once during system startup and remains static until the next reset.

Offset

Register	Offset
CR	800h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				RADDR											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RADDR								Reserved				LK	Reserved	Reserved	ENB
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-7 RADDR	Remap address It is added to the AHB system virtual address to form the physical address for the external FlexSPI memory. The address remap function is defined in terms of the VADDR (virtual address) and PADDR (physical address): PADDR[31:0] = {VADDR[31:28], (VADDR[27:7] + RADDR[27:7]), VADDR[6:0]} **NOTE** The AMBA-AHB bus specification defines that incrementing burst transfers cannot cross any 1 KB address boundary. Accordingly, the RADDR field must be set to a 0-modulo-1KB (or greater) value so the remap logic does not introduce any illegal boundary crossings.
6-4 —	Reserved
3 LK	Sticky lock bit Once set, it disables any subsequent writes to the CR register until the next reset .
2 —	Reserved
1 —	Reserved
0 ENB	Enable Remap 0 - The module is disabled and all input AHB addresses and attributes are simply routed to the output AHB address and attributes buses. 1 - All AHB addresses from the Cortex-M33 and ELE are remapped using the constant offset defined by the CR[RADDR] field. AHB accesses from other bus masters are simply passed through the module as if it was disabled.

Chapter 38
Ultra Secured Digital Host Controller (uSDHC)

38.1 Chip-specific USDHC information

Table 225. Reference links to related information

Topic	Related module	Reference
Full description	Arm Cortex CM33 Platform (CM33)	CM33 Platform
Full description	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing

The chip includes three instances of USDHC that provide simultaneous support for all three of 1x eMMC, 1x SD Card and 1x SDIO (for WIFI connectivity).

NOTE

Write/Read to the Reserved locations will not generate a transfer error.

NOTE

i.MX93 supports mechanical reset to reset eMMC, if uSDHC1 external signal RESET_B is used to reset the eMMC, only 1-bit/4-bit data transfer can be used.

38.2 Overview

uSDHC provides the interface between the host system and the eMMC, SD card, and SDIO as shown in Figure 126. The module acts as a bridge, passing host bus transactions to the eMMC, SD card, and SDIO by sending commands and performing data accesses to/from the cards. It handles the SD card/SDIO/eMMC protocols at the transmission level.

Table 226. Updated terms

Updated term	Deprecated term
Target	Slave
uSDHC controller	Master (related to device)
Initiator	Master (related to AXI bus)
Placeholder	Dummy

38.2.1 Block diagram

The following figure illustrates the block diagram for uSDHC. Dual port SRAM is FIFO for write/read. For more information, see Data buffer.

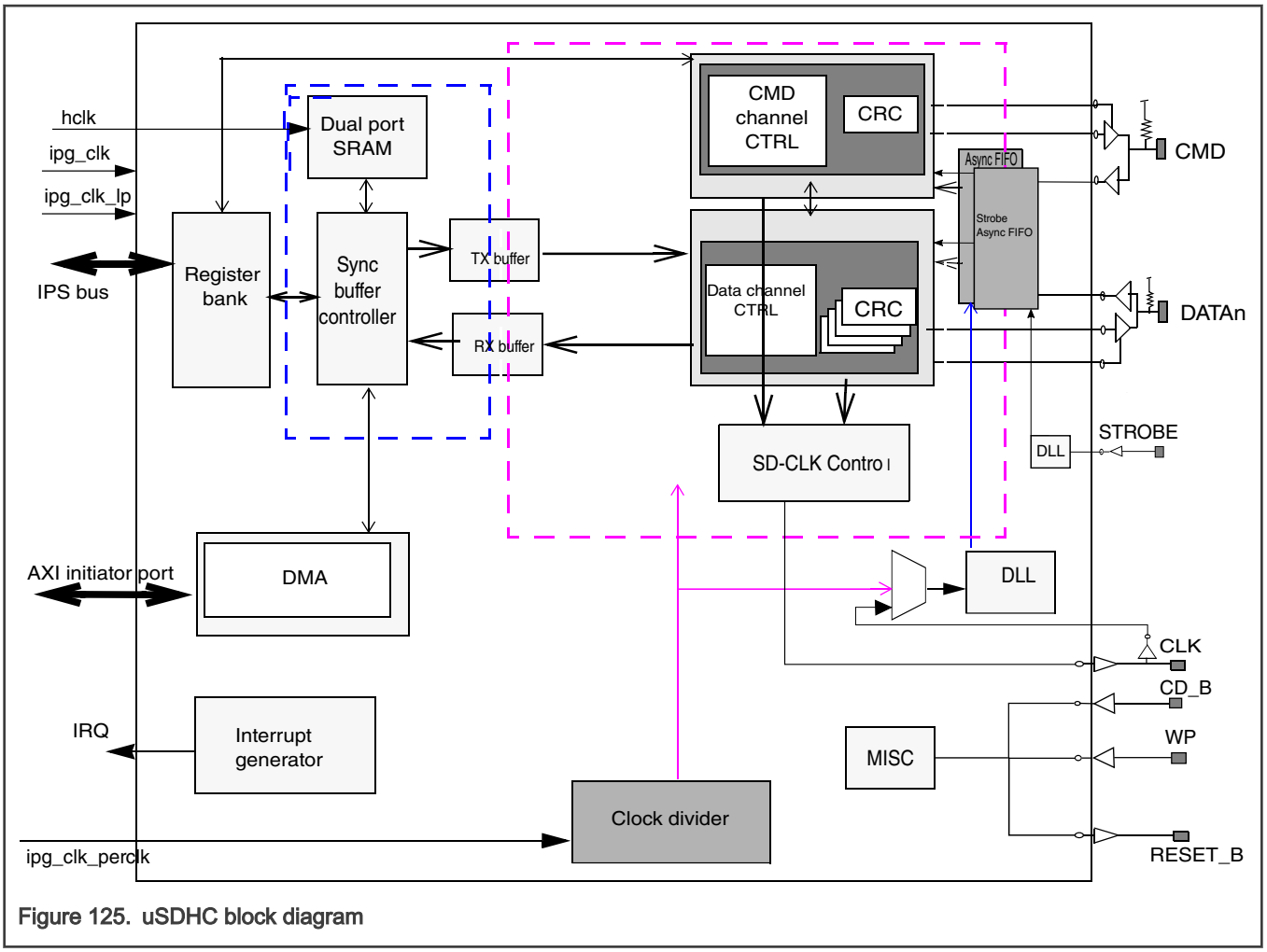


Figure 125. uSDHC block diagram

The figure below shows the System connection of uSDHC:

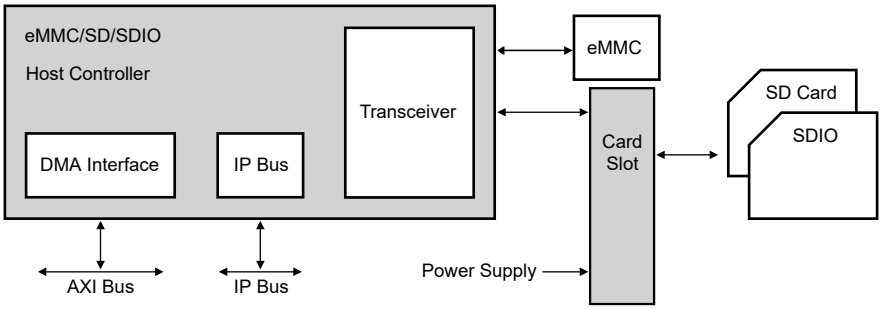


Figure 126. System connection of uSDHC

The following are brief descriptions of the cards supported by uSDHC:

- The Embedded MultiMediaCard (eMMC) is a universal low-cost data storage and communication media designed to cover a wide array of applications including mobile video and gaming. Previous eMMC were based on a 7-pin serial bus with a single data pin, while the new high speed eMMC communication is based on an advanced 11-pin serial bus designed to operate in the low-voltage range.
- The Secure Digital Card (SD) is an evolution of the old eMMC technology. It is specifically designed to meet the security, capacity, performance, and environmental requirements inherent in newly emerging audio and video consumer electronic

devices. The physical form factor, pin assignment, and data transfer protocol are backward compatible with the old eMMC (with some additions).

- Under the SD protocol, system connection can be categorized into memory card, I/O card, and combo card, which have both memory and I/O functions. The memory card invokes a copyright protection mechanism that complies with the security of the SDMI (Secure Digital Music Initiative) standard. The I/O card, which is also known as SDIO, provides high-speed data I/O with low-power consumption for mobile electronic devices.

38.2.2 Features

The list given below shows the features of the uSDHC module:

- Conforms to the SD Host Controller Standard Specification version 2.0/3.0
- Compatible with the eMMC System Specification version 4.2/4.3/4.4/4.41/4.5/5.0/5.1
- Compatible with the SD Memory Card Specification version 3.0 and supports the Extended Capacity SD Memory Card
- Compatible with the SDIO Specification version 2.0/3.0
- Designed to work with SD Memory, miniSD Memory, SDIO, miniSDIO, SD Combo, eMMC, MMC plus, and MMC RS cards
- For card bus clock frequency: See the product Data Sheet for the clock frequency of each supported mode.
- Supports 1-bit/4-bit SD and SDIO modes, and 1-bit/4-bit/8-bit eMMC modes
  - Up to 400 Mbps of data transfer for SDIO using four parallel data lines in the Single Data Rate (SDR) mode
  - Up to 400 Mbps of data transfer for SDIO using four parallel data lines in the Dual Data Rate (DDR) mode
  - Up to 400 Mbps of data transfer for SDXC cards using four parallel data lines in the Single Data Rate (SDR) mode
  - Up to 400 Mbps of data transfer for SDXC card using four parallel data lines in the Dual Data Rate (DDR) mode
  - Up to 1600 Mbps of data transfer for eMMC using eight parallel data lines in the Single Data Rate (SDR) mode
  - Up to 3200 Mbps of data transfer for eMMC using eight parallel data lines in the Dual Data Rate (DDR) mode
- Supports single block/multi-block read and write
- Supports block sizes of 1 ~ 4096 bytes
- Supports the write protection switch for write operations
- Supports both synchronous and asynchronous abort
- Supports pause during the data transfer at block gap
- Supports SDIO Read Wait and Suspend Resume operations
- Supports Auto CMD12 for multi-block transfer for memory device
- Host can initiate non-data transfer command while data transfer is in progress
- Allows cards to interrupt the host in 1-bit and 4-bit SDIO modes; also supports interrupt period
- Supports internal DMA capabilities
- Supports voltage selection by configuring vendor-specific register bit VEND_SPEC[VSELECT]
- Supports Advanced DMA to perform linked memory access
- Supports Command Queue mechanism

NOTE

This block can support all the above-listed speed mode and maximum data throughput. However, these may be specific to the device. See the corresponding chip-specific information or the device data sheet for accurate details.

38.3 Functional description

The following sections provide a brief functional description of the major system blocks, including the data buffer, DMA AXI interface, register bank as well as IP Bus interface, dual-port memory wrapper, data/command controller, clock and reset manager, and clock generator.

38.3.1 Modes and operations

38.3.1.1 Data transfer modes

The uSDHC module can select the following modes for data transfer:

- SD 1-bit
- SD 4-bit
- eMMC 1-bit
- eMMC 4-bit
- eMMC 8-bit
- Identification mode (up to 400 kHz)
- eMMC full-speed mode (up to 26 MHz)
- eMMC high-speed mode (up to 52 MHz)
- eMMC HS200 mode (up to 200 MHz)
- eMMC HS400 mode (200 MHz both edges)
- eMMC DDR mode (52 MHz both edges)
- SD card or SDIO full-speed mode (up to 25 MHz)
- SD card or SDIO high-speed mode (up to 50 MHz)
- SD card or SDIO UHS-I mode (up to 208 or 100 MHz in SDR mode, up to 50 MHz in the DDR mode)

NOTE

This block can support all the above listed speed mode and maximum clock frequency. However, these may be specific to the device. See the corresponding chip-specific information or the device data sheet for accurate details.

38.3.2 Data buffer

The uSDHC module uses one configurable data buffer to transfer data between the system bus (IP bus or advanced extensible interface (AXI) bus) and the SD card in an optimized manner, maximizing throughput between the two clock domains (IP peripheral clock and the AXI initiator clock). The buffer is used as a temporary storage for transferring data between the host system and the card. The watermark levels for read and write are both configurable and can range between 1 to 128 words. When using internal DMA mode, The watermark level for read should be configured to 16 and watermark level for write should be configured to integral multiple of 16 to get max AXI bus throughput. The next figure provides the uSDHC buffer scheme as buffer control and buffer RAM wrapper.

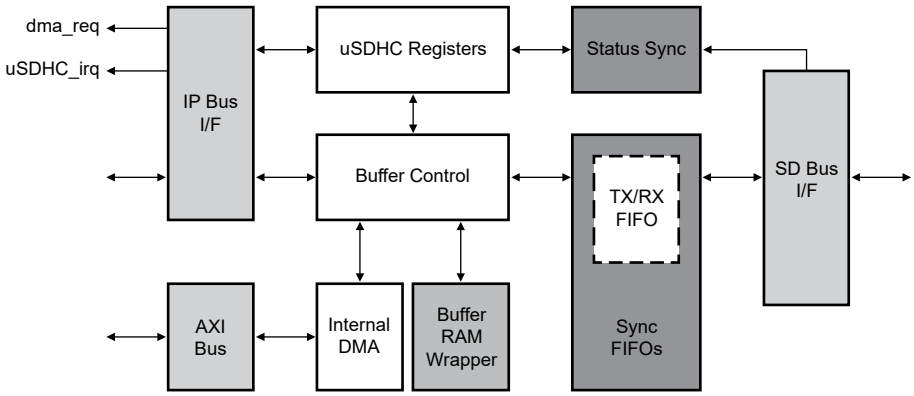


Figure 127. uSDHC buffer scheme

Here are 2 transfer modes to access the data buffer:

- CPU polling mode:
  - For a host-read operation, when the number of words received in the buffer meets or exceeds the RD_WML watermark value, by polling the BRR bit, the host driver can read the Buffer Data Port register to fetch the amount of words set in the RD_WML register from the buffer. The write operation is similar. For more information on the process of writing operation, see Write operation sequence.
- Internal DMA mode (includes simple and advanced DMA accesses):
  - The internal DMA access, either by simple or advanced DMA, is over the AXI bus.

For a read operation, when there are more words in the buffer than the amount set in WTMK_LVL[RD_WML], the internal DMA starts fetching data over the AXI bus.

The Write operation functions in a similar manner—sequential and contiguous access is necessary to ensure that the pointer address value is correct. Random or skipped access is not possible. The byte order, by reset, is little endian mode. The actual byte order is swapped inside the buffer, according to the endian mode configured by software (see the following figures). For a host write operation, the byte order is swapped after the data is fetched from the buffer and ready to send to the SD Bus. For a host read operation, the byte order is swapped before the data is stored in the buffer.

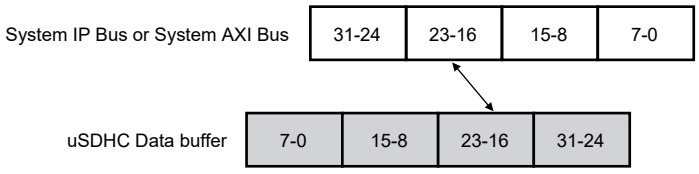


Figure 128. Data swap between system bus and uSDHC data buffer in the byte little endian mode

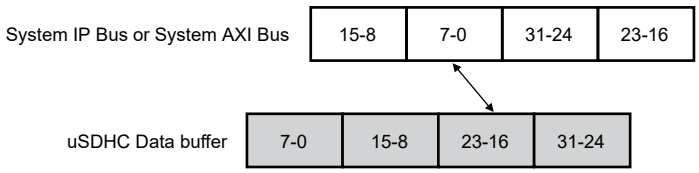


Figure 129. Data swap between system bus and uSDHC data buffer in half word big endian mode

38.3.2.1 Write operation sequence

There are 2 ways to write data into the buffer when the user transfers data to the card:

- Processor core polling through the BWR bit in the Interrupt Status register (interrupt or polling)
- Internal DMA

When the internal DMA is not used, the DMAEN bit in the Transfer Type register is not set when the command is sent, uSDHC asserts a DMA request when the amount of buffer space exceeds the value set in WTMK_LVL[WR_WML] and is ready for receiving new data. At the same time, uSDHC sets the BWR bit. The buffer write ready interrupt is generated if it is enabled by software.

When internal DMA is used, uSDHC does not inform the system before all the required number of bytes are transferred (if no error is encountered). When an error occurs during the data transfer, uSDHC stops the data transfer and abandons the current block. The host driver should read the contents of the DMA System Address register to obtain the starting address of the abandoned data block. If the current data transfer is in multi-block mode, uSDHC does not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The host driver sends CMD12 in this scenario and restarts the write operation from that address. It is recommended that a software reset for Data be applied before the transfer is restarted.

The uSDHC module does not start data transmission until the buffer has been filled with the number of words set in the WR_WML register. If the buffer is empty and the host system does not write data in time, uSDHC stops the CLK to avoid the data buffer underrun situation.

38.3.2.2 Read operation sequence

There are 2 ways to read data from the buffer when the data is received from the card:

- Processor core polling through the BRR bit in the Interrupt Status register (interrupt or polling)
- Internal DMA

When internal DMA is not used (DMAEN bit in Transfer Type register is not set when the command is sent), uSDHC asserts a DMA request when the amount of data exceeds the value set in the RD_WML register, which is available and ready for system fetching data. At the same time, uSDHC sets the BRR bit. The buffer read ready interrupt is generated if it is enabled by the software.

When internal DMA is used, uSDHC does not inform the system before all the required number of bytes are transferred (if no error is encountered). When an error occurs during the data transfer, uSDHC stops the data transfer and abandons the current block. The host driver should read the content of the DMA System Address register to get the starting address of the abandoned data block. If the current data transfer is in multi-block mode, uSDHC does not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The host driver sends CMD12 in this scenario and restarts the read operation from that address. It is recommended that a software reset (in register RSTD) for data be applied before the transfer is restarted.

For any read transfer mode, uSDHC does not start data transmission until the number of words set in the RD_WML register are in buffer. If the buffer is full and the host system does not read data in time, uSDHC stops the CLK to avoid the data buffer overrun situation.

38.3.2.3 Data buffer and block size

In the uSDHC module, the data buffer can hold up to 256 words (32-bit) and the watermark levels for write and read can be configured accordingly. The user needs to know the buffer size for the buffer operation during a data transfer to utilize it in the most optimized way. The host driver may configure the value according to the system and requirement.

During a multi-block data transfer, the block length can be set to any value between 1 and 4096 bytes, satisfying the requirements of the external card. The only restriction is from the external card, which can be limited in size or support of a partial block access (which is not the integer times of 512 bytes). That means, the largest block size is 512 bytes.

As uSDHC treats each block individually, for block sizes which are not multiples of four (not word-aligned), stuffed bytes are required at the end of each block. For example, if the block size is 7 bytes and there are 12 blocks to write, the software side must write twice for each block. For each block, the ending byte is abandoned by uSDHC because it only sends 7 bytes to the card and picks data from the following system write, resulting in 24 beats of write access in total.

38.3.2.4 Dividing large data transfer

This SDIO command CMD53 definition limits the maximum data size of data transfers according to the following formula:

Max data size = Block size x Block count

The length of a multiple block transfer needs to be in multiples of block size. If the total data length cannot be divided evenly into a multiple of the block size, then there are two options to transfer the data. These two options depend on the function and the card design. Option 1 is for the host driver to split the transaction. The remainder of the block size data is then transferred by using a single block command at the end. Option 2 is to add placeholder data in the last block to fill the block size. For option 2, the card must manage the removal of the placeholder data. Only for write, uSDHC sends data to the card.

See Figure 130 for an example showing the division of large data transfers, assuming a kind of WLAN SDIO that only supports a block size of up to 64 bytes. Although uSDHC supports a block size of up to 4096 bytes, SDIO can only accept a block size of less than 64 bytes, so the data must be divided (see the example below).

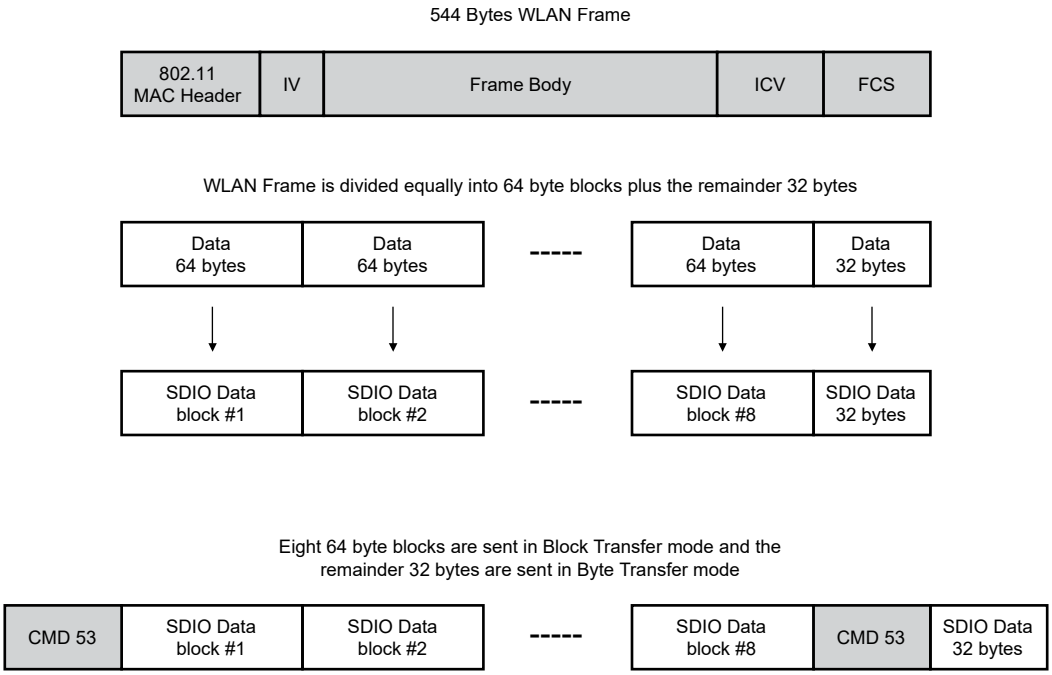


Figure 130. Example for dividing large data transfers

38.3.3 DMA AXI interface

The internal DMA implements a DMA engine and the AXI initiator. When the internal DMA is enabled, but the BWR and BRR bits are set if the BWRSEN and BRRSEN bits have been set in the Interrupt Status Enable register. See Figure 131 for an illustration of the DMA AXI interface block.

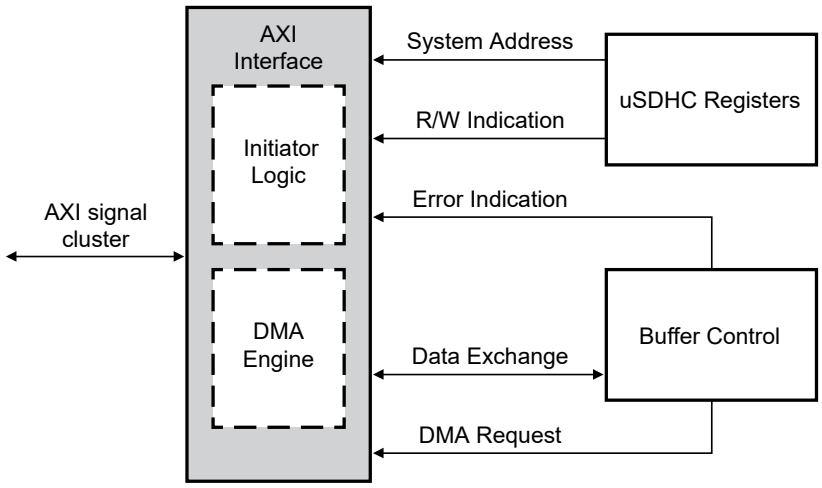


Figure 131. DMA AXI interface block

38.3.3.1 Internal DMA request

If the watermark level requirement is met in data transfer or if the last data of current block is ready in the data buffer, and the Internal DMA is enabled, the data buffer block sends a DMA request to AXI interface.

The delay in response from the internal DMA engine depends on the system AXI bus loading and the priority assigned to uSDHC. The DMA engine does not respond to the request during its burst transfer, but is ready to serve as soon as the burst is over. The data buffer de-asserts the request if the data buffer space (for write) or bytes in data buffer is smaller than the watermark level. Upon access to the buffer by internal DMA, the data buffer updates its internal buffer pointer, and when the watermark level is satisfied or the last data of the current block is ready in the data buffer, another DMA request is sent.

The data transfer is in the block unit, and the subsequent watermark level is always automatically set as the remaining number of words. For instance, for a multi block data read with each block size of 31 bytes, and the watermark level set to six words (24 bytes). After the first burst transfer, if there are greater than or equal to two words in the buffer, which might contain some data of the next block), another DMA request is sent. This is because the remaining number of words to transfer for the current block is ceiling((31 - 6 * 4) / 4) = 2. The uSDHC module reads two words out of the buffer, with seven valid bytes and one stuffed byte.

38.3.3.2 AXI initiator interface

It is possible that the internal AXI DMA engine could fail during the data transfer. Upon detection of an AXI bus error during DMA transfer, the DMA engine stops the transfer and goes to the idle state. At that point, the internal data buffer stops receiving incoming data and sending out data. The DMAE bit in the Interrupt Status register is generated to host CPU to report a bus error condition.

After the DMAE interrupt is received, the software sends a CMD12 to abort the current transfer and read the DS_ADDR bits of the DMA System Address register to get the starting address of the corrupted block. After the DMA error is fixed, the software should apply a data reset and restart the transfer from this address to recover the corrupted block.

38.3.3.3 ADMA engine

In the SD host controller standard, a new DMA transfer algorithm called the Advanced DMA (ADMA) is defined. For simple DMA, after the page boundary is reached, a DMA interrupt is generated and the new system address is programmed by the host driver. The ADMA defines the programmable descriptor table in the system memory. The host driver can calculate the system address at the page boundary and program the descriptor table before executing ADMA. It reduces the frequency of interrupts to the host system. Therefore, higher speed DMA transfers could be realized because the host MCU intervention is not be needed during long DMA-based data transfers.

There are two types of ADMA in host controller: ADMA1 and ADMA2. ADMA1 can support data transfer of 4KB aligned data in system memory, and ADMA2 eliminates the restriction so that the data of any location (word aligned) and any size (less than 64 KB) can be transferred in system memory. Their formats of Descriptor Table are different.

ADMA can recognize all kinds of descriptors defined in SD host controller Standard, and if the "End" flag is detected in the descriptor, ADMA stops after this descriptor is processed.

38.3.3.3.1 ADMA concept and descriptor format

ADMA1 includes the following descriptors:

- Valid/invalid descriptor
- Nop descriptor
- Set data length descriptor
- Set data address descriptor
- Link descriptor
- Interrupt flag and end flag in descriptor

ADMA2 includes the following descriptors:

- Valid/invalid descriptor
- Nop descriptor
- Rsv descriptor (new in ADMA2)
- Set data length and address descriptor
- Link descriptor
- Interrupt flag and end flag in descriptor

ADMA starts read/write operation after it reaches the tran state, using the data length and data address analyzed from most recent descriptor(s).

For ADMA1, the valid data length descriptor is the last set type descriptor before the tran type descriptor. Every tran type descriptor triggers a transfer, and the transfer data length is extracted from the most recent set type descriptor. If there is no set type descriptor after the previous Trans descriptor, the data length is the value for previous transfer, or 0 if no set descriptor is ever met.

For ADMA2, the tran type descriptor contains both data length and transfer data address, so the tran type descriptor itself can start a data transfer.

See Figure 132 for the format of the descriptor table for ADMA1.

Address/ Page Field		Address/ Page Field		Attribute Field					
31	12	11	6	5	4	3	2	1	0
Address or Data Length		000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	31- 28	27- 12
0	0	Nop	No Operation	Don't Care	
0	1	Set	Set Data Length	0000	Data Length
1	0	Tran	Transfer Data	Data Address	
1	1	Link	Link Descriptor	Descriptor Address	

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 132. Format of the ADMA1 descriptor table

See Figure 133 for the concept and access method of the descriptor table for ADMA1.

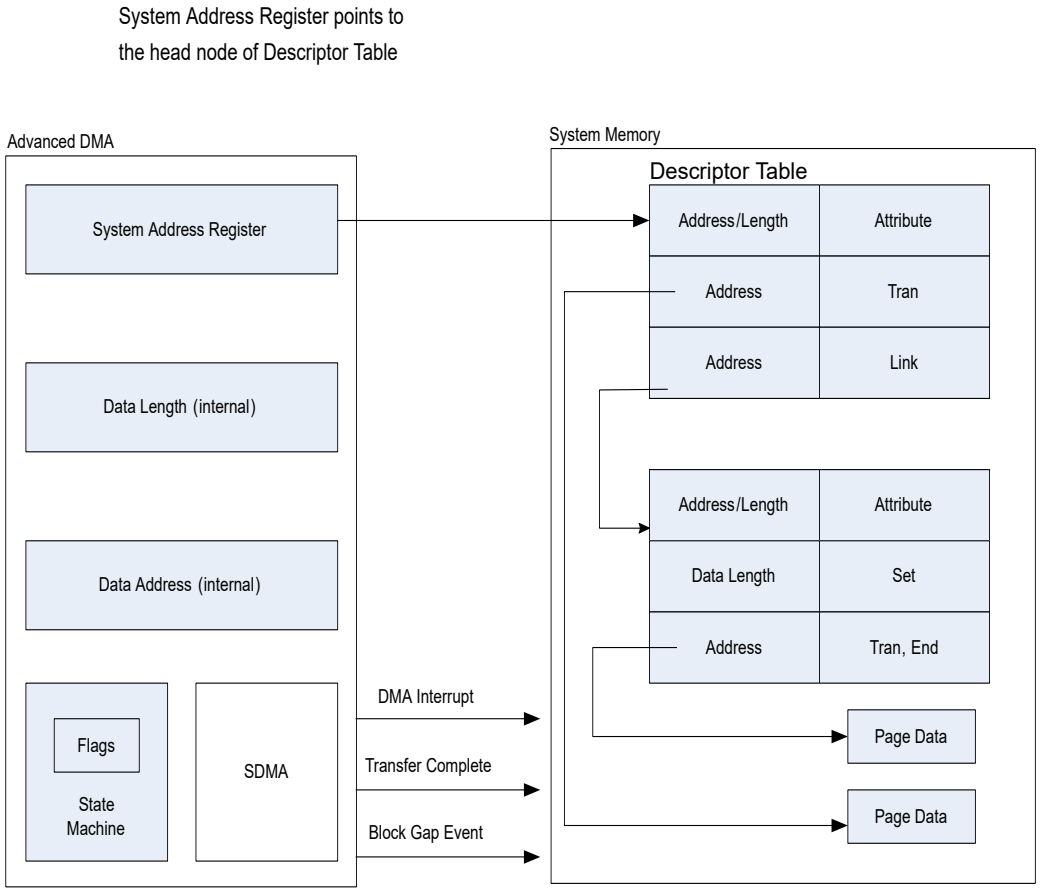


Figure 133. Concept and access method of the ADMA1 descriptor table

The figure below explains the ADMA2 format. ADMA2 deals with the lower 32-bit first, and then the higher 32-bit. If the 'Valid' flag of descriptor is 0, it ignores the higher 32-bit. The Address field should be set to word aligned (lower 2-bit is always set to 0). Data length is in byte unit.

Address Field		Length		Reserved		Attribute Field					
63	32	31	16	15	06	05	04	03	02	01	00
32-bit Address		16-bit length		0000000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	Operation
0	0	Nop	No Operation	Don't Care
0	1	Rsv	Reserved	Same as Nop. Read this line and go to next one
1	0	Tran	Transfer Data	Transfer data with address and length set in this descriptor line
1	1	Link	Link Descriptor	Link to another descriptor

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 134. Format of the ADMA2 descriptor table

See Figure 135 for the concept and access method of the descriptor table for ADMA2.

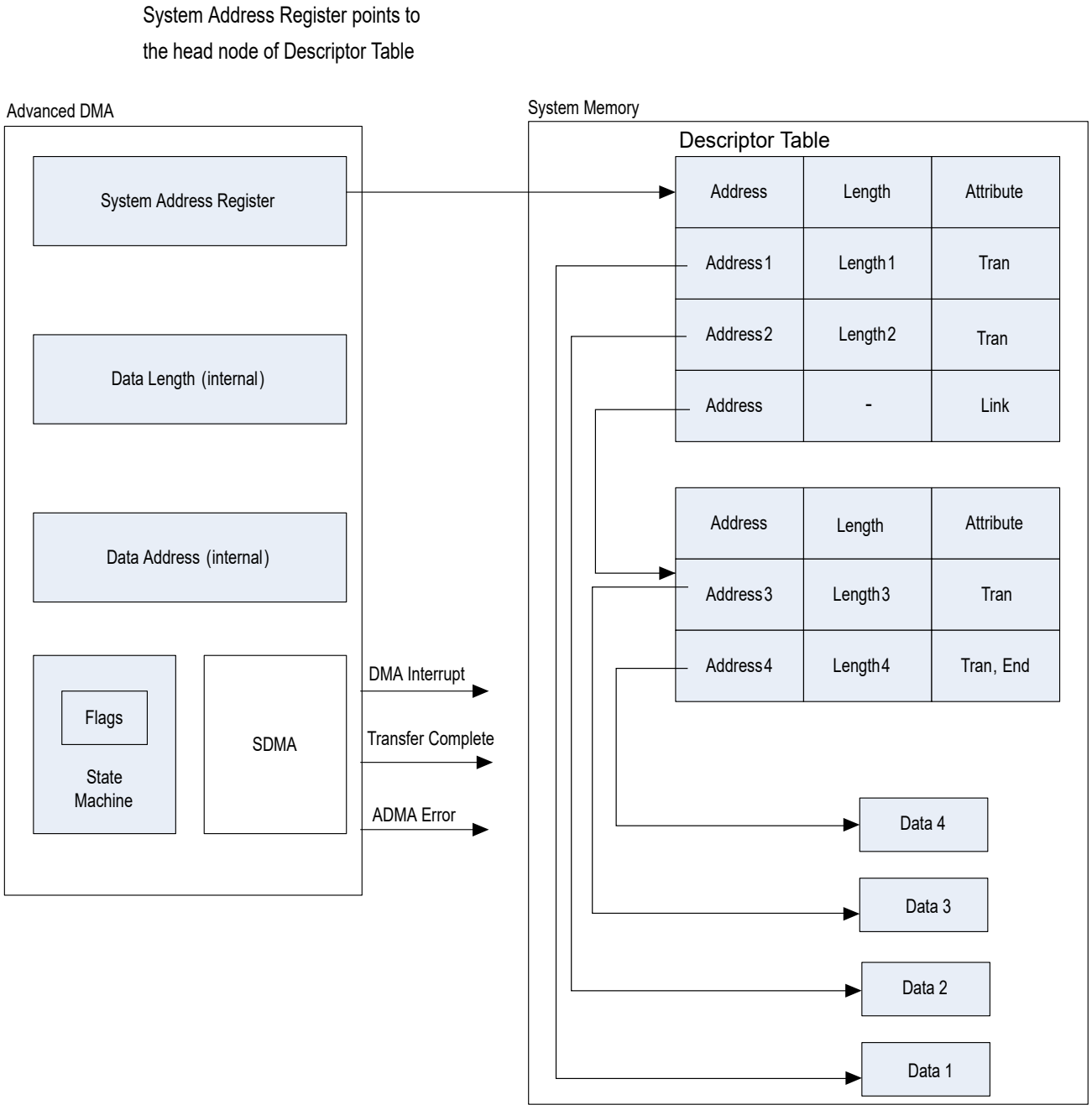


Figure 135. Concept and access method of the ADMA2 descriptor table

38.3.3.3.2 ADMA interrupt

If the interrupt flag descriptor is set, ADMA generates an interrupt according to various types of descriptors.

For ADMA1:

- Set type of descriptor: interrupt is generated when data length is set.
- Tran type descriptor: interrupt is generated when this transfer descriptor is complete, which means the data communication between the host system and uSDHC is finished.
- Link type of descriptor: interrupt is generated when new descriptor address is set.

- Nop type of descriptor: interrupt is generated just after this descriptor is fetched.

For ADMA2:

- Tran type of descriptor: interrupt is generated when this transfer descriptor is complete, which means the data communication between the host system and uSDHC is finished.
- Link type of descriptor: interrupt is generated when new descriptor address is set.
- Nop/Rsv type of descriptor: interrupt is generated just after this descriptor is fetched.

38.3.3.3.3 ADMA error

The ADMA stops whenever an error is encountered. These errors include:

- Fetching descriptor error
- AXI response error
- Data length mismatch error

An ADMA descriptor error is generated when it fails to detect a "valid" flag in the descriptor. If an ADMA descriptor error occurs, the interrupt is not generated even if the "Interrupt" flag of this descriptor is set.

When BLKCNTEN bit is set, data length set in register BLK_ATT must be equal to the whole data length set in descriptor, otherwise data length mismatch error is generated.

When BLKCNTEN bit is not set, then the whole data length set in descriptor is a multiple of block lengths; otherwise, when data set in the descriptor nodes are not performed at block boundaries, then data mismatch errors occur.

38.3.4 Register bank with IP bus interface

Register accesses through the IP bus interface are on the register bank. See Figure 136 for the block diagram.

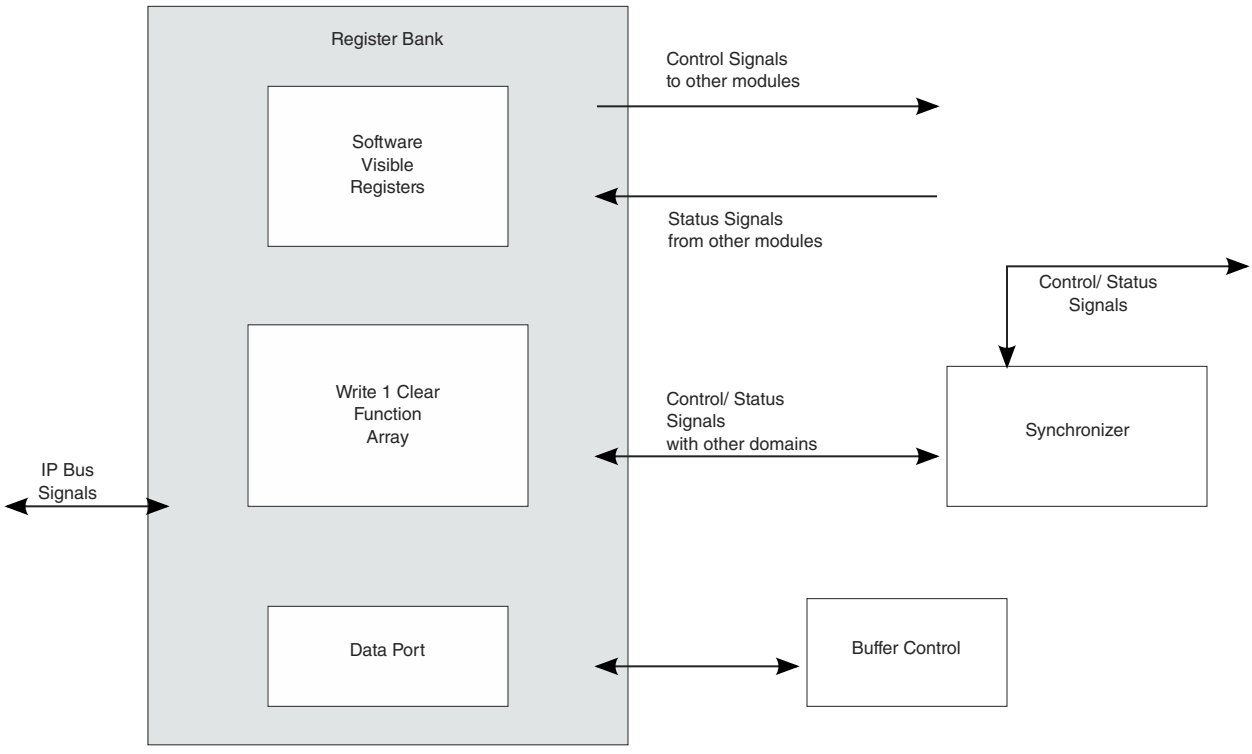


Figure 136. Register bank diagram

Only a 32-bit access is allowed, and no partial read/write is supported; therefore, all accesses are word aligned.

Access to an unimplemented address within the register memory map space does not generate a transfer error.

38.3.4.1 SD protocol unit

The SD protocol unit deals with all SD protocol affairs.

The SD protocol unit performs the following functions:

- Acts as the bridge between the internal buffer and the SD bus
- Sends the command data as well as its argument serially
- Stores the serial response bit stream into corresponding registers
- Detects the bus state on the CMD/DAT lines
- Monitors the interrupt from SDIO
- Asserts the read wait signal
- Gates off the SD clock when buffer is announcing danger status
- Detects the write protect state

The SD protocol unit consists of four sub-modules:

- SD control misc
- Command control
- Data control
- Clock control

38.3.4.2 SD control miscellaneous

In the SD control miscellaneous unit:

- The card detection (including CD_B and DATA3 for card detection) and card interrupt are implemented.
- The module monitors the signal level on all the eight data lines and the command lines. It directly routes the level values into the register bank.
- The module also detects the Write Protect (WP) line. If WP is activated as '1', writes to memory and combo cards are ignored by cards. In that case, there is no need to send write command anymore.

38.3.4.3 SD clock control

If the internal data buffer is near full (for read) or near empty (for write), the SD clock must be gated off to avoid buffer over/under-run. This module asserts the gate of the output SD clock to shut the clock off. After the buffer has space (for read) or has data (for write), the clock gate of this module opens, and the SD clock is active again.

38.3.4.4 Command control

The Command Control module deals with the transactions on the CMD line.

See Figure 137 for an illustration of the structure for the Command CRC shift register.

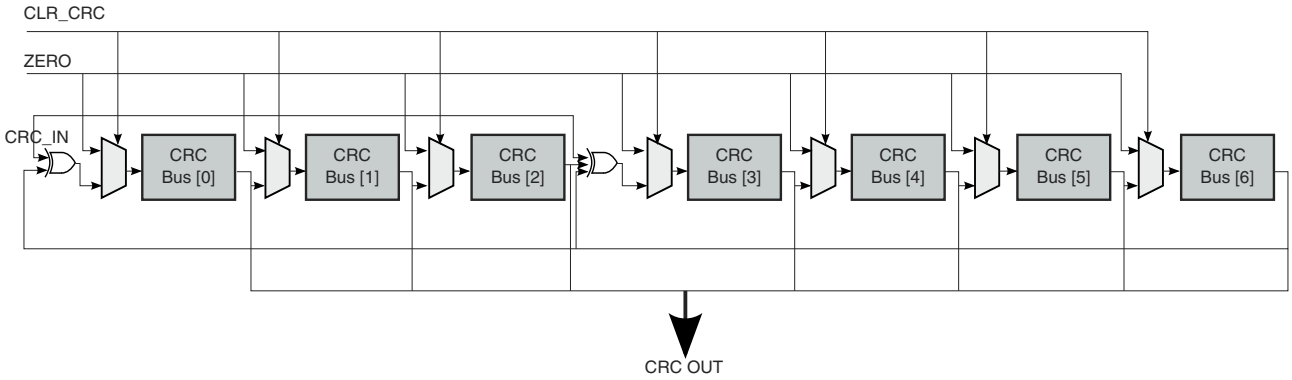


Figure 137. Command CRC shift register

The CRC polynomials for the CMD are as follows:

```
Generator polynomial: G(x) = x^7 + x^3 + 1
M(x) = (first bit) * x^n + (second bit) * x^n-1 +...+ (last bit) * x^0
CRC[6:0] = Remainder [(M(x) * x^7) / G(x)]
```

38.3.4.5 Data control

The data agent deals with the transactions on the eight data lines. Moreover, this module also detects the busy state on the DATA0 line and generates the read wait state by the request from the transceiver.

The CRC polynomials for the data are as follows:

```
Generator polynomial: G(x) = x^16 + x^12 + x^5 +1

M(x) = (first bit) * x^n + (second bit) * x^n-1 +...+ (last bit) * x^0

CRC[15:0] = Remainder [(M(x) * x^16) / G(x)]
```

38.3.5 Card insertion and removal detection

The uSDHC module uses either the DATA3 pin or the CD_B pin to detect card insertion or removal. It is controlled by SoC pad or other logic. When there is no card on the eMMC/SD bus, the DATA3 is pulled to a low voltage level by default.

When any card is inserted to or removed from the socket, uSDHC detects the logic value changes on the DATA3 pin and generates an interrupt. When the DATA3 pin is not used for card detection (for example, it is implemented in GPIO), the CD_B pin must be connected for card detection. Whether DATA3 is configured for card detection or not, the CD_B pin is always a reference for card detection. Whether the DATA3 pin or the CD_B pin is used to detect card insertion, uSDHC sends an interrupt (if enabled) to inform the Host system that a card is inserted.

38.3.6 Power management and wakeup events

When there is no operation between uSDHC and the card through the SD bus, the user can completely disable the peripheral clock and base clock in the chip-level clock control module to save power. When the user needs to use uSDHC to communicate with the card, it can enable the clock and start the operation.

In some circumstances, when the clocks to uSDHC are disabled, for instance, when the system is in low-power mode, there are some events for which the user needs to enable the clock and handle the event. These events are called wakeup interrupts. The uSDHC module can generate these interrupts even when there are no clocks enabled. The three interrupts that can be used as wakeup events are these:

- Card removal interrupt
- Card insertion interrupt
- Interrupt from SDIO (not available in multiple block data transfers)

These three wakeup events (or wakeup interrupts) can also be used to wakeup the system from low-power modes.

NOTE

To make the interrupt a wakeup event, when all the clocks to uSDHC are disabled or when the entire system is in low power mode, the corresponding wakeup enabled bit needs to be set. Refer to Protocol Control (PROT_CTRL) for more information on the uSDHC Protocol Control register.

38.3.6.1 Setting wakeup events

For uSDHC to respond to a wakeup event, the software must set the respective wakeup enable bit before the CPU enters the sleep mode. Before the software disables the host clock, it should ensure that all the following conditions have been met:

- No read or write transfer is active
- Data and command lines are not active
- No interrupts are pending
- Internal data buffer is empty

38.3.7 eMMC fast boot

The Embedded MultiMediaCard (eMMC4.3 or later) specification adds a fast boot feature that requires hardware support. There are two types of fast boot modes in the eMMC4.3 or later specification: boot operation and alternative boot operation. Each type also has with-acknowledge and without-acknowledge modes.

In the boot operation mode, the uSDHC controller (eMultiMediaCard host) can read boot data from the target (eMMC device) by keeping CMD line low after power-on, or sending CMD0 with argument + 0xFFFFFFFA (optional for target), before issuing CMD1.

NOTE

For the eMMC4.3 setting, please see the eMMC4.3 specification.

38.3.7.1 Boot operation

If the CMD line is held low for 74 clock cycles and more after power-up before the first command is issued, the target recognizes that boot mode is being initiated and starts preparing boot data internally.

Within one second after the CMD line goes low, the target starts to send the first boot data to the uSDHC controller on the DATA line(s). The uSDHC controller must keep the CMD line low to read all of the boot data.

NOTE

For the purposes of this documentation, fast boot is called "normal fast boot mode".

If boot acknowledge is enabled, the target has to send acknowledge pattern '010' to the uSDHC controller within 50ms after the CMD line goes low. If boot acknowledge is disabled, the target does not send out acknowledge pattern '010'.

The uSDHC controller can terminate the boot mode with the CMD line high.

The boot operation is terminated when all the contents of the enabled boot data are sent to the uSDHC controller. After the boot operation is executed, the target gets ready for the CMD1 operation and the uSDHC controller needs to start a normal eMMC initialization sequence by sending CMD1.

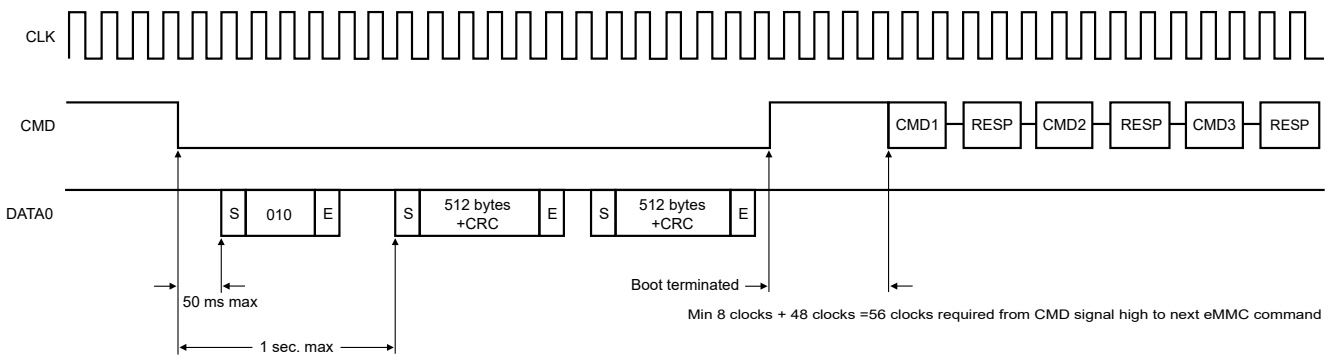


Figure 138. Embedded MultiMediaCard state diagram (normal boot mode)

38.3.7.2 Alternative boot operation

This boot function is optional for the device. If bit 0 in the extended CSD byte[228] is set to '1', the device supports the alternative boot operation.

After power-up, if the host issues CMD0 with the argument of 0xFFFFFFFA after 74 clock cycles, before CMD1 is issued or the CMD line goes low, the target recognizes that boot mode is being initiated and starts preparing boot data internally.

Within one second after CMD0 with the argument of 0xFFFFFFFA is issued, the target starts to send the first boot data to the uSDHC controller on the DATA line(s).

If boot acknowledge is enabled, the target must send the acknowledge pattern '010' to the uSDHC controller within 50ms after the CMD0 with the argument of 0xFFFFFFFA is received. If boot acknowledge is disabled, the target does not send out acknowledge pattern '010'.

The uSDHC controller can terminate the boot mode by issuing CMD0 (Reset).

Boot operation is terminated when all the contents of the enabled boot data are sent to the uSDHC controller. After the boot operation is executed, the target gets ready for the CMD1 operation and the uSDHC controller needs to start a normal eMMC initialization sequence by sending CMD1.

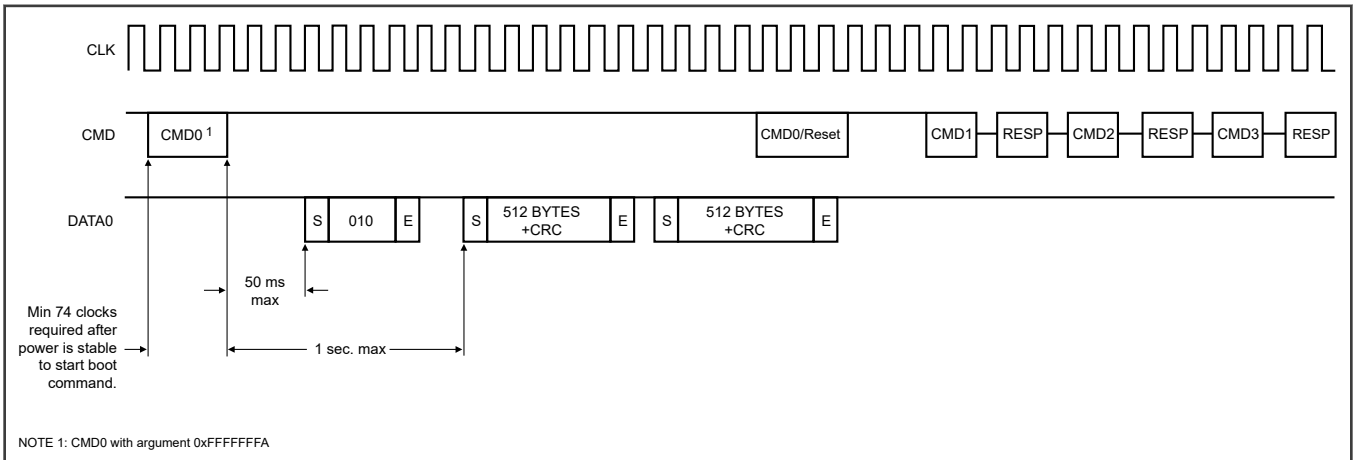


Figure 139. Embedded MultiMediaCard state diagram (alternative boot mode)

38.3.8 Commands for SD card, SDIO, and eMMC

A table containing the list of commands for the eMMC/SD card/SDIO is provided here.

Refer to the corresponding specifications for more details about the command information.

There are four kinds of commands defined to control the SD card, SDIO, and eMMC:

- Broadcast commands (bc), no response
- Broadcast commands with response (bcr), response from all cards simultaneously
- Addressed (point-to-point) commands (ac), no data transfer on the DATA
- Addressed (point-to-point) data transfer commands (adtc)

Response: A response is a token that is sent from the card to the host as an answer to a previously received command. A response is transferred serially on the CMD line.

Table 227. Commands for eMMC/SD card/SDIO

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all eMMC and SD memory cards to idle state.
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all eMMC and SD Memory cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.
CMD3[1]	ac	[31:16] RCA [15:0] stuff bits	R1 R6 (SDIO)	SET/ SEND_RELATIVE_ADDR	Assigns relative address to the card.
CMD4	bc	[31:16] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards.
CMD5	bc	[31:0] OCR without busy	R4	IO_SEND_OP_COND	Asks all SDIO in idle state to send them operation conditions register contents in the response on the CMD line.
CMD6[2]	adtc	[31] Mode 0: Check function 1: Switch function [30:8] Reserved for function groups 6 ~ 3 (All 0 or 0xFFFF) [7:4] Function group1 for command system [3:0] Function group2 for access mode	R1	SWITCH_FUNC	Checks switch ability (mode 0) and switch card function (mode 1). Refer to "SD Physical Specification V1.1" for more details.

Table continues on the next page...

Table 227. Commands for eMMC/SD card/SDIO (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD6[3]	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Switches the mode of operation of the selected card or modifies the EXT_CSD registers. Refer to "The Embedded MultiMediaCard System Specification Version 4.0 Final draft 2" for more details.
CMD7	ac	[31:16] RCA [15:0] stuff bits	R1b	SELECT/ DESELECT_CARD	Toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases, the card is selected by its own relative address and gets deselected by any other address. Address 0 deselects all.
CMD8	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	The card sends its EXT_CSD register as a block of data, with a block size of 512 bytes.
CMD9	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card-identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Forces the card to stop transmission. Only supported by memory device.
CMD13	ac	[31:16] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:16] RCA [15:0] stuff bits	-	GO_INACTIVE_STATE	Sets the card to inactive state in order to protect the card stack against communication breakdowns.

Table continues on the next page...

Table 227. Commands for eMMC/SD card/SDIO (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read and write). Default block length is specified in the CSD.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a stop command.
CMD19	adtc	[31:0] reserved bits(all 0)	R1	SEND_TUNING_BLOCK	64 bytes tuning pattern is sent for SDR50 and SDR104.
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISION follows.
CMD21	adtc	[31:0] stuff bits	R1	SEND_TUNING_BLOCK	128 clocks of tuning pattern (64 byte in 4 bit mode or 128 byte in 8 bit mode) is sent for HS200 optimal sampling point detection.
CMD22	Reserved				
CMD23	ac	[31] reliable write request [30:16] set to 0 [15:0] number of blocks	R1	SET_BLOCK_COUNT	Defines the number of blocks (read/write) and the reliable writer parameter (write) for a block read or write command.
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command is issued only once per card. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.

Table continues on the next page...

Table 227. Commands for eMMC/SD card/SDIO (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
CMD31	Reserved				
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selection of a single sector to be selected for erase.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection.
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erase all previously selected sectors.

Table continues on the next page...

Table 227. Commands for eMMC/SD card/SDIO (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD39	ac	[31:16] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8-bit (register) data fields. The command addresses a card, and a register, and provides the data for writing if the write flag is set. The R4 response contains data read from the address register. This command accesses application dependent registers which are not defined in the eMMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode.
CMD41	Reserved				
CDM42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43~51	Reserved				
CMD52	ac	[31:0] stuff bits	R5	IO_RW_DIRECT	Access a single register within the total 128k of register space in any I/O function.
CMD53	ac	[31:0] stuff bits	R5	IO_RW_EXTENDED	Accesses a multiple I/O register with a single command. Allows the reading or writing of a large number of I/O registers.
CMD54	Reserved				
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather than a standard command.
CMD56	adtc	[31:1] stuff bits [0]: RD/WR	R1b	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block is set by the SET_BLOCK_LEN command.
CMD57-59	Reserved				

Table continues on the next page...

Table 227. Commands for eMMC/SD card/SDIO (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD60	adtc	[31] WR [30:24] stuff bits [23:16] address [15:8] stuff bits [7:0] byte count	R1b	RW_MULTIPLE_REGISTER	These registers are used to control the behavior of the device and to retrieve status information regarding the operation of the device. All Status and Control registers are WORD (32-bit) in size and are WORD aligned. CMD60 is used to read and write these registers. MMC command defined by CE-ATA Digital Protocol.
CMD61	adtc	[31] WR [30:16] stuff bits [15:0] data unit count	R1b	RW_MULTIPLE_BLOCK	The host issues a RW_MULTIPLE_BLOCK (CMD61) to begin the data transfer. MMC command defined by CE-ATA Digital Protocol.
CMD62-63	Reserved				
ACMD6[4]	ac	[31:2] stuff bits [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4bit bus) to be used for data transfer. The allowed data bus widths are given in SCR register.
ACMD13[4]	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Memory Card status.
ACMD22[4]	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_SECTORS	Send the number of the written sectors (without errors). Responds with 32-bit plus the CRC data block.
ACMD23[4]	ac	[31:23] stuff bits [22:0] Number of blocks	R1	SET_WR_BLK_ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for fast Multiple Block WR command). "1"=default(one write block).
ACMD41[4]	bcr	[31:0] OCR	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) contents in the response on the CMD line.
ACMD42[4]	ac	[31:1] stuff bits [0] set_cd	R1	SET_CLR_CARD_DETECT	Connect(1)/Disconnect(0) the 50KOhm pull-up resistor on CD_B/DATA3 of the card.

Table continues on the next page...

Table 227. Commands for eMMC/SD card/SDIO (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
ACMD51[4]	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

1. CMD3 differs for eMMC and SD cards. For eMMC, it is referred to as SET_RELATIVE_ADDR, with a response type of R1. For SD cards, it is referred to as SEND_RELATIVE_ADDR, with a response type of R6 (with RCA inside).
2. CMD6 differs completely between high speed eMMC and high-speed SD cards. Command SWITCH_FUNC is for high-speed SD cards.
3. Command SWITCH is for high speed eMMC. The Index field can contain any value from 0-255, but only values 0-191 are valid. If the Index value is in the 192-255 range the card does not perform any modification and the SWITCH_ERROR status bit in the EXT_CSD register is set. The Access Bits are shown in Table 228.
4. ACMDs is preceded with the APP_CMD command. Commands listed are used for SD only, other SD commands not listed are not supported on this module.

The access bits for the EXT_CSD access modes are listed in the following table.

Table 228. EXT_CSD access modes

Bits	Access name	Operation
00	Command set	The command set is changed according to the Cmd Set field of the argument.
01	Set bits	The bits in the pointed byte are set, according to the bits set to 1 in the Value field.
10	Clear bits	The bits in the pointed byte are cleared, according to the bits set to 1 in the Value field.
11	Write byte	The Value field is written into the pointed byte.

38.3.9 SDIO interrupt

Information on interrupts in 1-bit mode, interrupts in 4-bit mode, and card interrupt handling are detailed in the sections below.

38.3.9.1 Interrupts in 1-bit mode

In this case, the DATA1 pin provides the interrupt function. An interrupt is asserted by pulling the DATA1 low from SDIO, until the interrupt service is finished to clear the interrupt.

38.3.9.2 Interrupt in 4-bit mode

As the interrupt and data line 1 share pin 8 in a 4-bit mode, an interrupt is only sent by the card and recognized by the host during a specific time. This is known as the interrupt period. The uSDHC module only provides samples the level on pin 8 during the interrupt period. At all other times, the host treats it as the data signal. The definition of the interrupt period is different for operations with single block and multiple block data transfers.

In the case of normal single data block transmissions, the interrupt period becomes active two clock cycles after the completion of a data packet. This interrupt period lasts until after the card receives the end bit of the next command that has a data block transfer associated with it.

For multiple block data transfers in a 4-bit mode, there is only a limited period of time that the interrupt period can be active because of the limited period of data line availability between the multiple blocks of data. This requires stricter definition of the interrupt period. For this case, the interrupt period is limited to two clock cycles. This begins two clocks after the end bit of the previous data block. During this 2-clock cycle interrupt period, if an interrupt is pending, the DATA1 line holds low for one clock cycle, then pulls high for the next clock cycle . On completion of the interrupt period, the card releases the DATA1 line into the high Z state. The uSDHC module provides sample of the DATA1 during the interrupt period when the IABG bit in the Protocol Control register is set.

Refer to SDIO Specification v1.10 for further information about the SDIO interrupt.

38.3.9.3 Card interrupt handling

When the CINTIEN bit in the Interrupt Signal Enable Register is set to 0, uSDHC clears the interrupt request to the host system. The host driver should clear this bit before servicing the SDIO interrupt, and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.

The SDIO Interrupt Status can be cleared by writing 1 to this bit. But as the interrupt source from SDIO does not clear, this bit is set again. To clear this bit, it is required to reset the interrupt source from the external card followed by writing 1 to this bit. In a 1-bit mode, uSDHC detects the SDIO interrupt with or without the SD clock (to support wakeup). In a 4-bit mode, the interrupt signal is sampled during the interrupt period, so there are some sample delays between the interrupt signal from SDIO and the interrupt to the Host System Interrupt Controller. When the SDIO status is set and the host driver needs to service this interrupt, the SDIO bit in the Interrupt Control Register of SDIO is cleared. This is required to clear the SDIO interrupt status latched in uSDHC and to stop driving the interrupt signal to the System Interrupt Controller. The host driver must issue a CMD52 to clear the card interrupt. After completion of the card interrupt service, the SDIO Interrupt Status Enable bit is set to 1 and uSDHC starts sampling the interrupt signal again.

See Figure 140 for an illustration of the SDIO interrupt scheme and for the sequences of software and hardware events that take place during a card interrupt handling procedure.

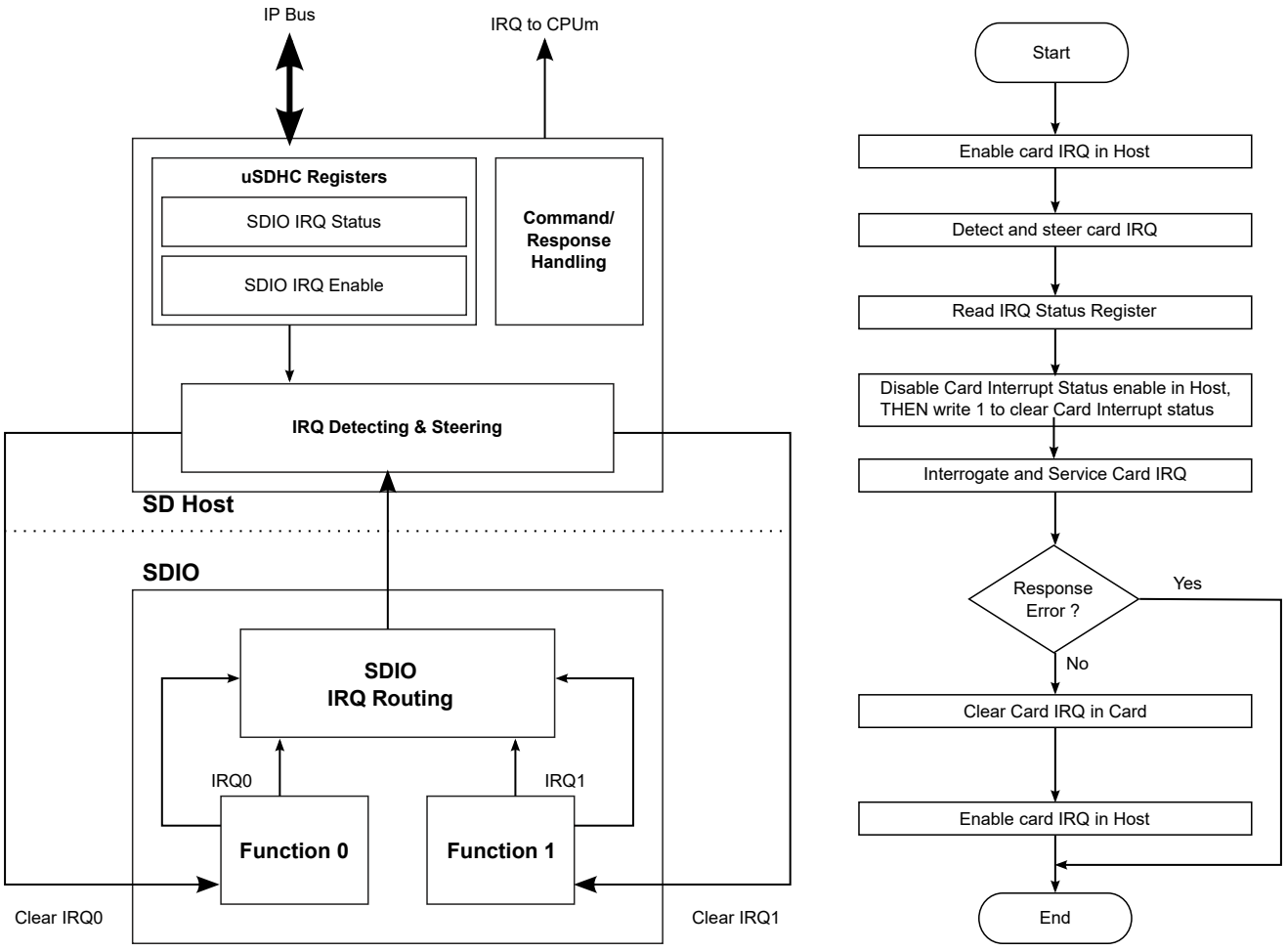


Figure 140. Card interrupt scheme, card interrupt detection, and handling procedure

38.3.10 Speed mode selection

To switch to HS200 or HS400 from High Speed (26 MHz-52 MHz), see the procedure below.

NOTE

While the actual timing change is done, the behavior of any command given (such as CMD13) cannot be guaranteed due to the asynchronous operation. Therefore, you must not use CMD13 to check the busy completion of the timing change indication. In case, CMD13 is used, the host must ignore CRC errors, if appear.

The max frequency for HS200 mode and HS400 mode could be slower than 200 MHz based on the chip implementation, you must set the target frequency for these two speed modes accordingly. Following sections describe the mode selections for HS200 mode and HS400 mode in detail.

38.3.10.1 HS200 mode selection

To switch to HS200, the Driver must perform the following steps:

1. Select the chip (through sending CMD7) and make sure it is unlocked (through CMD42).
2. Read the DEVICE_TYPE [196] field of the Extended CSD register to validate if the chip supports HS200 at the IO voltage appropriate for both host and chip (through CMD8).
3. Read the DRIVER_STRENGTH [197] field of the Extended CSD register to find the supported chip Driver Strengths (through CMD8). The default setting is 50 Ohm. The Host Designer might simulate its specific system, using a device driver models. Host can select the optimal Driver Type that might drive the host system load at the required operating frequency with the minimal noise generated.

   NOTE

   This step can be skipped if changes of Driver strength is not needed.

4. Check the setting of PROT_CTRL[DTW], make sure the data transfer width is set as expected.

   Make sure the I/O pad voltage is 1.8V and corresponding drive strength is set to proper value.

   Set HS200 bit and Driver Strength value in the HS_TIMING [185] field of the Extended CSD register by issuing CMD6. If the host attempts to write an invalid value, the HS_TIMING byte is not changed, the HS200 interface timing is not enabled, the Driver Strength is not changed, and the SWITCH_ERROR field is set. After the chip responds with R1, it might assert Busy signal. Once the busy signal gets de-asserted, the host may send a SEND_STATUS Command (CMD13) using the HS200 timing and after it receives a Trans State indication and No Error it means that the chip is set to HS200 timing and the Driver Strength is set to the selected settings.
5. At this point, the host can set the frequency to ≤ 200 MHz.
6. The host might invoke the HS200 standard tuning procedure, by sending CMD21 to the chip.

NOTE

The host must switch to the required bus width before starting the tuning operation to allow the tuning sequence to be done using the proper bus operating conditions.

38.3.10.2 Standard tuning procedure

By default, lower frequency operation, a fixed sampling clock is used to receive signals on CMD and DAT[3:0]. Before using the HS200, HS400, SDR104, or SDR50 modes, the Host Driver executes the tuning procedure at the mode switch sequence.

1. Issue uSDHC SW reset, set SYS_CTRL[RSTT] to 1.
2. Clear VEND_SPEC[FRC_SDCLK_ON] to 0.
3. Clean the history status by following these steps:
   a. Clear AUTOCMD12_ERR_STATUS[EXECUTE_TUNING] to 0.

b. Read Auto CMD12 Error Status (AUTOCMD12_ERR_STATUS) until AUTOCMD12_ERR_STATUS[EXECUTE_TUNING] reads 0.

c. Write 1 to clear INT_STATUS[BRR].

4. Set TUNING_CTRL to 0x01312880 to enable the standard tuning and disable the CMD line check.
5. Start the tuning procedure by setting AUTOCMD12_ERR_STATUS[EXECUTE_TUNING] to 1.
6. Issue CMD19(SD) / CMD21(eMMC) with the proper Command Transfer Type (CMD_XFR_TYP) and Mixer Control (MIX_CTRL) settings.
7. Wait for uSDHC BRR (Buffer Read Ready) interrupt signal is 1.
8. Check AUTOCMD12_ERR_STATUS[EXECUTE_TUNING]. If AUTOCMD12_ERR_STATUS[EXECUTE_TUNING] = 1, write 1 to clear INT_STATUS[BRR] and then repeat steps 6~7. If AUTOCMD12_ERR_STATUS[EXECUTE_TUNING] = 0, standard tuning has completed, or the tuning has not completed within 40 attempts. The Host Driver might abort this loop if the number of loops exceeds 40 or 150 ms timeout occurs. In this case, a fixed sampling clock should be used (AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] = 0).
9. Sampling Clock Select (AUTOCMD12_ERR_STATUS[SMP_CLK_SEL]) is valid after AUTOCMD12_ERR_STATUS[EXECUTE_TUNING] has changed from 1 to 0.
   AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] = 1, indicates tuning procedure passed.
   AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] = 0, indicates tuning procedure failed. The tuning result is applied to the delay chain, CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) [30:16], upon the successful tuning procedure completion.
10. Set MIX_CTRL[AUTO_TUNE_EN] to 1.

While the tuning sequence is being performed, the Host Controller does not generate interrupts (including Command Complete), except Buffer Read Ready. CMD19 response errors are not indicated.

Writing AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] to 0 forces the Host Controller to use a fixed sampling clock and resets the tuning circuit of the Host Controller.

NOTE

- There could be slight difference on delay cell delay in each project, and this difference can lead to different loop number needed. TUNING_CTRL[TUNING_STEP] can be used to control how may taps are to be added for each step.
- Manual tuning might be required in cases where standard tuning is not able to close passing window with CMD19/21 transfer fail.

38.3.10.3 Manual tuning procedure

Manual tuning controls more details in the tuning procedure, and some corner cases. Manual tuning provides control to tuning start point, tuning steps, and delay chains. To start manual tuning:

1. Clear TUNING_CTRL[STD_TUNING_EN] to 0.
2. Clear VEND_SPEC[FRC_SDCLK_ON] and MIX_CTRL[FBCLK_SEL] to 0, set MIX_CTRL[SMP_CLK_SEL] and MIX_CTRL[EXE_TUNE] to 1 to start tuning.
3. Set SYS_CTRL[RSTA] to 1, and wait for self-clear.
4. Config delay chain, CLK_TUNE_CTRL_STATUS[DLY_CELL_SET_PRE] with the valueA.

NOTE

The valueA must start from 0.

5. Set SYS_CTRL[RST_FIFO] to 1 and wait for self-clear.
6. Send CMD19/21 to card.

7. Poll for CC, CIE, CEBE, CCE and CTOE of Interrupt Status (INT_STATUS) assertion.
8. Poll for TC, DEBE, DCE and DTOE of Interrupt Status (INT_STATUS) assertion.
9. Check TP of Interrupt Status (INT_STATUS); TP = 1 & no command/data error indicates the current CMD19/21 transfer is passed; otherwise, indicates failed.
10. If TP = 0 or command/data error is observed, increase the valueA with the required tuning step(1 to 127) and repeat steps 3~9.
11. If TP = 1 & no command/data error is met after a loop with TP=0 (or command/data error), save the current valueA as value_start. Increase valueA with the required tuning step, and repeat steps 3~9 until TP = 0 (or command/data error) again. Save the current valueA as value_end.
12. Clear MIX_CTRL[EXE_TUNE] to 0.
13. Set SYS_CTRL[RSTA] to 1, and wait for self-clear.
14. Set MIX_CTRL[SMP_CLK_SEL] to 1.
15. Set bit[14:0] of CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) with the output of expression {[(((value_start+value_end)/2)< < 8)& 0xffffff00] -0x300 } | 0x33.
16. Poll bit[30:16] of CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) until the value written in the previous step is read.
17. Set MIX_CTRL[AUTO_TUNE_EN] to 1.

The manual tuning procedure defines a passing window, inside which the tuned clock can capture the correct data, and delay chain is to be configured as centered within the passing window.

It is not a fixed flow as the standard tuning. Any step can be changed based on the real case. In cases where passing window is not closed with a tuning fail, the center of the passing region must be applied for the delay chain setting. In cases where two passing windows separated by tuning fail(s) is observed, the center of the first passing window must be applied for the delay chain setting.

NOTE

Both the standard tuning and manual tuning are only for tuning of SD_CLK, cannot be used to tune STROBE clock of HS400 and HS400 enhanced mode.

38.3.10.4 Switch to HS400 mode

The valid IO voltage for HS400 is 1.8 V. The bus width is set to only DDR 8bit in HS400 mode. HS400 supports the same commands as DDR52. To switch to HS400 mode, the host must perform the following steps:

1. Initialize chip with “Backward Compatible Timings", after power-on or software reset (CMD0), the interface timing of the chip is set as the default "Backward Compatible Timing".
2. Select the chip with CMD7.
3. Read the DEVICE_TYPE [196] field of the Extended CSD register to validate whether the chip supports HS400 by sending CMD8.
4. Read the DRIVER_STRENGTH [197] field of the Extended CSD register to find the supported chip's Driver Strengths by sending CMD8.

NOTE

You may skip this step if changes of driver strength is not needed.

5. Set the "Selected Driver Strength" parameter in the HS_TIMING [185] field of the Extended CSD register to the appropriate driver strength for HS400 operation and set the "Timing Interface" parameter to 0x2 to switch to HS200 mode with CMD6. Send CMD13 to make sure mode switch to HS200 mode has been successfully done.

6. Perform the tuning procedure (standard or manual) at the HS400 target operating frequency, save the tuning result (CLK_TUNE_CTRL_STATUS[30:16]), clear TUNING_CTRL[STD_TUNING_EN] to 0, set SYS_CTRL[RSTT] to 1 and wait for self-clear.

   NOTE

   Tuning procedure in HS200 mode is required to synchronize the command response on the CMD line to CLK for HS400 operation.

7. Set the "Timing Interface" parameter in the HS_TIMING [185] field of the Extended CSD register to 0x1 to switch to High Speed mode with CMD6 and then set the clock frequency to a value not greater than 52 MHz.
8. Set BUS_WIDTH[183] to 0x06 to select the dual data rate x8 bus mode. Check the setting of PROT_CTRL[DTW]. Make sure the data transfer width is set as expected.
9. Set the "Timing Interface" parameter in the HS_TIMING [185] field of the Extended CSD register to 0x3 to switch to HS400 mode with CMD6. Set the MIX_CTRL[DDR_EN] and MIX_CTRL[HS400_MODE] to 1 to enable HS400 mode in controller.
10. Set bit[14:0] of CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) with saved tuning result value of 0x68 from standard tuning flow in step 6).
11. Poll bit[30:16] of CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) until the value written in the step 10 is read.
12. Set MIX_CTRL[SMP_CLK_SEL] to 1’b1 to apply the delay on capture clock of CMD channel, then set the clock frequency to ≤ 200 MHz.

    Ensure that uSDHC function clock source is 2x HS400 mode target frequency, and both SDCLKFS and DVS are configured to 0.
13. Set Strobe DLL control (STROBE_DLL_CTRL) to 0x00400039 to have strobe clock automatically tuned to 1/4 cycle of input ipp_strobe to capture data. Ensure that both STROBE_DLL_STATUS[STROBE_DLL_STS_REF_LOCK] and STROBE_DLL_STATUS[STROBE_DLL_STS_SLV_LOCK] are set before any transfer is started.

    NOTE

    For speed mode switch like from High Speed mode to HS200 mode or HS400(enhanced) mode, you must calibrate the right configuration of slew rate and drive strength for chip pad that are used for uSDHC.

38.3.10.5 Switch to HS400 enhanced mode

This selection flow describes how to initialize the eMMC device in HS400 mode while enabling enhanced strobe without the need for tuning procedure. To switch to HS400 mode with enhanced strobe, the host must perform the following steps:

1. Initialize device with "Backward Compatible Timings", after power-on or software reset (CMD0), the interface timing of the device is set as the default "Backward Compatible Timing".
2. Select the device with CMD7
3. Read the DEVICE_TYPE [196] field of the Extended CSD register to validate whether the device supports HS400 with CMD8.
4. Read the STROBE_SUPPORT[184] field of the Extended CSD register to validate whether the device supports enhanced strobe with CMD8.
5. Set the "Timing Interface" parameter in the HS_TIMING [185] field of the Extended CSD register to 0x1 to switch to High Speed mode by sending CMD6 and then set the clock frequency to a value not greater than 52 MHz.
6. Set BUS_WIDTH[183] to 0x86 to select the dual data rate x8 bus mode and enable enhanced strobe (this is active only after HS400 mode is selected) with CMD6. Check the setting of PROT_CTRL[DTW]. Make sure the data transfer width is set as expected.
7. Read the DRIVER_STRENGTH [197] field of the Extended CSD register to find the supported device Driver Strengths with CMD8.

NOTE

You may skip this step if changes of driver strength is not needed, if needed host may change the device Driver Strength.

8. Set the "Timing Interface" parameter in the HS_TIMING [185] field of the Extended CSD register to 0x3 to switch to HS400 mode with CMD6.
9. Set the MIX_CTRL[DDR_EN], MIX_CTRL[HS400_MODE], and MIX_CTRL[EN_HS400_MODE] bits to 1 to enable enhanced HS400 mode in controller.
10. Host may set the clock frequency to a value not greater than 200 MHz. Ensure uSDHC function clock source is 2x HS400 mode target frequency, and both SDCLKFS and DVS are configured to 0 .
11. Set Strobe DLL control (STROBE_DLL_CTRL) to 0x00400039 to have strobe clock automatically delayed 1/4 cycle of input ipp_strobe to capture data. Make sure both STROBE_DLL_STATUS[STROBE_DLL_STS_REF_LOCK] and STROBE_DLL_STATUS[STROBE_DLL_STS_SLV_LOCK] are set before any transfer is started.

NOTE

For speed mode switch like from high speed mode to HS200 mode or HS400(enhanced) mode, you must calibrate the right configuration of slew rate and drive strength for chip pad that are used for uSDHC.

38.3.11 Software restrictions

38.3.11.1 Initialization active

The driver cannot set INITA bit in System Control register when any of the command line or data lines are active, so the driver must ensure both CDIHB and CIHB bits are cleared.

38.3.11.2 Software polling procedure

For polling read or write, after the software begins a buffer read or write, it must access exactly the number of times as the values set in the Watermark Level Register; moreover, if the block size is not a multiple of the value in the Watermark Level Register (read and write respectively), the software must access exactly the remaining number of words at the end of each block. For example, for a read operation, if the RD_WML is 4, indicating the watermark level is 16 bytes, block size is 40 bytes, and the block number is 2, then the access times for the burst sequence in the whole transfer process must be 4, 4, 2, 4, 4, 2.

38.3.11.3 Suspend operation

To suspend the data transfer, the software must inform uSDHC that the suspend command is successfully accepted. To achieve this, after the Suspend command is accepted by SDIO, software must send another normal command marked as suspend command (CMDTYP bits set as '01') to inform uSDHC that the transfer is suspended.

If software needs to resume the suspended transfer, it should read the value in BLKCNT register to save the remaining number of blocks before sending the normal command marked as suspend, otherwise on sending such a 'suspend' command, uSDHC treats the current transfer is stopped and change the BLKCNT register to its original value, instead of retaining the remaining number of blocks.

38.3.11.4 Data length setting

For either ADMA (ADMA1 or ADMA2) transfer, the data in the data buffer must be word aligned, so the data length set in the descriptor must be a multiple of 4.

ADMA1 is 4KB aligned.

38.3.11.5 (A)DMA address setting

To configure the ADMA1/ADMA2/DMA address register, when TC bit is set, the register always updates itself with the internal address value to support dynamic address synchronization, so the software must ensure that the TC bit is cleared prior to configuring the ADMA1/ADMA2/DMA address register.

38.3.11.6 Data port access

Data port does not support parallel access. For example, during an internal DMA access, it is not allowed to write any data to the data port by CPU; or during a CPU read operation, it is also prohibited to write any data to the data port, by either CPU or internal DMA. Otherwise the data is corrupted inside the uSDHC buffer.

38.3.11.7 Change clock frequency

The uSDHC module does not automatically gate off the card clock when the host driver changes the clock frequency. To prevent possible glitch on the card clock, clear the FRC_SDCLK_ON bit when changing the clock divisor value (SDCLKFS or DVS in System Control Register) or setting the RSTA bit.

Also, before changing the clock divisor value, the host driver should make sure that the SDSTB bit is high.

38.3.11.8 Multi-block read

For pre-defined multi-block read operation, that is, the number of blocks to read has been defined by previous CMD23 for eMMC, or pre-defined number of blocks in CMD53 for SDIO/SDCombo, or whatever multi-block read without abort command at card side, an abort command, either automatic or manual CMD12/CMD52, is still required by uSDHC after the pre-defined number of blocks are done, to drive the internal state machine to idle mode. In this case, the card may not respond to this extra abort command and uSDHC gets response timeout. It is recommended to manually send an abort command with RSPTYP[1:0] both bits cleared.

38.3.12 Clocking

The table found here describes the clock sources for uSDHC. For more information on clocking, see the Clocking chapter.

Table 229. Clocks

Clock name	Description
hclk	Bus clock
ipg_clk	Peripheral clock
ipg_clk_perclk	Base clock
ipg_clk_lp	Low power clock

38.3.12.1 Clock and reset manager

This module controls all four kinds reset signals within uSDHC:

- Hardware reset
- Software reset for all logic
- Software reset for the data logic
- Software reset for the command logic

All these signals are fed into this module and stable signals are generated inside the module to reset all other modules. The module also gates off all the inside signals.

38.3.12.2 Clock generator

The clock generator generates the card CLK by peripheral source clock in two stages. The clock divisor can be configured through register SYS_CTRL[SDCLKFS] for prescaler configuration while the [DVS] for divisor configuration. Details can be found in the register function description. See the following figure for the structure of the divider. The term "Base" represents the frequency of peripheral source clock (see ipg_clk_perclk in Table 229).

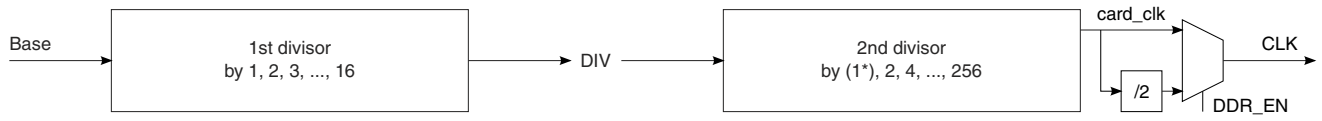


Figure 141. Two stages of the clock divider

The first stage outputs an intermediate clock (DIV) that can be Base, Base/2, Base/3, ..., or Base/16.

The second stage is a prescaler and outputs the actual internal working clock (card_clk). This clock is the driving clock for all the sub modules of the SD protocol unit, and helps in syncing FIFOs (see Figure 127) to synchronize with the data rate from the internal data buffer. The frequency of the clock output from this stage can be DIV, DIV/2, DIV/4,..., or DIV/256. Therefore, the highest frequency of the card_clk is base, and the next highest is Base/2, while the lowest frequency is Base/4096. If the duty cycle of the base clock is 50%, the duty cycle of card_clk is also 50%, even when the compound divisor is an odd value.

NOTE

CLK is different for the SDR and DDR modes.

- In the SDR mode, CLK is equal to the internal working clock (card_clk).

$$f_{card_clk} = \frac{f_{Base}}{SYS_CTRL[DVS] \times SYS_CTRL[SDCLKFS]}$$

$$f_{CLK} = f_{card_clk}$$

Equation 16. Equation for f_{card_clk} in SDR mode

For HS200 mode, if the source clock f_{Base} = 400 MHz, and the target clock f_{CLK} = 200 MHz, then configuring DVS to 'h00 and SDCLKFS to 'h01, the mapping value of SYS_CTRL[DVS] is 1 and the mapping value of SYS_CTRL[SDCLKFS] is 2.

$$f_{card_clk} = \frac{400MHz}{1x(2)} = 200MHz$$

$$f_{CLK} = f_{card_clk} = 200\ MHz$$

- In the DDR mode, CLK is equal to card_clk/2

$$f_{card_clk} = \frac{f_{Base}}{SYS_CTRL[DVS] \times (SYS_CTRL[SDCLKFS]/2)}$$

$$f_{CLK} = \frac{f_{card_clk}}{2}$$

Equation 17. Equation for f_{card_clk} in DDR mode

For HS400 mode, if the source clock f_{Base} = 400 MHz, and the target clock f_{CLK} = 200 MHz, then configuring DVS to 'h00 and SDCLKFS to 'h00, the mapping value of SYS_CTRL[DVS] is 1 and the mapping value of SYS_CTRL[SDCLKFS] is 2.

$$f_{card_clk} = \frac{400MHz}{1x(2/2)} = 400MHz$$

$$f_{CLK} = \frac{f_{\text{card_clk}}}{2} = \frac{400MHz}{2} = 200MHz$$

38.3.13 Command Queuing (CQE) operation

To facilitate command queuing in eMMC, the device manages an internal task queue that the host can queue data transfer tasks. Initially, the task queue is empty. Every task is issued by the host and initially queued as pending. The device controller works to prepare the pending tasks for execution. The software flow is described below:

1. Read CMDQ_SUPPORT[308] and CMDQ_DEPTH[307] of the extended CSD register (through CMD8) to verify eMMC chip support of command queue.
2. Set CMDQ_MODE_EN in CMD_MODE_EN[15] field of the extended CSD by issuing CMD6.
3. Send CMD13 to ensure a successful mode switch.
4. Define Task Descriptor List (TDL) comprised of task descriptor and transfer descriptor sets.
    a. 0x0010002b,0x00000000,0x20000023,0x34300000, //Write example
    b. 0x0010102b,0x00000000,0x20000023,0x34500000, //Read example
5. Set PROT_CTRL[DMASEL] to 2 to select ADMA2. CMDQ descriptors are in ADMA2 format.
6. Set CQSSC2[SSC2] to RCA.
7. Set Command Queuing Task Descriptor List Base Address (CQTDLBA) and Command Queuing Task Descriptor List Base Address Upper 32 Bits (CQTDLBAU) to TDL start address.
8. Set CQCFG[CQUE] to 1 to enable the command queuing.
9. Set CQTDBR[TDBR] field(s) to trigger the required task(s).
10. Read CQTCN[TCN] field(s) to confirm the task completion(s).
11. Set CQTCN[TCN] field(s) to 1 to clear status.

38.3.14 Abort transaction

Abort command

There are two types of abort command: CMD12 and I/O abort.

- CMD12 is used for memory device: such as eMMC, SD Card, SD Combo Card (Memory portion).
- I/O abort is used for SDIO card, SDCombo (I/O portion). I/O abort stops an infinite block transfer (block count = 0) by writing the ASx bit in CCCR.

If an exact number of blocks are to be transferred, it is recommended that the uSDHC issues a block command with the correct block count, rather than using an infinite count and aborting the data at the correct time.

Asynchronous and synchronous abort

There are two ways to issue an abort command (CMD12 or I/O abort): asynchronous abort and synchronous abort.

- In an asynchronous abort sequence, uSDHC can issue an abort command (CMD12 or I/O abort) at anytime unless Command Inhibit (PRES_STATE[CIHB]) in the Present State register is set to 1.
- In a synchronous abort sequence, uSDHC shall issue an abort command (CMD12 or I/O abort) after data transfer is stopped by using Stop At Block Gap Request (SABGREQ).

Read wait

This function uses DATA2 to allow uSDHC to signal the SDIO or Combo Card to temporarily halt the sending of read data by a card during the block gap.

38.4 External signals

The following table describes the uSDHC external signals:

Table 230. uSDHC external signals

Signal	Description	Direction
CLK	Is an Internally generated clock used to drive the eMMC, SD card, and SDIO.	O
CMD	Is used to send commands and receive responses to and from the card.	I/O
DAT7	DAT7 line in the 8-bit mode — Not used in other modes	I/O
DAT6	DAT6 line in the 8-bit mode — Not used in other modes	I/O
DAT5	DAT5 line in the 8-bit mode — Not used in other modes	I/O
DAT4	DAT4 line in the 8-bit mode — Not used in other modes	I/O
NOTE If uSDHC needs to support a 4-bit data transfer, DAT7~DAT4 can also be optional and tied to high.		
DAT3	DAT3 line in the 4/8-bit mode or configured as card detection pin. The bit may be configured as card detection pin in the 1-bit mode.	I/O
DAT2	DAT2 line or Read Wait in the 4-bit mode Read Wait in 1-bit mode	I/O
DAT1	DAT1 line in the 4/8-bit mode Also, used to detect interrupt in 1/4-bit mode	I/O
DAT0	DAT0 line in all the modes Also, used to detect busy state	I/O
CD_B	Card detection signal directly routed from the socket. Low value (0) indicates there is a card inserted. In the case the pin is not used (for the embedded memory), tie low. Optional for system implementation.	I
WP	Write protection signals directly routed from the socket. Low value (0) indicates it is not write protected. In the case the pin is not used (for the embedded memory), tie low.	I

Table continues on the next page...

Table 230. uSDHC external signals (continued)

Signal	Description	Direction
	Optional for system implementation.	
RESET_B	Is used to reset the eMMC or control Card power. Active Low. • eMMC: connect eMMC RST_n pin. • Card: instead of hardware reset to enable/disable Card power during warm reset.	O
VSELECT	Is used to change the voltage of the external power supplier. Optional for system implementation.	O
STROBE	Input clock for eMMC HS400 mode. **NOTE** If the uSDHC does not support the HS400 mode, STROBE can also be optional and tied to low.	I

38.5 Application information

All communication between the system and cards are controlled by the host. The host sends commands of two types: broadcast and addressed (point-to-point).

Broadcast commands are intended for all cards, such as GO_IDLE_STATE, SEND_OP_COND, and ALL_SEND_CID. In the Broadcast mode, eMMC are in the open-drain mode to avoid bus contention. See Commands for SD card, SDIO, and eMMC for the commands of bc and bcr categories.

After the broadcast command CMD3 is issued, the cards enter the standby mode. Addressed type commands are used from this point. In this mode, for eMMC, the CMD/DATA I/O pads turns to the push-pull mode to have the driving capability for maximum frequency operation. See Commands for SD card, SDIO, and eMMC for the commands of ac and adtc categories.

38.5.1 Command send and response receive basic operation

Assuming that the data type WORD is an unsigned 32-bit integer, the flow indicated below presents a guideline for sending a command to the card(s):

```
send_command(cmd_index, cmd_arg, other requirements)

{

    WORD wCmd; // 32-bit integer to make up the data to write into Transfer Type register, it is
recommended to implement in a bit-field manner

    wCmd = (<cmd_index> & 0x3f) << 24; // set the first 8 bits as '00'+<cmd_index>

    set CMDTYP, DPSEL, CICEN, CCCEN, RSTTYP, DTDSEL according to the command index;

    if (internal DMA is used) wCmd |= 0x1;

    if (multi-block transfer) {

        set MSBSEL bit;
```

```
        if (finite block number) {

            set BCEN bit;

            if (auto12 command is to use) set AC12EN bit;

        }

    }

    write_reg(CMDARG, <cmd_arg>); // configure the command argument

    write_reg(XFERTYP, wCmd); // set Transfer Type register as wCmd value to issue the command

}

wait_for_response(cmd_index)

{

    while (CC bit in IRQ Status register is not set); // wait until Command Complete bit is set

    read IRQ Status register and check if any error bits about Command are set

    if (any error bits are set) report error;

    write 1 to clear CC bit and all Command Error bits;

}
```

For the sake of simplicity, the function wait_for_response is implemented here by means of polling. For an effective and formal way, the response is usually checked after the Command Complete Interrupt is received. When doing this, make sure that the corresponding interrupt status bits are enabled.

For some scenarios, the response time-out is expected. For instance, after all cards respond to CMD3 and move to the Standby state no response to the Host when CMD2 is sent. The host driver deals with "fake" errors like this with caution.

38.5.2 Card identification mode

When a card is inserted to the socket or the card is reset by the host, the host needs to validate the operation voltage range, identify the cards, request the cards to publish the Relative Card Address (RCA) or to set the RCA for eMMC.

38.5.2.1 Card detect

See Figure 142 for a flow diagram showing the detection of SD card and SDIO using uSDHC.

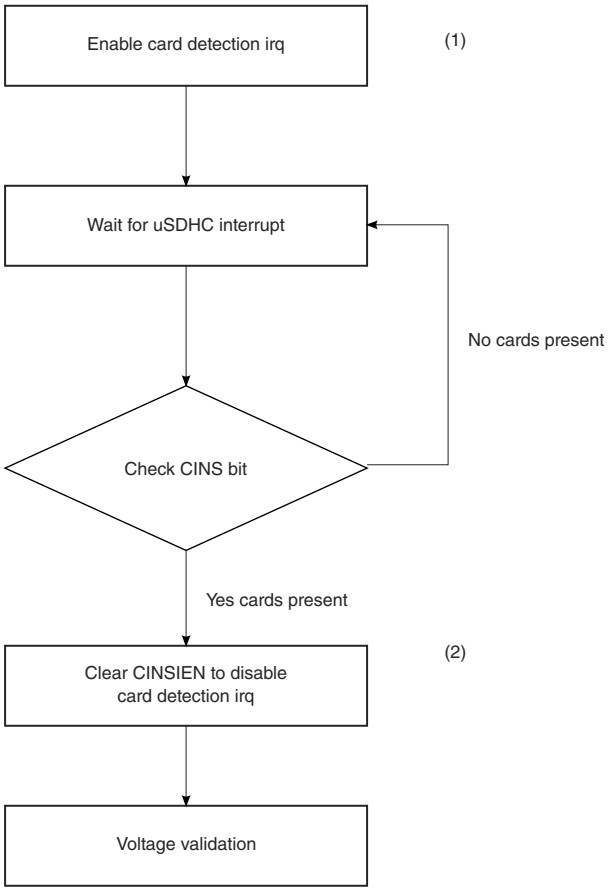


Figure 142. Flow diagram for card detection

Here is the card detect sequence:

- Set the CINSIEN bit to enable card detection interrupt.
- When an interrupt from uSDHC is received, check the CINS bit in the Interrupt Status register to see if it was caused by card insertion.
- Clear the CINSIEN bit to disable the card detection interrupt and ignore all card insertion interrupts afterwards.

38.5.2.2 Reset

The host consists of three types of resets:

- Hardware reset (Card and Host) that is driven by Power On Reset (POR).
- Software reset (Host only) is initiated by the write operation on the RSTD, RSTC, or RSTA bits of the System Control register to reset the data part, command part, or all parts of the host controller, respectively.
- Card reset (Card only): The command, "Go_Idle_State" (CMD0), is the software reset command for all types of eMMC and SD memory cards. This command sets each card into the idle state regardless of the current card state. For an SDIO, CMD52 is used to write an I/O reset in CCCR. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

After the card is reset, the host needs to validate the voltage range of the card. See the figure below for the software flow to reset both uSDHC and the card.

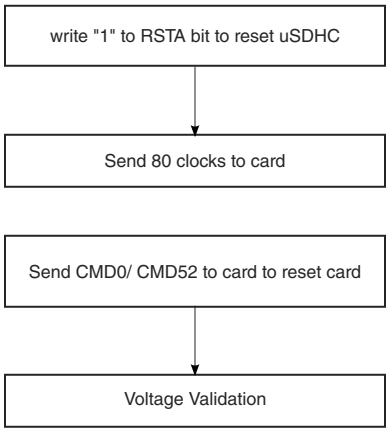


Figure 143. Flow chart for resetting uSDHC and SD card or SDIO

```
software_reset() {
    set_bit(SYSCTRL, RSTA); // software reset the Host set DTOCV and SDCLKFS fields to get the CLK of
    //frequency around 400kHz
    //configure IO pad to set the power voltage of external card to around 3.0V
    //poll bits CIHB and CDIHB
    //bits of PRSSTAT to wait both fields are cleared
    set_bit(SYSCTRL, INTIA); // send 80 clock ticks for card to power up
    send_command(CMD_GO_IDLE_STATE, <other parameter>); // reset the card with CMD0 or
    send_command(CMD_IO_RW_DIRECT, <other parameter>);
}
```

38.5.2.3 Voltage validation

All cards should be able to establish communication with the host using any operation voltage in the maximum allowed voltage range specified in the card specification. However, the supported minimum and maximum values for VDD are defined in the Operation Conditions Register (OCR) and may not cover the whole range. Cards that store the CID and CSD data in the preload memory are only able to communicate this information under data transfer VDD conditions. This means that if the host and card have non-common VDD ranges, the card is neither able to complete the identification cycle nor able to send CSD data.

Therefore, special commands Send_Op_Cont (CMD1 for eMMC), SD_Send_Op_Cont (ACMD41 for SD Memory), and IO_Send_Op_Cont (CMD5 for SD I/O) are used. The voltage validation procedure is designed to provide a mechanism to identify and reject cards that do not match the VDD range(s) desired by the host. This is accomplished when the host sends the desired VDD voltage window as the operand of this command. Cards that cannot perform the data transfer in the specified range must discard themselves from further bus operations and go into the Inactive state. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before sending out-of-range cards into the Inactive state. This query should be used if the host is able to select a common voltage range or if a notification is sent to the system when a non-usable card in the stack is detected.

The following steps show how to perform voltage validation when a card is inserted:

```
voltage_validation(voltage_range_argument) {
    label the card as UNKNOWN;
    send_command(IO_SEND_OP_COND, 0x0, <other parameters are omitted>); // CMD5, check SDIO operation
voltage, command argument is zero
    if (RESP_TIMEOUT != wait_for_response(IO_SEND_OP_COND)) { // SDIO command is accepted
        if (0 < number of IO functions) {
            label the card as SDIO;
            IORDY = 0;
```

```
            while (!(IORDY in IO OCR response)) { // set voltage range for each IO function
                send_command(IO_SEND_OP_COND, <voltage range>, <other parameter>);
                wait_for_response(IO_SEND_OP_COND);
            } // end of while ...
        } // end of if (0 < ...
        if (memory part is present inside SDIO)
            Label the card as SDCombo; //this is an SD-Combo card
    } // end of if (RESP_TIMEOUT ...
    if (the card is labeled as SDIO)
        return; // card type is identified and voltage range is set, so exit the function;
    send_command(APP_CMD, 0x0, <other parameter>); // CMD55, Application specific CMD prefix
    if (no error calling wait_for_response(APP_CMD, <...>) { // CMD55 is accepted
        send_command(SD_APP_OP_COND, <voltage range>, <...>); // ACMD41, to set voltage range for
memory part or SD card
        wait_for_response(SD_APP_OP_COND); // voltage range is set
        if (card type is UNKNOWN)
           label the card as SD;
        return; //
    } // end of if (no error ...
    else if (errors other than time-out occur) { // command/response pair is corrupted
        deal with it by program specific manner;
    } // of else if (response time-out
    else { // CMD55 is refuse, it must be eMMC
        if (card is already labeled as SDCombo) { // change label
            re-label the card as SDIO;
            ignore the error or report it;
            return; // card is identified as SDIO
        } // end of if (card is ...
        send_command(SEND_OP_COND, <voltage range>, <...>);
        if(RESP_TIMEOUT == wait_for_response(SEND_OP_COND)) { // CMD1 is not accepted,
            either label the card as UNKNOWN;
            return;
        } // end of if (RESP_TIMEOUT ...
    } // end of else
}
```

38.5.2.4 Card registry

This section briefly describes the registry flow. For details, please refer to the card specifications. Card registry for the eMMC and SD/SDIO/SD combo cards are different. For the SD card, the identification process starts at a clock rate lower than 400 kHz and the power voltage higher than 2.7 V (as defined by the card spec). Currently, the CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated, the host requests the card to send their valid operation conditions. The response to ACMD41 is the operation condition register of the card. The same command is sent to all the new cards in the system. Incompatible cards are put into the Inactive state. The host then issues the command, All_Send_CID (CMD2), to each card to get its unique card identification (CID) number. Cards that are currently unidentified (in the Ready state), send their CID number as the response. After the CID is sent by the card, the card goes into the Identification state.

The host then issues Send_Relative_Addr (CMD3), requesting the card to publish a new relative card address (RCA) that is shorter than the CID. This RCA is used to address the card for future data transfer operations. After the RCA is received, the card changes its state to the Standby state. At this point, if the host wants the card to have an alternative RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process with CMD2 and CMD3 for each card in the system until the last CMD2 gets no response from any of the cards in the system.

For eMMC operation, the host starts the card identification process in the open-drain mode with the identification clock rate lower than 400 kHz and the power voltage higher than 2.7 V. The open drain driver stages on the CMD line to allow parallel card operation during card identification. After the bus is activated, the host requests the cards to send their valid operation conditions

(CMD1). The response to CMD1 is the "wired AND " operation on the condition restrictions of all cards in the system. Incompatible cards are sent into the Inactive state. The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards (the cards in the Ready state) simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods, stop sending their CID immediately and must wait for the next identification cycle. As the CID is unique for each card, only one card can be successfully sent its full CID to the host. This card then goes into the Identification state. Thereafter, the host issues Set_Relative_Addr (CMD3) to assign a relative card address (RCA) to the card. After the RCA is received, the state of the card changes to standby, and the card does not react in further identification cycles. Also, its output driver switches from open-drain to push-pull. The host repeats the process, mainly CMD2 and CMD3, until the host receives a time-out condition to recognize the completion of the identification process.

The following steps show how to perform an operation using eMMC:

```
card_registry() {
    do { // decide RCA for each card until response time-out
        if(card is labelled as SDCombo or SDIO) { // for SDIO like device
            send_command(SET_RELATIVE_ADDR, 0x00, <...>); // ask SDIO to publish its RCA
            retrieve RCA from response;
        } // end if (card is labelled as SDCombo ...
        else if (card is labelled as SD) { // for SD card
            send_command(ALL_SEND_CID, <...>);
            if(RESP_TIMEOUT == wait_for_response(ALL_SEND_CID))
                break;
            send_command(SET_RELATIVE_ADDR, <...>);
            retrieve RCA from response;
        } // else if (card is labelled as SD ...
        else if (card is labelled as eMMC) {
            send_command(ALL_SEND_CID, <...>);
            rca = 0x1; //arbitrarily set RCA, 1 here for example
            send_command(SET_RELATIVE_ADDR, 0x1 << 16, <...>); // send RCA at upper 16 bits
        } // end of else if (card is labelled as eMMC...
    } while (response is not time-out);
}
```

38.5.3 Card access

Information about Block Write, Block Read, Suspense Resume, ADMA Usage, Transfer Error, and Card Interrupt are detailed in the sections below.

38.5.3.1 Block write

Information on Normal Write, DDR Write, and Write with Pause are detailed in the sections below.

38.5.3.1.1 Normal Write

During a block write (CMD24 - 27, CMD60, CMD61), one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. If the CRC fails, the card indicates that the failure on the DATA line. The transferred data is discarded and not written, and all further transmitted blocks (in multiple block write mode) are ignored.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card detects the block misalignment error and stops the programming before the beginning of the first misaligned block. The card sets the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, waits in the Receive-data-State for a stop command. The write operation also stops if the host tries to write over a write-protected area.

For eMMC and SD cards, programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card reports an error and does not change any register contents.

For all types of cards, some may require long and unpredictable periods of time to write a block of data. After receiving a block of data and completing the CRC check, the card begins writing and holds the DATA line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) or other means for SDIO at any time, and the card responds with its status. The responded status indicates whether the card can accept new data or whether the write process is still in progress. The host may deselect the card by issuing a CMD7 (to select a different card) to place the card into the Standby state and release the DATA line without interrupting the write operation. When re-selecting the card, it reactivates the busy indication by pulling data to low if the programming is still in progress and the write buffer is unavailable.

The software flow to write to a card that incorporates the internal DMA and the write operation is a multi-block write. For the other methods (CPU polling status with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to write to a card is described below:

1. Check the card status, wait until the card is ready for data.
2. Set the card block length/size:
   - For eMMC and SD cards, use SET_BLOCKLEN (CMD16).
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT (CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Clear INT_SIGNAL_EN[BWRIEN] to 0 to disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer.

   For multi-block write, set AC12EN to 1 if the device is memory device: such as eMMC, SD Card, SD Combo Card (memory portion). SDIO does not support CMD12, but only supports using IO_RW_DIRECT (CMD52) to abort.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred, or any other error that occurred during the auto12 command sending and response receiving, if the device is memory device: such as eMMC, SD Card, SD Combo Card (memory portion).

38.5.3.1.2 DDR write

uSDHC supports the dual data rate mode.

The software flow to write to a card in the DDR mode is described as below:

1. Check the card status and wait until the card is ready for data.

   NOTE

   For eMMC, block length can only be set to 512 Byte.
2. Set the uSDHC number block register (NOB), where nob is 5, for instance.
3. Set the eMMC, SD card to high-speed mode and use SWITCH (CMD6); or set SDIO to high-speed mode and use IO_RW_DIRECT (CMD52).
4. Set the eMMC bus or SD with 4-bit/8-bit DDR mode and use SWITCH (CMD6); or set SDIO to with 4-bit DDR mode and use IO_RW_DIRECT (CMD52).
5. Clear INT_SIGNAL_EN[BWRIEN] to 0 to disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. Set DDR_EN to 1.

   For multi-block write, set AC12EN to 1 if the device is memory device: such as eMMC, SD Card, SD Combo Card (Memory portion). SDIO does not support CMD12, but only supports using IO_RW_DIRECT (CMD52) to abort.
6. Wait for the Transfer Complete interrupt.

7. Check the status bit to see if a write CRC error occurred, or any other error that occurred during the auto12 command sending and response receiving if the device is memory device: such as eMMC, SD Card, SD Combo Card (Memory portion).

38.5.3.1.3 Write with Pause

The write operation can be paused during the transfer. Instead of stopping the CLK at any time to pause all the operations, which is also inaccessible to the host driver, the driver can set the Stop At Block Gap Request (SABGREQ) bit in the Protocol Control register to pause the transfer between the data blocks. As there is no time-out condition in a write operation during the data blocks, a write to all types of cards can be paused in this way, and if the DATA0 line is not required to de-assert to release the busy state, no suspend command is needed.

Similar to the flow described in Normal Write, the write with pause is shown with the same kind of write operation:

1. Check the card status and wait until the card is ready for data.
2. Set the card block length/size:
   - For eMMC and SD cards, use SET_BLOCKLEN (CMD16).
   - For SDIO or the I/O portion of SD Combo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Clear INT_SIGNAL_EN[BWRIEN] to 0 to disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer.

   For multi-block write, set AC12EN to 1 if the device is memory device: such as eMMC, SD Card, SD Combo Card (Memory portion). SDIO does not support CMD12, but only supports using IO_RW_DIRECT (CMD52) to abort.
6. Set the SABGREQ bit.
7. Wait for the Transfer Complete interrupt.
8. Clear the SABGREQ bit.
9. Check the status bit to see if a write CRC error occurred.
10. Set the CREQ bit to continue the write operation.
11. Wait for the Transfer Complete interrupt.
12. Check the status bit to see if a write CRC error occurred, or any other error that occurred during the auto12 command sending and response receiving if the device is memory device: such as eMMC, SD Card, SD Combo Card (Memory portion).

The number of blocks left during the data transfer is accessible by reading the contents of BLK_ATT[BLKCNT]. As the data transfer and the setting of the SABGREQ bit are concurrent, and the delay of register read and the register setting, the actual number of blocks left may not be exactly the value read earlier. The host driver reads the value of BLKCNT after the transfer is paused and the Transfer Complete interrupt (INT_STATUS[TC]) is received.

It is also possible that the last block has begun when the Stop At Block Gap Request is sent to the buffer. In this case, the next block gap is the end of the transfer. These types of requests are ignored, the driver should treat these as a non-pause transfer, and deal with it as a common write operation.

38.5.3.2 Block read

Information about Normal read, DDR read, Read with Pause, and Delay Line (DLL) in Read Path are detailed in the sections below.

38.5.3.2.1 Normal read

For block reads, the basic unit of data transfer is a block whose maximum size is stored in areas defined by the corresponding card specification. A CRC is appended to the end of each block, ensuring data transfer integrity. The CMD17, CMD18, CMD53, CMD60, CMD61, and so on, can initiate a block read. After completing the transfer, the card returns to the Transfer state. For multi blocks read, data blocks are continuously transferred until a stop command is issued.

The software flow to read from a card that incorporates the internal DMA and the read operation is a multi-block read with the Auto CMD12 enabled. For the other methods (CPU polling status with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to read from a card is described below:

1. Check the card status and wait until card is ready for data.
2. Set the card block length/size:
   - For eMMC and SD cards, use SET_BLOCKLEN (CMD16).
   - For SDIO or the I/O portion of SD Combo Cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Clear INT_SIGNAL_EN[BRRIEN] to 0 to disable the buffer read ready interrupt, configure the DMA settings, and enable the uSDHC DMA when sending the command with data transfer.

   For multi-block read, set AC12EN to 1 if the device is memory device: such as eMMC, SD Card, Combo Card (Memory portion). SDIO does not support CMD12, but only supports using IO_RW_DIRECT (CMD52) to abort.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a read CRC error occurred, or any other error occurred during the auto12 command sending and response receiving if the device is memory device: such as eMMC, SD Card, Combo Card (Memory portion).

38.5.3.2.2 DDR read

The uSDHC module supports dual data rate mode.

The software flow to write to a card in the DDR mode is described below:

1. Check the card status and wait until the card is ready for data.

   NOTE

   For eMMC, block length can only be set to 512 Bytes.
2. Set the uSDHC number block register (NOB) where nob is 5, for instance.
3. Set the eMMC, SD card to high-speed mode and use SWITCH (CMD6); or set SDIO to high-speed mode and use IO_RW_DIRECT (CMD52).
4. Set the eMMC bus or SD with 4-bit/8-bit DDR mode and use SWITCH (CMD6); or set SDIO to with 4-bit DDR mode and use IO_RW_DIRECT (CMD52).
5. Clear INT_SIGNAL_EN[BRRIEN] to 0 to disable the buffer write ready interrupt, configure the DMA settings, and enable the uSDHC DMA when sending the command with data transfer. Set DDR_EN to 1.

   For multi-block read, set AC12EN to 1 if the device is memory device: such as eMMC, SD Card, Combo Card (Memory portion). SDIO does not support CMD12, but only supports using IO_RW_DIRECT (CMD52) to abort.
6. Wait for the Transfer Complete interrupt.

7. Check the status bit to see if a write CRC error occurred, or any other error that occurred during the auto12 command sending and response receiving if the device is memory device: such as eMMC, SD Card, Combo Card (Memory portion).

38.5.3.2.3 Read with Pause

The read operation can be paused during the transfer. Instead of stopping the CLK at any time to pause all the operations, which is also inaccessible to the host driver, the driver can set the Stop At Block Gap Request (SABGREQ) bit in Control register to pause the transfer between the data blocks.

Only SDIO and SD Combo Card support Read Wait feature while Read with Pause. Read Wait feature means that it uses DATA2 to allow uSDHC to signal the SDIO or SD Combo Card to temporarily halt the sending of read data by a card during the block gap. Before setting SABGREQ to 1, ensure to set the RWCTL bit to 1 after recognizing the read wait capability of SDIO (SRW bit in CCCR register is 1).

Similar to the flow described in Normal read, the read with pause is shown with the same kind of read operation:

1. Check the SRW bit in the CCCR register on SDIO or SD Combo Card (I/O portion) to confirm that the card supports the Read Wait mode.
2. If SDIO or Combo Card (I/O portion) supports Read Wait, set RWCTL to 1.

   If the device is memory device: such as eMMC, SD Card, or SD Combo Card (Memory portion), set RD_DONE_NO_8CLK to 1.
3. Check the card status and wait until the card is ready for data.
4. Set the card block length/size:
   - For eMMC and SD cards, use SET_BLOCKLEN (CMD16).
   - For SDIO or the I/O portion of SD Combo Cards, use IO_RW_DIRECT (CMD52) to set the I/O Block Size bitfield in the CCCR register (for function 0) or FBR register (for functions 1~7).
5. Set the uSDHC block length register to be the same as the block length set for the card in step 4.
6. Set the uSDHC number block register (NOB), where nob is 5, for instance.
7. Clear INT_SIGNAL_EN[BRRIEN] to 0 to disable the buffer read ready interrupt, configure the DMA setting, and enable the uSDHC DMA when sending the command with data transfer.

   For multi-block read, set AC12EN to 1 if the device is memory device: such as eMMC, SD Card, Combo Card (Memory portion). SDIO does not support CMD12, but only supports using IO_RW_DIRECT (CMD52) to abort.
8. Set the SABGREQ bit.
9. Wait for the Transfer Complete interrupt.
10. Clear the SABGREQ bit.
11. Check the status bit to see if read CRC error occurred.
12. Set the CREQ bit to continue the read operation.
13. Wait for the Transfer Complete interrupt.
14. Check the status bit to see if a read CRC error occurred, or any other error occurred during the auto12 command sending and response receiving if the device is memory device: such as eMMC, SD Card, Combo Card (Memory portion).

Similar to the Write operation, it is possible to meet the ending block of the transfer when paused. In this case, uSDHC ignores the Stop At Block Gap Request and treats it as a command read operation.

If the Suspend command is sent and the transfer is later resumed by means of a Resume command, uSDHC takes the command as a normal one accompanied with data transfer. It is left for the host driver to set all the relevant registers before the transfer

is resumed. For memory device, if there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set, as well as the AC12EN bit. Then uSDHC automatically sends CMD12 to mark the end of multi-block transfer.

38.5.3.2.4 Delay Line (DLL) in read path

The DLL is newly added to assist in sampling read data. The DLL provides the ability to programmatically select a quantized delay (in fractions of the clock period) regardless of on-chip variations such as process, voltage, and temperature (PVT).

The reasons why DLL is needed for uSDHC are these:

- The path of read data traveling from card to host varies.
- In the eMMC and SD cards DDR mode, the minimum input setup and hold time are both at 2.5 ns.

The data sampling window is so small that the delay of loopback clock needs to be accurate and consistent regardless of PVT. The DLL takes the divided card_clk as the reference clock and loopback clock as the input clock. It then generates a delayed version of the input clock according to the programmed target delay.

The DLL can be disabled or bypassed, and it can also be manually set for a fixed delay in the override mode. The override value set is the number of delay cells. In the override mode, there is no need to set the DLL_enable. Another DLL mode is target value mode. In this mode, the DLL automatically adjusts the number of delay cells according to the target value set by the user and PVT changes. Be aware that the target value is in units of 1/32 of the clock reference period. If the card_clk is 100Mhz, then the reference clock period is 10ns; setting target value of 16 means 5ns =(16/32)*10ns. The software can disable automatic update by the setting dll_gate_update bit.

As you might change the frequency of card_clk from time to time by changing SDCLKFS[7:0]/DVS[3:0], the software must adjust the delay value to ensure it works correctly when the reference clock (card_clk) is changed. If DLL is to be used, make sure SDCLKFS is used (at least be set to divided by 2) so that the REF_CLK generated are the same frequency as card_clk. There are two DLLs, PARTS DLL and STROBE DLL. PARTS DLL controls the SD_CLK loop back delay for which DLL (Delay Line) Control (DLL_CTRL) and DLL Status (DLL_STATUS) are used. STROBE DLL controls the STROBE loop back delay for which Strobe DLL control (STROBE_DLL_CTRL) and Strobe DLL status (STROBE_DLL_STATUS) are used. For PARTS DLL, below flow can be taken as reference. For STROBE DLL, similar register field with prefix STROBE in Strobe DLL control (STROBE_DLL_CTRL) and Strobe DLL status (STROBE_DLL_STATUS) must be used.

Step 1: Set the DLL_CTRL_RESET and DLL_CTRL_ENABLE fields

Step 2: Configure the SDCLKFS[7:0] and DVS[3:0]

Step 3: Wait until SDSTB is asserted

Step 4: Clear the DLL_CTRL_RESET field

Step 5: Wait until both the DLL_STS_SLV_LOCK and DLL_STS_REF_LOCK are asserted

Step 6: Set the DLL_CTRL_SLV_FORCE_UPD

Step 7: Clear the DLL_CTRL_SLV_FORCE_UPD

NOTE

The software should make sure that the DLL_CTRL_SLV_FORCE_UPD lasts for at least one card_clk. So, the software may need to add some delay between step 6 and step 7.

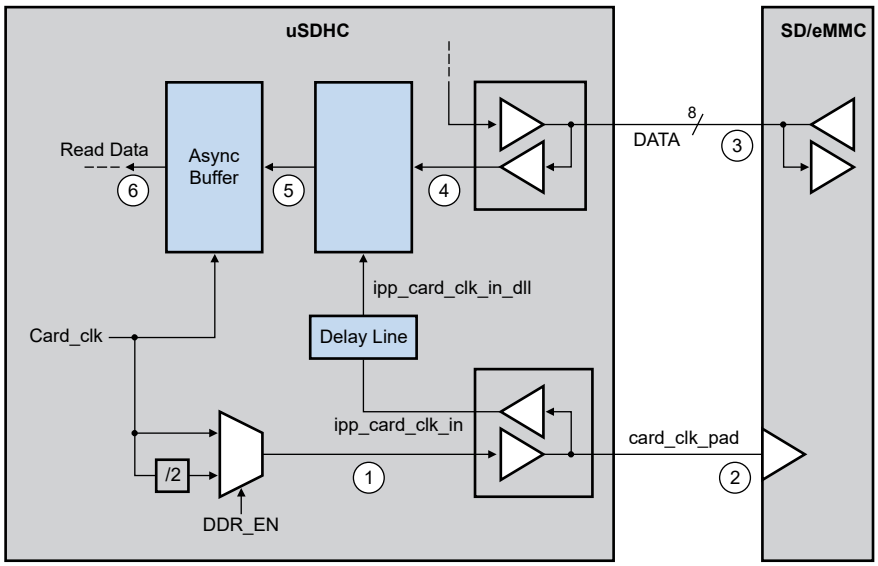


Figure 144. DLL in read path

38.5.3.3 Suspend Resume

The uSDHC module supports the Suspend Resume operations of SDIO and Combo Card, although slightly different than the suggested implementation of Suspend in the SDIO specification. Both Read and Write support Suspend and Resume, however ADMA operation does not support this function.

38.5.3.3.1 Suspend

After setting the SABGREQ bit, the host driver may send a Suspend command to switch to another function of SDIO and Combo Card. The uSDHC module does not monitor the content of the response, so it does not know if the Suspend command succeeded or not. Accordingly, it does not de-assert Read Wait for read pause. To solve this problem, the host driver does not mark the Suspend command as "Suspend", i.e., setting the CMDTYP bits to 01. Instead, the driver sends this command as if it were a normal command, and only when the command succeeds, and the BS bit is set in the response, can the Driver send another command marked as "Suspend" to inform uSDHC that the current transfer is suspended. Here is the sequence for the Suspend operation:

1. Check CCCR[SRW] on SDIO or SD Combo Card (I/O portion) to confirm that the card supports the Read Wait mode. If SDIO or Combo Card (I/O portion) supports Read Wait, set the RWCTL bit to 1.
2. Set the SABGREQ bit to 1 to pause the current data transfer at block gap. Wait for the BGE bit set as 1, and wait for the Transfer Complete interrupt. Save the context registers in the system memory for later use, including the DMA System Address Register DS_ADDR (for internal DMA operation), and the Block Attribute Register BLK_ATT.
3. Send the Suspend command to suspend the active function. The CMDTYP bit field must be 2'b00.

   The following steps show an example of CMD52 function0 suspend with CMDTYP=2'b00:

   a. Set CMD_ARG to 0x88001802

   b. Set MIX_CTRL to 0x80000000

   c. Set CMD_XFR_TYP to 0x341A0000
4. Check the BS bit of the CCCR in the response (CMD_RSP0[0]). If it is 1, repeat this step until the BS bit is cleared or abandon the suspend operation according to the Driver strategy.
5. Send another normal I/O command to the suspended function. The CMDTYP of this command must be 2'b01, so uSDHC can detect this special setting and be informed that the paused operation has successfully suspended.

The following steps show an example of CMD52 function0 suspend with CMDTYP=2'b01:

a. Set CMD_ARG to 0x00001800

b. Set MIX_CTRL to 0x80000000

c. Set CMD_XFR_TYP to 0x345A0000

6. Clear the BGE bit to 0, and clear the SABGREQ bit to 0. Begin operation for another function on SDIO and Combo Card (I/O portion).

38.5.3.3.2 Resume

To resume the data transfer, a Resume command is issued:

1. To resume the suspended function, restore the context registers (DS_ADDR and BLK_ATT) with the saved value in the step #2 of the Suspend operation.
2. Send the Resume command: In the Transfer Type register, all the bitfields are set to the value as if this were another ordinary data transfer instead of a transfer resume (except the CMDTYP is set to 2'b10).

   The following steps show an example of CMD52 function0 resume with CMDTYP=2'b10:

   a. Set CMD_ARG to 0x88001A01

   b. Set MIX_CTRL either to 0x80000023 (Suspend Write, Host to Card) or to 0x80000033 (Suspend Read, Card to Host)

   c. Set CMD_XFR_TYP to 0x34BA0000

3. If the Resume command (INT_STATUS[CC]) has responded, the data transfer is resumed.

38.5.3.4 ADMA usage

To use the ADMA in a data transfer, the host driver must prepare the correct descriptor chain prior to sending the read/write command. The steps to prepare the correct descriptor chain are as follows:

1. Create a descriptor to set the data length that the current descriptor group is about to transfer. The data length should be even numbers of the block size.
2. Create another descriptor to transfer the data from the address setting in this descriptor. The data address must be at a page boundary (4KB address aligned).
3. If necessary, create a Link descriptor containing the address of the next descriptor. The descriptor group is created in steps 1 ~ 3.
4. Repeat steps 1 ~ 3 until all descriptors are created.
5. In the last descriptor, set the End flag to 1 and make sure the total length of all descriptors matches the product of the block size and block number configured in the Block Attribute Register.
6. Set the ADMA System Address Register to the address of the first descriptor and set the DMAS field in the Protocol Control Register to 01 to select the ADMA.
7. Issue a write or read command with the DMAEN bit set to 1 in the Transfer Type Register.
8. Once the ADMA transfer is complete, clear PROT_CTRL[DMASEL] to 00b.

Steps 1 ~ 5 are independent of step 6, so step 6 can finish before steps 1 ~ 5. Regarding the descriptor configuration, it is recommended not to use the Link descriptor as it requires extra system memory access.

38.5.3.5 Transfer error

Information about CRC, Internal DMA, Transfer ADMA, and Auto CMD12 errors are detailed in the sections below.

38.5.3.5.1 CRC error

It is possible at the end of a block transfer that a write CRC status error or read CRC error occurs. For this type of error, the latest block received is discarded. This is because the integrity of the data block is not guaranteed. It is recommended to discard the following data blocks and re-transfer the block from the corrupted one. For a multi-block transfer, the host driver issues a CMD12 to abort the current process and start the transfer by a new data command. In this scenario, even when the AC12EN and BCEND bits are set, uSDHC does not automatically send a CMD12 because the last block is not transferred. On the other hand, if it is within the last block that the CRC error occurs, an Auto CMD12 is sent by uSDHC. In this case, the host driver re-sends or re-obtains the last block with a single block transfer.

38.5.3.5.2 Internal DMA error

During the data transfer with internal Simple DMA, if the DMA engine encounters an error on the AXI bus, the DMA operation stops, and the DMA error interrupt is sent to the host system. When acknowledged by such an interrupt, the host driver calculates the start address of data block in which the error occurs. The start address can be calculated by either:

- Reading the DMA System Address register: The error occurs during the previous burst. Considering the block size, and the start address of the next burst transfer, it is straight forward to obtain the start address of the corrupted block.
- Reading the BLKCNT field of the Block Attribute register: Considering the number of blocks left, the total number to transfer, the start address of transfer, and the size of each block, the start address of corrupted block can be determined.To use this method, MIX_CTRL[BCEN] bit has to be set to enable Block Attribute register update.

When a DMA error occurs, it is recommended to abort the current transfer by means of a CMD12 (for multi block transfer), apply a reset for data, and restart the transfer from the corrupted block to recover from the error.

38.5.3.5.3 Transfer ADMA error

There are three kinds of possible ADMA errors: The AXI transfer, invalid descriptor, and data-length mismatch. Whenever these errors occur, the DMA transfer stops and the corresponding error status bit is set. For acknowledging the status, the host driver should recover the error as shown below and re-transfer from the place of interruption.

- AXI transfer error: Such errors may occur during data transfer or descriptor fetch. For either scenario, it is recommended to retrieve the transfer context, reset for the data part and re-transfer the block that was corrupted, or to the next block if no block is corrupted.
- Invalid descriptor error: For such errors, it is recommended to retrieve the transfer context, reset for the data part and recreate the descriptor chain from the invalid descriptor and issue a new transfer. As the data to transfer now may be less than the previous setting, the data length configured in the new descriptor chain should match the new value.
- Data-length mismatch error: It is similar to recover from this error. The Host Driver polls relating registers to retrieve the transfer context, apply a reset for the data part, configure a new descriptor chain and make another transfer if there is data left. Like the previous scenario of the invalid descriptor error, the data length must match the new transfer.

38.5.3.5.4 Auto CMD12 error

After the last block of the multi-block transfer is sent or received, and the AC12EN bit is set when the data transfer is initiated by the data command, uSDHC automatically sends a CMD12 to the card to stop the transfer. When errors with this command occur, it is recommended to the host driver to deal with the situations in the following manner:

- Auto CMD12 response time-out: It is not certain whether the command is accepted by the card or not. The host driver clears the auto CMD12 error status bits and re-send CMD12 until it is accepted by the card.
- Auto CMD12 response CRC error: As the card responds to CMD12, it stops the transfer. The host driver may ignore the error and clear the error status bit.
- Auto CMD12 conflict error or not sent: The command is not sent; therefore, the host driver sends a CMD12 manually.

38.5.3.6 Card interrupt

The external cards can inform the host controller by means of some special signals. For SDIO, it can be the low-level on the DATA1 line during some special period. The uSDHC module only monitors the DATA1 line and supports the SDIO interrupt.

When the SDIO interrupt is captured by uSDHC, and the host system is informed by uSDHC asserting the uSDHC interrupt line, the interrupt service from the host driver is called.

As the interrupt source is controlled by the external card, the interrupt from SDIO must be serviced before the CINT bit is cleared. Refer to Card interrupt handling for the card interrupt handling flow.

38.5.4 Switch function

A switch command is issued by the host driver to enable new features added to the eMMC and SD cards. The eMMC and SD cards can transfer data at bus widths other than 1-bit. Different speed modes are also defined. To enable these features, a switch command is issued by the host driver.

For SDIO, the high-speed mode /DDR50/SDR50/SDR104 are enabled by writing the EHS bit in the CCCR register after the SHS bit is confirmed as high. For SD cards, the high-speed mode/DDR50/SDR50/SDR104 are queried and enabled by a CMD6 (with the mnemonic symbol as SWITCH_FUNC). For eMMC, the high-speed mode HS200 or HS400 is queried by a CMD8 and enabled by a CMD6 (with the mnemonic symbol as SWITCH).

The SDR4-bit, SDR8-bit, DDR4-bit, and DDR8-bit width of eMMC is also enabled by the SWITCH command, but with a different argument.

These new functions can also be disabled by a software reset. For SDIO, it can be done by setting the RES bit in the CCCR register. For other cards, it can be accomplished by issuing a CMD0. This method of restoring to the normal mode is not recommended because a complete identification process is needed before the card is ready for data transfer.

For the sake of simplicity, the following pseudo-code examples do not show current capability check, which is recommended in the function switch process.

38.5.4.1 Query, enable, and disable SDIO high-speed mode

```
enable_sdio_high_speed_mode(void)
{
send CMD52 to query bit SHS at address 0x13;
if (SHS bit is '0') report SDIO does not support high speed mode and return;
send CMD52 to set bit EHS at address 0x13 and read after write to confirm EHS bit is set;
change clock divisor value or configure the system clock feeding into uSDHC to generate the card_clk
of around 50MHz;
(data transactions like normal peers)
}
disable_sdio_high_speed_mode(void)
{
send CMD52 to clear bit EHS at address 0x13 and read after write to confirm EHS bit is cleared;
change clock divisor value or configure the system clock feeding into uSDHC to generate the card_clk
of the desired value below 25MHz;
(data transactions like normal peers)
}
```

38.5.4.2 Query, enable, and disable SD high-speed mode/DDR50 /SDR50 /SDR104

```
enable_sd_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0xFFFFFx and read 64 bytes of data accompanying the R1 response;(high speed
mode,x=1; SDR50,x=2; SDR104,x=3; DDR50,x=4;)
wait data transfer done bit is set;
check if the bit x of received 512 bits is set;
```

```
if (bit 401 is '0') report the SD card does not support high speed mode and return;
if (bit 402 is '0') report the SD card does not support SDR50 mode and return;
if (bit 403 is '0') report the SD card does not support SDR104 mode and return;
if (bit 404 is '0') report the SD card does not support DDR50 mode and return;
      send CMD6, with argument 0x80FFFFFx and read 64 bytes of data accompanying the R1 response;
(high speed mode,x=1; SDR50,x=2; SDR104,x=3; DDR50,x=4;)
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the card_clk
of around 50MHz for high speed mode, 100MHz for SDR50, 200MHz for SDR104, 50MHz for DDR50;
(data transactions like normal peers)
}
disable_sd_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0x80FFFFF0 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

38.5.4.3 Query, enable, and disable eMMC high-speed mode

```
enable_mmc_high_speed_mode(void)
{
send CMD9 to get CSD value of eMMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the eMMC does not support high speed mode and return;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of eMMC;
extract the value of CARD_TYPE field to check the 'high speed mode' in this eMMC is 26MHz or
52MHz;
send CMD6 with argument 0x1B90100;
send CMD13 to wait card ready (busy line released);
send CMD8 to get EXT_CSD value of eMMC;
check if HS_TIMING byte (byte number 185) is 1;
if (HS_TIMING is not 1) report eMMC switching to high speed mode failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 26MHz or 52MHz according to the CARD_TYPE;
(data transactions like normal peers)
}
disable_mmc_high_speed_mode(void)
{
send CMD6 with argument 0x2B90100;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of eMMC;
check if HS_TIMING byte (byte number 185) is 0;
if (HS_TIMING is not 0) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 20MHz;
(data transactions like normal peers)
}
```

38.5.4.4 Set eMMC bus width

```
change_mmc_bus_width(void)
{
send CMD9 to get CSD value of eMMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the eMMC does not support multiple bit width and return;
send CMD6 with argument 0x3B70x00; (8-bit(dual data rate), x=6; 4-bit(dual data rate), x=5;8-bit,
x=2; 4-bit, x=1; 1-bit, x=0)
send CMD13 to wait card ready (busy line released);
(data transactions like normal peers)
}
```

38.5.5 ADMA operation

Here are the codes for the ADMA1 and ADMA2 operations.

38.5.5.1 ADMA1 operation

```
Set_adma1_descriptor
{
if (to start data transfer) {
// Make sure the address is 4KB align.
Set 'Set' type descriptor;
{
Set Act bits to 01;
Set [31:12] bits data length (byte unit);
}
Set 'Tran' type descriptor;
{
Set Act bits to 10;
Set [31:12] bits address (4KB align);
}
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to 11;
Set [31:12] bits the next descriptor address (4KB aligned);
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set End bit to 1;
}
if (to generate interrupt for this descriptor) {
Set Int bit to 1;
}
Set Valid bit to 1;
}
```

38.5.5.2 ADMA2 operation

```
Set_adma2_descriptor
{
if (to start data transfer) {
// Make sure the address is a 32-bit boundary (lower 2-bit are always '00').
Set higher 32-bit of descriptor for this data transfer initial address;
Set [31:16] bits data length (byte unit);
```

```
Set Act bits to '10';
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to '11';
// Make sure the address is 32-bit boundary (lower 2-bit are always set to '00').
Set higher 32-bit of descriptor for the next descriptor address;
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set 'End' bit '1';
}
if (to generate interrupt for this descriptor) {
Set 'Int' bit '1';
}
Set the 'Valid' bit to '1';
}
```

38.5.6 Fast boot operation

38.5.6.1 Normal fast boot flow

Here are the steps for normal fast boot flow:

1. Software must configure the SYS_CTRL[INITA] bit to make sure that 74 card clocks are finished.
2. Software must configure the eMMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 0 (normal fast boot), and bit 4 to select the ack mode. If the data is sent through the DMA mode, the software should configure bit 7 to enable the automatic stop at block gap feature, and configure bit 3-bit 0 to select the ack timeout value according to the SD CLK frequency.
3. Software then needs to configure the Block Attributes Register to set the block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes.
4. Software must configure the Protocol control register to set Data Transfer Width (DTW). If in the DDR fast boot mode, DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software needs to configure the Command Argument Register to set argument if needed (no need in normal fast boot).
6. Software must configure the Transfer Type Register to start the boot process. In normal boot mode, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN retain the default value, where DPSEL bit is set to 1, DTDSEL is set to 1 and MSBSEL is set to 1.
7. DMAEN should be configured as 0 in the polling mode and if BCEN is configured as 1, it is recommended to configure the number of blocks in the Block Attributes Register to the maximum value. If in DDR fast boot mode, DDR_EN needs to be set to 1.
8. When step 6 is configured, the boot process begins. The software needs to poll the data buffer ready status to read the data from the buffer in time. If a boot timeout happens (ack times out or the first data read times out), an interrupt is triggered, and the software must configure eMMC Boot Register to bit 6 to 0 to disable boot. This makes CMD high, then after at least 56 clocks, it is ready to begin a normal initialization process.
9. If there is no timeout, software needs to determine when the data read is finished and then configure eMMC Boot Register bit 6 to 0 to disable boot. This render CMD line high and command completed asserted. After at least 56 clocks, it is ready to begin the normal initialization process.
10. You must reset the host and begin the normal process.

38.5.6.2 Alternative fast boot flow

Here are the steps for alternative fast boot flow:

1. Software needs to configure SYS_CTRL[INITA] to make sure 74 card clocks are finished.
2. Software needs to configure eMMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 1 (alternative boot), and bit 4 to select the ack mode or not. If data needs to be sent through the DMA mode, then configure bit 7 to enable the automatic stop at block gap feature. Software should also configure bit 3-bit 0 to select the ack timeout value according to the SD clock frequency.
3. Software then needs to configure the Block Attributes Register to set the block size and count. If in the DDR fast boot mode, the block size only can be configured to 512 bytes.
4. Software needs to configure the Protocol control register to set the data transfer width (DTW). If in the DDR fast boot mode, DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software needs to configure Command Argument Register to set argument to 0xFFFFFFFA.
6. Software needs to configure the Transfer Type Register to start the boot process by CMD0 with the 0xFFFFFFFA argument. In alternative boot, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN, and DMAEN retain the default value. DPSEL bit is set to 1, DTDSEL is set to 1, and MSBSEL is set to 1. Note DMAEN should be configured as 0 in the polling mode, and if BCEN is configured as 1 in polling mode, it is recommended to configure the block count in the Block Attributes Register to the maximum value. If in the DDR fast boot mode, DDR_EN needs to be set to 1.
7. When step 6 is configured, the boot process begins. Software needs to poll the data buffer ready status to read the data from the buffer in time. If there is a boot timeout (ack data timeout in 50ms or data timeout in 1s), the host sends out the interrupt and the software needs to send CMD0 with reset and then configure the boot enable bit to 0 to stop this process.
8. If there is no time out, the software needs to decide when to stop the boot process, and send out the CMD0 with reset and then after the command is completed, configure the eMMC Boot Register bit 6 to stop the process. After 8 clocks from the command completion, the target (card) is ready for the identification step.
9. You must reset the host and begin the normal process.

38.5.6.3 Fast boot application case (in DMA mode)

In the boot application case, because the image destination and the image size are contained in the beginning of the image, it is necessary to switch DMA parameters on the fly during eMMC fast boot.

In fast boot, the host can use Advanced DMA2 (ADMA2) with two destinations.

The detailed flow is described below:

1. The software needs to configure INIT_ACTIVE bit (system control register bit 27) to make sure that 74 card clocks are finished.
2. The software needs to configure the eMMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot) or 1 (alternative boot); and bit 4 to select the ack mode. In DMA mode, configure bit 7 to 1 to enable the automatic stop at block gap feature. Also configure bits[31-16] to set the (BLK_CNT - VALUE1). Here VALUE1 is the value of the block count that needs to transfer the first time, so that the host stops at the block gap when the uSDHC controller gets VAULE1 blocks from the device. Also, configure bits[3-0] to select the ack timeout value according to the SD clock frequency.
3. The software then needs to configure the Block Attributes Register to set block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes. In DMA mode, it is recommended to set the block count (BLK_CNT) to the max value (16'hffff).
4. The software needs to configure Protocol Control Register to set DTW (data transfer width). If in DDR fast boot mode, the DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software enable ADMA2 by configuring Protocol Control Register bits [9-8].
6. The software needs to set at least three pairs of ADMA2 descriptor in boot memory (that is, in IRAM, at least six words). The first pair descriptor defines the start address (that is, IRAM) and data length (that is,512byte*VALUE1) of the first part boot code. The software also needs to set the second pair descriptor, the second start address (any value that

is writable), and data length is suggested to set 1~2word (record as VALUE2). Note that the second couple desc also transfers useful data even at lease 1 word, because our ADMA2 cannot support 0 data_length data transfer descriptor.

7. The software needs to configure Command Argument Register to set argument to 0xFFFFFFFA in alternative fast boot and do not need to be set in normal fast boot.
8. The software needs to configure Transfer Type Register to start the boot process. CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN, and DMAEN retain the default value. DPSEL bit is set to 1, DTDSEL is set to 1, and MSBSEL is set to 1. DMAEN is configured as 1 in the DMA mode. And, if BCEN is configured as 1, then configure blk no in Bock Attributes Register to the max value. And, if in the DDR fast boot mode, DDR_EN needs to be set to 1.
9. When step 8 is configured, boot process begins, the first VALUE1 block number data gets transferred. The software needs to poll the TC bit (bit1 in Interrupt Status Register) to determine first transfer is ended. Also, the software needs to polling the BGE bit (bit2 in Interrupt Status Register) to determine if the first transfer stops at the block gap.
10. When TC and BGE bits are set to 1, the software can analyze the first code of VALUE1 block, initializes the new memory device, if required, and sets the third pair of descriptors to define the start address and length of the remaining part of the boot code (VALUE3, the remain boot code block). Remember to set the last descriptor with END.
11. The software needs to configure the eMMC Boot Register (offset 0xc4) again. Set bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot), to 1 (alternative boot); and bit 4 to select the ack mode or not. In the DMA mode, configure bit 7 to 1 for enabling the automatically stop at block gap feature. Also, configure bit31-bit16 to set the (BLK_CNT - (VALUE1+1+VALUE3)), that host stops at block gap when the uSDHC controller gets (VALUE1+1+VALUE3)) blocks from device totally include the blocks received in step 9. And need to configure bit 3-bit0 to select the ack timeout value according to the sd clk frequency. Note that the software does not need to configure the BLK_CNT again, because it is counted down automatically by the uSDHC controller.
12. The software needs to clear the TC and BGE bits and the software needs to clear SABGREQ (bit 16 in the Protocol control register) and set CREQ (bit17 in the Protocol control register) to 1 to resume the data transfer. Host transfers the VALUE2 and VALUE3 data to the destination that is set by descriptor.
13. The software needs to do poll BGE bit to determine if the fast boot is over.

Note:

- When ADMA boot flow starts, for uSDHC, it is like a normal ADMA read operation. So, set ADMA2 descriptor as the normal ADMA2 transfer.
- Need a few words length memory to keep descriptor.
- For the 1~2-word data in second descriptor setting, it is a useful data, so the software needs to deal the data because of the application case.

38.6 uSDHC memory map and register definition

38.6.1 USDHC register descriptions

This section includes the module memory map and detailed descriptions of all registers.

See the table below for the register memory map of uSDHC. All these registers only support 32-bit accesses.

NOTE

The uSDHC registers are 32-bit wide and only support 32-bit access.

38.6.1.1 USDHC memory map

uSDHC1 base address: 4285_0000h

uSDHC2 base address: 4286_0000h

uSDHC3 base address: 428B_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	DMA System Address (DS_ADDR)	32	RW	0000_0000
4	Block Attributes (BLK_ATT)	32	RW	0001_0000
8	Command Argument (CMD_ARG)	32	RW	0000_0000
C	Command Transfer Type (CMD_XFR_TYP)	32	RW	0000_0000
10	Command Response0 (CMD_RSP0)	32	R	0000_0000
14	Command Response1 (CMD_RSP1)	32	R	0000_0000
18	Command Response2 (CMD_RSP2)	32	R	0000_0000
1C	Command Response3 (CMD_RSP3)	32	R	0000_0000
20	Data Buffer Access Port (DATA_BUFF_ACC_PORT)	32	RW	0000_0000
24	Present State (PRES_STATE)	32	R	See section
28	Protocol Control (PROT_CTRL)	32	RW	0880_0020
2C	System Control (SYS_CTRL)	32	RW	0080_800F
30	Interrupt Status (INT_STATUS)	32	RW	0000_0000
34	Interrupt Status Enable (INT_STATUS_EN)	32	RW	0000_0000
38	Interrupt Signal Enable (INT_SIGNAL_EN)	32	RW	0000_0000
3C	Auto CMD12 Error Status (AUTOCMD12_ERR_STATUS)	32	RW	0000_0000
40	Host Controller Capabilities (HOST_CTRL_CAP)	32	RW	03F3_B407
44	Watermark Level (WTMK_LVL)	32	RW	0810_0810
48	Mixer Control (MIX_CTRL)	32	RW	8000_0000
50	Force Event (FORCE_EVENT)	32	RW	0000_0000
54	ADMA Error Status (ADMA_ERR_STATUS)	32	R	0000_0000
58	ADMA System Address (ADMA_SYS_ADDR)	32	RW	0000_0000
60	DLL (Delay Line) Control (DLL_CTRL)	32	RW	0000_0000
64	DLL Status (DLL_STATUS)	32	R	0000_0200
68	CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS)	32	RW	0000_0000
70	Strobe DLL control (STROBE_DLL_CTRL)	32	RW	0000_0000
74	Strobe DLL status (STROBE_DLL_STATUS)	32	R	0000_0200
C0	Vendor Specific Register (VEND_SPEC)	32	RW	3000_7809
C4	eMMC Boot (MMC_BOOT)	32	RW	0000_0000
C8	Vendor Specific 2 Register (VEND_SPEC2)	32	RW	0001_9006
CC	Tuning Control (TUNING_CTRL)	32	RW	0021_2800
100	Command Queuing Version (CQVER)	32	R	0000_0510

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
104	Command Queuing Capabilities (CQCAP)	32	RW	0000_310A
108	Command Queuing Configuration (CQCFG)	32	RW	0000_0000
10C	Command Queuing Control (CQCTL)	32	RW	0000_0000
110	Command Queuing Interrupt Status (CQIS)	32	RW	0000_0000
114	Command Queuing Interrupt Status Enable (CQISTE)	32	RW	0000_0000
118	Command Queuing Interrupt Signal Enable (CQISGE)	32	RW	0000_0000
11C	Command Queuing Interrupt Coalescing (CQIC)	32	RW	0000_0000
120	Command Queuing Task Descriptor List Base Address (CQTDLBA)	32	RW	0000_0000
124	Command Queuing Task Descriptor List Base Address Upper 32 Bits (CQTDLBAU)	32	RW	0000_0000
128	Command Queuing Task Doorbell (CQTDBR)	32	RW	0000_0000
12C	Command Queuing Task Completion Notification (CQTCN)	32	RW	0000_0000
130	Command Queuing Device Queue Status (CQDQS)	32	R	0000_0000
134	Command Queuing Device Pending Tasks (CQDPT)	32	R	0000_0000
138	Command Queuing Task Clear (CQTCLR)	32	RW	0000_0000
140	Command Queuing Send Status Configuration 1 (CQSSC1)	32	RW	0001_1000
144	Command Queuing Send Status Configuration 2 (CQSSC2)	32	RW	0000_0000
148	Command Queuing Command Response for Direct-Command Task (CQCRDCT)	32	R	0000_0000
150	Command Queuing Response Mode Error Mask (CQRMEM)	32	RW	FDF9_A080
154	Command Queuing Task Error Information (CQTERRI)	32	R	0000_0000
158	Command Queuing Command Response Index (CQCRI)	32	R	0000_0000
15C	Command Queuing Command Response Argument (CQCRA)	32	R	0000_0000

38.6.1.2 DMA System Address (DS_ADDR)

This register contains the physical system memory address used for DMA transfers.

Offset

Register	Offset
DS_ADDR	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DS_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DS_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 DS_ADDR	System address **DMA system address / argument 2** When ACMD23_ARGU2_EN is set to 0, SDMA uses this register as system address and supports only 32-bit addressing mode. Auto CMD23 cannot be used with SDMA. When ACMD23_ARGU2_EN is set to 1, SDMA uses ADMA System Address register (05Fh – 058h) instead of this register to support both ADMA1 and ADMA2. This register is used only for Argument2 and SDMA may use Auto CMD23. 1. **SDMA system address** Because the address must be word (4 bytes) aligned, the least two bits are reserved and always set to 0. When uSDHC stops a DMA transfer, this register points out the system address of the next contiguous data position. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operation during transfers may return an invalid value. The host driver initializes this register before starting a DMA transaction. After DMA has stopped, the system address of the next contiguous data position can be read from this register. This register is protected during a data transfer. When data lines are active, write to this register is ignored. The host driver waits until the DLA field in the Present State register is cleared, before writing to this register. The uSDHC internal DMA does not support a virtual memory system. It only supports continuous physical memory access. Because this register supports dynamic address reflecting, when TC field is set, it automatically alters the value of internal address counter, so the software cannot change this register when TC field is set. Such restriction is also listed in Software restrictions. 2. **Argument 2** This register is used with the Auto CMD23 to set a 32-bit block count value to the argument of the CMD23 while executing Auto CMD23. If Auto CMD23 is used with ADMA, the full 32-bit block count value can be used. If Auto CMD23 is used without ADMA, the available block count value is limited by the Block Count register. 65535 blocks is the maximum value in this case.

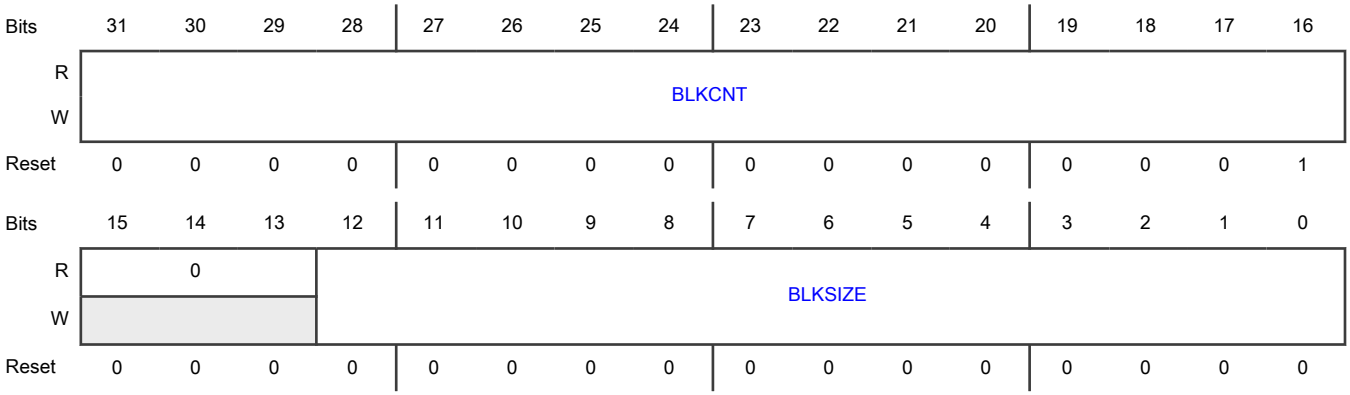
38.6.1.3 Block Attributes (BLK_ATT)

This register is used to configure the number of data blocks and the number of bytes in each block.

Offset

Register	Offset
BLK_ATT	4h

Diagram



Fields

Field	Description
31-16 BLKCNT	Blocks count for current transfer This field is enabled when the Block Count Enable field in the Transfer Mode register is set to 1 and is valid only for multiple block transfers. For single block transfer, this field always reads as 1. The host driver sets this field to a value between 1 and the maximum block count. The uSDHC module decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to zero results in no data blocks being transferred. This field should be accessed only when no transaction is executing (that is, after transactions are stopped). During data transfer, read operations on this field may return an invalid value and write operations are ignored. When saving transfer content because of a Suspend command, the number of blocks yet to be transferred can be determined by reading this field. The reading of this field should be applied after transfer is paused by stop at block gap operation and before sending the command marked as suspend. This is because when the Suspend command is sent out, uSDHC treats the current transfer as stopped and change the BLKCNT field back to its original value instead of keeping the dynamical indicator of the remaining block count. When restoring transfer content prior to issuing a Resume command, the host driver restores the previously saved block count. **NOTE** Although the BLKCNT field is 0 after reset, the read of reset value is 0x1. This is because when MSBSEL field is indicating a single block transfer, the read value of BLKCNT is always 1. 0000_0000_0000_0000 - Stop count 0000_0000_0000_0001 - 1 block 0000_0000_0000_0010 - 2 blocks

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1111_1111_1111_1111 - 65535 blocks
15-13 —	Reserved
12-0 BLKSIZE	Transfer block size This field specifies the block size for block data transfers. Values ranging from 1 byte up to the maximum buffer size can be set. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations are ignored. 0_0000_0000_0000 - No data transfer 0_0000_0000_0001 - 1 byte 0_0000_0000_0010 - 2 bytes 0_0000_0000_0011 - 3 bytes 0_0000_0000_0100 - 4 bytes 0_0001_1111_1111 - 511 bytes 0_0010_0000_0000 - 512 bytes 0_1000_0000_0000 - 2048 bytes 1_0000_0000_0000 - 4096 bytes

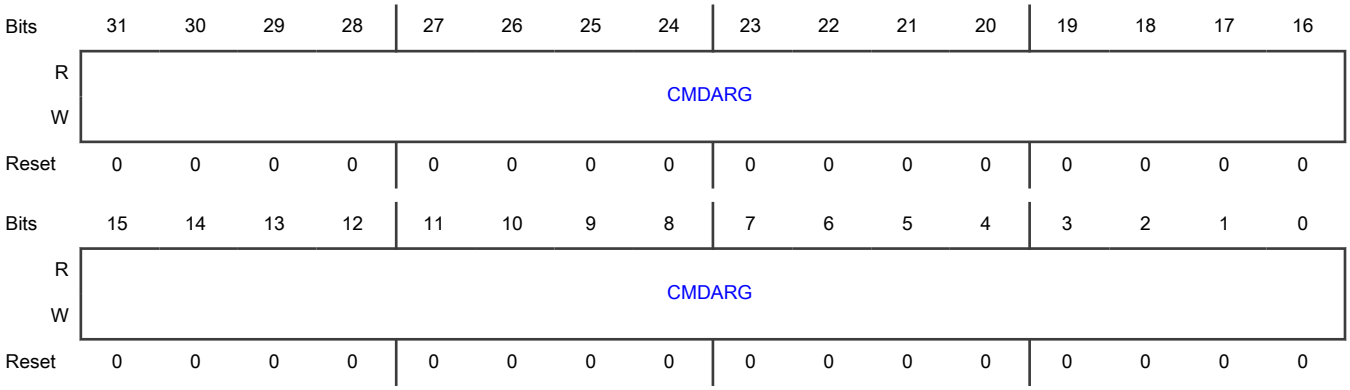
38.6.1.4 Command Argument (CMD_ARG)

This register contains the SD/eMMC command argument.

Offset

Register	Offset
CMD_ARG	8h

Diagram



Fields

Field	Description
31-0 CMDARG	Command argument The SD/eMMC command argument is specified as bits 39-8 of the command format in the SD or eMMC specification. This field is write protected when the Command Inhibit (CMD) field in the Present State register is set.

38.6.1.5 Command Transfer Type (CMD_XFR_TYP)

This register is used to control the operation of data transfers. The host driver sets this register before issuing a command followed by a data transfer or before issuing a Resume command. To prevent data loss, uSDHC prevents writing to the bits, which are involved in the data transfer of this register, when data transfer is active. These fields are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

The host driver checks the Command Inhibit DAT field (PRES_STATE[CDIHB]) and the Command Inhibit CMD field (PRES_STATE[CIHB]) in the Present State register before writing to this register. When the CDIHB field in the Present State register is set, any attempt to send a command with data by writing to this register is ignored; when the CIHB field is set, any write to this register is ignored.

On sending commands with data transfer involved, it is mandatory that the block size is non-zero. Block count must also be non-zero, or indicated as a single block transfer (bit 5 of this register is '0' when written), or block count is disabled (bit 1 of this register is '0' when written), otherwise uSDHC ignores the sending of this command and do nothing. For write command, with all above restrictions, it is also mandatory that the write protect switch is not active (PRES_STATE[WPSPL] field of Present State register is '1'); otherwise, WP is activated as '1', write to memory and combo cards are ignored by cards (in that case, there is no need to send write command anymore).

If the commands with data transfer do not receive the response in 64 clock cycles, that is, if response time-out happens, uSDHC treats the external device, does not accept the command, and stops the data transfer. In this scenario, the driver should issue the command again to retry the transfer. It is also possible that for some reason the card responds to the command but uSDHC does not receive the response, and if it is internal DMA (either simple DMA or ADMA) read operation, the external system memory is over-written by the internal DMA with data sent back from the card.

The table below shows the summary of how register settings determine the type of data transfer.

Table 231. Transfer type register setting for various transfer types

Multi/single block select	Block count enable	Block count	Function
0	Do not care	Do not care	Single transfer
1	0	Do not care	Infinite transfer
1	1	Positive number	Multiple transfer
1	1	Zero	No data transfer

The table below shows the relationship between the Command Index Check Enable and the Command CRC Check Enable, regarding the Response Type bits as well as the name of the response type.

Table 232. Relationship between parameters and the name of the response type

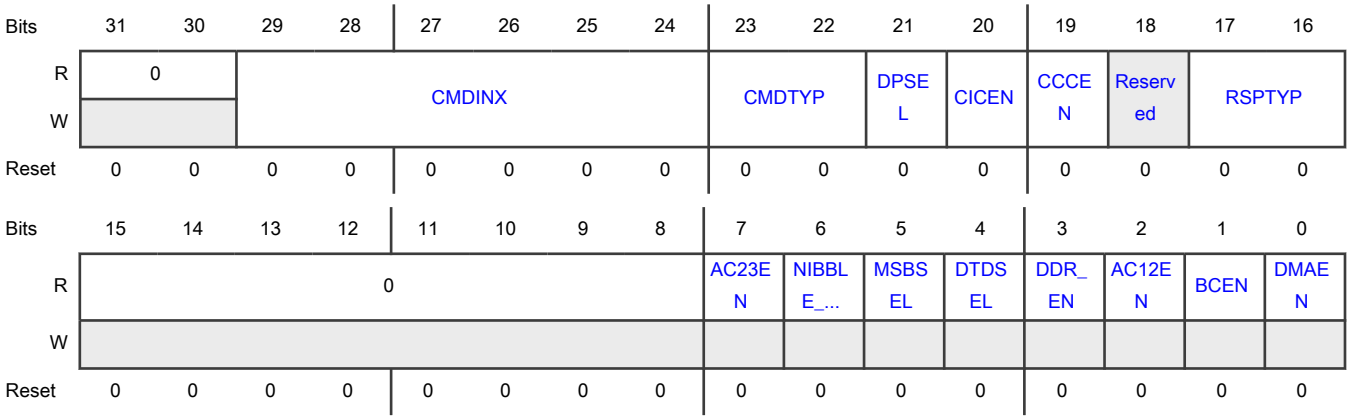
Response type	Index check enable	CRC check enable	Name of response type
00	0	0	No response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R5, R6
11	1	1	R1b,R5b

- In the SDIO specification, response type notation for R5b is not defined. R5 includes R5b in the SDIO specification, but R5b is defined in this specification to specify that uSDHC checks the busy status after receiving a response. For example, usually CMD52 is used with R5, but the I/O abort command is used with R5b.
- The CRC fields for R3 and R4 are expected to be all 1 bits. The CRC check is disabled for these response types.

Offset

Register	Offset
CMD_XFR_TYP	Ch

Diagram



Fields

Field	Description
31-30 —	Reserved
29-24 CMDINX	Command index

Table continues on the next page...

Table continued from the previous page...

Field	Description
	These fields are set to the command number that is specified in bits 45-40 of the command-format in the SD Memory Card Physical Layer Specification and SDIO Specification.
23-22 CMDTYP	Command type There are three types of special commands: Suspend, Resume, and Abort. These fields are set to 00b for all other commands. • Suspend command: If the Suspend command succeeds, uSDHC assumes that the card bus has been released and that it is possible to issue the next command that uses the DATA line. Because uSDHC does not monitor the content of command response, it does not know if the Suspend command succeeded or not. It is the host driver's responsibility to check the status of the Suspend command and send another command marked as Suspend to inform uSDHC that a Suspend command was successfully issued. See Suspend Resume for more details. After the end bit of command is sent, uSDHC deasserts Read Wait for read transactions and stops checking busy for write transactions. In a 4-bit mode, the interrupt cycle starts. If the Suspend command fails, uSDHC maintains its current state, and the host driver restarts the transfer by setting the Continue Request field in the Protocol Control register. • Resume command: The host driver re-starts the data transfer by restoring the registers saved before sending the Suspend command and then sends the Resume command. The uSDHC module checks for a pending busy state before starting write transfers. • Abort command: If this command is set when executing a read transfer, uSDHC stops reads to the buffer. If this command is set when executing a write transfer, uSDHC stops driving the DATA line. After issuing the Abort command, the host driver should issue a software reset (Abort Transaction). 00 - Normal other commands 01 - Suspend CMD52 for writing bus suspend in CCCR 10 - Resume CMD52 for writing function select in CCCR 11 - Abort CMD12, CMD52 for writing I/O Abort in CCCR
21 DPSEL	Data present select This field is set to 1 to indicate that data is present and is transferred using the DATA line. It is set to 0 for the following: • Commands using only the CMD line (for example, CMD52) • Commands with no data transfer, but using the busy signal on DATA0 line (R1b or R5b (for example, CMD38)) **NOTE** In resume command, this field is set, and other bits in this register is set the same as when the transfer was initially launched. When the write protect switch is on, (that is, the WPSPL field is active as '0'), any command with a write operation ignored. When this field is set, while the DTDSEL field is 0, writes to the register Transfer Type are ignored. 0 - No data present 1 - Data present
20	Command index check enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
CICEN	If this field is set to 1, uSDHC checks the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this field is set to 0, the Index field is not checked. 0 - Disable command index check 1 - Enables command index check
19 CCCEN	Command CRC check enable If this field is set to 1, uSDHC checks the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this field is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. See RSPTYP[1:0] and Command Transfer Type (CMD_XFR_TYP). 0 - Disables command CRC check 1 - Enables command CRC check
18 —	Reserved
17-16 RSPTYP	Response type select 00 - No response 01 - Response length 136 10 - Response length 48 11 - Response length 48, check busy after response
15-8 —	Reserved
7 AC23EN	AC23EN This field is read when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is tied to '0'. When this field is set to 1, the host controller issues a CMD23 automatically before issuing a command specified in the Command Register. 0 - Disable 1 - Enable
6 NIBBLE_POS	NIBBLE_POS This field indicates the nibble position in the DDR 4-bit mode. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. 0- the sequence is 'odd high nibble -> even high nibble -> odd low nibble -> even low nibble'; 1- the sequence is 'odd high nibble -> odd low nibble -> even high nibble -> even low nibble'. 0 - Disable 1 - Enable
5	MSBSEL

Table continues on the next page...

Table continued from the previous page...

Field	Description
MSBSEL	This field enables multiple block DATA line data transfers. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. For any other commands, this field can be set to 0. If this field is 0, it is not necessary to set the Block Count register. See Command Transfer Type (CMD_XFR_TYP). 0 - Disable 1 - Enable
4 DTDSEL	DTDSEL This field defines the direction of DATA line data transfers. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. The field is set to 1 by the host driver to transfer data from the SD card to uSDHC and is set to 0 for all other commands. 0 - Disable 1 - Enable
3 DDR_EN	DDR_EN Dual data rate mode selection. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. 0 - Disable 1 - Enable
2 AC12EN	AC12EN Multiple block transfers for memory require a CMD12 to stop the transaction. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. When this field is set to 1, uSDHC issues a CMD12 automatically when the last block transfer has completed. The host driver is not set this field to issue commands that do not require CMD12 to stop a multiple block data transfer. In particular, secure commands defined in File Security Specification (see reference list) do not require CMD12. In single block transfer, uSDHC ignores this field no matter it is set or not. 0 - Disable 1 - Enable
1 BCEN	BCEN This field is used to enable the Block Count register, which is only relevant for multiple block transfers. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. When this field is 0, the internal counter for block is disabled, which is useful in executing an infinite transfer. 0 - Disable 1 - Enable
0 DMAEN	DMAEN This field enables DMA functionality. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is enabled; otherwise, this field is read-only. If this field is set to 1, a DMA operation begins when the host driver sets the DPSEL field of this register. Whether the simple DMA or the advanced DMA is active depends on the DMA Select field of the Protocol Control register.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Disable 1 - Enable

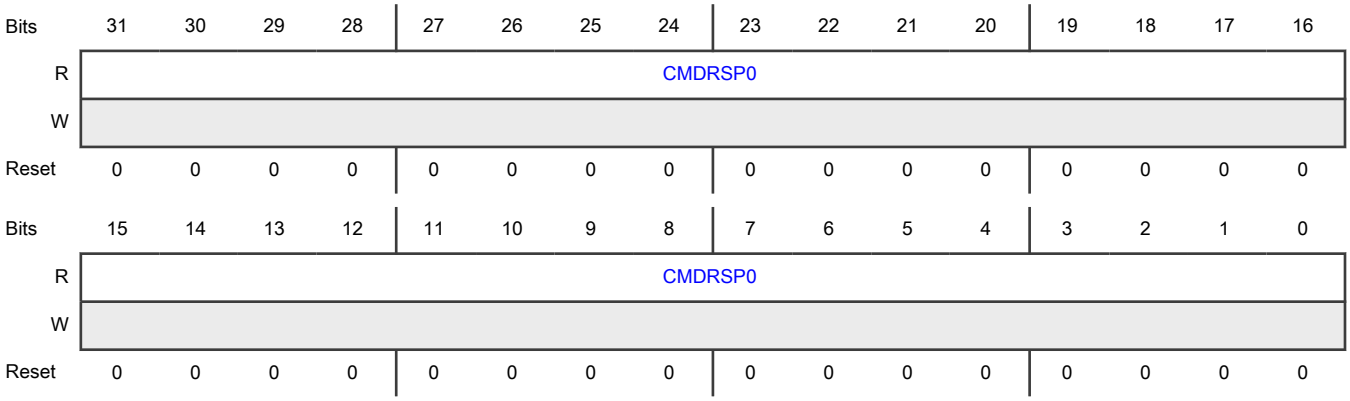
38.6.1.6 Command Response0 (CMD_RSP0)

This register is used to store part 0 of the response bits from the card.

Offset

Register	Offset
CMD_RSP0	10h

Diagram



Fields

Field	Description
31-0 CMDRSP0	Command response 0 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

38.6.1.7 Command Response1 (CMD_RSP1)

This register is used to store part 1 of the response bits from the card.

Offset

Register	Offset
CMD_RSP1	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CMDRSP1	Command response 1 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

38.6.1.8 Command Response2 (CMD_RSP2)

This register is used to store part 2 of the response bits from the card.

Offset

Register	Offset
CMD_RSP2	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CMDRSP2	Command response 2 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

38.6.1.9 Command Response3 (CMD_RSP3)

This register is used to store part 3 of the response bits from the card.

The table below describes the mapping of command responses from the SD bus to Command Response registers for each response type. In this table, R[ ] refers to a bit range within the response data as transmitted on the SD bus.

Table 233. Response bit definition for each response type

Response type	Meaning of response	Response field	Response register
R1,R1b (normal response)	Card status	R[39:8]	CMDRSP0
R1b (auto CMD12 response)	Card status for auto CMD12	R[39:8]	CMDRSP3
R2 (CID, CSD register)	CID/CSD register [127:8]	R[127:8]	{CMDRSP3[23:0], CMDRSP2, CMDRSP1, CMDRSP0}
R3 (OCR register)	OCR register for memory	R[39:8]	CMDRSP0
R4 (OCR register)	OCR register for I/O etc.	R[39:8]	CMDRSP0
R5, R5b	SDIO response	R[39:8]	CMDRSP0
R6 (publish RCA)	New published RCA[31:16] and card status[15:0]	R[39:9]	CMDRSP0

This table shows that most responses with a length of 48 (R[47:0]) have 32-bits of the response data (R[39:8]) stored in the CMDRSP0 register. Responses of type R1b (Auto CMD12 responses) have response data bits (R[39:8]) stored in the CMDRSP3 register. Responses with length 136 (R[135:0]) have 120-bits of the response data (R[127:8]) stored in the CMDRSP0, 1, 2, and 3 registers.

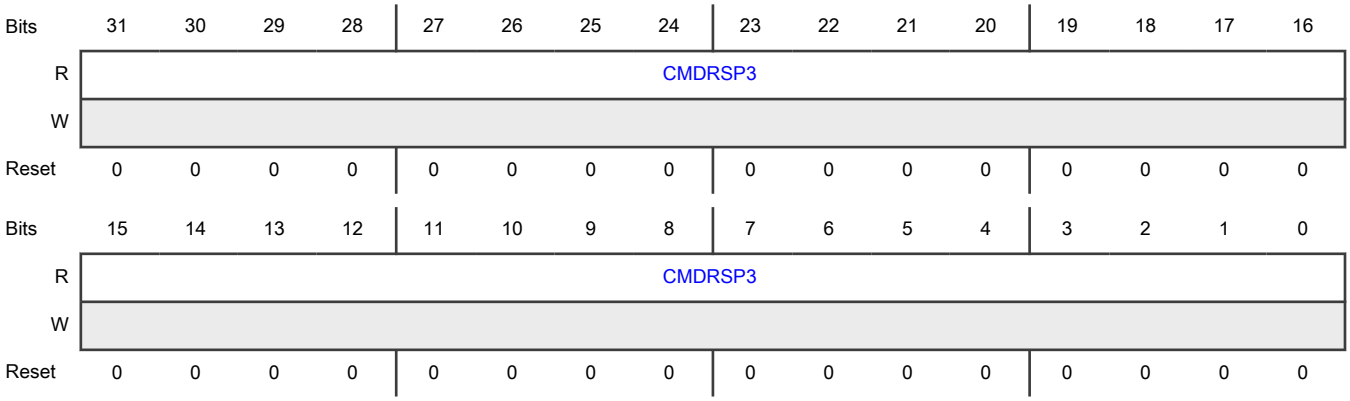
To be able to read the response status efficiently, uSDHC only stores part of the response data in the Command Response registers. This enables the host driver to efficiently read 32-bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by uSDHC (as specified by the Command Index Check Enable and the Command CRC Check Enable bits in the Transfer Type register) and generate an error interrupt if any error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, uSDHC checks R[47:1], and if the response length is 136 the uSDHC checks R[119:1].

Because uSDHC may have a multiple block data transfer executing concurrently with a CMD_wo_DAT command, uSDHC stores the Auto CMD12 response in the CMDRSP3 register. The CMD_wo_DAT response is stored in CMDRSP0. This allows uSDHC to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa. When uSDHC modifies part of the Command Response registers, as shown in the table above, it preserves the unmodified bits.

Offset

Register	Offset
CMD_RSP3	1Ch

Diagram



Fields

Field	Description
31-0 CMDRSP3	Command response 3 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

38.6.1.10 Data Buffer Access Port (DATA_BUFF_ACC_PORT)

The Buffer Data Port register is for 32-bit data access by the Arm platform. When the internal DMA is enabled, any write to this field is ignored, and any read from this field always yields 0s.

Offset

Register	Offset
DATA_BUFF_ACC_PORT	20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATCONT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATCONT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 DATCONT	Data content This field is used to access the internal buffer.

38.6.1.11 Present State (PRES_STATE)

The host driver can get status of uSDHC from this 32-bit read only register.

The host driver can issue CMD0, CMD12, CMD13 (for memory) and CMD52 (for SDIO) when the DATA lines are busy during a data transfer. These commands can be issued when Command Inhibit (CMD) is set to zero. Other commands are issued when Command Inhibit (DATA) is set to zero. Possible changes to the SD Physical Specification may add other commands to this list in the future.

NOTE

The reset value of Present State register depends on board connectivity.

Offset

Register	Offset
PRES_STATE	24h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLSL								CLSL	Reserved		Reserved	WPSPL	CDPL	0	CINST
W																
Reset	u	u	u	u	u	u	u	u	u	0	0	u	u	u	0	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSCD	0		RTR	BREN	BWEN	RTA	WTA	Reserved				SDSTB	DLA	CDIHB	CIHB
W																
Reset	1	0	0	0	0	0	0	0	1	0	0	0	u	0	0	0

Fields

Field	Description
31-24 DLSL	DATA[7:0] line signal level This field is used to check the DATA line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DATA0. The reset value is affected by the external pull-up / pull-down resistors. By default, the read value of this field after reset is 8'b11110111, when DATA3 is pulled down and the other lines are pulled up. 0000_0001 - Data 0 line signal level 0000_0010 - Data 1 line signal level 0000_0100 - Data 2 line signal level 0000_1000 - Data 3 line signal level 0001_0000 - Data 4 line signal level 0010_0000 - Data 5 line signal level 0100_0000 - Data 6 line signal level 1000_0000 - Data 7 line signal level
23 CLSL	CMD line signal level This field is used to check the CMD line level to recover from errors, and for debugging. The reset value is affected by the external pull-up / pull-down resistor, by default, the read value of this field after reset is 1'b1, when the command line is pulled up.
22-21 —	Reserved
20 —	Reserved
19 WPSPL	Write protect switch pin level

Table continues on the next page...

Table continued from the previous page...

Field	Description
	The Write Protect switch is supported for memory and combo cards. This field reflects the inverted value of the WP pin of the card socket. A software reset does not affect this field. The reset value is affected by the external write protect switch. If the WP pin is not used, it should be tied low, so that the reset value of this field is high and write is enabled. 0 - Write protected (WP = 1) 1 - Write enabled (WP = 0)
18 CDPL	Card detect pin level This field reflects the inverse value of the CD_B pin for the card socket. Debouncing is not performed on this field. This field may be valid, but is not guaranteed, because of propagation delay. Use of this field is limited to testing because it must be debounced by software. A software reset does not affect this field. A write to the Force Event Register does not affect this field. The reset value is affected by the external card detection pin. This field shows the value on the CD_B pin (that is, when a card is inserted in the socket, it is 0 on the CD_B input, and consequently, the CDPL reads 1. 0 - No card present (CD_B = 1) 1 - Card present (CD_B = 0)
17 —	Reserved
16 CINST	Card inserted This field indicates whether a card has been inserted. The uSDHC module debounces this signal so that the host driver does not need to wait for it to stabilize. Changing from a 0 to 1 generates a Card Insertion interrupt in the Interrupt Status register. Changing from a 1 to 0 generates a Card Removal interrupt in the Interrupt Status register. A write to the Force Event Register does not affect this field. The Software Reset For All in the System Control register does not affect this field. A software reset does not affect this field. 0 - Power on reset or no card 1 - Card inserted
15 TSCD	Tap select change done This field indicates the delay setting is effective after write CLK_TUNE_CTRL_STATUS register. 0 - Delay cell select change is not finished. 1 - Delay cell select change is finished.
14-13 —	Reserved
12 RTR	Re-Tuning Request (only for SD3.0 SDR104 mode, and eMMC HS200 mode) Host controller may request host driver to execute re-tuning sequence by setting this field when the data window is shifted by temperature drift and a tuned sampling point does not have a good margin to receive correct data.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field is cleared when a command is issued with setting Execute Tuning field in MIX_CTRL register. Changing of this field from 0 to 1 generates Re-Tuning Event. See Interrupt status registers for more detail. This field isn't set to 1 if Sampling Clock Select in the MIX_CTRL register is set to 0 (using fixed sampling clock). 0 - Fixed or well tuned sampling clock 1 - Sampling clock needs re-tuning
11 BREN	Buffer read enable This status field is used for non-DMA read transfers. The uSDHC module implements an internal buffer to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer. If this field is high, valid data greater than the watermark level exist in the buffer. A change of this field from 1 to 0 occurs when some reads from the buffer (read DATPORT (Base + 0x20)) are made and the buffer hasn't valid data greater than the watermark level. A change of this field from 0 to1 occurs when there is enough valid data ready in the buffer and the Buffer Read Ready interrupt has been generated and enabled. 0 - Read disable 1 - Read enable
10 BWEN	Buffer write enable This status field is used for non-DMA write transfers. The uSDHC module implements an internal buffer to transfer data efficiently. This read only flag indicates if space is available for write data. If this field is 1, valid data greater than the watermark level can be written to the buffer. A change of this field from 1 to 0 occurs when some writes to the buffer (write DATPORT (Base + 0x20)) are made and the buffer hasn't valid space greater than the watermark level. A change of this field from 0 to 1 occurs when the buffer can hold valid data greater than the write watermark level and the Buffer Write Ready interrupt is generated and enabled. 0 - Write disable 1 - Write enable
9 RTA	Read transfer active This status field is used for detecting completion of a read transfer. This field is set for either of the following conditions: • After the end field of the read command • When writing a 1 to the Continue Request field in the Protocol Control register to restart a read transfer A transfer complete interrupt is generated when this field changes to 0. This field is cleared for either of the following conditions: • When the last data block as specified by block length is transferred to the System, that is, all data are read away from uSDHC internal buffer. • When all valid data blocks have been transferred from uSDHC internal buffer to the System and no current block transfers are being sent because of the Stop At Block Gap Request being set to 1. 0 - No valid data

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Transferring data
8 WTA	Write transfer active This status field indicates a write transfer is active. If this field is 0, it means no valid write data exists in uSDHC. This field is set in either of the following cases: • After the end field of the write command • When writing 1 to the Continue Request field in the Protocol Control register to restart a write transfer This field is cleared in either of the following cases: • After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple) • After getting the CRC status of any block where data transmission is about to be stopped by a Stop At Block Gap Request During a write transaction, a Block Gap Event interrupt is generated when this field is changed to 0, as result of the Stop At Block Gap Request being set. This status is useful for the host driver in determining when to issue commands during Write Busy state. 0 - No valid data 1 - Transferring data
7-4 —	Reserved
3 SDSTB	SD clock stable This status field indicates that the internal card clock is stable. This field is for the host driver to poll clock status when changing the clock frequency. It is recommended to clear FRC_SDCLK_ON field in System Control register to remove glitches on the card clock when the frequency is changing. Before changing clock divisor value (SDCLKFS or DVS), host driver should make sure the SDSTB field is high. 0 - Clock is changing frequency and not stable. 1 - Clock is stable.
2 DLA	Data line active This status field indicates whether one of the DATA lines on the SD bus is in use. In the case of read transactions: This status indicates if a read transfer is executing on the SD bus. Changes in this value from 1 to 0, between data blocks, generates a Block Gap Event interrupt in the Interrupt Status register. This field is set in either of the following cases: • After the end field of the read command

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• When writing a 1 to the Continue Request field in the Protocol Control register to restart a read transfer This field is cleared in either of the following cases: • When the end field of the last data block is sent from the SD bus to uSDHC. • When the Read Wait state is stopped by a Suspend command and the DATA2 line is released. The uSDHC module waits at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), uSDHC can wait for a current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait to use the suspend / resume function. This field remains 1 during Read Wait. In the case of write transactions: This status indicates that a write transfer is executing on the SD bus. Changes in this value from 1 to 0 generate a Transfer Complete interrupt in the Interrupt Status register. This field is set in either of the following cases: • After the end field of the write command • When writing to 1 to the Continue Request field in the Protocol Control register to continue a write transfer This field is cleared in either of the following cases: • When the SD card releases Write Busy of the last data block, uSDHC also detects if the output is not busy. If the SD card does not drive the busy signal after the CRC status is received, uSDHC assumes the card drive "Not Busy". • When the SD card releases write busy, prior to waiting for write transfer, and because of a Stop At Block Gap Request. In the case of command with busy pending: This status indicates that a busy state follows the command and the data line is in use. This field is cleared when the DATA0 line is released. 0 - DATA line inactive 1 - DATA line active
1 CDIHB	Command Inhibit Data (DATA) This status field is generated if either the DAT Line Active or the Read Transfer Active is set to 1. If this field is 0, it indicates that uSDHC can issue the next SD / eMMC Command. Commands with a busy signal belong to Command Inhibit (DATA) (for example. R1b, R5b type). Changing from 1 to 0 generates a Transfer Complete interrupt in the Interrupt Status register. **NOTE** The SD host driver can save registers for a suspend transaction after this field has changed from 1 to 0. 0 - Can issue command that uses the DATA line 1 - Cannot issue command that uses the DATA line

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 CIHB	Command inhibit (CMD) If this status bit is 0, it indicates that the CMD line is not in use and uSDHC can issue a SD / eMMC command using the CMD line. This field is set also immediately after the Transfer Type register is written. This field is cleared when the command response is received. Even if the Command Inhibit (DATA) is set to 1, commands using only the CMD line can be issued if this field is 0. Changing from 1 to 0 generates a Command Complete interrupt in the Interrupt Status register. If uSDHC cannot issue the command because of a command conflict error (see Command CRC Error) or because of a Command Not Issued By Auto CMD12 Error, this field remains 1 and the Command Complete is not set. The Status of issuing an auto CMD12 does not show on this field. 0 - Can issue command using only CMD line 1 - Cannot issue command

38.6.1.12 Protocol Control (PROT_CTRL)

This register controls three cases to restart the transfer after stop at the block gap. Which case is appropriate depends on whether uSDHC issues a Suspend command or the SD card accepts the Suspend command.

- If the host driver does not issue a Suspend command, the Continue request is used to restart the transfer.
- If the host driver issues a Suspend command and the SD card accepts it, a Resume command is used to restart the transfer.
- If the host driver issues a Suspend command and the SD card does not accept it, the Continue request is used to restart the transfer.

Any time stop at block gap request stops the data transfer, the host driver waits for a Transfer Complete (in the Interrupt Status register), before attempting to restart the transfer. When restarting the data transfer by Continue Request, the host driver clears the Stop At Block Gap Request before or simultaneously.

Offset

Register	Offset
PROT_CTRL	28h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	NON_EXA...	Reserved			WECRM	WECINS	WECINT	Reserved			RD_DONE...	IABG	RWCTL	CREQ	SABGREQ
Reset	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						DMASEL		Reserved	Reserved	EMODE		D3CD	DTW		Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Fields

Field	Description
31 —	Reserved Always write as 0
30 NON_EXACT_BLK_RD	Non-exact block read Current block read is non-exact block read. It is only used for SDIO. 0 - The block read is exact block read. Host driver does not need to issue abort command to terminate this multi-block read. 1 - The block read is non-exact block read. Host driver needs to issue abort command to terminate this multi-block read.
29-27 —	Reserved
26 WECRM	Wakeup event enable on SD card removal This field enables a wakeup event, via a card removal, in the Interrupt Status register. FN_WUS (Wakeup Support) in CIS does not affect this field. When this field is set, the Card Removal Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Removal Status and uSDHC interrupt. 0 - Disables wakeup event enable on SD card removal 1 - Enables wakeup event enable on SD card removal
25 WECINS	Wakeup event enable on SD card insertion This field enables a wakeup event, via a card insertion, in the Interrupt Status register. FN_WUS (Wakeup Support) in CIS does not affect this field. When this field is set, the Card Insertion Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Insertion Status and uSDHC interrupt. 0 - Disable wakeup event enable on SD card insertion 1 - Enable wakeup event enable on SD card insertion
24 WECINT	Wakeup event enable on card interrupt This field enables a wakeup event, via a card interrupt, in the Interrupt Status register. This field can be set to 1 if FN_WUS (Wakeup Support) in CIS is set to 1. When this field is set, the Card Interrupt Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Interrupt Status and uSDHC interrupt. 0 - Disables wakeup event enable on card interrupt 1 - Enables wakeup event enable on card interrupt
23-21 —	Reserved Always write as 3'b100
20	Read performed number 8 clock

Table continues on the next page...

Table continued from the previous page...

Field	Description
RD_DONE_NO_8CLK	According to the SD/eMMC spec, for read data transaction, 8 clocks are needed after the end field of the last data block. So, by default(RD_DONE_NO_8CLK=0), eight clocks are active after the end field of the last read data transaction. However, these 8 clocks should not be active if user wants to use stop at block gap (include the auto stop at block gap in boot mode) feature for read and the RWCTL field (bit18) is not enabled. In this case, software should set RD_DONE_NO_8CLK to avoid these 8 clocks. Otherwise, the device might send extra data to uSDHC while uSDHC ignores these data. In a summary, this field should be set only if the use case needs to use stop at block gap feature while the device can't support the read wait feature.
19 IABG	Interrupt at block gap This field is valid only in 4-bit mode, of SDIO, and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If SDIO cannot signal an interrupt during a multiple block transfer, this field should be set to 0 to avoid an inadvertent interrupt. When the host driver detects an SDIO insertion, it sets this field according to the CCCR of the card. 0 - Disables interrupt at block gap 1 - Enables interrupt at block gap
18 RWCTL	Read wait control The read wait function provided by this field is optional for SDIO. If the card supports read wait, set this field to enable use of the read wait protocol to stop read data using the DATA2 line. Otherwise, uSDHC has to stop the SD clock to hold read data, which restricts commands generation. When the host driver detects an SDIO insertion, it sets this field according to the CCCR of the card. If the card does not support read wait, this field should never be set to 1; otherwise, DATA line conflicts might occur. If this field is set to 0, stop at block gap during read operation is also supported, but uSDHC stops the SD clock to pause reading operation. 0 - Disables read wait control and stop SD clock at block gap when SABGREQ field is set 1 - Enables read wait control and assert read wait without stopping SD clock at block gap when SABGREQ field is set
17 CREQ	Continue request This field is used to restart a transaction which was stopped using the stop at block gap request. When a suspend operation is not accepted by the card, it is also by setting this field to restart the paused transfer. To cancel stop at the block gap, set stop at block gap request to 0 and set this field to 1 to restart the transfer. The uSDHC module automatically clears this field, therefore it is not necessary for the host driver to set this field to 0. If both stop at block gap request and this field are 1, the continue request is ignored. 0 - No effect 1 - Restart
16 SABGREQ	Stop at block gap request This field is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the transfer complete is set to 1, indicating a transfer completion, the host driver leaves this field set to 1. Clearing both the stop at block gap request and continue request does not cause the transaction to

Table continues on the next page...

Table continued from the previous page...

Field	Description
	restart. Read Wait is used to stop the read transaction at the block gap. The uSDHC module supports the stop at block gap request for write transfers, but for read transfers it requires that SDIO support read wait. Therefore, the host driver does not set this field during read transfers unless SDIO supports Read Wait and has set the read wait control to 1; otherwise, uSDHC stops the SD bus clock to pause the read operation during block gap. In the case of write transfers in which the host driver writes data to the Data Port register, the host driver sets this field after all block data is written. If this field is set to 1, the host driver does not write data to the Data Port register after a block is sent. Once this field is set, the host driver does not clear this field before the Transfer Complete field in Interrupt Status register is set, otherwise uSDHC's behavior is undefined. This field effects read transfer active, write transfer active, DATA Line Active and Command Inhibit (DATA) in the Present State register. 0 - Transfer 1 - Stop
15-10 —	Reserved
9-8 DMASEL	DMA select This field is valid while DMA (SDMA or ADMA) is enabled and selects the DMA operation. 00 - No DMA or simple DMA is selected. 01 - ADMA1 is selected. 10 - ADMA2 is selected. 11 - Reserved
7 —	Reserved
6 —	Reserved
5-4 EMODE	Endian mode This field supports all three endian modes in data transfer. See Data buffer for more details. 00 - Big endian mode 01 - Half word big endian mode 10 - Little endian mode 11 - Reserved
3 D3CD	DATA3 as card detection pin If this field is set, DATA3 should be pulled down to act as a card detection pin. Be cautious when using this feature, because DATA3 is also a chip-select for the SPI mode. A pull-down on this pin and CMD0 might set the card into the SPI mode, which uSDHC does not support.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - DATA3 does not monitor card insertion 1 - DATA3 as card detection pin
2-1 DTW	Data transfer width This field selects the data width of the SD bus for a data transfer. The host driver sets it to match the data width of the card. Possible data transfer width is 1-bit, 4-bits or 8-bits. 00 - 1-bit mode 01 - 4-bit mode 10 - 8-bit mode 11 - Reserved
0 —	Reserved

38.6.1.13 System Control (SYS_CTRL)

This register provides control of the system. See detail in the field description.

Offset

Register	Offset
SYS_CTRL	2Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			RSTT	INITA	RSTD	RSTC	RSTA	IPP_RST...	RST_FIFO	Reserved	0	DTOCV			
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SDCLKFS								DVS				Reserved			
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Fields

Field	Description
31-29	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
28 RSTT	Reset tuning When set this field to 1, it resets tuning circuit. After tuning circuits are reset, bit value is 0. Clearing execute_tuning field in AUTOCMD12_ERR_STATUS also sets this field to 1 to reset tuning circuit
27 INITA	Initialization active When this field is set, 80 SD-clocks are sent to the card. After the 80 clocks are sent, this field is self cleared. This field is very useful during the card power-up period when 74 SD-clocks are needed and the clock auto gating feature is enabled. Writing 1 to this bit when this field is already 1 has no effect. Writing 0 to this field at any time has no effect. When either of the CIHB and CDIHB fields in the Present State register are set, writing 1 to this field is ignored (that is, when command line or data lines are active, write to this field is not allowed). On the other-hand, when this field is set, that is, during initialization active period, it is allowed to issue command, and the command bit stream appears on the CMD pad after all 80 clock cycles are done. So, when this command ends, the driver can make sure the 80 clock cycles are sent out. This is very useful when the driver needs to send 80 cycles to the card and does not want to wait till this field is self cleared.
26 RSTD	Software reset for data line Only part of the data circuit is reset. DMA circuit is also reset. After this field is set, the software waits for self-clear. The following registers and bits are cleared by this field: • Data Port register — Buffer is cleared and initialized • Present State register — Buffer read enable — Buffer write enable — Read transfer active — Write transfer active — DATA line active — Command Inhibit (DATA) • Protocol Control register — Continue request • Interrupt Status register — Buffer read ready — Buffer write ready — DMA interrupt — Block gap event — Transfer complete

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE When reset, the software must make sure there is no incomplete data transferring. If there is data transfer going on, the software needs to wait TC or DC INT_STATUS register is set. 0 - No reset 1 - Reset
25 RSTC	Software reset for CMD line Only part of the command circuit is reset. After this field is set, the software waits for self-clear. The following registers and bits are cleared by this field: • Present State Register — Command Inhibit (CMD) • Interrupt Status Register — Command Complete 0 - No reset 1 - Reset
24 RSTA	Software reset for all This reset affects the entire host controller except for the card detection circuit. RSTA resets all the registers that can be reset by RSTC/RSTD. During its initialization, the host driver is set this field to 1 to reset uSDHC. The uSDHC module resets this field to 0 when the capabilities registers are valid and the host driver can read them. Additional use of Software Reset For All does not affect the value of the capabilities registers. After this field is set, it is recommended that the host driver reset the external card and re-initialize it. After this field is set, the software should wait for self-clear. In tuning process, after every CMD19 is finished, this field is set to retest uSDHC. **NOTE** When reset, the software must make sure there is no incomplete data transferring. If there is data transfer going on, the software needs to wait TC or DC INT_STATUS register is set. 0 - No reset 1 - Reset
23 IPP_RST_N	Hardware reset This field has alternative function as below: • eMMC: be used to reset eMMC, through pad directly to eMMC RST_n pin. • Card: be used to control SD Card power enable or disable during warm reset, through pad directly to hardware reset on board.
22 RST_FIFO	Reset the async FIFO Reset the async FIFO between card interface and the internal logic. After this field is set, the software waits for self-clear.

Table continues on the next page...

Table continued from the previous page...

Field	Description
21 —	Reserved
20 —	Reserved
19-16 DTOCV	Data timeout counter value This value determines the interval by which DAT line timeouts are detected. See the Data Timeout Error field in the Interrupt Status register for information on factors that dictate time-out generation. Time-out clock frequency is generated by dividing the base clock SDCLK value by this value. The host driver can clear the Data Timeout Error Status Enable (in the Interrupt Status Enable register) to prevent inadvertent time-out events. 0000 - SDCLK x 2^{32} 0001 - SDCLK x 2^{33} 0010 - SDCLK x 2^{18} 0011 - SDCLK x 2^{19} 1101 - SDCLK x 2^{29}, recommend to use for supported speed modes except HS200, HS400, SDR104 mode 1110 - SDCLK x 2^{30}, recommend to use for HS200 and SDR104 mode 1111 - SDCLK x 2^{31}, recommend to use for HS400 mode
15-8 SDCLKFS	SDCLK frequency select This field is used to select the frequency of the SDCLK pin. The frequency is not programmed directly, rather this field holds the prescaler (of this register) and divisor (next register) of the Base Clock Frequency register. In Single Data Rate mode (DDR_EN field of MIXERCTRL is '0') Only the following settings are allowed: 80h) Base clock divided by 256 40h) Base clock divided by 128 20h) Base clock divided by 64 10h) Base clock divided by 32 08h) Base clock divided by 16 04h) Base clock divided by 8 02h) Base clock divided by 4 01h) Base clock divided by 2 00h) Base clock divided by 1 While in Dual Data Rate mode (DDR_EN field of MIXERCTRL is '1') Only the following settings are allowed:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	80h) Base clock divided by 512 40h) Base clock divided by 256 20h) Base clock divided by 128 10h) Base clock divided by 64 08h) Base clock divided by 32 04h) Base clock divided by 16 02h) Base clock divided by 8 01h) Base clock divided by 4 00h) Base clock divided by 2 When the software changes the DDR_EN field, SDCLKFS might need to be changed also. In Single Data Rate mode, setting 00h bypasses the frequency prescaler of the SD clock. Multiple bits must not be set, or the behavior of this prescaler is undefined. The two default divider values can be calculated by the frequency of ipg_perclk and the following Divisor bits. The frequency of SDCLK is set by the following formula: Clock Frequency = (Base Clock) / (prescaler x divisor) For example, in Single Data Rate mode, if the Base Clock Frequency is 96 MHz, and the target frequency is 25 MHz, then choosing the prescaler value of 01h and divisor value of 1h yields 24 MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400 kHz, the prescaler value of 08h and divisor value of eh yields the exact clock value of 400 kHz. The reset value of this field is 80h, so if the input Base Clock (ipg_perclk) is about 96 MHz, the default SD clock after reset is 375 kHz. Before changing clock divisor value (SDCLKFS or DVS), host driver should make sure the SDSTB field is high. If setting SDCLKFS and DVS can generate the same clock frequency,(for example, in SDR mode, SDCLKFS = 01h is same as DVS = 01h.), SDCLKFS is highly recommended. For HS400 mode, SDCLKFS must be used so that REF_CLK, generated for STROBE DLL, is at the same frequency as STROBE clock. For HS200 mode, if DLL is to be used, make sure SDCLKFS is used so that REF_CLK, generated for PARTS DLL, is at the same frequency as SD_CLK. For SDR104 mode, if DLL is to be used, make sure SDCLKFS is used so that REF_CLK, generated for PARTS DLL, is at the same frequency as SD_CLK.
7-4 DVS	Divisor This field is used to provide a more exact divisor to generate the desired SD clock frequency. Note the divider can even support odd divisors without deterioration of duty cycle. Before changing clock divisor value (SDCLKFS or DVS), Host driver should make sure the SDSTB field is high. The settings are as follows:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0000 - Divide-by-1 0001 - Divide-by-2 1110 - Divide-by-15 1111 - Divide-by-16
3-0 —	Reserved Always write as 1.

38.6.1.14 Interrupt Status (INT_STATUS)

An interrupt is generated when the normal interrupt signal enable is enabled and at least one of the status fields is set to 1. For all fields, writing 1 to a bit clears it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. For card interrupt, before writing 1 to clear, it is required that the card stops asserting the interrupt, meaning that when the card driver services the interrupt condition; otherwise, the CINT field is asserted again.

The table below shows the relationship between the command timeout error and the command complete.

Table 234. uSDHC status for command timeout error/command complete bit combinations

Command complete	Command timeout error	Meaning of the status
0	0	X
X	1	Response not received within 64 SDCLK cycles
1	0	Response received

The table below shows the relationship between the transfer complete and the data timeout error.

Table 235. uSDHC status for data timeout error/transfer complete bit combinations

Transfer complete	Data timeout error	Meaning of the status
0	0	X
0	1	Timeout occurred during transfer
1	X	Data transfer complete

The table below shows the relationship between the command CRC error and command timeout error.

Table 236. uSDHC status for command CRC error/command timeout error bit combinations

Command CRC error	Command timeout error	Meaning of the status
0	0	No error
0	1	Response timeout error

Table continues on the next page...

Table 236. uSDHC status for command CRC error/command timeout error bit combinations (continued)

Command CRC error	Command timeout error	Meaning of the status
1	0	Response CRC error
1	1	CMD line conflict

Offset

Register	Offset
INT_STATUS	30h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	Reserved		DMAE	0	TNE	0	AC12E	0	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE
W		W1C		W1C		W1C		W1C		W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_INT...	CQI	TP	RTE	0	Reserved		CINT	CRM	CINS	BRR	BWR	DINT	BGE	TC	CC
W		W1C	W1C	W1C		W1C		W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30-29 —	Reserved
28 DMAE	DMA error Occurs when an Internal DMA transfer has failed. This field is set to 1, when some error occurs in the data transfer. This error can be caused by either simple DMA or ADMA, depending on which DMA is in use. The value in DMA System Address register is the next fetch address where the error occurs. Because any error corrupts the whole data block, the host driver restarts the transfer from the corrupted block boundary. The address of the block boundary can be calculated either from the current DS_ADDR value or from the remaining number of blocks and the block size. 0 - No error 1 - Error

Table continues on the next page...

Table continued from the previous page...

Field	Description
27 —	Reserved
26 TNE	Tuning error: (only for SD3.0 SDR104 mode and eMMC HS200 mode) This field is set when an unrecoverable error is detected in a tuning circuit. By detecting Tuning Error, host driver needs to abort a command executing and perform tuning. This field is relevant in combination with the bus error only and must be cleared prior to all the write and read transactions.
25 —	Reserved
24 AC12E	Auto CMD12 error Occurs when detecting that one of the fields in the Auto CMD12 Error Status register has changed from 0 to 1. This field is set to 1, not only when the errors in Auto CMD12 occur, but also, when the Auto CMD12 is not executed due to the previous command error. 0 - No error 1 - Error
23 —	Reserved
22 DEBE	Data end bit error Occurs either when detecting 0 at the end field position of read data that uses the DATA line, or at the end field position of the CRC. This field is not asserted in tuning process. 0 - No error 1 - Error
21 DCE	Data CRC error Occurs when detecting a CRC error when transferring read data that uses the DATA line, or when detecting the Write CRC status having a value other than 010. This field is not asserted in tuning process. 0 - No error 1 - Error
20 DTOE	Data timeout error Occurs when detecting one of following time-out conditions. • Busy time-out for R1b, R5b type • Busy time-out after Write CRC status • Read Data time-out.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field is not asserted in tuning process. 0 - No error 1 - Time out
19 CIE	Command index error Occurs if a command index error occurs in the command response. This field is not asserted in tuning process. 0 - No error 1 - Error
18 CEBE	Command end bit error Occurs when detecting that the end field of a command response is 0. This field is not asserted in tuning process. 0 - No error 1 - End bit error generated
17 CCE	Command CRC error Command CRC Error is generated in two cases. • If a response is returned and the Command Timeout Error is set to 0 (indicating no time-out), this field is set when detecting a CRC error in the command response. • The uSDHC module detects a CMD line conflict by monitoring the CMD line when a command is issued. If uSDHC drives the CMD line to 1, but detects 0 on the CMD line at the next SDCLK edge, then uSDHC stops the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error should also be set to 1 to distinguish CMD line conflict. This field is not asserted in tuning process. 0 - No error 1 - CRC error generated
16 CTOE	Command timeout error Occurs only if no response is returned within 64 SDCLK cycles from the end field of the command. If uSDHC detects a CMD line conflict, in which case a Command CRC Error is also be set (as shown in Interrupt Status (INT_STATUS)), this field is set without waiting for 64 SDCLK cycles. This is because the command is stopped by uSDHC. This field is not asserted in tuning process. 0 - No error 1 - Time out
15	Error Interrupt Status

Table continues on the next page...

Table continued from the previous page...

Field	Description
ERR_INT_STATUS	This bit is set when any of the error status bit DMAE, AC12E, DEBE, DCE, DTOE, CIE, CEBE, CCE and CTOE is set.
14 CQI	Command queuing interrupt This interrupt is asserted when at least one of the bits in CQIS register is set. This interrupt is cleared only by clearing the source interrupt in CQIS register.
13 TP	Tuning pass:(only for SD3.0 SDR104 mode and eMMC HS200 mode) Current CMD19 transfer is done successfully. That is, current sampling point is correct.
12 RTE	Re-tuning event: (only for SD3.0 SDR104 mode and eMMC HS200 mode) This status is set if Re-Tuning Request in the Present State register changes from 0 to 1. host controller requests host driver to perform re-tuning for next data transfer. Current data transfer (not large block count) can be completed without re-tuning. 0 - Re-tuning is not required. 1 - Re-tuning should be performed.
11 —	Reserved
10-9 —	Reserved
8 CINT	Card interrupt This status field is set when an interrupt signal is detected from the external card. In 1-bit mode, uSDHC detects the Card Interrupt without the SD clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so the interrupt from card can only be sampled during interrupt cycle, introducing some delay between the interrupt signal from SDIO and the interrupt to the host system. Writing this field to 1 can clear this field, but as the interrupt source from SDIO does not clear, this field is set again. to clear this field, it is required to reset the interrupt source from the external card followed by a writing 1 to this field. When this status has been set, and the host driver needs to service this interrupt, the Card Interrupt Signal Enable in the Interrupt Signal Enable register should be 0 to stop driving the interrupt signal to the host system. After completion of the card interrupt service (It should reset the interrupt sources in SDIO and the interrupt signal might not be asserted), write 1 to clear this field, set the Card Interrupt Signal Enable to 1, and start sampling the interrupt signal again. 0 - No card interrupt 1 - Generate card interrupt
7 CRM	Card removal This status field is set if the Card Inserted field in the Present State register changes from 1 to 0. When the host driver writes this field to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state might possibly be changed when the host driver clears this field

Table continues on the next page...

Table continued from the previous page...

Field	Description
	and the interrupt event might not be generated. When this field is cleared, it is set again if no card is inserted. to leave it cleared, clear the Card Removal Status Enable field in Interrupt Status Enable register. 0 - Card state unstable or inserted 1 - Card removed
6 CINS	Card insertion This status field is set if the Card Inserted field in the Present State register changes from 0 to 1. When the host driver writes this field to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state might possibly be changed when the host driver clears this field and the interrupt event might not be generated. When this field is cleared, it is set again if a card is inserted. to leave it cleared, clear the Card Inserted Status Enable field in Interrupt Status Enable register. 0 - Card state unstable or removed 1 - Card inserted
5 BRR	Buffer read ready This status field is set if the Buffer Read Enable field, in the Present State register, changes from 0 to 1. See the Buffer Read Enable field in the Present State register for additional information. This field indicates that cmd19 is finished in tuning process. 0 - Not ready to read buffer 1 - Ready to read buffer
4 BWR	Buffer write ready This status field is set if the Buffer Write Enable field, in the Present State register, changes from 0 to 1. See the Buffer Write Enable field in the Present State register for additional information. 0 - Not ready to write buffer 1 - Ready to write buffer
3 DINT	DMA interrupt Occurs only when the internal DMA finishes the data transfer successfully. Whenever errors occur during data transfer, this field does not be set. Instead, the DMAE field is set. Either Simple DMA or ADMA finishes data transferring, this field is set. 0 - No DMA interrupt 1 - DMA interrupt is generated.
2 BGE	Block gap event If the Stop At Block Gap Request field in the Protocol Control register is set, this field is set when a read or write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this field is not set to 1. In the case of a Read Transaction: This field is set at the falling edge of the DATA Line Active Status (When the transaction is stopped at SD bus timing). The Read Wait must be supported to use this function.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	In the case of Write Transaction: This field is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing). 0 - No block gap event 1 - Transaction stopped at block gap
1 TC	Transfer complete This field is set when a read or write transfer is completed. In the case of a Read Transaction: This field is set at the falling edge of the Read Transfer Active Status. There are two cases in which this interrupt is generated. The first is when a data transfer is completed as specified by the data length (after the last data has been read to the host system). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request field in the Protocol Control register (after valid data has been read to the host system). In the case of a Write Transaction: This field is set at the falling edge of the DATA Line Active Status. There are two cases in which this interrupt is generated. The first is when the last data is written to the SD card as specified by the data length and the busy signal is released. The second is when data transfers are stopped at the block gap, by setting the Stop At Block Gap Request field in the Protocol Control register, and the data transfers are completed. (after valid data is written to the SD card and the busy signal released). In the case of a command with busy, this field is set when busy is deasserted. This field is not asserted in tuning process. 0 - Transfer does not complete 1 - Transfer complete
0 CC	Command complete This field is set when you receive the end field of the command response (except auto CMD12). See the Command Inhibit (CMD) in the Present State register. This field is not asserted in tuning process. 0 - Command not complete 1 - Command complete

38.6.1.15 Interrupt Status Enable (INT_STATUS_EN)

Setting any bit in this register to 1 enables the corresponding interrupt status be requested to the system. If any bit is set to 0, the corresponding interrupt request gets blocked.

- Depending on IABG field setting, uSDHC might be programmed to sample the card interrupt signal during the interrupt period and hold its value in the flip-flop. There are some delays on the Card Interrupt, asserted from the card, to the time the host system is informed.
- To detect a CMD line conflict, the host driver must set both Command Timeout Error Status Enable and Command CRC Error Status Enable to 1.

Offset

Register	Offset
INT_STATUS_EN	34h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	Reserved		DMAESEN	0	TNESEN	0	AC12ESEN	0	DEBESEN	DCESEN	DTOESEN	CIESEN	CEBESEN	CCESEN	CTOESEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	CQISEN	TPSEN	RTESEN	0	Reserved		CINTSEN	CRMSEN	CINSSEN	BRRSEN	BWRSEN	DINTSEN	BGESEN	TCSEN	CCSEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30-29 —	Reserved
28 DMAESEN	DMA error status enable 0 - Masked 1 - Enabled
27 —	Reserved
26 TNESEN	Tuning error status enable 0 - Masked 1 - Enabled
25 —	Reserved
24 AC12ESEN	Auto CMD12 error status enable 0 - Masked 1 - Enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
23 —	Reserved
22 DEBESEN	Data end bit error status enable 0 - Masked 1 - Enabled
21 DCESEN	Data CRC error status enable 0 - Masked 1 - Enabled
20 DTOESEN	Data timeout error status enable 0 - Masked 1 - Enabled
19 CIESEN	Command index error status enable 0 - Masked 1 - Enabled
18 CEBESEN	Command end bit error status enable 0 - Masked 1 - Enabled
17 CCESEN	Command CRC error status enable 0 - Masked 1 - Enabled
16 CTOESEN	Command timeout error status enable 0 - Masked 1 - Enabled
15 —	Reserved
14 CQISEN	Command queuing status enable 0 - Masked 1 - Enabled
13 TPSEN	Tuning pass status enable 0 - Masked 1 - Enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 RTESEN	Re-tuning event status enable 0 - Masked 1 - Enabled
11 —	Reserved
10-9 —	Reserved
8 CINTSEN	Card interrupt status enable If this field is set to 0, uSDHC clears the interrupt request to the system. The Card Interrupt detection is stopped when this field is cleared and restarted when this field is set to 1. The host driver should clear the Card Interrupt Status Enable before servicing the Card Interrupt and should set this field again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. 0 - Masked 1 - Enabled
7 CRMSEN	Card removal status enable 0 - Masked 1 - Enabled
6 CINSSEN	Card insertion status enable 0 - Masked 1 - Enabled
5 BRRSEN	Buffer read ready status enable 0 - Masked 1 - Enabled
4 BWRSEN	Buffer write ready status enable 0 - Masked 1 - Enabled
3 DINTSEN	DMA interrupt status enable 0 - Masked 1 - Enabled
2 BGESEN	Block gap event status enable 0 - Masked 1 - Enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 TCSEN	Transfer complete status enable 0 - Masked 1 - Enabled
0 CCSEN	Command complete status enable 0 - Masked 1 - Enabled

38.6.1.16 Interrupt Signal Enable (INT_SIGNAL_EN)

This register is used to select which interrupt status is indicated to the host system as the interrupt. These status fields all share the same interrupt lines. Setting any of these fields to 1 enables interrupt generation. The corresponding Status register field generates an interrupt when the corresponding interrupt signal enable field is set.

Offset

Register	Offset
INT_SIGNAL_EN	38h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	Reserved		DMAEIEN	0	TNEIEIN	0	AC12EIEN	0	DEBEIEN	DCEIEIN	DTOEIEN	CIEIEIN	CEBEIEN	CCEIEIN	CTOEIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	CQIIEN	TPIEN	RTEIEN	0	Reserved		CINTIEN	CRMIEN	CINSIEN	BRRIEN	BWRIEN	DINTIEN	BGEIEN	TCIEN	CCIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30-29 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
28 DMAEIEN	DMA error interrupt enable 0 - Masked 1 - Enable
27 —	Reserved
26 TNEIEN	Tuning error interrupt enable 0 - Masked 1 - Enabled
25 —	Reserved
24 AC12EIEN	Auto CMD12 error interrupt enable 0 - Masked 1 - Enabled
23 —	Reserved
22 DEBEIEN	Data end bit error interrupt enable 0 - Masked 1 - Enabled
21 DCEIEN	Data CRC error interrupt enable 0 - Masked 1 - Enabled
20 DTOEIEN	Data timeout error interrupt enable 0 - Masked 1 - Enabled
19 CIEIEN	Command index error interrupt enable 0 - Masked 1 - Enabled
18 CEBEIEN	Command end bit error interrupt enable 0 - Masked 1 - Enabled
17	Command CRC error interrupt enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
CCEIEN	0 - Masked 1 - Enabled
16 CTOEIEN	Command timeout error interrupt enable 0 - Masked 1 - Enabled
15 —	Reserved
14 CQIIEN	Command queuing signal enable 0 - Masked 1 - Enabled
13 TPIEN	Tuning pass interrupt enable 0 - Masked 1 - Enabled
12 RTEIEN	Re-tuning event interrupt enable 0 - Masked 1 - Enabled
11 —	Reserved
10-9 —	Reserved
8 CINTIEN	Card interrupt enable 0 - Masked 1 - Enabled
7 CRMIEN	Card removal interrupt enable 0 - Masked 1 - Enabled
6 CINSIEN	Card insertion interrupt enable 0 - Masked 1 - Enabled
5 BRRIEN	Buffer read ready interrupt enable 0 - Masked

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enabled
4 BWRIEN	Buffer write ready interrupt enable 0 - Masked 1 - Enabled
3 DINTIEN	DMA interrupt enable 0 - Masked 1 - Enabled
2 BGEIEN	Block gap event interrupt enable 0 - Masked 1 - Enabled
1 TCIEN	Transfer complete interrupt enable 0 - Masked 1 - Enabled
0 CCIEN	Command complete interrupt enable 0 - Masked 1 - Enabled

38.6.1.17 Auto CMD12 Error Status (AUTOCMD12_ERR_STATUS)

When the Auto CMD12 Error Status field in the Status register is set, the host driver checks this register to identify what kind of error the Auto CMD12 / CMD 23 indicated. Auto CMD23 errors are indicated in field 04-01. This register is valid only when the Auto CMD12 Error status field is set.

The table below shows the relationship between the Auto CMGD12 CRC Error and the Auto CMD12 Command Timeout Error.

Table 237. Relationship between command CRC error and command timeout error for auto CMD12

Auto CMD12 CRC error	Auto CMD12 timeout error	Type of error
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

Changes in Auto CMD12 Error Status register can be classified in three scenarios:

- When uSDHC is going to issue an Auto CMD12
  - Set field 0 to 1 if the Auto CMD12 cannot be issued due to an error in the previous command
  - Set field 0 to 0 if the Auto CMD12 is issued

- At the end field of an Auto CMD12 response
  - — Check errors correspond to fields 1-4
  - — Set fields 1-4 corresponding to detected errors
  - — Clear fields 1-4 corresponding to detected errors
- Before reading the Auto CMD12 Error Status field 7
  - — Set field 7 to 1 if there is a command that can't be issued
  - — Clear field 7 if there is no command to issue

The timing for generating the Auto CMD12 Error and writing to the Command register are asynchronous. After that, field 7 is sampled when the driver is not writing to the Command register. So, it is suggested to read this register only when the AC12E field in Interrupt Status register is set. An Auto CMD12 Error Interrupt is generated when one of the error fields (0-4) is set to 1. The Command Not Issued By Auto CMD12 Error does not generate an interrupt.

Offset

Register	Offset
AUTOCMD12_ERR_STATUS	3Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						0	0	SMP_CLK...	EXECUTE...	0			0	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CNIBAC1...	0		AC12IE	AC12EBE	AC12CE	AC12TOE	AC12NE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-26 —	Reserved
25 —	Reserved
24 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
23 SMP_CLK_SEL	Sample clock select This field is used to select sampling clock to receive CMD and DATA. This field is functional only when TUNING_CTRL[STD_TUNING_EN] is set. Otherwise, this field is reserved. This field is controlled by re-tuning procedure and is valid after the tuning procedure is complete (AUTOCMD12_ERR_STATUS[EXECUTE_TUNING]=0). A value of '1' means that tuning has been completed successfully and a value of '0' implies the tuning has failed. Writing '1' to this field is meaningless and ignored when the tuning has failed. The tuning circuit is reset by writing '0' to this field. This field can be cleared by setting AUTOCMD12_ERR_STATUS[EXECUTE_TUNING]=1. Once the tuning circuit is reset, it takes time to complete tuning sequence. Therefore, host driver should keep this field as 1 to perform the re-tuning sequence in a short time. This field cannot be changed while the Host controller is receiving response or a read data block. 0 - Fixed clock is used to sample data 1 - Tuned clock is used to sample data
22 EXECUTE_TUNING	Execute tuning This field is used to start the tuning procedure. This field is functional only when TUNING_CTRL[STD_TUNING_EN] is set. Otherwise, this field is reserved. This field is set to start tuning procedure and automatically cleared when tuning procedure is completed. The result of tuning is indicated to AUTOCMD12_ERR_STATUS[SMP_CLK_SEL]. 0 - Tuning procedure is stopped 1 - Start tuning procedure
21-19 —	Reserved
18 —	Reserved
17-8 —	Reserved
7 CNIBAC12E	Command not issued by Auto CMD12 error Setting this field to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error (D04-D01) in this register. 0 - No error 1 - Not issued
6-5 —	Reserved
4 AC12IE	Auto CMD12 / 23 index error Occurs if the command index error occurs in response to a command.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - No error 1 - Error, the CMD index in response is not CMD12/23
3 AC12EBE	Auto CMD12 / 23 end bit error Occurs when detecting that the end field of command response is 0 which should be 1. 0 - No error 1 - End bit error generated
2 AC12CE	Auto CMD12 / 23 CRC error Occurs when detecting a CRC error in the command response. 0 - No CRC error 1 - CRC error met in Auto CMD12/23 response
1 AC12TOE	Auto CMD12 / 23 timeout error Occurs if no response is returned within 64 SDCLK cycles from the end field of the command. If this field is set to1, the other error status fields (2-4) have no meaning. 0 - No error 1 - Time out
0 AC12NE	Auto CMD12 not executed If memory multiple block data transfer is not started, due to a command error, this field is not set because it is not necessary to issue an Auto CMD12. Setting this field to 1 means uSDHC cannot issue the Auto CMD12 to stop a memory multiple block data transfer due to some error. If this field is set to 1, other error status fields (1-4) have no meaning. 0 - Executed 1 - Not executed

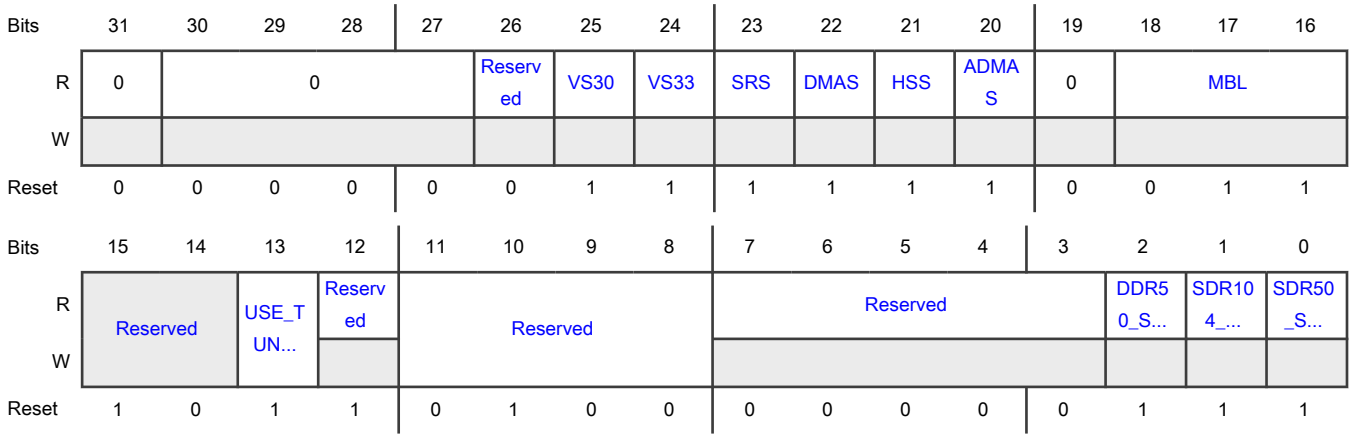
38.6.1.18 Host Controller Capabilities (HOST_CTRL_CAP)

This register provides the host driver with information specific to uSDHC implementation. The value in this register is the power-on-reset value and does not change with a software reset.

Offset

Register	Offset
HOST_CTRL_CAP	40h

Diagram



Fields

Field	Description
31 —	Reserved
30-27 —	Reserved
26 —	Reserved
25 VS30	Voltage support 3.0 V This field depends on the host system ability. 0 - 3.0 V not supported 1 - 3.0 V supported
24 VS33	Voltage support 3.3 V This field depends on the host system ability. 0 - 3.3 V not supported 1 - 3.3 V supported
23 SRS	Suspend / resume support This field indicates whether uSDHC supports Suspend / Resume functionality. If this field is 0, the Suspend and Resume mechanism, as well as the read wait, are not supported, and the host driver does not issue either Suspend or Resume commands. 0 - Not supported 1 - Supported
22	DMA support

Table continues on the next page...

Table continued from the previous page...

Field	Description
DMAS	This field indicates whether uSDHC can use the internal DMA to transfer data between system memory and the data buffer directly. 0 - DMA not supported 1 - DMA supported
21 HSS	High speed support This field indicates whether uSDHC supports High Speed mode and the host system can supply a SD clock frequency from 25 MHz to 50 MHz. 0 - High speed not supported 1 - High speed supported
20 ADMAS	ADMA support This field indicates whether uSDHC supports the ADMA feature. 0 - Advanced DMA not supported 1 - Advanced DMA supported
19 —	Reserved
18-16 MBL	Max block length This field indicates the maximum block size that the host driver can read and write to the buffer in uSDHC. The buffer transfers block size without wait cycles. 000 - 512 bytes 001 - 1024 bytes 010 - 2048 bytes 011 - 4096 bytes
15-14 —	Reserved
13 USE_TUNING_SDR50	Use Tuning for SDR50 This field is used to enable tuning for SDR50 mode 0 - SDR50 does not support tuning 1 - SDR50 supports tuning
12 —	Reserved
11-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
7-3 —	Reserved
2 DDR50_SUPPORT	DDR50 support This field indicates support of DDR50 mode.
1 SDR104_SUPPORT	SDR104 support This field indicates support of SDR104 mode.
0 SDR50_SUPPORT	SDR50 support This field indicates support of SDR50 mode.

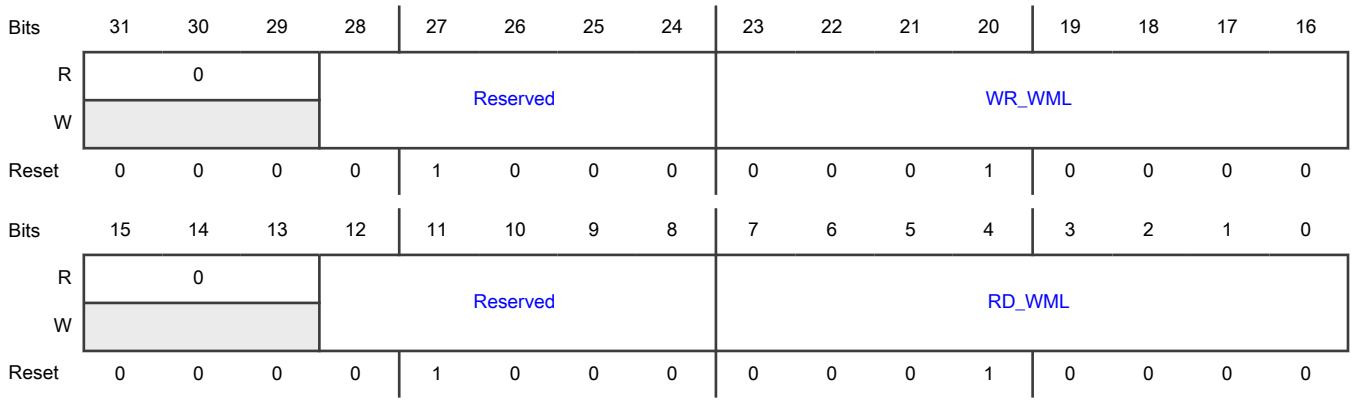
38.6.1.19 Watermark Level (WTMK_LVL)

This register indicates configurability of write and read watermark levels (FIFO threshold). Their value can range from 1 to 128 words.

Offset

Register	Offset
WTMK_LVL	44h

Diagram



Fields

Field	Description
31-29 —	Reserved
28-24 —	Reserved
23-16 WR_WML	Write watermark level This field indicates the number of words used as the watermark level (FIFO threshold) in a DMA write operation. Also, the number of words as a sequence of write bursts in back-to-back mode. The maximum legal value for the write watermark level is 128.When using internal DMA, WR_WML can be set any value between 1 and 128. To get max AXI throughput, we recommend to use integer multiple of 16.
15-13 —	Reserved
12-8 —	Reserved
7-0 RD_WML	Read watermark level This field indicates the number of words used as the watermark level (FIFO threshold) in a DMA read operation. Also, the number of words as a sequence of read bursts in back-to-back mode. The maximum legal value for the read water mark level is 128.When using internal DMA mode, RD_WML must be set to the value less than 16. To get max AXI throughput, recommend to use 16.

38.6.1.20 Mixer Control (MIX_CTRL)

This register is used to DMA and data transfer. To prevent data loss, the software should check if data transfer is active before writing this register. These fields are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

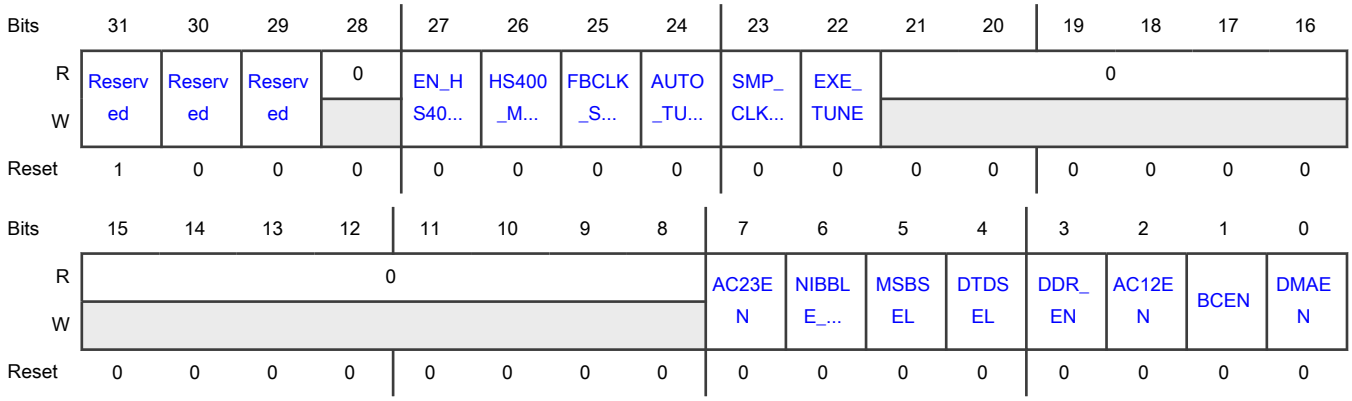
Table 238. Transfer type register setting for various transfer types

Multi/single block select	Block count enable	Block count	Function
0	Do not care	Do not care	Single transfer
1	0	Do not care	Infinite transfer
1	1	Positive number	Multiple transfer
1	1	Zero	No data transfer

Offset

Register	Offset
MIX_CTRL	48h

Diagram



Fields

Field	Description
31 —	Reserved Always write as 1.
30 —	Reserved Always write as 0.
29 —	Reserved Always write as 0.
28 —	Reserved
27 EN_HS400_MODE	Enable enhance HS400 mode Enhance HS400 enable
26 HS400_MODE	Enable HS400 mode HS400 Enable
25 FBCLK_SEL	Feedback clock source selection (Only used for SD3.0, SDR104 mode and eMMC HS200 mode) 0 - Feedback clock comes from the loopback CLK 1 - Feedback clock comes from the ipp_card_clk_out
24 AUTO_TUNE_EN	Auto tuning enable (Only used for SD3.0, SDR104 mode and eMMC HS200 mode) 0 - Disable auto tuning 1 - Enable auto tuning
23 SMP_CLK_SEL	Clock selection When TUNING_CTRL[STD_TUNING_EN] is 0, this field is used to select Tuned clock or Fixed clock to sample data / cmd (Only used for SD3.0, SDR104 mode and eMMC HS200 mode)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Fixed clock is used to sample data / cmd 1 - Tuned clock is used to sample data / cmd
22 EXE_TUNE	Execute tuning: (Only used for SD3.0, SDR104 mode and eMMC HS200 mode) When TUNING_CTRL[STD_TUNING_EN] is 0, this field is set to 1 to indicate the host driver is starting tuning procedure. Tuning procedure is stopped by writing 0. 0 - Not tuned or tuning completed 1 - Execute tuning
21-8 —	Reserved
7 AC23EN	Auto CMD23 enable This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. When this field is set to 1, the host controller issues a CMD23 automatically before issuing a command specified in the Command Register.
6 NIBBLE_POS	Nibble position indication This field indicates the nibble position in the DDR 4-bit mode. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. 0- the sequence is 'odd high nibble -> even high nibble -> odd low nibble -> even low nibble'; 1- the sequence is 'odd high nibble -> odd low nibble -> even high nibble -> even low nibble'.
5 MSBSEL	Multi / Single block select This field enables multiple block DATA line data transfers. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. For any other commands, this field can be set to 0. If this field is 0, it is not necessary to set the Block Count register. See Command Transfer Type (CMD_XFR_TYP). 0 - Single block 1 - Multiple blocks
4 DTDSEL	Data transfer direction select This field defines the direction of DATA line data transfers. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. The field is set to 1 by the host driver to transfer data from the SD card to uSDHC and is set to 0 for all other commands. 0 - Write (Host to card) 1 - Read (Card to host)
3 DDR_EN	Dual data rate mode selection This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only.
2	Auto CMD12 enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
AC12EN	Multiple block transfers for memory require a CMD12 to stop the transaction. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. When this field is set to 1, uSDHC issues a CMD12 automatically when the last block transfer has completed. The host driver is not set this field to issue commands that do not require CMD12 to stop a multiple block data transfer. In particular, secure commands defined in File Security Specification (see reference list) do not require CMD12. In single block transfer, uSDHC ignores this field no matter it is set or not. 0 - Disable 1 - Enable
1 BCEN	Block count enable This field is used to enable the Block Count register, which is only relevant for multiple block transfers. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. When this field is 0, the internal counter for block is disabled, which is useful in executing an infinite transfer. 0 - Disable 1 - Enable
0 DMAEN	DMA enable This field enables DMA functionality. This field is read/write when VEND_SPEC[CMD_BYTE_EN] is disabled; otherwise, this field is read-only. If this field is set to 1, a DMA operation begins when the host driver sets the DPSEL field of this register. Whether the simple DMA or the advanced DMA is active depends on the DMA Select field of the Protocol Control register. 0 - Disable 1 - Enable

38.6.1.21 Force Event (FORCE_EVENT)

This register is not a physically implemented register. Rather, it is an address at which the Interrupt Status register can be written if the corresponding field of the Interrupt Status Enable Register is set. Writing 1 to this register sets the corresponding field of Interrupt Status register, while writing 0 to it has no effect. A read from this register always results in 0's. To change the corresponding status fields in the Interrupt Status register, make sure to set IPGEN field in System Control Register so that peripheral clock is always active.

Forcing a card interrupt generates a short pulse on the DATA1 line, and the driver might treat this interrupt as a normal interrupt. The interrupt service routine might skip polling the card interrupt factor as the interrupt is self cleared.

Offset

Register	Offset
FORCE_EVENT	50h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
W	FEVTCINT			FEVTDMAE		FEVTTNE		FEVTAC1...		FEVTDEBE	FEVTDCE	FEVTDTOE	FEVTCIE	FEVTCEBE	FEVTCCE	FEVTCTOE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0		0	0	0	0	0
W									FEVTCNI...			FEVTAC1...	FEVTAC1...	FEVTAC1...	FEVTAC1...	FEVTAC1...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 FEVTCINT	Force event card interrupt Writing 1 to this field generates a short low-level pulse on the internal DATA1 line, as if a self-clearing interrupt was received from the external card. If enabled, the CINT field is set and the interrupt service routine might treat this interrupt as a normal interrupt from the external card.
30-29 —	Reserved
28 FEVTDMAE	Force event DMA error Forces the DMAE field of Interrupt Status register to be set
27 —	Reserved
26 FEVTTNE	Force tuning error Forces the TNE field of Interrupt Status register to be set
25 —	Reserved
24 FEVTAC12E	Force event Auto Command 12 error Forces the AC12E field of Interrupt Status register to be set
23 —	Reserved
22 FEVTDEBE	Force event data end bit error Forces the DEBE field of Interrupt Status register to be set

Table continues on the next page...

Table continued from the previous page...

Field	Description
21 FEVTDCE	Force event data CRC error Forces the DCE field of Interrupt Status register to be set
20 FEVTDTOE	Force event data time out error Force the DTOE field of Interrupt Status register to be set
19 FEVTCIE	Force event command index error Forces the CCE field of Interrupt Status register to be set
18 FEVTCEBE	Force event command end bit error Forces the CEBE field of Interrupt Status register to be set
17 FEVTCCE	Force event command CRC error Forces the CCE field of Interrupt Status register to be set
16 FEVTCTOE	Force event command time out error Forces the CTOE field of Interrupt Status register to be set
15-8 —	Reserved
7 FEVTCNIBAC12E	Force event command not executed by Auto Command 12 error Forces the CNIBAC12E field in the Auto Command12 Error Status register to be set
6-5 —	Reserved
4 FEVTAC12IE	Force event Auto Command 12 index error Forces the AC12IE field in the Auto Command12 Error Status register to be set
3 FEVTAC12EBE	Force event Auto Command 12 end bit error Forces the AC12EBE field in the Auto Command12 Error Status register to be set
2 FEVTAC12CE	Force event auto command 12 CRC error Forces the AC12CE field in the Auto Command12 Error Status register to be set
1 FEVTAC12TOE	Force event auto command 12 time out error Forces the AC12TOE field in the Auto Command12 Error Status register to be set
0 FEVTAC12NE	Force event auto command 12 not executed Forces the AC12NE field in the Auto Command12 Error Status register to be set

38.6.1.22 ADMA Error Status (ADMA_ERR_STATUS)

When an ADMA Error Interrupt has occurred, the ADMA error sates field in this register holds the ADMA state and the ADMA System Address register holds the address around the error descriptor.

For recovering from this error, the host driver requires the ADMA state to identify the error descriptor address as follows:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address.
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address.
- ST_CADR: This state is never set because it only increments the descriptor pointer and does not generate an ADMA error.
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address.

In case of a write operation, the host driver should use the ACMD22 to get the number of the written block, rather than using this information, because unwritten data might exist in the host controller.

The host controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid=0) in the ST_FDS state. The host driver can distinguish this error by reading the Valid field of the error descriptor.

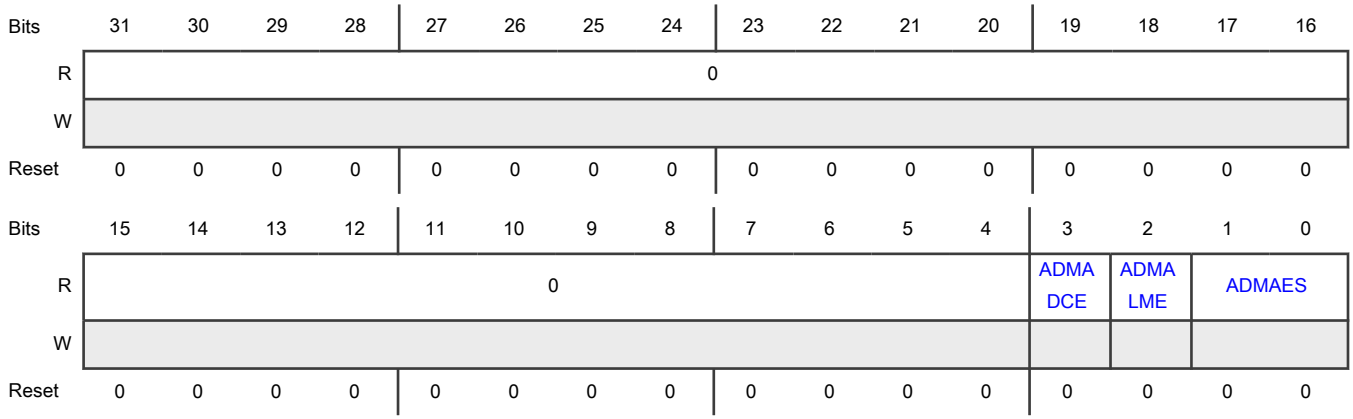
Table 239. ADMA error state coding

D01-D00	ADMA error state (when error has occurred)	Contents of ADMA System Address register
00	ST_STOP (Stop DMA)	Holds the address of the next executable descriptor command
01	ST_FDS (Fetch descriptor)	Holds the valid descriptor address
10	ST_CADR (Change address)	No ADMA error is generated
11	ST_TFR (Transfer data)	Holds the address of the next executable descriptor command

Offset

Register	Offset
ADMA_ERR_STATUS	54h

Diagram



Fields

Field	Description
31-4 —	Reserved
3 ADMADCE	ADMA descriptor error This error occurs when invalid descriptor fetched by ADMA. 0 - No error 1 - Error
2 ADMALME	ADMA length mismatch error This error occurs in the following two cases: • While the block count enable is being set, the total data length specified by the descriptor table is different from that specified by the block count and block length. • Total data length cannot be divided by the block length. 0 - No error 1 - Error
1-0 ADMAES	ADMA error state (when ADMA error is occurred) This field indicates the state of the ADMA when an error has occurred during an ADMA data transfer. See ADMA Error Status (ADMA_ERR_STATUS) for more details.

38.6.1.23 ADMA System Address (ADMA_SYS_ADDR)

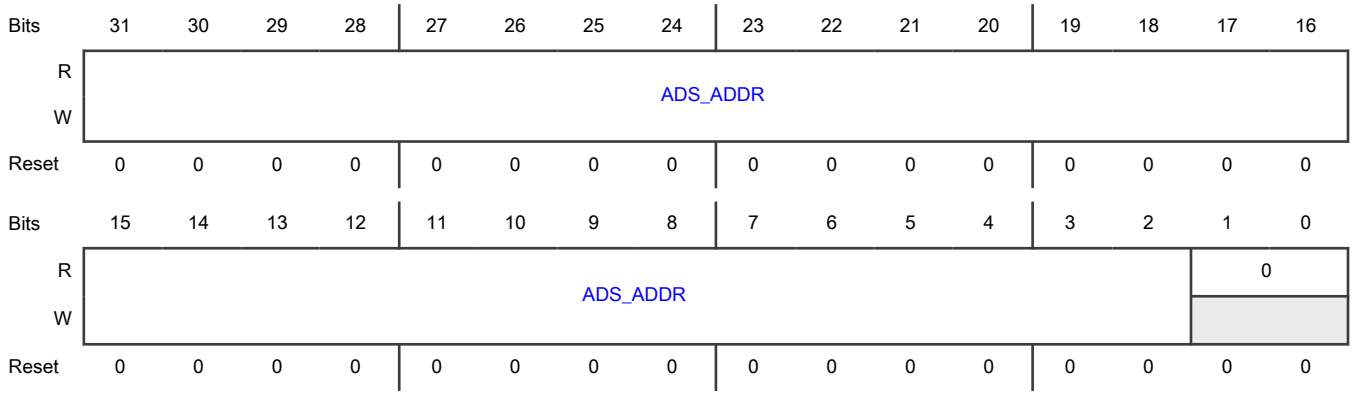
This register holds the word address of the executing command in the Descriptor table. At the start of ADMA, the host driver sets the start address of the Descriptor table. The ADMA engine increments this register address whenever fetching a Descriptor command. When the ADMA is stopped at the Block Gap, this register indicates the address of the next executable Descriptor command. When the ADMA Error Interrupt is generated, this register holds the valid Descriptor address depending on the ADMA state. The lower 2 bits of this register is tied to '0' so the ADMA address is always word aligned.

Because this register supports dynamic address reflecting, when TC field is set, it automatically alters the value of internal address counter, so the software cannot change this register when TC field is set. Such restriction is also listed in Software restrictions.

Offset

Register	Offset
ADMA_SYS_ADDR	58h

Diagram



Fields

Field	Description
31-2 ADS_ADDR	ADMA system address This field contains the physical system memory address used for ADMA transfers.
1-0 —	Reserved

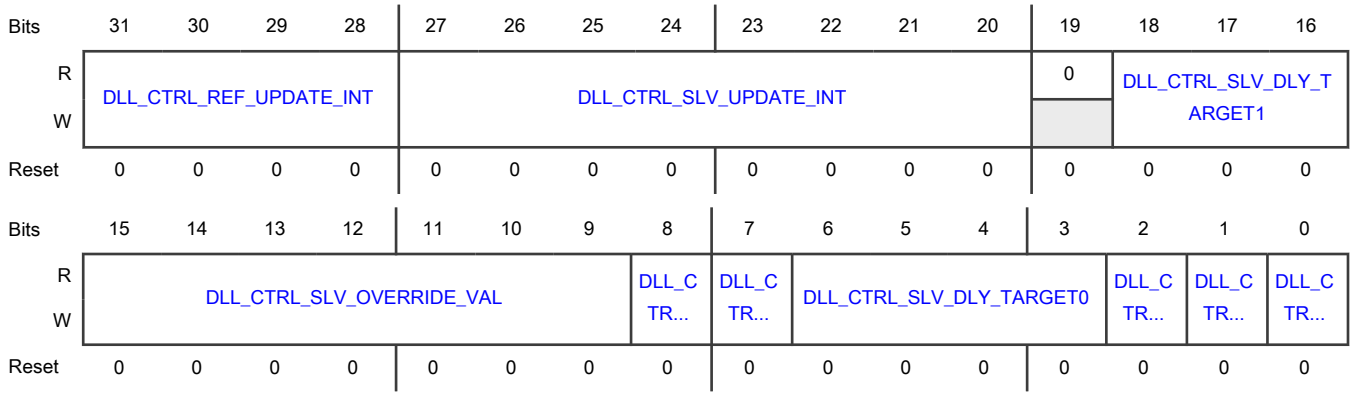
38.6.1.24 DLL (Delay Line) Control (DLL_CTRL)

This register contains control fields for DLL.

Offset

Register	Offset
DLL_CTRL	60h

Diagram



Fields

Field	Description
31-28 DLL_CTRL_REF_UPDATE_INT	DLL control loop update interval The interval cycle is (2 + REF_UPDATE_INT) * REF_CLOCK. By default, the DLL control loop updates every two REF_CLOCK cycles. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that might effect the delay (such as voltage and temperature)
27-20 DLL_CTRL_SLV_UPDATE_INT	Target delay line update interval If default 0 is used, it means 256 cycles of REF_CLOCK. A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the target-delay line using the SLV_FORCE_UPDATE register. Note that the target delay line also updates automatically when the reference DLL transitions to a locked state (from an un-locked state).
19 —	Reserved
18-16 DLL_CTRL_SLV_DLY_TARGET1	DLL target delay target1 See DLL_CTRL_SLV_DLY_TARGET0 below.
15-9 DLL_CTRL_SLV_OVERRIDE_VAL	DLL target override val When SLV_OVERRIDE = 1, this field is used to select 1 of 128 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 128.
8 DLL_CTRL_SLV_OVERRIDE	DLL target override Set this field to 1 to Enable manual override for target delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to be enabled using the ENABLE field. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE = 0
7 DLL_CTRL_GATE_UPDATE	DLL gate update Set this field to 1 to prevent the DLL from updating (because when clock_in exists, glitches might appear during DLL updates). This field might be used by software if such a condition occurs. Clear the bit to 0 to allow the DLL to update automatically.
6-3 DLL_CTRL_SLV_DLY_TARGET0	DLL target delay target0 The delay target for uSDHC loopback read clock can be programmed in 1/16th increments of an REF_CLOCK half-period. The delay is (DLL_CTRL_SLV_DLY_TARGET + 1)* REF_CLOCK_PERIOD[1] / 2) / 16, where DLL_CTRL_SLV_DLY_TARGET is a 7-bit value obtained by concatenating {DLL_CTRL_SLV_DLY_TARGET1, DLL_CTRL_SLV_DLY_TARGET0}. Therefore, the input read-clock can be delayed relative to the input data by (REF_CLOCK_PERIOD / 2) / 16 up to REF_CLOCK_PERIOD * 4.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE For the restrictions of delay cell implementation, the delay target must be set between REF_CLOCK_PERIOD/16 and REF_CLOCK_PERIOD*2 when REF_CLOCK is running at 200 MHz. When REF_CLOCK frequency is slower than 100 MHz, the maximum delay target might not reach REF_CLOCK_PERIOD*2.
2 DLL_CTRL_SLV_FORCE_UPD	DLL target delay line Setting this field to 1, forces the target delay line to update to the DLL calibrated value immediately. The target delay line updates automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the target-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when uSDHC is idle. This function might not work when uSDHC is working on data / cmd / response.
1 DLL_CTRL_RESET	DLL reset Setting this field to 1 force a reset on DLL. This causes the DLL to lose lock and re-calibrate to detect an REF_CLOCK half-period phase shift. This signal is used by the DLL as edge-sensitive, so to create a subsequent reset, RESET must be taken low and then asserted again.
0 DLL_CTRL_ENABLE	DLL and delay chain Set this field to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the target delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled.

1. REF_CLOCK_PERIOD is 1/REF_CLOCK.

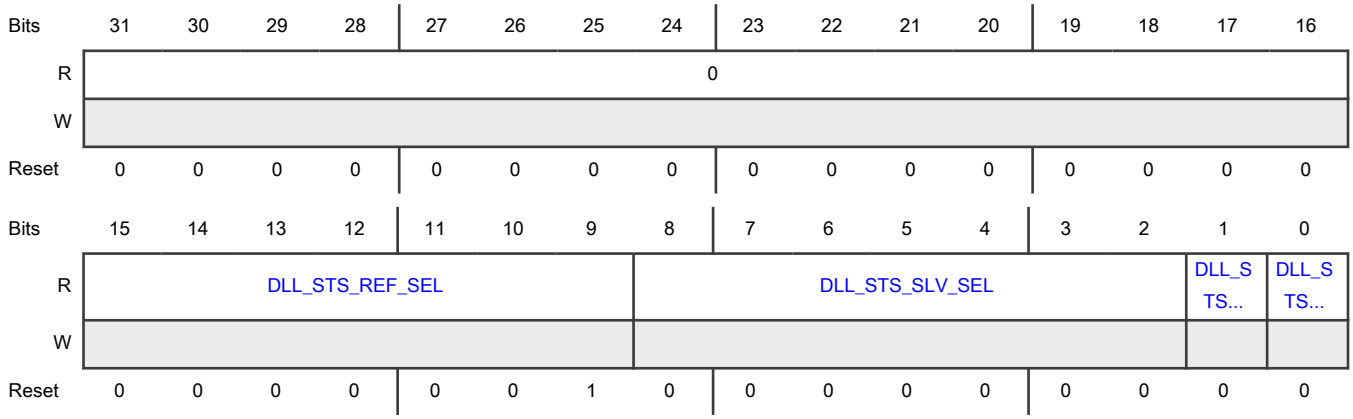
38.6.1.25 DLL Status (DLL_STATUS)

This register contains the DLL status information. All fields are read only and reads the same as the power-reset value.

Offset

Register	Offset
DLL_STATUS	64h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-9 DLL_STS_REF_SEL	Reference delay line select taps This is encoded by 7 fields for 127 taps.
8-2 DLL_STS_SLV_SEL	Target delay line select status This is the instant value generated from reference chain. Because the reference chain can only be updated when REF_CLOCK is detected, this value should be the right value to be updated when the reference is locked.
1 DLL_STS_REF_LOCK	Reference DLL lock status This signifies that the DLL has detected and locked to an REF_CLOCK half-period phase shift , allowing the target delay-line to perform programmed clock delays
0 DLL_STS_SLV_LOCK	Target delay-line lock status This signifies that a valid calibration has been set to the target-delay line and that the target-delay line is implementing the programmed delay value

38.6.1.26 CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS)

This register contains the Clock Tuning Control status information. This register is added to support SD3.0 UHS-I SDR104 mode and eMMC HS200 mode.

Offset

Register	Offset
CLK_TUNE_CTRL_STATUS	68h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PRE_ERR	TAP_SEL_PRE							TAP_SEL_OUT				TAP_SEL_POST			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NXT_ERR	DLY_CELL_SET_PRE							DLY_CELL_SET_OUT				DLY_CELL_SET_POST			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 PRE_ERR	PRE error PRE error which means the number of delay cells added on the feedback clock is too small. It is valid only when SMP_CLK_SEL of Mix control register (bit23 of 0x48) is enabled.
30-24 TAP_SEL_PRE	TAP_SEL_PRE Reflects the number of delay cells added on the feedback clock between the feedback clock and CLK_PRE. When AUTO_TUNE_EN (bit24 of 0x48) is disabled, TAP_SEL_PRE is always equal to DLY_CELL_SET_PRE. When AUTO_TUNE_EN (bit24 of 0x48) is enabled, TAP_SEL_PRE is updated automatically according to the status of the auto tuning circuit to adjust the sample clock phase.
23-20 TAP_SEL_OUT	Delay cells added on the feedback clock between CLK_PRE and CLK_OUT Reflects the number of delay cells added on the feedback clock between CLK_PRE and CLK_OUT.
19-16 TAP_SEL_POST	Delay cells added on the feedback clock between CLK_OUT and CLK_POST Reflects the number of delay cells added on the feedback clock between CLK_OUT and CLK_POST.
15 NXT_ERR	NXT error NXT error which means the number of delay cells added on the feedback clock is too large. It's valid only when SMP_CLK_SEL of Mix control register (bit23 of 0x48) is enabled.
14-8 DLY_CELL_SET_PRE	delay cells on the feedback clock between the feedback clock and CLK_PRE Set the number of delay cells on the feedback clock between the feedback clock and CLK_PRE.
7-4 DLY_CELL_SET_OUT	Delay cells on the feedback clock between CLK_PRE and CLK_OUT Set the number of delay cells on the feedback clock between CLK_PRE and CLK_OUT.

Table continues on the next page...

Table continued from the previous page...

Field	Description
3-0 DLY_CELL_SET_POST	Delay cells on the feedback clock between CLK_OUT and CLK_POST Set the number of delay cells on the feedback clock between CLK_OUT and CLK_POST.

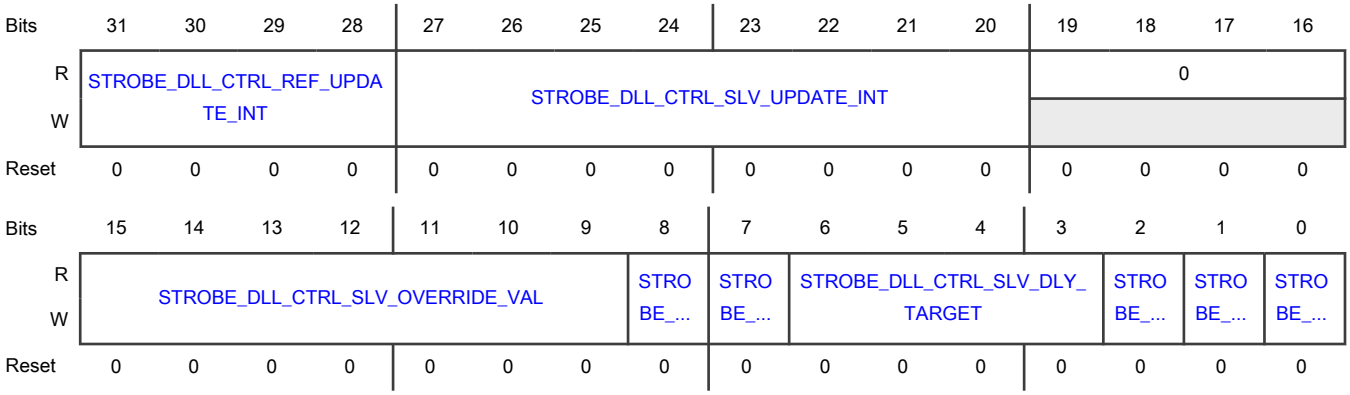
38.6.1.27 Strobe DLL control (STROBE_DLL_CTRL)

This register contains the strobe DLL control information.

Offset

Register	Offset
STROBE_DLL_CTRL	70h

Diagram



Fields

Field	Description
31-28 STROBE_DLL_CTRL_REF_UPDATE_INT	Strobe DLL control reference update interval The interval cycle is (2 + STROBE_REF_UPDATE_INT) * STROBE_REF_CLOCK. By default, the DLL control loop updates every two STROBE_REF_CLOCK cycles. **NOTE** Increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that might effect the delay (such as voltage and temperature)
27-20 STROBE_DLL_CTRL_SLV_UPDATE_INT	Strobe DLL control target update interval Target delay line update interval. If default 0 is used, it means 256 cycles of STROBE_REF_CLOCK. A value of 0x0F results in 15 cycles and so on.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE software can always cause an update of the target-delay line using the STROBE_SLV_FORCE_UPDATE register. The target delay line also updates automatically when the reference DLL transitions to a locked state (from an un-locked state).
19-16 —	This field is reserved. This read-only field is reserved and always has the value 0.
15-9 STROBE_DLL_CTRL_SLV_OVERRIDE_VAL	Strobe DLL control target Override value When STROBE_SLV_OVERRIDE = 1, this field is used to manually select one of 128 physical taps. A value of 0 selects tap 1, and a value of 0x7F selects tap 128.
8 STROBE_DLL_CTRL_SLV_OVERRIDE	Strobe DLL control target override Set this field to 1 to Enable manual override for target delay chain using STROBE_SLV_OVERRIDE_VAL; set this field to 0 to disable manual override. This feature does not require the DLL to be enabled using the ENABLE field. In fact to reduce power, if STROBE_SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE = 0.
7 STROBE_DLL_CTRL_GATE_UPDATE	Strobe DLL control gate update Set this field to 1 to prevent the DLL from updating (because when STROBE_CLOCK_IN exists, glitches might appear during DLL updates). This field can be used by software if such a condition occurs. Clear the field to 0 to allow the DLL to update automatically.
6-3 STROBE_DLL_CTRL_SLV_DLY_TARGET	Strobe DLL Control Target Delay Target The delay target for uSDHC loopback read clock can be programmed in 1/16th increments of an STROBE_REF_CLOCK half-period. The delay is ((STROBE_DLL_CTRL_SLV_DLY_TARGET + 1) * STROBE_REF_CLOCK_PERIOD[1] / 2) / 16, Therefore, the input read-clock can be delayed relative to the input data by (STROBE_REF_CLOCK_PERIOD / 2) / 16 up to (STROBE_REF_CLOCK_PERIOD *8)/16. **NOTE** For the restrictions of delay cell implementation, the minimum delay target that can be set is REF_CLOCK_PERIOD/16.
2 STROBE_DLL_CTRL_SLV_FORCE_UPD	Strobe DLL control target force updated Setting this field to 1, forces the target delay line to update to the DLL calibrated value immediately. The target delay line should automatically update the STROBE_SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the target-line update can only occur if STROBE_SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when uSDHC is idle. This function might not work when uSDHC is working on data / cmd / response.
1	Strobe DLL control reset

Table continues on the next page...

Table continued from the previous page...

Field	Description
STROBE_DLL_CTRL_RESET	Setting this field to 1 to force a reset on DLL. This causes the DLL to lose lock and re-calibrate to detect an REF_CLOCK half-period phase shift. This signal is used by the DLL as edge-sensitive, to create a subsequent reset, RESET must be taken low and then asserted again.
0 STROBE_DLL_CTRL_ENABLE	Strobe DLL control enable Set this field to 1 to enable the DLL and delay chain; otherwise, set to 0 to bypasses DLL. **NOTE** Using the target delay line override feature with STROBE_SLV_OVERRIDE and STROBE_SLV_OVERRIDE VAL, the DLL does not need to be enabled.

1. which is 1/STROBE_REF_CLOCK.

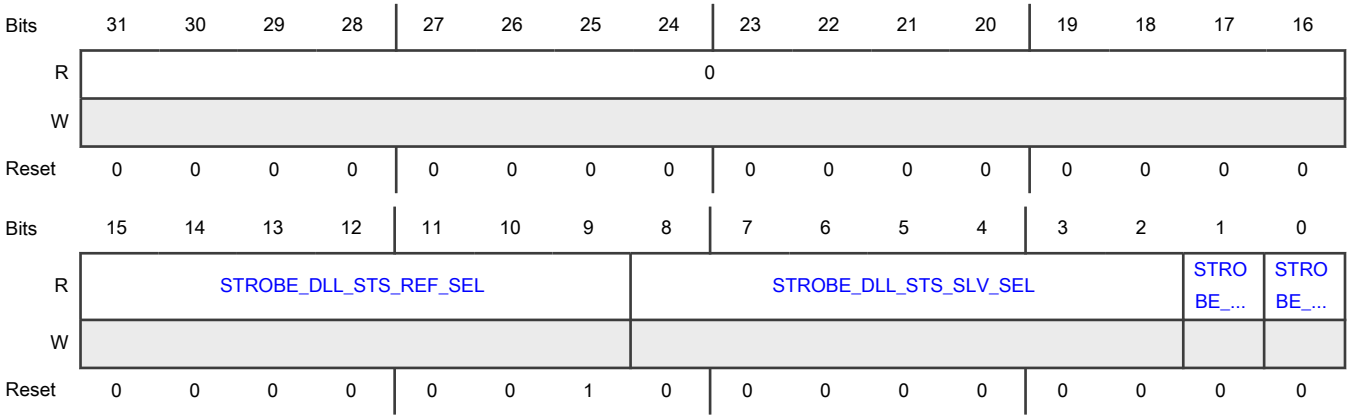
38.6.1.28 Strobe DLL status (STROBE_DLL_STATUS)

This register contains the strobe DLL status information. All fields are read only and read the same as the power-reset value.

Offset

Register	Offset
STROBE_DLL_STATUS	74h

Diagram



Fields

Field	Description
31-16 —	This field is reserved. This read-only field is reserved and always has the value 0.
15-9	Strobe DLL status reference select

Table continues on the next page...

Table continued from the previous page...

Field	Description
STROBE_DLL_STS_REF_SEL	Reference delay line select taps. This is encoded by 7 fields for 127 taps.
8-2 STROBE_DLL_STS_SLV_SEL	Strobe DLL status target select Target delay line select status. This is the instant value generated from reference chain. Because the reference chain can only be updated when STROBE_REF_CLOCK is detected, this value can be updated with the right value when the reference is locked.
1 STROBE_DLL_STS_REF_LOCK	Strobe DLL status reference lock This signifies that the DLL has detected and locked to a half-phase REF_CLOCK shift, it allows the target delay-line to perform programmed clock delays.
0 STROBE_DLL_STS_SLV_LOCK	Strobe DLL status target lock Target delay-line lock status. This signifies that a valid calibration has been set to the target-delay line, and the target-delay line is implementing the programmed delay value.

38.6.1.29 Vendor Specific Register (VEND_SPEC)

This register contains the vendor specific control/status register.

Offset

Register	Offset
VEND_SPEC	C0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMD_BYT...	Reserved	Reserved	Reserved	Reserved				Reserved							
W																
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CRC_CHK...	Reserved			Reserved	Reserved	FRC_SDC...	Reserved	Reserved	Reserved	Reserved	AC12_WR...	Reserved	VSELECT	Reserved	
W																
Reset	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	1

Fields

Field	Description
31 CMD_BYTE_EN	Register byte access for CMD_XFR_TYP This field controls the register byte access for Command Transfer Type (CMD_XFR_TYP). If this field is enabled, the register can be configured through byte write enable. IPS_bus can write only one byte once for one write operation. 0 - Disable. MIX_CTRL[7:0] is read/write and CMD_XFR_TYP[7:0] is read-only. 1 - Enable. MIX_CTRL[7:0] is read-only and CMD_XFR_TYP[7:0] is read/write.
30 —	Reserved Always write as 0.
29 —	Reserved Always write as 1
28 —	Reserved Always write as 0.
27-24 —	Reserved Always write as 4'b0000.
23-16 —	Reserved Always write as 8'h00.
15 CRC_CHK_DIS	CRC Check Disable 0 - Check CRC16 for every read data packet and check CRC fields for every write data packet 1 - Ignore CRC16 check for every read data packet and ignore CRC fields check for every write data packet
14-11 —	Reserved Reserved
10 —	Reserved Always write as 0.
9 —	Reserved Always write as 0.
8 FRC_SDCLK_ON	Force CLK Force CLK output active Do not set this bit to 1 unless it is necessary. Also, make sure that this bit is cleared when uSDHC's clock is about to be changed (frequency change, clock source change, or delay chain tuning). 0 - CLK active or inactive is fully controlled by the hardware. 1 - Force CLK active

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 AC12_WR_CHKBUSY_EN	Check busy enable Check busy enable after auto CMD12 for write data packet 0 - Do not check busy after auto CMD12 for write data packet 1 - Check busy after auto CMD12 for write data packet
2 —	Reserved
1 VSELECT	Voltage selection Change the value of output signal VSELECT to control the voltage on pads for external card. There must be a control circuit out of uSDHC to change the voltage on pads. 0 - Change the voltage to high voltage range, around 3.0 V 1 - Change the voltage to low voltage range
0 —	Reserved Always write as 0.

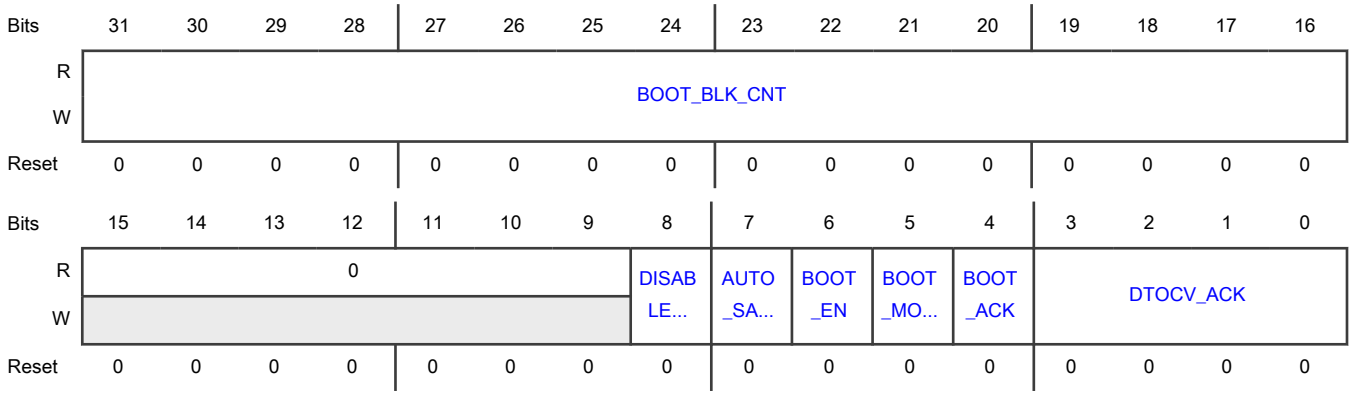
38.6.1.30 eMMC Boot (MMC_BOOT)

This register contains the eMMC Fast Boot control register.

Offset

Register	Offset
MMC_BOOT	C4h

Diagram



Fields

Field	Description
31-16 BOOT_BLK_CNT	Stop At Block Gap value of automatic mode The value defines the Stop At Block Gap value of automatic mode. When received, card block cnt is equal to (BLK_CNT - BOOT_BLK_CNT) and AUTO_SABG_EN is 1, then Stop At Block Gap. Here, BLK_CNT is defined in the Block Attributes Register, field 31 - 16 of 0x04.
15-9 —	Reserved
8 DISABLE_TIME_OUT	Time out **NOTE** When this field is set, there is no timeout check no matter whether BOOT_EN is set or not. 0 - Enable time out 1 - Disable time out
7 AUTO_SABG_EN	Auto stop at block gap During boot, enable auto stop at block gap function. This function is triggered, and host stops at block gap when received card block cnt is equal to (BLK_CNT - BOOT_BLK_CNT).
6 BOOT_EN	Boot enable Boot mode enable 0 - Fast boot disable 1 - Fast boot enable
5 BOOT_MODE	Boot mode Boot mode select 0 - Normal boot 1 - Alternative boot

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 BOOT_ACK	BOOT ACK Boot ACK mode select 0 - No ack 1 - Ack
3-0 DTOCV_ACK	DTOCV_ACK Boot ACK time out counter value 0000 - SDCLK x 2^32 0001 - SDCLK x 2^33 0010 - SDCLK x 2^18 0011 - SDCLK x 2^19 0100 - SDCLK x 2^20 0101 - SDCLK x 2^21 0110 - SDCLK x 2^22 0111 - SDCLK x 2^23 1110 - SDCLK x 2^30 1111 - SDCLK x 2^31

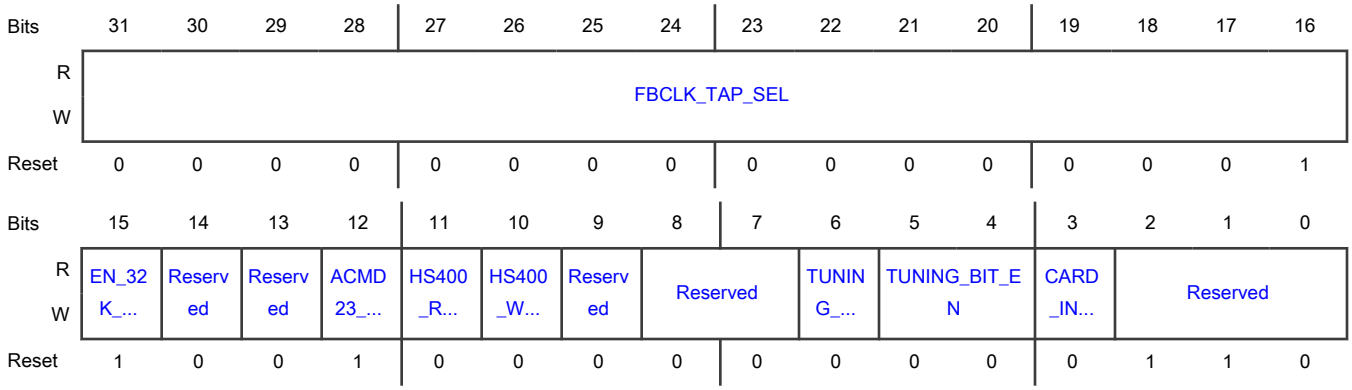
38.6.1.31 Vendor Specific 2 Register (VEND_SPEC2)

This register contains the vendor specific control 2 register.

Offset

Register	Offset
VEND_SPEC2	C8h

Diagram



Fields

Field	Description
31-16 FBCLK_TAP_SEL	Enable extra delay on internal feedback clock It is a hot code. The default value is 16'h0001 that means the length of feedback delay chain is 1. The value of 16'h8000 means the length of feedback delay chain is 16.
15 EN_32K_CLK	Select the clock source for host card detection. It can use low power clock for card detection, by setting this bit to 1. 0 - Use the peripheral clock (ipg_clk) for card detection. 1 - Use the low power clock (ipg_clk_lp) for card detection.
14 —	Reserved
13 —	Reserved
12 ACMD23_ARGU2_EN	Argument2 register enable for ACMD23 0 - Disable 1 - Argument2 register enable for ACMD23 sharing with SDMA system address register. Default is enabled.
11 HS400_RD_CLK_STOP_EN	HS400 read clock stop enable Only stop clock at read block gap.
10 HS400_WR_CLK_STOP_EN	HS400 write clock stop enable Only stop clock at write block gap.
9 —	Reserved
8-7 —	Reserved
6 TUNING_CMD_EN	Tuning command enable Enable the auto tuning circuit to check the CMD line. 0 - Auto tuning circuit does not check the CMD line. 1 - Auto tuning circuit checks the CMD line.
5-4 TUNING_BIT_EN	Tuning bit enable Enable the auto tuning circuit to check the DATA[7:0]. The value option is as the list below. 00 - Enable Tuning circuit for DATA[3:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	01 - Enable Tuning circuit for DATA[7:0] 10 - Enable Tuning circuit for DATA[0] 11 - Invalid
3 CARD_INT_D3_TEST	Card interrupt detection test This field is used only for debugging. 0 - Check the card interrupt only when DATA3 is high. 1 - Check the card interrupt by ignoring the status of DATA3.
2-0 —	Reserved

38.6.1.32 Tuning Control (TUNING_CTRL)

The register contains configuration of tuning circuit.

Offset

Register	Offset
TUNING_CTRL	CCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved							STD_TUN...	Reserved	TUNING_WINDOW			Reserved	TUNING_STEP		
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TUNING_COUNTER								DIS_CMD...	TUNING_START_TAP						
W																
Reset	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-25 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
24 STD_TUNING_EN	Standard tuning circuit and procedure enable This field is used to enable standard tuning circuit and procedure.
23 —	Reserved
22-20 TUNING_WINDOW	Data window Select data window value for auto tuning
19 —	Reserved
18-16 TUNING_STEP	TUNING_STEP The increasing delay cell steps in tuning procedure.
15-8 TUNING_COUNTER	Tuning counter The MAX repeat CMD19 times in tuning procedure.
7 DIS_CMD_CHK_FOR_STD_TUNING	Disable command check for standard tuning Writing 1 to this field disables command check (command CRC, CMD end bit error, and CMD index error or CMD timeout error) for standard tuning flow after each tuning command is sent.
6-0 TUNING_START_TAP	Tuning start The start delay cell point when send first CMD19 in tuning procedure.

38.6.1.33 Command Queuing Version (CQVER)

This register provides information about the version of the eMMC CQ standard that is implemented by CQE, in BCD format. The current version is 5.1.

Offset

Register	Offset
CQVER	100h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				MAJOR_VN				MINOR_VN				VERSION_SUFFIX			
W																
Reset	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0

Fields

Field	Description
31-12 —	Reserved
11-8 MAJOR_VN	eMMC major version number Digit left of decimal point, in BCD format
7-4 MINOR_VN	eMMC minor version number Digit right of decimal point, in BCD format
3-0 VERSION_SUFFIX	eMMC version suffix The second digit right of decimal point, in BCD format

38.6.1.34 Command Queuing Capabilities (CQCAP)

This register indicates hardware capabilities.

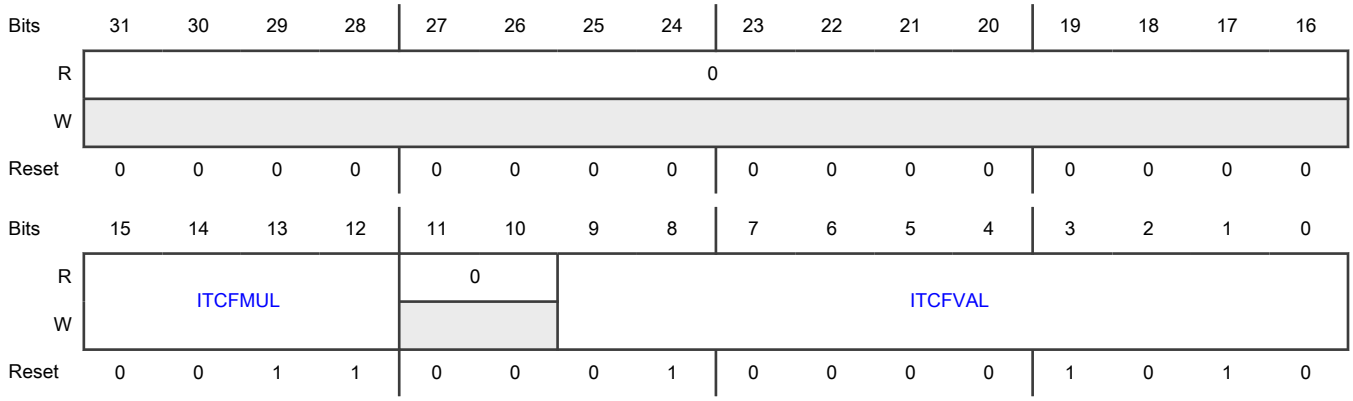
NOTE

This register must not be configured and must be used as a read-only register. Writing to the read-write fields of this register does not affect any functionality. Reset values reflect the correct capability information.

Offset

Register	Offset
CQCAP	104h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-12 ITCFMUL	Internal timer clock frequency multiplier ITCFMUL and ITCFVAL indicate the frequency of the clock used for interrupt coalescing timer and for determining the SQS polling period. See ITCFVAL definition for details. 0001 - 0.001 MHz 0010 - 0.01 MHz 0011 - 0.1 MHz 0100 - 1 MHz 0101 - 10 MHz 0110-1001 - Reserved
11-10 —	Reserved
9-0 ITCFVAL	Internal timer clock frequency value ITCFMUL and ITCFVAL indicate the frequency of the clock used for interrupt coalescing timer and for determining the polling period when using periodic SEND_QUEUE_ STATUS (CMD13) polling. The clock frequency is calculated as ITCFVAL* ITCFMUL. For example, to encode 19.2 MHz, ITCFVAL is C0h (= 192 decimal) and ITCFMUL is 3h (0.1 MHz): 192 * 0.1 MHz = 19.2 MHz.

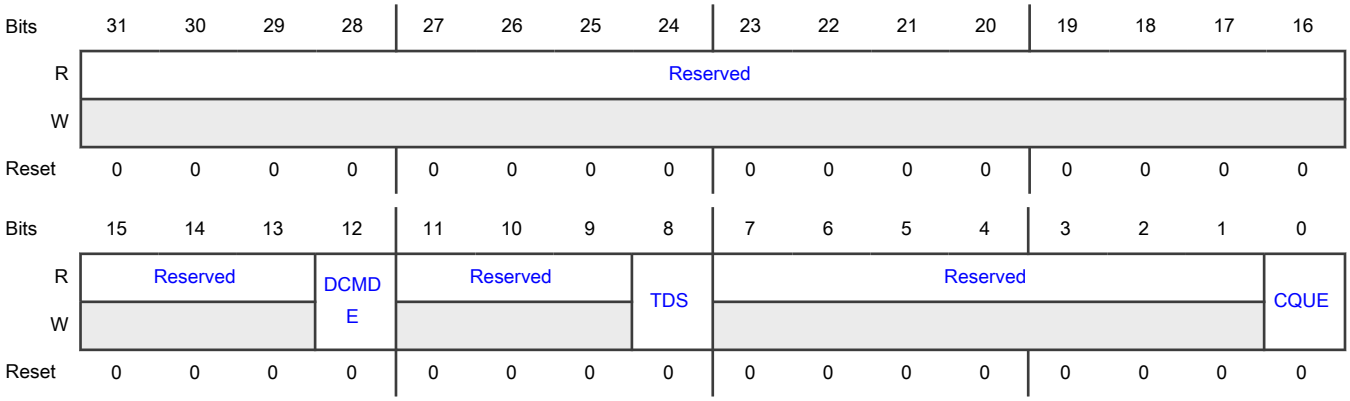
38.6.1.35 Command Queuing Configuration (CQCFG)

This register controls CQE behavior affecting the general operation of command queuing module or operation of multiple tasks in the same time.

Offset

Register	Offset
CQCFG	108h

Diagram



Fields

Field	Description
31-13 —	Reserved
12 DCMDE	Direct command (DCMD) enable This bit indicates to the hardware whether the Task Descriptor in slot #31 of the TDL is a Data Transfer Task Descriptor, or a Direct Command Task Descriptor. CQE uses this bit when a task is issued in slot #31, to determine how to decode the Task Descriptor. 0 - Task descriptor in slot #31 is a Data Transfer Task Descriptor 1 - Task descriptor in slot #31 is a DCMD Task Descriptor
11-9 —	Reserved
8 TDS	Task descriptor size This bit indicates whether the task descriptor size is 128 bits or 64 bits as detailed in Data Structures section. This bit can only be configured when Command Queuing Enable bit is ‘0’ (command queuing is disabled) 0 - Task descriptor size is 64 bits 1 - Task descriptor size is 128 bits
7-1 —	Reserved
0	Command queuing enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
CQUE	Software shall write '1' this bit when in order to enable command queuing mode (for example, enable CQE). When this bit is 0, CQE is disabled and software controls the eMMC bus using the legacy eMMC host controller. Before software writes '1' to this bit, software shall verify that the eMMC host controller is in idle state and there are no commands or data transfers ongoing. When software wants to exit command queuing mode, it shall clear all previous tasks if such exist before setting this bit to 0.

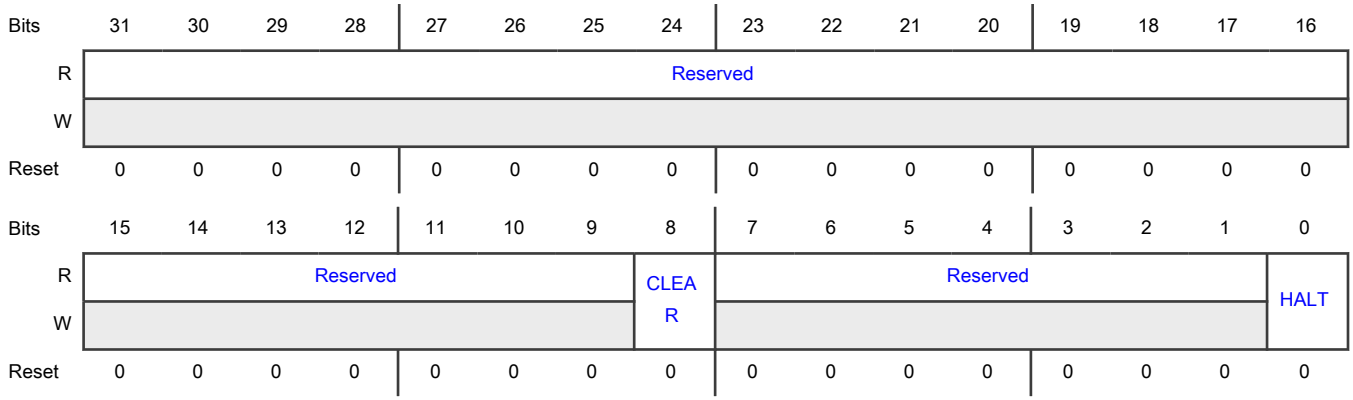
38.6.1.36 Command Queuing Control (CQCTL)

This register controls CQE behavior affecting the general operation of command queuing module or operation of multiple tasks in the same time.

Offset

Register	Offset
CQCTL	10Ch

Diagram



Fields

Field	Description
31-9 —	Reserved
8 CLEAR	Clear all tasks Software shall write '1' this bit when it wants to clear all the tasks sent to the chip. This bit can only be written when CQE is in halt state (for example, Halt bit is 1).

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When software writes 1, the value of the register is updated to '1', and CQE shall reset CQTDBR register and all other context information for all unfinished tasks. Then CQE will clear this bit. Software should poll on this bit until it is set to back 0 and may then resume normal operation, by clearing the Halt bit. CQE does not communicate to the chip that the tasks were cleared. It is software's responsibility to order the chip to discard the tasks in its queue using CMDQ_TASK_MGMT command. Writing '0' to this register shall have no effect.
7-1 —	Reserved
0 HALT	Halt Host software shall write '1' to the bit when it wants to acquire software control over the eMMC bus and disable CQE from issuing commands on the bus. For example, issuing a Discard Task command (CMDQ_TASK_MGMT) When software writes '1', CQE shall complete the ongoing task if such a task is in progress. Once the task is completed and CQE is in idle state, CQE shall not issue new commands and shall indicate so to software by setting this bit to 1. Software may poll on this bit until it is set to 1, and may only then send commands on the eMMC bus. In order to exit halt state (for example, resume CQE activity), software shall clear this bit (write '0'). Writing '0' when the value is already '0' shall have no effect.

38.6.1.37 Command Queuing Interrupt Status (CQIS)

This register indicates pending interrupts that require service. Each bit in this registers is asserted in response a specific event, only if the respective bit is set in CQISTE register.

Offset

Register	Offset
CQIS	110h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												TCL	RED	TCC	HAC
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved
3 TCL	Task cleared This status bit is asserted (if CQISTE[TCL_STE]=1) when a task clear operation is completed by CQE. The completed task clear operation is either an individual task clear (CQTCLR) or clearing of all tasks (CQCTL).
2 RED	Response error detected interrupt This status bit is asserted (if CQISTE[RED_STE]=1) when a response is received with an error bit set in the device status field. The contents of the device status field are listed in 6.13. Software uses CQRMEM register to configure which device status bit fields may trigger an interrupt, and which are masked.
1 TCC	Task complete interrupt This status bit is asserted (if CQISTE[TCC_STE]=1) when at least one of the following two conditions are met: • A task is completed and the INT bit is set in its Task Descriptor • Interrupt caused by Interrupt Coalescing logic (see 0)
0 HAC	Halt complete interrupt This status bit is asserted (if CQISTE[HAC_STE]=1) when halt bit in CQCTL register transitions from 0 to 1 indicating that host controller has completed its current ongoing task and has entered halt state.

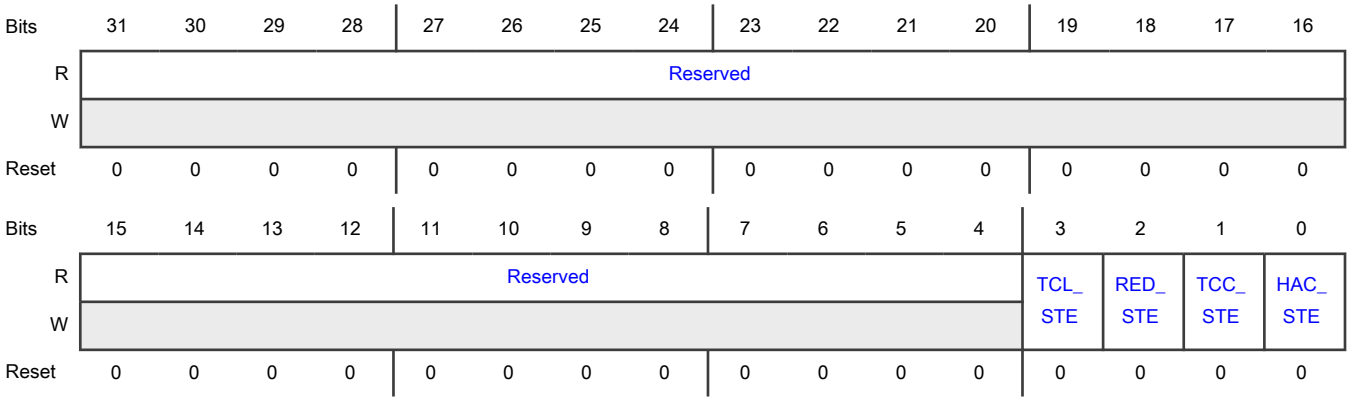
38.6.1.38 Command Queuing Interrupt Status Enable (CQISTE)

This register enables and disables the reporting of the corresponding interrupt to host software in CQIS register. When a bit is set ('1') and the corresponding interrupt condition is active, then the bit is CQIS is asserted. Interrupt sources that are disabled (‘0’) are not indicated in the CQIS register. This register is bit-index matched to CQIS register.

Offset

Register	Offset
CQISTE	114h

Diagram



Fields

Field	Description
31-4 —	Reserved
3 TCL_STE	Task cleared status enable Related to CQIS[TCL] 0 - CQIS[TCL] is disabled 1 - CQIS[TCL] is set when its interrupt condition is active
2 RED_STE	Response error detected status enable Related to CQIS[RED] 0 - CQIS[RED]is disabled 1 - CQIS[RED] is set when its interrupt condition is active
1 TCC_STE	Task complete status enable Related to CQIS[TCC] 0 - CQIS[TCC] is disabled 1 - CQIS[TCC] is set when its interrupt condition is active
0 HAC_STE	Halt complete status enable Related to CQIS[HAC] 0 - CQIS[HAC] is disabled 1 - CQIS[HAC] is set when its interrupt condition is active

38.6.1.39 Command Queuing Interrupt Signal Enable (CQISGE)

This register enables and disables the generation of interrupts to host software. When a bit is set ('1') and the corresponding bit in CQIS is set, then an interrupt is generated. Interrupt sources that are disabled (‘0’) are still indicated in the CQIS register. This register is bit-index matched to CQIS register.

Offset

Register	Offset
CQISGE	118h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												TCL_SGE	RED_SGE	TCC_SGE	HAC_SGE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved
3 TCL_SGE	Task cleared signal enable When set and CQIS[TCL] is asserted, the CQE generates an interrupt.
2 RED_SGE	Response error detected signal enable When set and CQIS[RED] is asserted, the CQE generates an interrupt.
1 TCC_SGE	Task complete signal enable When set and CQIS[TCC] is asserted, the CQE generates an interrupt.
0 HAC_SGE	Halt complete signal enable When set and CQIS[HAC] is asserted, the CQE generates an interrupt.

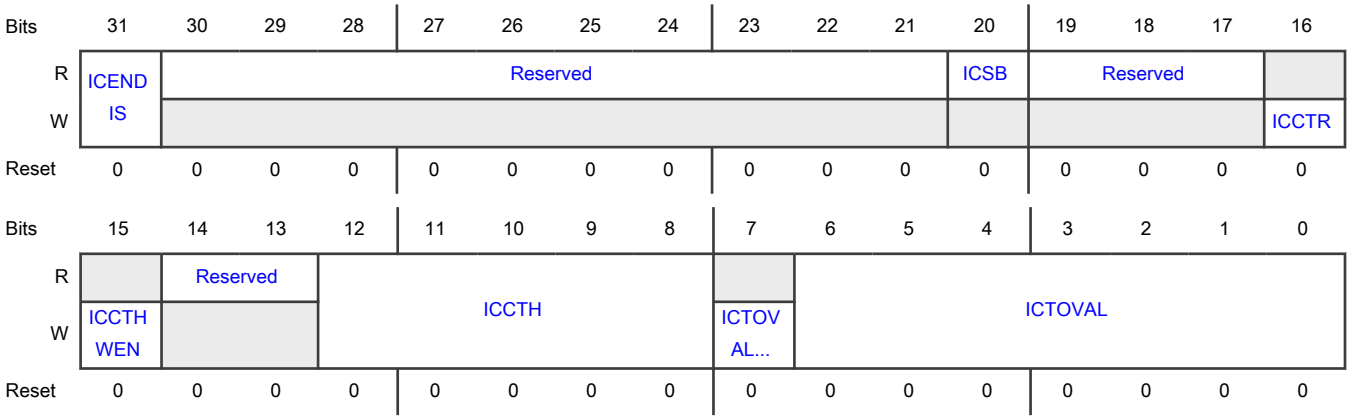
38.6.1.40 Command Queuing Interrupt Coalescing (CQIC)

This register controls the interrupt coalescing feature.

Offset

Register	Offset
CQIC	11Ch

Diagram



Fields

Field	Description
31 ICENDIS	Interrupt coalescing enable/disable When host driver writes ‘1’, the interrupt coalescing timer and counter are reset.
30-21 —	Reserved
20 ICSB	Interrupt coalescing status This bit indicates to software whether any tasks (with INT=0) have completed and counted towards interrupt coalescing (for example, ICSB is set if and only if IC counter > 0). 0 - No task completions have occurred since last counter reset (IC counter =0) 1 - At least one task completion has been counted (IC counter >0)
19-17 —	Reserved
16 ICCTR	Counter and timer reset When host driver writes ‘1’, the interrupt coalescing timer and counter are reset.
15 ICCTHWEN	Interrupt coalescing counter threshold write enable When software writes ‘1’, the value ICCTH is updated with the contents written at the same cycle. When software writes ‘0’, the value in ICCTH is not updated.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE Write operations to ICCTH are only allowed when the task queue is empty.
14-13 —	Reserved
12-8 ICCTH	Interrupt coalescing counter threshold Software uses this field to configure the number of task completions (only tasks with INT=0 in the Task Descriptor) which are required in order to generate an interrupt. Counter Operation: As data transfer tasks with INT=0 complete, they are counted by CQE. The counter is reset by software during the interrupt service routine. The counter stops counting when it reaches the value configured in ICCTH. The maximum allowed value is 31. **NOTE** When ICCTH is 0, task completions are not counted, and counting-based interrupts are not generated. In order to write to this field, the ICCTHWEN field must be set at the same write operation.
7 ICTOVALWEN	Interrupt coalescing timeout value write enable When software writes ‘1’, the value ICTOVAL is updated with the contents written at the same cycle. When software writes ‘0’, the value in ICTOVAL is not updated. **NOTE** Write operations to ICTOVAL are only allowed when the task queue is empty.
6-0 ICTOVAL	Interrupt coalescing timeout value Software uses this field to configure the maximum time allowed between the completion of a task on the bus and the generation of an interrupt. Timer Operation: The timer is reset by software during the interrupt service routine. It starts running when a data transfer task with INT=0 is completed, after the timer was reset. When the timer reaches the value configured in ICTOVAL field it generates an interrupt and stops. The timer’s unit is equal to 1024 clock periods of the clock whose frequency is specified in the Internal Timer Clock Frequency field CQCAP register. The minimum value is 01h (1024 clock periods) and the maximum value is 7Fh (127*1024 clock periods). For example, a CQCAP field value of 0 indicates a 19.2 MHz clock frequency (period = 52.08 ns). If the setting in ICTOVAL is 10h, the calculated polling period is 16*1024*52.08 ns= 853.33 us **NOTE** When ICTOVAL is 0, the timer is not running, and timer-based interrupts are not generated. In order to write to this field, the ICTOVALWEN bit must be set at the same write operation.

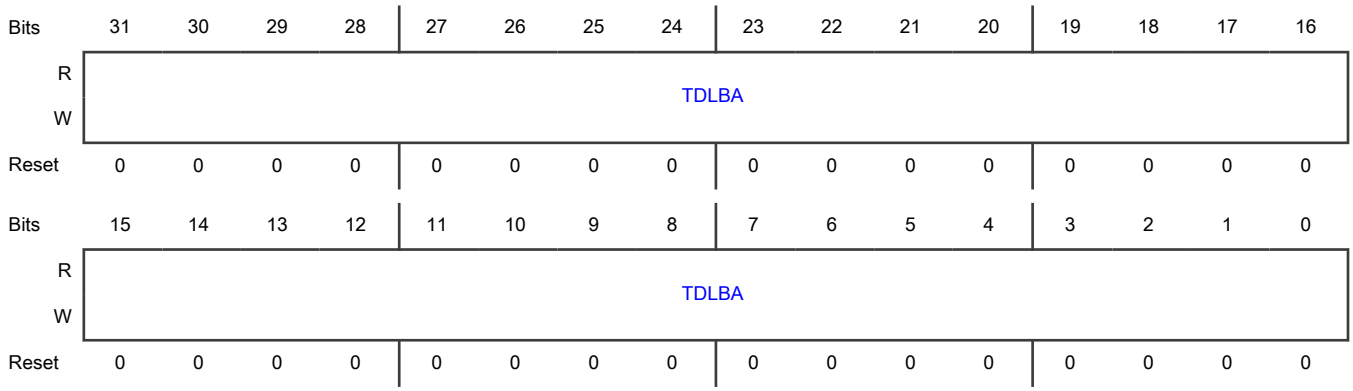
38.6.1.41 Command Queuing Task Descriptor List Base Address (CQTDLBA)

This register is used for configuring the lower 32 bits of the byte address of the head of the Task Descriptor List in the host memory.

Offset

Register	Offset
CQTDLBA	120h

Diagram



Fields

Field	Description
31-0 TDLBA	Task descriptor list base address This register stores the LSB bits (bits 31:0) of the byte address of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This address shall be set on 1 KByte boundary: The lower 10 bits of this register shall be set to 0 by software and shall be ignored by CQE.

38.6.1.42 Command Queuing Task Descriptor List Base Address Upper 32 Bits (CQTDLBAU)

This register is used for configuring the upper 32 bits of the byte address of the head of the Task Descriptor List in the host memory.

Offset

Register	Offset
CQTDLBAU	124h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDLBAU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDLBAU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TDLBAU	Task descriptor list base address This register stores the MSB bits (bits 63:32) of the byte address of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This register is reserved when using 32-bit addressing mode.

38.6.1.43 Command Queuing Task Doorbell (CQTDBR)

Using this register, software triggers CQE to process a new task.

Offset

Register	Offset
CQTDBR	128h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TDBR	Task doorbell Software configures TDLBA and TDLBAU, and enable CQE in CQCFG before using this register. Writing 1 to bit n of this register triggers CQE to start processing the task encoded in slot n of the TDL. CQE always processes tasks in-order according to the order submitted to the list by CQTDBR write transactions. CQE processes Data Transfer tasks by reading the Task Descriptor and sending QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) commands to the device. CQE processes DCMD tasks (in slot #31, when enabled) by reading the Task Descriptor, and generating the command encoded by its index and argument. The corresponding bit is cleared to '0' by CQE in one of the following events: 1. When a task execution is completed (with success or error) 2. The task is cleared using CQTCLR register 3. All tasks are cleared using CQCTL register 4. CQE is disabled using CQCFG register Software may initiate multiple tasks at the same time (batch submission) by writing 1 to multiple bits of this register in the same transaction. In the case of batch submission: CQE shall process the tasks in order of the task index, starting with the lowest index. If one or more tasks in the batch are marked with QBR, the ordering of execution will be based on said processing order. Writing 0 by software shall have no impact on the hardware, and will not change the value of the register bit.

38.6.1.44 Command Queuing Task Completion Notification (CQTCN)

This register is used by CQE to notify software about completed tasks.

Offset

Register	Offset
CQTCN	12Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TCN															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TCN															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TCN	Task complete notification CQE shall set bit n of this register (at the same time it clears bit n of CQTDBR) when a task execution is completed (with success or error). When receiving interrupt for task completion, software may read this register to know which tasks have finished. After reading this register, software may clear the relevant bit fields by writing 1 to the corresponding bits.

38.6.1.45 Command Queuing Device Queue Status (CQDQS)

This register stores the most recent value of the device's queue status.

Offset

Register	Offset
CQDQS	130h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DQS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DQS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 DQS	Device queue status Every time the Host controller receives a queue status register (QSR) from the device, it updates this register with the response of status command (For example, the chip's queue status).

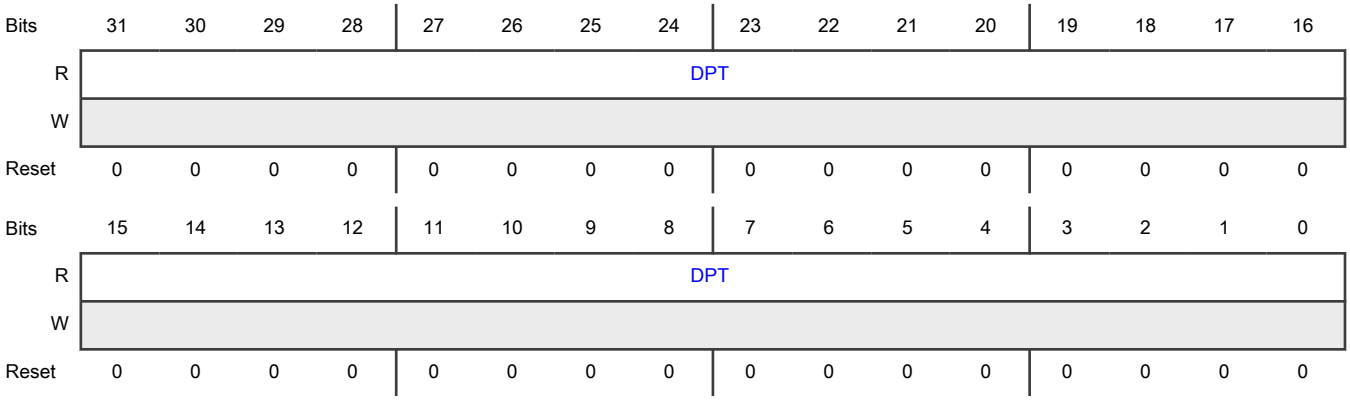
38.6.1.46 Command Queuing Device Pending Tasks (CQDPT)

This register indicates to software which tasks are queued in the device, awaiting execution.

Offset

Register	Offset
CQDPT	134h

Diagram



Fields

Field	Description
31-0 DPT	Device pending tasks Bit n of this register is set if and only if QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) were sent for this specific task and if this task hasn't been executed yet. CQE shall set this bit after receiving a successful response for CMD45. CQE shall clear this bit after the task has completed execution. Software needs to read this register in the task-discard procedure, when the controller is halted, to determine if the task is queued in the device. If the task is queued, the driver sends a CMDQ_TASK_MGMT (CMD48) to the device ordering it to discard the task. Then software clears the task in the CQE. Only then the software orders CQE to resume its operation using CQCTL register.

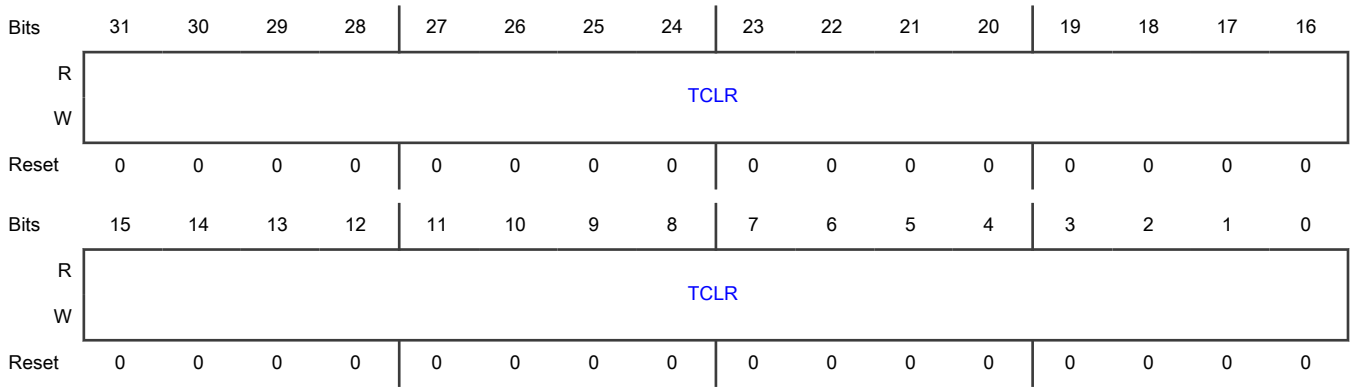
38.6.1.47 Command Queuing Task Clear (CQTCLR)

This register is used for removing an outstanding task in the CQE. The register should be used only when CQE is in Halt state.

Offset

Register	Offset
CQTCLR	138h

Diagram



Fields

Field	Description
31-0 TCLR	Task clear Writing 1 to bit n of this register orders CQE to clear a task which software has previously issued. This bit can only be written when CQE is in Halt state as indicated in CQCFG register Halt bit. When software writes '1' to a bit in this register, CQE updates the value to '1', and starts clearing the data structures related to the task. CQE clears the bit fields (sets a value of 0) in CQTCLR and in CQTDBR once clear operation is complete. Software should poll on the CQTCLR until it is cleared to verify clear operation was complete. Writing to this register only clears the task in the CQE and does not have impact on the device. In order to discard the task in the device, host software shall send CMDQ_TASK _MGMT while CQE is still in Halt state. Host driver is not allowed to use this register to clear multiple tasks at the same time. Clearing multiple tasks can be done using CQCTL register. Writing 0 to a register bit shall have no impact.

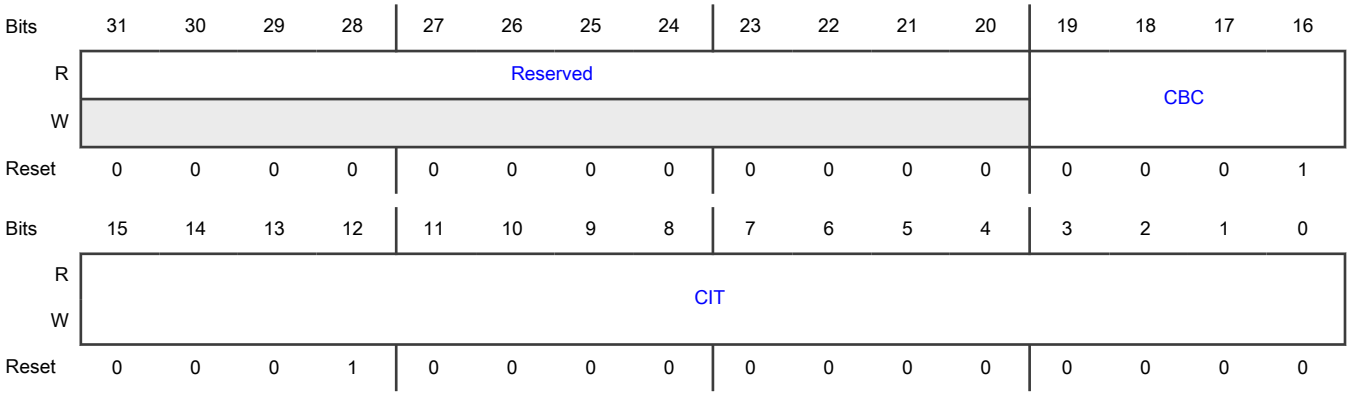
38.6.1.48 Command Queuing Send Status Configuration 1 (CQSSC1)

The register controls the when SEND_QUEUE_STATUS commands are sent.

Offset

Register	Offset
CQSSC1	140h

Diagram



Fields

Field	Description
31-20 —	Reserved
19-16 CBC	Send status command block counter This field indicates to CQE when to send SEND_QUEUE_STATUS (CMD13) command to inquire the status of the device's task queue. A value of *n* means CQE sends status command on the CMD line, during the transfer of data block BLOCK_CNT-n, on the data lines, where BLOCK_CNT is the number of blocks in the current transaction. A value of 0 means that SEND_QUEUE_STATUS (CMD13) command is not sent during the transaction. Instead it will be sent only when the data lines are idle. A value of 1 means that STATUS command is to be sent during the last block of the transaction.
15-0 CIT	Send status command idle timer This field indicates to CQE the polling period to use when using periodic SEND_QUEUE_STATUS (CMD13) polling. Periodic polling is used when tasks are pending in the device, but no data transfer is in progress. When a SEND_QUEUE_STATUS response indicating that no task is ready for execution, CQE counts the configured time until it issues the next SEND_QUEUE_STATUS. Timer units are clock periods of the clock whose frequency is specified in the Internal Timer Clock Frequency field CQCAP register. The minimum value is 0001h (1 clock period) and the maximum value is FFFFh (65535 clock periods). Default interval is: 4096 clock periods. For example, a CQCAP field value of 0 indicates a 19.2 MHz clock frequency (period = 52.08 ns). If the setting in CQSSC1.CIT is 1000h, the calculated polling period is 4096*52.08 ns= 213.33 us

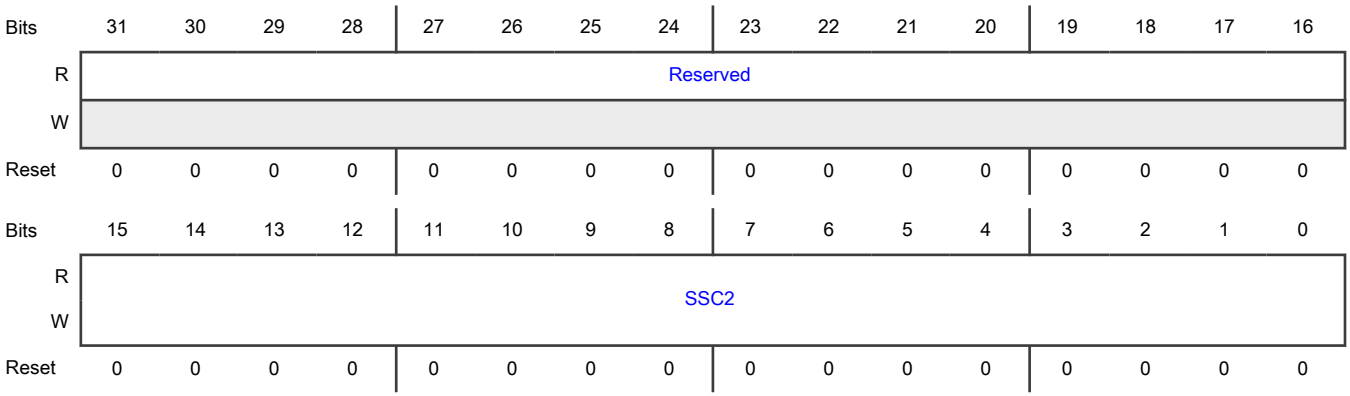
38.6.1.49 Command Queuing Send Status Configuration 2 (CQSSC2)

This register is used for configuring RCA field in SEND_QUEUE_STATUS command argument.

Offset

Register	Offset
CQSSC2	144h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 SSC2	Send queue status RCA This field provides CQE with the contents of the 16-bit RCA field in SEND_QUEUE_ STATUS (CMD13) command argument. CQE copies this field to bits 31:16 of the argument when transmitting SEND_ QUEUE_STATUS (CMD13) command.

38.6.1.50 Command Queuing Command Response for Direct-Command Task (CQCRDCT)

This register is used for passing the response of a DCMD task to software.

Offset

Register	Offset
CQCRDCT	148h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CRDCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CRDCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CRDCT	Direct command last response This register contains the response of the command generated by the last direct-command (DCMD) task which was sent. CQE updates this register when it receives the response for a DCMD task. This register is considered valid only after bit 31 of CQTDBR register is cleared by CQE.

38.6.1.51 Command Queuing Response Mode Error Mask (CQRMEM)

This register controls the generation of Response Error Detection (RED) interrupt.

Offset

Register	Offset
CQRMEM	150h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RMEM															
W																
Reset	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RMEM															
W																
Reset	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RMEM	Response mode error mask This bit is used as in interrupt mask on the device status filed which is received in R1/R1b responses. The reset value of this register is set to trigger an interrupt on all “Error” type bits in the device status, as defined in 6.13. **NOTE** Responses to CMD13 (SQS) encode the QSR, so they are ignored by this logic. 0000_0000_0000_0000_0000_0000_0000_0000 - When a R1/R1b response is received, bit i in the device status is ignored 0000_0000_0000_0000_0000_0000_0000_0001 - When a R1/R1b response is received, with bit i in the device status set, a RED interrupt is generated

38.6.1.52 Command Queuing Task Error Information (CQTERRI)

This register is updated by CQE when an error occurs on data or command related to a task activity.

When such error is detected by CQE or indicated by the eMMC controller CQE stores in CQTERRI the task IDs and the command indices of the commands which were executed on the command line and data lines when the error occurred.

Software is expected to use this information in the error recovery procedure. See B.2.8 for more details.

Offset

Register	Offset
CQTERRI	154h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DTEFV	Reserved			DTETID				Reserved		DTECI					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RMEFV	Reserved			RMETID				Reserved		RMECI					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 DTEFV	Data transfer error fields valid This bit is updated when an error is detected by CQE, or indicated by eMMC controller. If a data transfer is in progress when the error is detected/indicated, the bit is set to 1. If a no data transfer is in progress when the error is detected/indicated, the bit is cleared to 0.
30-29 —	Reserved
28-24 DTETID	Data transfer error task ID This field indicates the ID of the task which was executed on the data lines when an error occurred. The field is updated if a data transfer is in progress when an error is detected by CQE, or indicated by eMMC controller.
23-22 —	Reserved
21-16 DTECI	Data transfer error command index This field indicates the index of the command which was executed on the data lines when an error occurred. The index shall be set to EXECUTE_READ_TASK (CMD46) or EXECUTE_WRITE_TASK (CMD47) according to the data direction. The field is updated if a data transfer is in progress when an error is detected by CQE, or indicated by eMMC controller.
15 RMEFV	Response mode error fields valid This bit is updated when an error is detected by CQE, or indicated by eMMC controller. If a command transaction is in progress when the error is detected/indicated, the bit is set to 1. If a no command transaction is in progress when the error is detected/indicated, the bit is cleared to 0.
14-13 —	Reserved
12-8 RMETID	Response mode error task ID This field indicates the index of the command which was executed on the command line when an error occurred. The field is updated if a command transaction is in progress when an error is detected by CQE, or indicated by eMMC controller.
7-6 —	Reserved
5-0	Response mode error command index

Table continues on the next page...

Table continued from the previous page...

Field	Description
RMECI	This field indicates the index of the command which was executed on the command line when an error occurred. The field is updated if a command transaction is in progress when an error is detected by CQE, or indicated by eMMC controller.

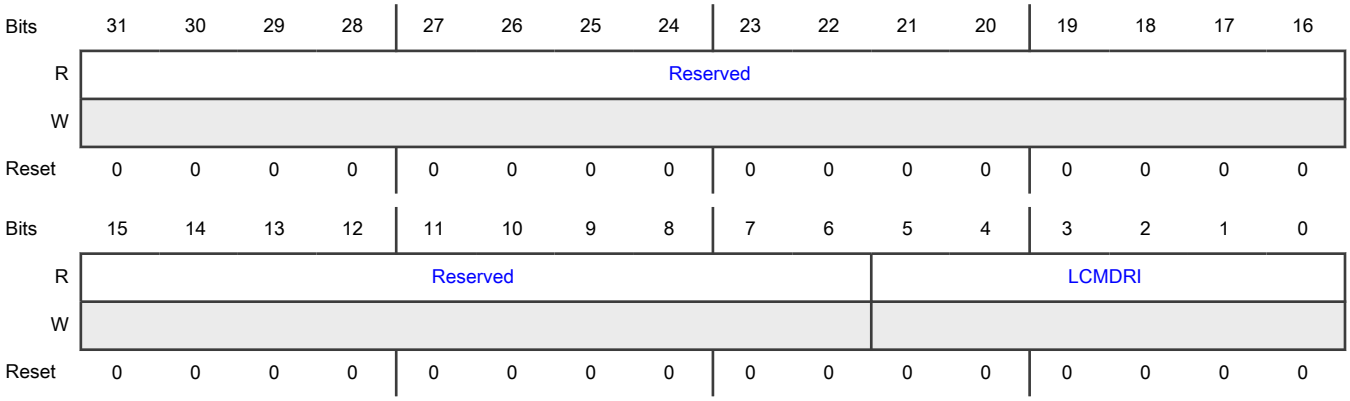
38.6.1.53 Command Queuing Command Response Index (CQCRI)

This register stores the index of the last received command response.

Offset

Register	Offset
CQCRI	158h

Diagram



Fields

Field	Description
31-6 —	Reserved
5-0 LCMDRI	Last command response index This field stores the index of the last received command response. CQE shall update the value every time a command response is received..

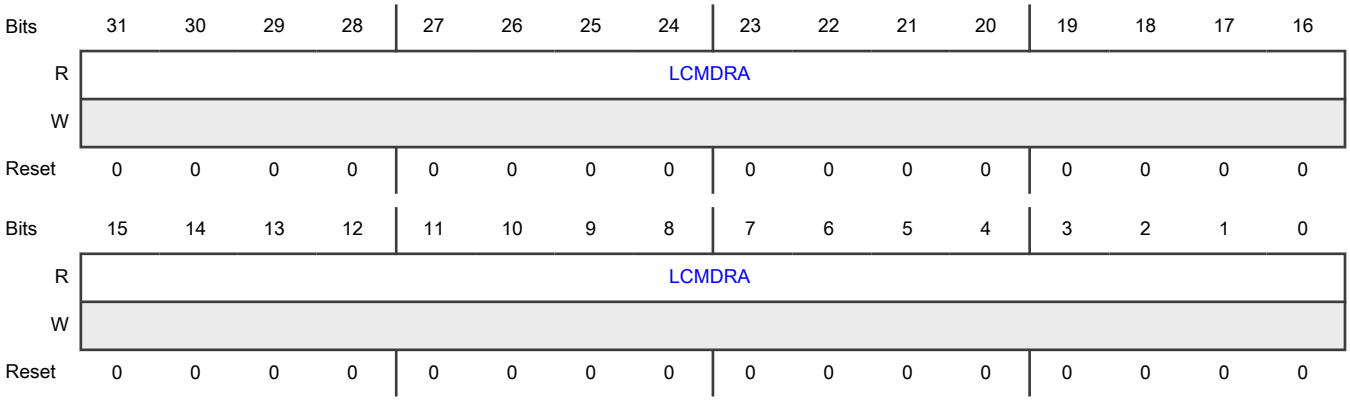
38.6.1.54 Command Queuing Command Response Argument (CQCRA)

This register stores the argument of the last received command response.

Offset

Register	Offset
CQCRA	15Ch

Diagram



Fields

Field	Description
31-0 LCMDRA	Last command response argument This field stores the argument of the last received command. CQE shall update the value every time a command response is received.

38.7 Glossary

eMMC	Embedded Multi-Media Card
SD Card	Secure Digital Card
eSD	embedded SD Memory Device
SDIO	SD Input/Output
SD Combo	Memory and I/O SD Card
SDSC	SD memory card with Standard Capacity, up to and including 2 GB
SDHC	SD memory card with High Capacity, more than 2 GB, up to and including 32 GB
SDXC	SD memory card with Extended Capacity, more than 32 GB, up to and including 2 TB
Default Speed mode	up to 12.5 MB/s (3.3 V signaling, frequency up to 25 MHz)
High Speed mode	up to 25 MB/s (3.3 V signaling, frequency up to 50 MHz)
SDR12	up to 12.5 MB/s (1.8 V signaling, frequency up to 25 MHz)
SDR25	up to 25 MB/s (1.8 V signaling, frequency up to 50 MHz)
SDR50	up to 50 MB/s (1.8 V signaling, frequency up to 100 MHz)
SDR104	up to 104 MB/s (1.8 V signaling, frequency up to 208 MHz)

DDR50	up to 50 MB/s (1.8 V signaling, frequency up to 50 MHz, sampled on both clock edges)
DLL	Delay Line, which is used to tune SDCLK
UHS-I	Ultra High Speed - I
CMD	Command line
ACMD	Application CMD
DATA	Data line
SDCLK	Clock line
DMA	Direct Memory Access
ADMA	Advanced DMA
ADMA1	Advanced DMA1, only supports 4 KB aligned data transfer
ADMA2	Advanced DMA2, no 4 KB aligned limitation
Rsv	Reserved
Nop	None operation
CLK	Clock
CQE	Command Queuing Engine
TDL	Task Descriptor List
SDR	Single Data Rate signaling
DDR	Double Data Rate signaling
ASx	Abort Select
CCCR	SDIO or Combo Card Common Control Registers
CID	Card Identification number register
CIS	Card Information Structure register
CSD	Card Specific Data register

Chapter 39
CAN (FlexCAN)

39.1 Chip-specific FLEXCAN information

Table 240. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The chip includes two instances of CAN-FD Controllers (CAN-FD1 is in AONMIX and CAN-FD2 is in WAKEUPMIX).

CAN1 features a low-pass filter (=glitch filter) on the CAN RX input. This filter is fed by the can1_root_clk, and enabled by the WAKSRC field in the MCR register. If the filter is enabled and can1_clk_root is gated, then the CAN RX input will not reach the FlexCAN logic. This condition must be avoided either by disabling the glitch filter, or by ensuring can1_clk_root is always running when the CAN functionality is required.

CAN2 features a low-pass filter (=glitch filter) on the CAN RX input. This filter is fed by the WAKEUPMIX 24MHz clock and enabled by the WAKSRC field in the MCR register. If the filter is enabled its clock is gated, then the CAN RX input will not reach the FlexCAN logic. This condition must be avoided either by disabling the glitch filter, or by ensuring its clock is always running when the CAN functionality is required.

NOTE

The FlexCAN clock description mentions an "oscillator domain". This domain is connected either to can_clk_root for the CAN module (CAN_CTRL1[CLKSRC] = 0) or to the WAKEUPMIX bus clock (CAN_CTRL1[CLKSRC]=1). It is not directly connected to any oscillator.

39.2 Overview

FlexCAN is a communication controller implementing the CAN protocol according to the ISO 11898-1_2024 standard and CAN 2.0 Part B protocol specifications.

The CAN protocol was primarily designed to be used as a vehicle serial data bus, meeting the specific real-time processing and reliable operation requirements in the electromagnetic interference (EMI) environment of a vehicle. FlexCAN is a full implementation of the CAN protocol specification, the CAN with flexible data rate (CAN FD) protocol, and the CAN version 2.0 Part B protocol. It supports both standard and extended message frames and long payloads.

NOTE

Legacy Receive (RX) FIFO cannot be used in CAN Flexible Data (FD) mode. In CAN FD mode, use the Enhanced Receive FIFO feature instead of the Legacy Receive FIFO.

NOTE

FD mode is limited to Message Buffer (MB) mode only, the Enhanced Receive FIFO with DMA is supported.

39.2.1 Block diagram

Figure 145 shows the main submodules implemented in FlexCAN.

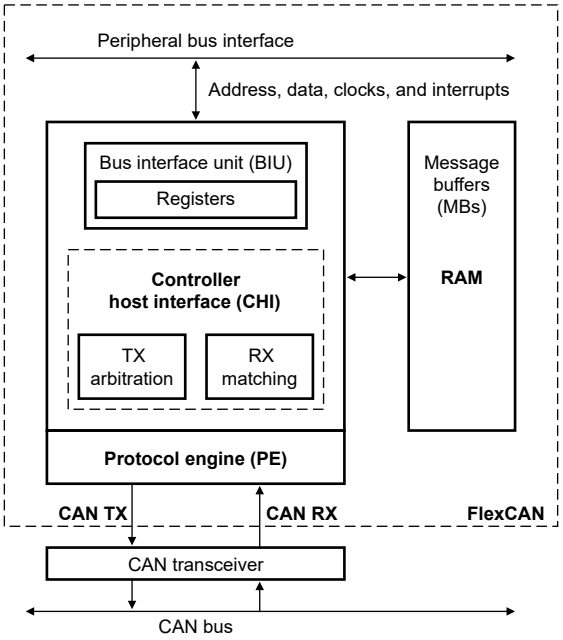


Figure 145. Block diagram

The Protocol Engine (PE) submodule manages the serial communication on the CAN bus:

- RAM access for receiving and transmitting message frames
- Receive message validation
- Error handling
- CAN FD message detection

Controller Host Interface (CHI) manages:

- Message buffer (MB) selection for reception and transmission
- Arbitration and ID matching algorithms for both CAN FD and non-CAN FD message formats

Bus Interface Unit (BIU) controls access to and from the internal interface bus to establish connection to the CPU and to other blocks. The BIU manages access to:

- Clocks
- Address and data buses
- Interrupt outputs
- DMA
- Test signals

39.2.2 Features

- Full implementation of *CAN with Flexible data rate (CAN FD) protocol specification* and *CAN Specification Version 2.0, Part B*:
  - Standard data frames
  - Extended data frames

  - Data length of 0–64 bytes
  - Content-related addressing
- Compliance with the ISO 11898-1_2024 standard
- Flexible message buffers that can be configured to store a payload of 0, 8, 16, 32, or 64 bytes
  - Increasing the payload size decreases the number of supported message buffers (see FlexCAN memory partition for CAN FD).
  - Message buffers are configurable as receive or transmit, supporting standard and extended messages.
- Individual Receive Mask registers for each message buffer
- Full-featured Legacy RX FIFO with storage capacity for six frames and automatic internal pointer handling with DMA support
- Full-featured Enhanced RX FIFO with storage capacity of 20 CAN FD frames and automatic internal pointer handling with DMA support
- Transmission abort capability
- Programmable clock source to CAN Protocol Interface, either peripheral clock or oscillator clock
- Optional general purpose RAM space, using RAM not used by reception or transmission structures
- Listen-Only mode
- Programmable loopback mode supporting self-test operation
- Programmable transmission priority scheme: lowest ID, lowest buffer number, or highest priority
- Timestamp based on 32-bit free-running timer, with optional external time tick
- Global network time, synchronized by specific message
- Maskable interrupts
- Independence from transmission medium (external transceiver is assumed)
- Short latency time due to arbitration scheme for high-priority messages
- Low-power modes, with programmable wake-up on bus activity
- Transceiver delay compensation when transmitting CAN FD messages at faster data rates
- Management of remote request frames, automatically or by software
- Restriction only to write CAN bit time settings and configuration bits in Freeze mode
- Transmit message buffer status (lowest-priority buffer or empty buffer)
- Identifier Acceptance Filter Hit Indicator (IDHIT) register for received frames
- ESR1[SYNCH] to indicate module is synchronous with CAN bus
- CRC status for transmitted message
- Legacy RX FIFO Global Mask register
- Selectable priority between message buffers and receive FIFO during matching process
- Powerful legacy RX FIFO ID filtering, capable of matching incoming IDs against either 128 extended IDs, 256 standard IDs, or 512 partial (8-bit) IDs, with 32 individual masking capability
- Powerful enhanced RX FIFO ID filtering, capable of matching incoming IDs against either 64 extended or 128 standard ID filter elements with three filtering schemes: mask plus filter, range, and two filters without mask
- Complete backward compatibility with previous FlexCAN version
- Detection and correction of errors in memory read accesses:

— Each byte of FlexCAN memory is associated to five parity bits.

— The error correction mechanism ensures that errors in one bit of this 13-bit word can be corrected (correctable errors).

— Errors in two bits can be detected but not corrected (noncorrectable errors).

39.3 Functional description

FlexCAN is a CAN protocol engine with a flexible message buffer system for transmitting and receiving CAN frames. The system is a set of message buffers (MBs) that stores configuration and control data, timestamp, message ID, and data (see Message buffer structure).

For classical CAN frames, FlexCAN supports simultaneous reception through Legacy FIFO and message buffer. For CAN FD frames, FlexCAN supports reception through message buffer and enhanced receive FIFO.

For message buffer reception, a matching algorithm makes it possible to store received frames only into MBs that have the same ID programmed in the ID field. A masking scheme makes it possible to match the ID programmed on the message buffer with a range of IDs on received CAN frames. For transmission, an arbitration algorithm decides the prioritization of message buffers to be transmitted based on the message ID (optionally augmented by three local priority bits) or the message buffer ordering.

A message buffer is active at a given time if it can participate in both the matching and arbitration processes. A receive message buffer with a 0b code is inactive (see Table 278). A transmit message buffer with a 1000b or 1001b code is also inactive (see Table 279).

FlexCAN can receive and transmit messages in CAN FD format. The message buffers are sized to store the quantity of data bytes selected by FDCTRL[MBDSRn]. The quantity of FD message buffers available for a given quantity of data bytes is described in the CAN FD Control (FDCTRL) register. See also FlexCAN memory partition for CAN FD.

39.3.1 Modes of operation

FlexCAN has these functional modes:

Table 241. Functional modes

Mode	Description
Normal (User or Supervisor)	In Normal mode, FlexCAN receives and transmits message frames, manages errors normally, and enables all CAN Protocol functions. User and Supervisor modes differ in the access to some restricted control registers.
Freeze	Freeze mode is enabled when MCR[FRZ] = 1. If enabled, FlexCAN enters Freeze mode when MCR[HALT] is 1 or when is requested at chip level and FlexCAN writes 1 to MCR[FRZACK]. In this mode, no transmission or reception of frames is done, and synchronicity to the CAN bus is lost. See Freeze mode.
Loopback	FlexCAN enters this mode when CTRL1[LPB] becomes 1. In this mode, FlexCAN performs an internal loopback that can be used for self-test. The bit stream output of the transmitter is internally fed back to the receiver input. The receiving CAN input pin is ignored and the transmitting CAN output goes to the recessive state (logic 1). FlexCAN behaves as it normally does when transmitting and treats its own transmitted message as a message received from a remote node. To ensure proper reception of its own message, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field. Both transmit and receive interrupts are generated.
Listen-Only	FlexCAN enters this mode when CTRL1[LOM] becomes 1. In this mode, transmission is disabled, all error counters are frozen, and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station are received. If FlexCAN detects an unacknowledged message, it flags a BIT0 error (without changing the REC), as if it was trying to acknowledge the message.

Table continues on the next page...

Table 241. Functional modes (continued)

Mode	Description
CAN FD Active	In this mode, FlexCAN can transmit and receive all messages formatted according to the CAN FD Standard (2.0) and 2.0B Protocol in an interleaved fashion. The CPU can put FlexCAN into CAN FD Active mode by configuring MCR[FDEN] in Freeze mode.

Some features available in classical CAN are unavailable in CAN FD mode.

Table 242. Differences between classical CAN and CAN FD

Feature	Classical CAN	CAN FD
Legacy RX FIFO DMA	Yes	No
Legacy RX FIFO	Yes	No
Enhanced RX FIFO DMA	Yes	Yes
Enhanced RX FIFO	Yes	Yes

FlexCAN can operate in these low-power modes:

Table 243. Low-power modes

Mode	Description
Module Disable	FlexCAN enters this mode when the CPU writes 1 to MCR[MDIS] and FlexCAN writes 1 to MCR[LPMACK]. After FlexCAN is disabled, it issues a request to disable the clocks to the CAN Protocol Engine and Controller Host Interface submodules. Writing 0 to MCR[MDIS] exits this mode. See Module Disable mode.
Doze	FlexCAN enters this mode when MCR[DOZE] is 1, Doze mode is requested at chip level, and FlexCAN writes 1 to MCR[LPMACK]. When in Doze mode, FlexCAN issues a request to disable the clocks to the CAN Protocol Engine and the CAN Controller-Host Interface submodules. This mode is exited when: • MCR[DOZE] becomes 0 • The chip is removed from Doze mode • Or activity is detected on the CAN bus and the Self-Wake-up mechanism is enabled. See Doze mode.
Stop	FlexCAN enters this mode when Stop mode is requested at chip level and FlexCAN writes 1 to MCR[LPMACK]. When in Stop mode, FlexCAN puts itself in an inactive state and then informs the CPU that the clocks can be shut down globally. Exit from this mode occurs when the Stop mode request is removed, or when activity is detected on the CAN bus and the Self-Wake-up mechanism is enabled. See Stop mode.

39.3.1.1 Modes of operation details

FlexCAN has functional modes and low-power modes. See Modes of operation for an introductory description of all the modes of operation. The following subsections contain functional details about Freeze mode and low-power modes.

CAUTION

FlexCAN does not support "Permanent Dominant" failure on the CAN bus line. If a Low-Power request or Freeze mode request occurs during a "Permanent Dominant" condition, the corresponding acknowledgment field can never be 1.

39.3.1.1.1 Freeze mode

This mode is requested either by the CPU writing 1 to MCR[HALT] or when the chip is put into Debug mode. MCR[FRZ] must be 1 and the module must not be in a low-power mode.

When MECR[NCEFAFRZ] becomes 1 and a noncorrectable error is detected in a memory read access performed by FlexCAN internal processes (see Error response), FlexCAN also requests Freeze mode. This request occurs via both MCR[HALT] and MCR[FRZ] automatically becoming 1.

To obtain acknowledgment, FlexCAN writes 1 to MCR[FRZACK]. The CPU must only consider FlexCAN to be in Freeze mode when both request and acknowledgment conditions are satisfied.

When Freeze mode is requested, FlexCAN:

1. Waits to be in either Intermission, Error Passive, Bus Off, or Idle state.
2. Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in does not prevent entering Freeze mode.
3. Ignores the receive input pin and drives the transmit pin as recessive.
4. Stops the prescaler, halting all CAN protocol activities.
5. Grants write access to the Error Counters register, which is read-only in other modes.
6. Writes 1 to MCR[NOTRDY] and MCR[FRZACK].

After requesting Freeze mode, you must wait for MCR[FRZACK] to become 1 before executing any other action, otherwise FlexCAN may operate unpredictably. In Freeze mode, all memory-mapped registers are accessible, except for CTRL1[CLKSRC], which can be read but cannot be written.

Freeze mode is exited in one of these conditions:

- CPU writes 0 to MCR[FRZ].
- The chip is removed from Debug mode or the MCR[HALT] becomes 0.

MCR[FRZACK] becomes 0 after the protocol engine recognizes the negation of the freeze request. After leaving Freeze mode, FlexCAN tries to resynchronize to the CAN bus by waiting for 11 consecutive recessive bits.

39.3.1.1.2 Module Disable mode

This low-power mode is normally used to disable a complete FlexCAN block temporarily, with no power consumption. The CPU requests this mode by writing 1 to MCR[MDIS], and FlexCAN acknowledges the request by writing 1 to MCR[LPMACK]. The CPU must only consider FlexCAN to be in Disable mode when both the request and acknowledgment conditions are satisfied.

If FlexCAN is disabled during Freeze mode, the module requests to disable the clocks to the PE and CHI submodules, writes 1 to MCR[LPMACK] and writes 0 to MCR[FRZACK].

If the module is disabled during transmission or reception, FlexCAN:

1. Waits to be in either Idle or Bus Off state, or waits for the third bit of Intermission and then checks that it is recessive.
2. Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. FlexCAN does not take a pending move-in into account.
3. Ignores its receive input pin and drives its transmit pin as recessive.
4. Shuts down the clocks to the PE and CHI submodules.
5. Writes 1 to MCR[NOTRDY] and MCR[LPMACK].

In this mode, the Bus Interface Unit continues to operate, enabling the CPU to access memory-mapped registers, except for:

- The Receive Message Buffers Global Mask registers
- The Receive Buffer 14 Mask register
- The Receive Buffer 15 Mask register

When in Disable mode, these items may not be accessible:

- The message buffers
- The Receive Individual Mask registers
- The reserved words within RAM

To exit this mode, the CPU writes 0 to MCR[MDIS], causing FlexCAN to request to resume the clocks and write 0 to MCR[LPMACK]. This write occurs after the CAN protocol engine recognizes the negation of disable mode requested by the CPU.

39.3.1.1.3 Doze mode

This is a system low-power mode in which the CPU bus is kept alive and a global Doze mode request is sent to all peripherals asking them to enter low-power mode. When Doze mode is globally requested, MCR[DOZE] needs to have been asserted previously for Doze mode to be triggered. The acknowledgement is obtained through the assertion of MCR[LPMACK] by FlexCAN. The CPU must only consider FlexCAN to be in Doze mode when both request and acknowledgement conditions are satisfied.

If Doze mode is triggered during Freeze mode, FlexCAN requests to shut down the clocks to the PE and CHI submodules, sets MCR[LPMACK] and negates MCR[FRZACK]. If Doze mode is triggered during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive.
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive.
- Shuts down the clocks to the PE and CHI submodules.
- Sets MCR[NOTRDY] and MCR[LPMACK].

The Bus Interface Unit continues to operate, enabling the CPU to access memory-mapped registers, except the Rx Mailboxes Global Mask registers, the Rx Buffer 14 Mask register, the Rx Buffer 15 Mask register.

Also, the message buffers, the Rx Individual Mask registers, and the reserved words within RAM may not be accessed when the module is in Doze mode.

Exiting Doze mode is done in one of the following ways:

- CPU removing the Doze mode request
- CPU negating MCR [DOZE]
- Self-Wake mechanism

In the Self-Wake mechanism, if MCR[SLFWAK] was set at the time FlexCAN entered Doze mode, then after detection of a recessive to dominant transition on the CAN bus, FlexCAN negates the DOZE bit, requests to resume its clocks and negates the LPMACK after the CAN protocol engine recognizes the negation of the Doze mode request. It also sets ESR [WAKINT] and, if enabled by MCR[WAKMSK], generates a Wake-Up interrupt to the CPU. FlexCAN then waits for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it does not receive the frame that woke it up. The following table details the effect of SLFWAK and WAKMSK after wakeup from Doze mode.

Table 244. Wakeup from Doze mode

SLFWAK	WAKINT	WAKMSK	FlexCAN clocks enabled	Wakeup interrupt generated
0	—	—	No	No
0	—	—	No	No

Table continues on the next page...

Table 244. Wakeup from Doze mode (continued)

SLFWAK	WAKINT	WAKMSK	FlexCAN clocks enabled	Wakeup interrupt generated
1	0	0	No	No
1	0	1	No	No
1	1	0	Yes	No
1	1	1	Yes	Yes

The sensitivity to CAN bus activity can be modified by applying a low-pass filter function to the Rx CAN input line when in Doze mode. (See MCR[WAKSRC].) This feature can be used to protect FlexCAN from waking up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference within noisy environments.

39.3.1.1.4 Stop mode

In this low-power mode for the system, all chip clocks can be stopped for maximum power savings. Stop mode is globally requested by the CPU and FlexCAN writes 1 to a Stop Acknowledgment signal to indicate acknowledgment. The CPU must only consider FlexCAN to be in Stop mode when both request and acknowledgment conditions are satisfied.

If FlexCAN receives the global Stop mode request during Freeze mode, it shuts down the clocks globally by:

1. Writing 1 to MCR[LPMACK].
2. Writing 0 to MCR[FRZACK].
3. Sending the Stop Acknowledge signal to the CPU.

If Stop mode is requested during transmission or reception, FlexCAN:

1. Waits to be in either Idle or Bus Off state, or waits for the third bit of Intermission and verifies that it is recessive.
2. Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. FlexCAN does not take a pending move-in into account.
3. Ignores its receive input pin and drives its transmit pin as recessive.
4. Writes 1 to MCR[NOTRDY] and MCR[LPMACK].
5. Sends a Stop Acknowledge signal to the CPU, so that the CPU can shut down the clocks globally.

FlexCAN exits Stop mode when the CPU resumes the clocks and removes the Stop mode request. This exit can occur as a result of the Self-Wake mechanism.

In Self-Wake, if MCR[SLFWAK] = 1 when FlexCAN enters Stop mode, then upon detecting a recessive-to-dominant transition on the CAN bus, FlexCAN writes 1 to ESR1[WAKINT]. If enabled by MCR[WAKMSK], FlexCAN generates a wake-up interrupt to the CPU. Upon receiving the interrupt, the CPU should resume the clocks and remove the Stop mode request. FlexCAN waits for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, FlexCAN does not receive the frame that woke it up. Table 245 details the effect of MCR[SLFWAK] and MCR[WAKMSK] upon wakeup from Stop mode. Waking from Stop mode only works when both fields are 1.

After the CAN protocol engine recognizes the negation of the Stop mode request, FlexCAN writes 0 to MCR[LPMACK]. FlexCAN then waits for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, FlexCAN does not receive the frame that woke it up.

Table 245. Waking from Stop mode

SLFWAK	WAKINT	WAKMSK	Chip clocks enabled	Wake-up interrupt generated
0	—	—	No	No
0	—	—	No	No
1	0	0	No	No
1	0	1	No	No
1	1	0	No	No
1	1	1	Yes	Yes

The sensitivity to CAN bus activity can be modified by applying a low-pass filter function to the receive CAN input line when in Stop mode. (See the description in MCR[WAKSRC].) This feature can protect FlexCAN from waking up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference within noisy environments.

NOTE

When FlexCAN is in the Bus Off state and Low-Power mode, FlexCAN may take an extra 128 IDLE cycles to leave the Bus Off state after exiting Low-Power mode.

39.3.2 Transmission process

NOTE

Instances of MB_CS in this topic refer to items in message buffers. See Message buffer structure for details.

To transmit a CAN frame, the CPU must prepare a message buffer for transmission by executing the following procedure:

1. Check whether the respective interrupt flag is set, and clear it if necessary.
2. If the message buffer is active (transmission pending), request an abort of the transmission. Write the ABORT code (1001b) to the CODE field of the Control and Status word.
3. Poll the IFLAG register until the corresponding IFLAG flag is set, or wait for the interrupt request (if enabled by the respective IMASK field).
4. Read the CODE field to identify whether the transmission was aborted or transmitted (see Transmission abort mechanism).
5. Clear the corresponding interrupt flag.
6. Write the ID register (containing the local priority if enabled via MCR[LPRIOEN]).
7. Write payload data bytes.
8. Configure the Control and Status word as needed.
    a. Set ID type via MB_CS[IDE].
    b. Set Remote Transmission Request (if needed) via MB_CS[RTR].
    c. If MCR[FDEN] = 1, configure MB_CS[EDL] and MB_CS[BRS]. For details about the relationship between the written value and transmitted value of MB_CS[ESI], see Table 255.[3]
    d. Configure Data Length Code in bytes via MB_CS[DLC]. See Table 282 for detailed information.

[3] There is no need to write the ESI field; it is automatically transmitted as dominant by error active nodes and as recessive by error passive nodes. If FlexCAN is operating as a network gateway, however, the CPU writes MB_CS[ESI] according to the error status of the node that sent the message.

e. To transmit the CAN frame, activate the message buffer by writing Ch to MB_CS[CODE].

NOTE

To maximize software performance, configure all the fields in the MB_CS word in a single 32-bit write operation. If the fields are configured in separate writes, MB_CS[CODE] must be the last write in the Control and Status word.

When the MB is activated, it participates in the arbitration process and its contents are eventually transmitted according to its priority. When the DLC value stored in the MB selected for transmission exceeds the MB payload size, FlexCAN completes the expected DLC by adding a constant CCh pattern.

After a successful transmission:

1. The value of the free-running timer is written into the timestamp field.
2. The CODE field in the Control and Status word is updated.
3. Both Cyclic Redundancy Check (CRCR) and CAN FD CRC (FDCRC) are updated.
4. A status flag is set in the Interrupt Flag register.
5. If allowed by the corresponding Interrupt Mask Register field, an interrupt is generated.

After transmission, the new CODE field depends on the code used to activate the message buffer (see Table 278 and Table 279 in Message buffer structure).

When the Abort feature is enabled (MCR[AEN] is 1), after the Interrupt flag is set for an MB configured as transmit buffer, the message buffer is blocked. The CPU cannot update the message buffer until the Interrupt Flag is cleared by the CPU. The CPU must clear the corresponding IFLAG flag before preparing this MB for a new transmission or reception.

NOTE

For backward compatibility (MCR[AEN] is 0), write the INACTIVE code (1000b) to the CODE field to deactivate the MB. In this case, the pending frame may be transmitted without notification (see Message buffer inactivation).

39.3.3 Arbitration process

The arbitration process scans the message buffers, searching for the transmission MB that holds the message to be sent at the next opportunity. This MB is called the arbitration winner. The scan starts from the lowest number MB and continues to the higher ones.

The arbitration process is triggered when at least one of the following occur:

- CRC field of the CAN frame arrives. The starting point depends on the value of CTRL2[TASD].
- Error Delimiter field of a CAN frame is in progress.
- Overload Delimiter field of a CAN frame is in progress.
- The winner of the arbitration is deactivated, and the CAN bus has not reached the first bit of the Intermission field.
- CPU writes to the Control and Status word of a winner MB, and the CAN bus has not reached the first bit of the Intermission field.
- CHI is in the Idle state and the CPU writes to the Control and Status word of any message buffer.
- FlexCAN exits Bus Off state.
- FlexCAN leaves Freeze mode or a low-power mode.

If the arbitration process does not evaluate all message buffers before the CAN bus reaches the first bit of the Intermission field, the temporary arbitration winner is invalidated. FlexCAN will not compete for the CAN bus at the next opportunity.

The arbitration process selects the winner among the active transmission message buffers at the end of the scan according to the values of CTRL1[LBUF] and MCR[LPRIOEN].

See Arbitration process (continued) for more information about this process.

39.3.3.1 Lowest-number message buffer first

If CTRL1[LBUF] is 1, the first (lowest number) active transmission message buffer found is the arbitration winner. MCR[LPRIOEN] has no effect when CTRL1[LBUF] is 1.

39.3.3.2 Highest-priority message buffer first

If CTRL1[LBUF] is 0, the arbitration process searches for the active transmission message buffer with the highest priority. The frame of this message buffer has a higher probability of winning the arbitration on the CAN bus when multiple external nodes compete for the bus simultaneously.

The sequence of bits considered for this arbitration is called the arbitration value of the message buffer. The transmission message buffer with the lowest arbitration value among all transmission message buffers has the highest priority.

If two or more message buffers have equivalent arbitration values, the message buffer with the lowest number is the arbitration winner.

The composition of the arbitration value depends on MCR[LPRIOEN].

39.3.3.2.1 Local priority disabled

If MCR[LPRIOEN] = 0, the arbitration value is built using the exact sequence of bits that would be transmitted in a CAN frame where local priority is disabled.

Table 246. Composition of the arbitration value when local priority is disabled

Format	Message buffer arbitration value (32 bits)				
Standard (IDE = 0)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	— (18 bits)	— (1 bit)
Extended (IDE = 1)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

39.3.3.2.2 Local priority enabled

To enable local priority, MCR[LPRIOEN] must be 1. In this case, the message buffer PRIO field (see Message buffer structure) is included at the very left of the arbitration value.

Table 247. Composition of the arbitration value when local priority is enabled

Format	Message buffer arbitration value (35 bits)					
Standard (IDE = 0)	PRIO (3 bits)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	— (18 bits)	— (1 bit)
Extended (IDE = 1)	PRIO (3 bits)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

Because the PRIO field is the most significant part of the arbitration value, message buffers with low PRIO values have higher priority than message buffers with high PRIO values. This priority is maintained regardless of the rest of their arbitration values.

The PRIO field is not part of the frame on the CAN bus. Its purpose is only to affect the internal arbitration process.

39.3.3.3 Arbitration process (continued)

After the arbitration winner is found (see Arbitration process), its content is copied to a hidden auxiliary message buffer called a transmit serial message buffer (TX SMB). The TX SMB has the same structure as a normal message buffer, but is not user-accessible. This copy operation is called move-out. After it is done, write access to the Control and Status word of the corresponding MB is blocked (if MCR[AEN] = 1). Write access is restored in one of the following events:

- The CPU clears the corresponding IFLAG flag after the message buffer is transmitted.
- FlexCAN enters Freeze mode or Bus Off state.
- FlexCAN loses the bus arbitration, or there is an error during the transmission.

At the first opportunity window on the CAN bus, the message on the TX SMB is transmitted according to the CAN protocol rules.

The arbitration process can be triggered under the following conditions:

- During RX and TX frames from CAN CRC field to end of frame. The value of CTRL2[TASD] may be changed to optimize the arbitration start point.
- During CAN Bus Off state from TX_ERR_CNT = 124 to 128. The value of CTRL2[TASD] may be changed to optimize the arbitration start point.
- During Control and Status write by CPU in Bus Idle. The first Control and Status write starts the arbitration process, and a second Control and Status write during this same arbitration restarts the process. If other Control and Status writes are performed, the transmission arbitration process is pending. If there is no arbitration winner after the arbitration process has finished, then the TX arbitration machine begins a new arbitration process. If there is a pending arbitration and Bus Idle state starts, then an arbitration process is triggered. In this case, the first and second Control and Status writes in Bus Idle do not restart the arbitration process. If there is not enough time to finish arbitration in the Wait for Bus Idle state and the next state is Idle, the scan is not interrupted. That scan is completed during Bus Idle state. During this arbitration, a Control and Status write does not cause an arbitration restart.
- Deactivation of an arbitration winner during a valid arbitration window.
- Upon exiting Freeze mode (first bit of the Wait for Bus Idle state). If a resynchronization occurs during the Wait for Bus Idle state, the arbitration process is restarted.

The arbitration process stops when:

- All message buffers are scanned.
- A transmission message buffer is found (if lowest buffer feature is enabled).
- An arbitration winner deactivates or aborts during any arbitration process.
- There is not enough time to finish the transmission arbitration process (for instance, when a deactivation is performed near the end of frame). In this case, the arbitration process is pending.
- An error or overload flag occurs in the bus.
- A low-power or Freeze mode request occurs in Idle state.

Arbitration is considered pending when:

- It is not possible to finish arbitration process in time.
- A Control and Status write occurs during arbitration, when that write is performed in an MB whose number is lower than the transmission arbitration pointer.
- Any Control and Status write occurs when there is no transmission arbitration process in progress.
- RX Match has updated an RX code to TX code.
- FlexCAN enters the Bus Off state.

If a Control and Status write is performed in the arbitration winner, a new process is restarted immediately.

If a Control and Status write is performed in an MB whose number is higher than the transmission arbitration pointer, the ongoing arbitration process scans this MB as normal.

39.3.4 Receive process

To be able to receive CAN frames into a message buffer, the CPU must prepare it for reception by executing the following steps:

1. If the message buffer is active (either TX or RX), deactivate it (see Message buffer inactivation), preferably with a safe deactivation (see Transmission abort mechanism).

2. Write the ID word into the message buffer.
3. To activate the message buffer, write the EMPTY code (0100b) to the CODE field of the Control and Status word. No setup is required for EDL, BRS, and ESI fields. The respective fields in the received message overwrite these fields.

After the MB is activated, it can receive frames that match the programmed filter. At the end of a successful reception, the move-in process updates the message buffer (see Move-in) as follows:

1. The received data field (up to eight bytes for Classical CAN message format and up to 64 bytes for CAN FD message format) is stored.
2. The received Identifier field is stored.
3. The value of the free-running timer when the second bit of the Identifier field of the frame is written into the Timestamp field of the MB.
4. The received SRR, IDE, RTR, EDL, BRS, ESI, and DLC fields are stored.
5. The CODE field in the Control and Status word is updated (see Table 278 and Table 279 in Section Message buffer structure).
6. If allowed by the corresponding Interrupt Mask field, a status flag is set in the Interrupt Flag register and an interrupt is generated.

The recommended way for the CPU to service (read) the frame received in a message buffer is:

1. Read the Control and Status word of that message buffer.
2. Verify that the BUSY bit is 0, indicating that the message buffer is locked. If it is not 0, repeat step 1 until it becomes 0. See Message buffer lock mechanism.
3. Read the contents of the message buffer. After the message buffer is locked, FlexCAN move-in processes do not modify its contents. See Move-in.
4. Acknowledge the proper flag in the IFLAG registers.
5. Unlock the message buffer by reading the free-running timer.

To verify frame reception, the CPU should poll the status flag bit for that specific message buffer in one of the IFLAG registers, not the CODE field of that message buffer. Polling the CODE field does not work in this case. After a frame is received and the CPU services the message buffer (by reading the Control and Status word and unlocking the message buffer), the CODE field does not return to EMPTY. It remains FULL, as explained in Table 278. If the CPU tries to work around this behavior by writing to the Control and Status word to force an EMPTY code after reading the message buffer without a prior safe deactivation, a newly received frame matching the filter of that message buffer may be lost.

CAUTION

In summary: never poll by reading the Control and Status word of the message buffers directly. Instead, read the IFLAG registers.

The Identifier field of the received frame is always stored in the matching message buffer. If the match was due to masking, the contents of the ID field in a message buffer may change. When MCR[SRXDIS] becomes 1, FlexCAN does not store frames transmitted by itself in any MB, even if it contains a matching receive MB. Also, no interrupt flag or interrupt signal is generated. Otherwise, when MCR[SRXDIS] becomes 0, if a matching receive MB exists, FlexCAN can receive frames transmitted by itself.

When MCR[DMA]) becomes 1, upon receiving a frame in the Legacy FIFO, IFLAG1[BUF5I] generates a DMA request and does not generate a CPU interrupt (see Legacy RX FIFO in DMA operation). The IMASK1 fields in the Legacy RX FIFO region are not used.

The DMA controller must service the received frame using the following procedure:

1. Read the Control and Status word (read 80h address, optional).
2. Read the ID field (read 84h address, optional).
3. Read all data bytes (start read at 88h address, optional).

39.3.5 Matching process

The matching process scans the message buffer memory for RX MBs programmed with the same ID as the one received from the CAN bus. The matching starts from the lowest number message buffer and continues toward the higher-numbered ones.

For enhanced RX FIFO, see Enhanced RX FIFO.

For legacy RX FIFO, see Legacy RX FIFO.

As the frame is received, it is stored in a hidden auxiliary MB called Receive Serial Message Buffer (RX SMB).

The starting point of the matching process depends on the following conditions:

- If the received frame is a remote frame, the starting point is the CRC field of the frame.
- If the received frame is a data frame with DLC field equal to zero, the starting point is the CRC field of the frame.
- If the received frame is a data frame and the DLC field has a nonzero value, the starting point is the DATA field of the frame.

If a matching ID is found in one of the message buffers, the move-in process transfers the contents of the RX SMB to the matched MB. If any CAN protocol error is detected, no match results are transferred to the matched MB at the end of reception.

The matching process scans all matching elements of the active receive MBs (CODE is EMPTY, FULL, OVERRUN, or RANSWER). The process searches for a successful comparison to the matching elements of the RX SMB that is receiving the frame on the CAN bus. The RX SMB has the same structure as a message buffer. The reception structures (message buffers) associated with the matching elements that had a successful comparison are the matched structures. The matching winner is selected at the end of the scan among those matched structures. The matching winner depends on conditions described in Table 248.

Table 248. Matching architecture

Structure	SMB[RTR]	CTRL2[RRS]	CTRL2[EACEN]	MB[IDE]	MB[RTR]	MB[ID[1]]	MB[CODE]
Message buffer	0	—	0	cmp[2]	no_cmp[3]	cmp_msk[4]	EMPTY, FULL, or OVERRUN
Message buffer	0	—	1	cmp_msk	cmp_msk	cmp_msk	EMPTY, FULL, or OVERRUN
Message buffer	1	0	—	cmp	no_cmp	cmp	RANSWER
Message buffer	1	1	0	cmp	no_cmp	cmp_msk	EMPTY, FULL, or OVERRUN
Message buffer	1	1	1	cmp_msk	cmp_msk	cmp_msk	EMPTY, FULL, or OVERRUN

1. For message buffer structure, if SMB[IDE] is 1, the ID is 29 bits (ID Standard plus ID Extended). If SMB[IDE] is 0, the ID is 11 bits (ID Standard). For Legacy RX FIFO structure, the ID depends on IDAM.
2. cmp: Compares the RX SMB contents to the MB contents regardless of masks.
3. no_cmp: The RX SMB contents are not compared to the MB contents.
4. cmp_msk: Compares the RX SMB contents to MB contents, accounting for masks.

NOTE

For Enhanced RX FIFO, see Enhanced RX FIFO matching process.

A reception structure is free-to-receive when any of the following conditions is satisfied:

- The CODE field of the message buffer is EMPTY.
- The CODE field of the message buffer is either FULL or OVERRUN, and it has already been serviced (the CPU has read the Control and Status word and unlocked it as described in Message buffer lock mechanism).
- The CODE field of the message buffer is either FULL or OVERRUN and an inactivation (see Message buffer inactivation) is performed.

The scan order for message buffers is from the matching element with the lowest number to the higher ones.

39.3.5.1 Matching priority

MCR[IRMQ] affects the matching winner search for MBs. If the field is 0, the matching winner is the first matched MB whether it is free-to-receive or not. If it is 1, the matching winner is selected according to this priority:

1. The first free-to-receive matched message buffer
2. The last non-free-to-receive matched message buffer

39.3.5.2 Special cases

If a non-safe MB inactivation (see Message buffer inactivation) occurs during the matching process and the inactivated MB is the temporary matching winner, the temporary matching winner is invalidated. The matching elements scan is not stopped and not restarted; it continues normally. The consequence is that the current matching process works as if the matching elements compared before the inactivation did not exist. In this case, a message may be lost.

Consider an example where:

- MCR[IRMQ] is 1.
- There are two message buffers with the same ID: the second and fifth MBs in the array.
- FlexCAN starts receiving messages with that ID.

When the first message arrives, the matching algorithm finds the first match in message buffer number 2. The code of this message buffer is EMPTY, so the message is stored in that MB. When the second message arrives, the matching algorithm finds MB number 2 again, but it is not "free-to-receive." It continues looking, finds MB number 5, and stores the message in that MB. If yet another message with the same ID arrives, the matching algorithm finds no matching free-to-receive MBs, so it overwrites the last matched message buffer (MB number 5). In doing so, it updates the CODE field of the message buffer to OVERRUN.

The ability to match the same ID in more than one MB can be used to implement a reception queue to allow more time for the CPU to service the MBs. By programming more than one MB with the same ID, received messages are queued into the message buffers. The CPU can examine the Timestamp field of the message buffers to determine the order in which the messages arrived.

Matching a range of IDs is possible via ID acceptance masks. FlexCAN supports individual masking per message buffer (see Receive Individual Mask (RXIMR0 - RXIMR95)). During the matching algorithm, if a mask field is 1, the corresponding ID bit is compared. If the mask field is 0, the corresponding ID bit is a "don't care". Individual Mask Registers are implemented in RAM, so they are not initialized out of reset. Also, they can only be programmed when the module is in Freeze mode; otherwise, the module blocks them.

FlexCAN also supports an alternate masking scheme with only RX Message Buffers Global Mask (RXMGMASK), Receive 14 Mask (RX14MASK), and Receive 15 Mask (RX15MASK) for backward compatibility with legacy applications. This alternate masking scheme is enabled when the MCR[IRMQ] = 0.

39.3.6 Move process

There are two types of move process: move-in and move-out.

39.3.6.1 Move-in

The move-in process is the copying of a message received by an RX SMB to an RX message buffer that has matched it. Each RX SMB has its own move-in process, but only one is performed at a given time. The move-in starts only when the message held by the RX SMB has a corresponding match (see Matching process) and all of the following conditions are true:

- The CAN bus has reached or already gone past:
  - The second bit of Intermission field next to the frame that carried the message that is in the RX SMB.
  - The first bit of an overload frame next to the frame that carried the message in the RX SMB.
- There is no ongoing matching process.
- The CPU is not locking the destination message buffer.
- No move-in process from another RX SMB is ongoing. If more than one move-in process is to be started at the same time, both are performed and the newest process substitutes for the oldest.

The term "pending move-in" is used throughout the documentation and stands for a move-to-be that does not satisfy all of the above conditions.

If any of the following conditions is satisfied, the move-in is canceled and the RX SMB is able to receive another message:

- The destination message buffer is inactivated after the CAN bus has reached the first bit of the Intermission field next to the frame that carried the message. Also, its matching process has finished.
- There is a previous pending move-in to the same destination message buffer.
- The RX SMB is receiving a frame transmitted by FlexCAN itself and self-reception is disabled (MCR[SRXDIS] = 1).
- Any CAN protocol error is detected.

If the module enters Freeze or Low-Power mode, the pending move-in is not canceled. It remains on hold, waiting for Freeze and Low-Power mode to be exited and for the module to be unlocked. If a message buffer is unlocked during Freeze mode, the move-in occurs immediately.

The move-in process is FlexCAN executing the following steps:

1. Read all data words from the RX SMB in accordance with the selected payload size for the RX storage element.
2. Write all data words to the RX message buffer according to the selected payload size for the RX storage element. If the data size of the storage element is smaller than the original payload size described in the DLC field of the message, the payload is truncated. The high-order bytes that do not fit the destination size are lost.
3. Read the Control and Status and ID words from the RX SMB.
4. Write the Control and Status and ID words to the RX message buffer, and update the CODE field.

The move-in process is not atomic; the inactivation of the destination message buffer immediately cancels it (see Message buffer inactivation). In this case, the message buffer may remain partially updated, and therefore incoherent.

To alert the CPU that the message buffer content is temporarily incoherent, the BUSY Bit (least significant bit of the CODE field) of the destination message buffer becomes 1 during move-in.

39.3.6.2 Move-out

The move-out process is the copying of content from a TX message buffer to the TX SMB when a message for transmission is available (see Arbitration process). The move-out occurs in the following conditions:

- In the first bit of the Intermission field
- During the Bus Off state, when the TX error counter is in the 124 to 128 range
- During the Bus Idle state
- During the Wait for Bus Idle state

The move-out process is not atomic. Only the CPU has priority to access the memory concurrently outside the Bus Idle state. In Bus Idle, the move-out has the lowest priority of the concurrent memory accesses.

39.3.7 Data coherence

In order to maintain data coherency and proper FlexCAN operation, the CPU must obey the rules described in Transmission process and Receive process.

39.3.7.1 Transmission abort mechanism

The abort mechanism provides a safe way to request the abortion of a pending transmission. A feedback mechanism is provided to inform the CPU whether the transmission was aborted or the frame could not be aborted and was transmitted instead.

These primary conditions must be fulfilled in order to abort a transmission:

- MCR[AEN] must be 1.
- The first CPU action must be the writing of abort code (1001b) into the CODE field of the Control and Status word.

Active message buffers configured for transmission must be aborted before they can be updated. The write operation is blocked and the transmission is not disturbed when the abort code is written to:

- A message buffer currently being transmitted.
- A message buffer that was already loaded into the TX SMB for transmission.

In this case, the abort request is captured and kept pending until one of the following conditions is satisfied:

- The module loses the bus arbitration.
- There is an error during the transmission.
- The module is put into Freeze mode.
- The module enters the Bus Off state.
- There is an overload frame.

If none of the conditions above are reached:

1. The message buffer is transmitted correctly.
2. The interrupt flag is set in the proper IFLAG register.
3. If enabled, an interrupt to the CPU is generated.

The abort request is automatically cleared when the interrupt flag is set. If only one of the above conditions is reached, the frame is not transmitted. In this case:

1. The abort code is written into the CODE field.
2. The interrupt flag is set in the proper IFLAG register.
3. Optionally, an interrupt is generated to the CPU.

If the CPU writes the abort code before the transmission begins internally, the write operation is not blocked. The MB is updated and the interrupt flag is set. In this way, the CPU only needs to read the abort code to verify that the active MB was safely inactivated. In this case, although MCR[AEN] = 1 and the CPU wrote the abort code, the MB is inactivated and not aborted, because the transmission did not start yet. A message buffer is aborted only when the abort request is captured and kept pending until one of the previous conditions is satisfied.

39.3.7.2 Message buffer inactivation

Inactivation protects the message buffer against updates by FlexCAN internal processes. It allows the CPU to rely on message buffer data coherence after having updated it, even in Normal mode.

Inactivation of transmission message buffers must be performed only when MCR[AEN] = 0.

If a message buffer is inactivated, it does not participate in the arbitration process or the matching process until it is reactivated. See Transmission process and Receive process for detailed instructions on how to inactivate and reactivate a message buffer.

To inactivate a message buffer, the CPU must update its CODE field to INACTIVE (either 0b or 1000b).

Because you cannot synchronize the CODE field update with FlexCAN internal processes, an inactivation can have the following consequences:

- A frame in the bus that matches the filtering of an inactivated RX message buffer may be lost without notice. This loss can occur even if there are other message buffers with the same filter.
- A frame containing the message within an inactivated TX message buffer may be transmitted without setting the respective IFLAG flag.

To perform a safe inactivation and avoid the above consequences for TX message buffers, the CPU must use the transmission abort mechanism (see Transmission abort mechanism).

The inactivation automatically unlocks the message buffer (see Message buffer lock mechanism).

39.3.7.3 Message buffer lock mechanism

In addition to message buffer inactivation, FlexCAN uses a message buffer lock mechanism to maintain data coherence for the receive process. When the CPU reads the Control and Status word of an RX message buffer with codes FULL or OVERRUN, FlexCAN is configured to allow the CPU to read the whole message buffer in an atomic operation. FlexCAN sets an internal lock flag for that message buffer.

The lock is released in any of the following events:

- The CPU reads the free-running timer (global unlock operation).
- The CPU reads the Control and Status word of another message buffer, regardless of its code.
- The CPU writes into the Control and Status word. This procedure is not recommended for normal unlocking, because it cancels a pending move and may lose a received message.

The message buffer lock prevents a new frame from being written into the message buffer when the CPU is reading it.

NOTE

The locking mechanism applies only to RX message buffers that have a code other than INACTIVE (0b) or EMPTY[1] (0100b). TX message buffers cannot be locked.

Consider an example where:

- The second and the fifth message buffers of the array are programmed with the same ID.
- FlexCAN has already received and stored messages into these two message buffers.
- The CPU reads message buffer number 5 while another message with the same ID is arriving.

When the CPU reads the Control and Status word of message buffer number 5, this message buffer is locked. The new message arrives and the matching algorithm finds no free-to-receive message buffers, so it overrides message buffer number 5. This message buffer is locked, so the new message cannot be written to it. The message remains in the RX SMB until the message buffer is unlocked, and only then is it written to the message buffer.

If the message buffer remains locked and another new message with the same ID arrives, the new message overwrites the one in the RX SMB. There is no indication of lost messages in the CODE field of the message buffer or in Error and Status 1 (ESR1).

When the message is moved from the RX SMB to the message buffer, the BUSY bit on the CODE field becomes 1. If the CPU reads the Control and Status word and identifies that the BUSY bit is 1, it must wait until the BUSY bit becomes 0 to access the MB.

NOTE

If the BUSY bit is 1 or the message buffer is empty, reading the Control and Status word does not lock the message buffer.

Inactivation takes precedence over locking. If the CPU inactivates a locked receive message buffer, then its lock status is negated and the message buffer is marked as invalid for the current matching round. Any pending message on the RX SMB is not

[1] In previous FlexCAN versions, reading the Control and Status word locked the message buffer even when CODE indicates it is EMPTY. This behavior is maintained when the IRMQ bit is 0.

transferred to the message buffer. A message buffer is unlocked when the CPU reads Free-Running Timer (TIMER) or the Control and Status word of another message buffer.

The lock and unlock mechanisms have the same functionality in Normal and Freeze modes.

An unlock during Normal or Freeze mode results in the move-in of the pending message. If unlocking occurs during a low-power mode, however, the move-in is postponed (see Modes of operation). Move-in takes place only when the module returns to Normal or Freeze mode.

39.3.8 Enhanced RX FIFO

FlexCAN supports an enhanced RX FIFO engine which can store up to 20 CAN FD messages. The region 2000h–204Fh contains the output of the FIFO, which the CPU should read. To enable the enhanced RX FIFO, write 1 to ERFCR[ERFEN]. FlexCAN has two FIFO options, Legacy RX FIFO and Enhanced RX FIFO, but both options cannot be enabled at the same time. See Legacy RX FIFO for additional information.

To configure the enhanced RX FIFO watermark, write a value to ERFCR[ERFWM]. If ERFCR[ERFWM] is configured, the CPU is notified only if a minimum number of messages is stored in the FIFO. When the number of stored messages is greater than the value in ERFCR[ERFWM], the module sets ERFSR[ERFWMI]. Optionally, if MCR[DMA] or ERFIER[ERFWMIIE] are enabled, a DMA transfer or an interrupt can be triggered, respectively.

For the enhanced RX FIFO to receive, the CPU must execute the configuration procedure below. If the CPU must change any configurations of the Enhanced RX FIFO, the same procedure must be followed.

Prerequisites

MCR[RFEN] must be 0.

Procedure

Step	Purpose	Programming	Notes
1	Enter Freeze mode.	See Freeze mode.	—
2	If enhanced RX FIFO is not already enabled, enable it.	Write 1 to ERFCR[ERFEN].	—
3	Reset enhanced RX FIFO engine.	Write 1 to ERFSR[ERFCLR].	—
4	If the enhanced RX FIFO error flags are set, clear them.	Write 1 to these flags: • ERFSR[ERFUFW] • ERFSR[ERFOVF] • ERFSR[ERFWMI] • ERFSR[ERFDA]	—
5	Specify the total number of enhanced RX FIFO filter elements to be used in Enhanced RX FIFO reception.	Write the number to ERFCR[NFE].	—
6	Specify the number of extended ID and standard ID filter elements to be used in Enhanced RX FIFO reception.	Write the number to ERFCR[NEXIF].	ERFCR[NEXIF] ≤ ERFCR[NFE] + 1.
7	If you are using DMA, enable DMA.	Write 1 to MCR[DMA].	—
8	If you are using DMA, specify the number of words to transfer for each Enhanced RX FIFO data element.	Write the number to ERFCR[DMALW].	—

Table continues on the next page...

Table continued from the previous page...

Step	Purpose	Programming	Notes
9	Specify the Enhanced RX FIFO watermark.	Write the number to ERFCR[ERFWM].	If MCR[DMA] = 1, ERFCR[ERFWM] should be 0h.
10	If you are using interrupts, enable the interrupts.	Write 1 to the interrupt enables in Enhanced RX FIFO Interrupt Enable (ERFIER).	—
11	Configure the filter elements.	Write to the ERFFEL*n* registers.	ERFFEL*n* registers are implemented in RAM; you must explicitly initialize them before prior any reception.
12	Exit Freeze mode.	See Freeze mode.	—

There are two types of enhanced RX FIFO filter elements that can be stored in ERFFEL*n* registers: extended-ID filter elements and standard-ID filter elements. Each extended-ID filter element is stored in two ERFFEL*n* registers, and each standard-ID filter element is stored in one ERFFEL*n* register. ERFCR[NFE] defines the total number of Enhanced RX FIFO filter elements.

In addition, the filter memory space can be split into two regions: one for extended-ID filter elements and another for standard-ID filter elements, according to ERFCR[NEXIF]. Figure 146 shows how the enhanced RX filter elements are defined. See Enhanced RX FIFO matching process for information about the Enhanced RX FIFO matching process and filter element formats.

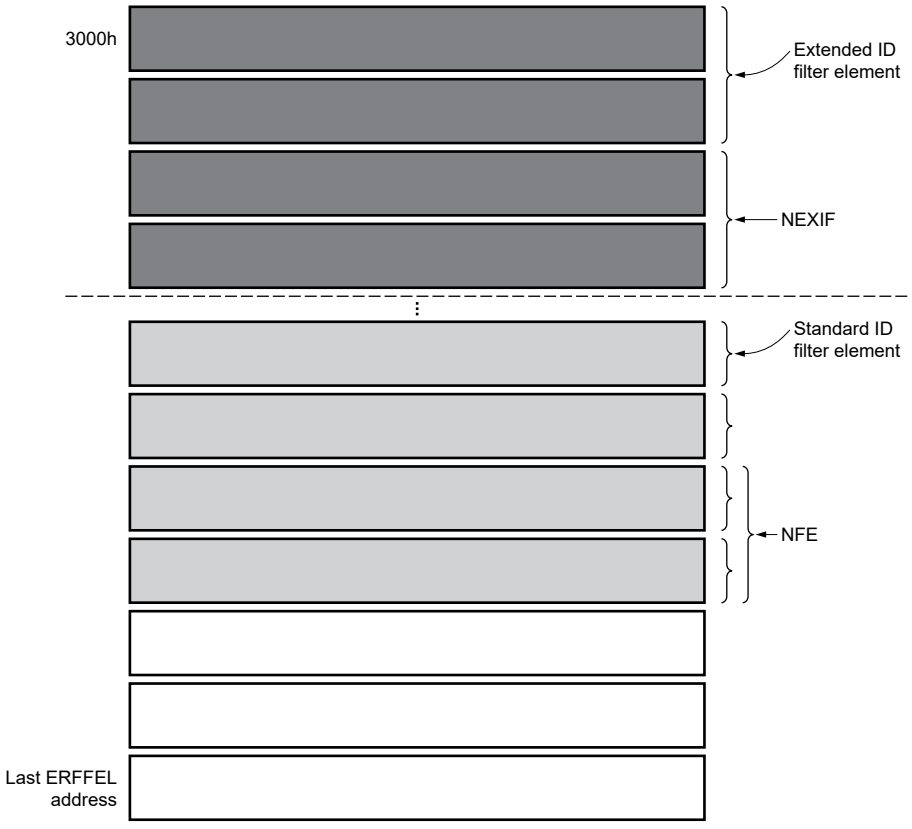


Figure 146. Number of enhanced RX FIFO filter elements

39.3.8.1 Enhanced RX FIFO matching process

When ERFCR[ERFEN] = 1, FlexCAN scans the ERFFEL*n* memory region. If at least one filter element satisfies the matching criteria, the CAN message content is transferred to the enhanced RX FIFO memory. If multiple filters match the incoming message ID, the first matching filter found by the matching process must be indicated in IDHIT.

Each ERFFEL*n* register can store one standard filter element. ERFFEL*n*[FEL][31:30], also called FSCH, determines the matching criteria in this way:

- If FSCH = b00, the filter scheme is based on mask and filter. A CAN message matches a standard ID filter element only if these criteria are reached:
  1. CAN message is base-frame format (IDE = 0).
  2. (ID[*n*] = STD ID filter [*n*]) or (STD ID Mask[*n*] = 0) for each bit *n* from 0 to 10.
  3. (RTR = RTR Filter) or (RTR MASK = 0).

In this explanation, RTR and ID are the Remote Transmit Request field and the ID from a CAN message, respectively.

If FSCH = b00, the filters and masks are defined as shown in Table 249.

Table 249. Standard ID filter element with filter and mask scheme (FSCH = b00)

31	30	29	28	27	26	16	15	12	11	10	0
FSCH = b00		Reserved		RTR Filter	STD ID Filter		Reserved		RTR MASK	STD ID MASK	

- If FSCH = b01, the filter scheme is based on range. A CAN message matches a standard ID filter element only if these criteria are reached:
  1. CAN message is base frame format (IDE = 0).
  2. ID ≥ STD ID Filter1.
  3. ID ≤ STD ID Filter2.
  4. (RTR = RTR filter) or (RTR MASK = 0).

RTR and ID are the Remote Transmit Request bit and ID from a CAN message, respectively. If FSCH = b01, the filters and mask are defined as shown in Table 250.

Table 250. Standard ID filter element with range scheme (FSCH = b01)

31	30	29	28	27	26	16	15	12	11	10	0
FSCH = b01		Reserved		RTR Filter	STD ID Filter2		Reserved		RTR MASK	STD ID Filter1	

- If FSCH = b10, the filter scheme is based on two filters without masks. A CAN message matches a standard ID filter element only if these criteria are reached:
  1. CAN message is base frame format (IDE = 0).
  2. (ID[*n*] = STD ID Filter1[*n*]) or (ID[*n*] = STD ID Filter2[*n*]) for each bit *n* from 0 to 10.
  3. (RTR = RTR Filter1) or (RTR = RTR Filter2).

RTR and ID are the Remote Transmit Request bit and ID from a CAN message, respectively. If FSCH = b10, the filters are defined as shown in Table 251.

Table 251. Standard ID filter element with two-filter scheme (FSCH = b10)

31	30	29	28	27	26	16	15	12	11	10	0
FSCH = b10		Reserved		RTR Filter 2	STD ID Filter2		Reserved		RTR Filter 1	STD ID Filter1	

Each pair of ERFFEL*n* registers can store one extended filter element. ERFFEL*n*[FSCH] determines the matching criteria in this way:

- If FSCH = b00, the filter scheme is based on mask and filter. A CAN message matches an extended ID filter element only if these criteria are reached:
  1. CAN message is extended frame format (IDE = 1).
  2. (ID[*n*] = EXT ID filter [*n*]) or (EXT ID Mask[*n*] = 0) for each bit *n* from 0 to 28.
  3. (RTR = RTR Filter) or (RTR MASK = 0).

If FSCH = b00, the filters and masks are defined as shown in Table 252.

Table 252. Extended ID filter element with filter + mask scheme (FSCH = b00)

31	30	29	28	0
FSCH		RTR Filter	EXT ID Filter	
Reserved		RTR MASK	EXT ID MASK	

- If FSCH = b01, the filter scheme is based on range. A CAN message matches an extended ID filter element only if the following criteria are reached:
  1. CAN message is extended frame format (IDE = 1).
  2. ID ≥ EXT ID Filter1.
  3. ID ≤ EXT ID Filter2.
  4. (RTR = RTR Filter) or (RTR MASK = 0).

If FSCH = b01, the filters and masks are defined as shown in Table 253.

Table 253. Extended ID filter element with range scheme (FSCH = b01)

31	30	29	28	0
FSCH		RTR Filter	EXT ID Filter2	
Reserved		RTR MASK	EXT ID Filter 1	

- If FSCH = b10, the filter scheme is based on two filters without masks. A CAN message matches an extended ID filter element only if these criteria are reached:
  1. CAN message is extended frame format (IDE = 1).
  2. (ID[*n*] = EXT ID Filter1[*n*]) or (ID[*n*] = EXT ID Filter2[*n*]) for each bit *n* from 0 to 28.
  3. (RTR = RTR Filter1) or (RTR = RTR Filter2).

If FSCH = b10, the filters are defined as shown in Table 254.

Table 254. Extended ID filter element with two-filter scheme (FSCH = b10)

31	30	29	28	0
FSCH		RTR Filter2	EXT ID Filter2	
Reserved		RTR Filter1	EXT ID Filter 1	

39.3.8.2 Enhanced RX FIFO under DMA operation

You can enable the DMA feature by writing 1 to both ERFCR[ERFEN] and MCR[DMA]. The DMA controller can read the received message by reading a message buffer structure at the enhanced FIFO output port at the address range defined in Enhanced RX FIFO structure.

NOTE

FlexCAN supports 32-bit access only for DMA transfers.

For proper FIFO engine operation, the CPU should not access the Enhanced FIFO output port address range during DMA operation. Before writing 1 to MCR[DMA], the CPU must service Enhanced RX FIFO status bits. Otherwise, these bits may show that the FIFO has data to be serviced, and mistakenly generate a DMA request. Before writing 0 to MCR[DMA], the CPU must first clear the ERFSR[ERFUFW], ERFSR[ERFOVF], ERFSR[ERFWMI], and ERFSR[ERFDA] flags. It must then clear the enhanced RX FIFO engine by writing one to ERFSR[ERFCLR].

When there is one frame available to be read from the Enhanced RX FIFO, FlexCAN sets ERFSR[ERFDA]. Upon receiving the request, the DMA controller can read the message in the Enhanced RX FIFO output. Each message reading process must end by the address defined in ERFCR[DMALW].

Follow these rules for Enhanced RX FIFO DMA operation:

- Because a DMA transfer cannot be changed dynamically, program ERFCR[DMALW] so the enhanced RX FIFO element can store the largest CAN message present on the CAN bus.
- Data bytes are valid according to the DLC field. See Table 282.

Each time the DMA controller reads one message from the FIFO, FlexCAN clears ERFSR[ERFDA]. If there is at least one message stored in the FIFO, FlexCAN sets it again.

Consider an example where the maximum number of bytes in the data field of a CAN frame for a certain application is eight, and high-resolution timestamp is enabled. In that case, the last enhanced RX FIFO address offset can be found in Table 292 and Table 293. Using this address offset, ERFCR[DMALW] can be determined in this way:

- Maximum number of data bytes = 8
- HR_TIME_STAMP enabled
- Last address offset = TS_OFF = 2014h
- DMALW = 5

39.3.8.3 Enhanced RX FIFO clear operation

When ERFCR[ERFEN] is 1, the CPU can clear the Enhanced RX FIFO by writing 1 to ERFSR[ERFCLR]. The clear operation resets the internal FIFO pointers, but the FIFO content stored in RAM is not changed. This operation can only be performed in Freeze mode; the module blocks the operation in other modes. This operation does not clear ERFSR[ERFUFW], ERFSR[ERFOVF], ERFSR[ERFDA], or ERFSR[ERFWMI]. The CPU must service all these fields before executing the clear FIFO operation.

39.3.9 Legacy RX FIFO

The Legacy RX FIFO is receive-only. To enable it, write 1 to MCR[RFEN]. To maintain software backward compatibility with previous versions of FlexCAN that did not have the Legacy FIFO feature, the reset value of this field is zero.

CAUTION

Do not enable Legacy RX FIFO when the CAN FD feature is enabled.

The Legacy FIFO is six messages deep. The memory region the Legacy FIFO structure occupies (both message buffers and Legacy FIFO engine) is described in Legacy RX FIFO structure. The CPU can read the received messages sequentially, in the order they were received, by repeatedly reading a message buffer structure at the output of the Legacy FIFO.

IFLAG1[BUF5I] (Frames Available in Legacy RX FIFO) is set when at least one frame is available to be read from the Legacy FIFO. If the corresponding mask bit enables it, an interrupt is generated. Upon receiving the interrupt, the CPU can read the message (accessing the output of the Legacy FIFO as a message buffer) and Legacy RX FIFO Information (RXFIR), then clear the interrupt. If there are more messages in the Legacy FIFO, clearing the interrupt:

1. Updates the output of the Legacy FIFO with the next message.
2. Updates RXFIR with the attributes of that message.
3. Reissues the interrupt to the CPU.

Otherwise, the flag remains cleared. The output of the Legacy FIFO is valid only when IFLAG1[BUF5I] is set.

IFLAG1[BUF6I] (Legacy RX FIFO Warning) is set when the Legacy RX FIFO receives a new message that increases the number of unread messages from four to five. This change means that the Legacy RX FIFO is almost full. The flag remains set until the CPU clears it.

IFLAG1[BUF7I] (Legacy RX FIFO Overflow) is set when an incoming message is lost because the Legacy RX FIFO is full. The flag is not set when the Legacy RX FIFO is full and a message buffer captures the message. The flag remains set until the CPU clears it.

Clearing one of the three flags above does not affect the state of the other two.

If an IFLAG flag is set and the corresponding mask bit is 1, an interrupt is generated.

A powerful filtering scheme is provided to accept only frames intended for the target application, reducing the interrupt servicing workload. The filtering criteria are specified by programming a table of up to 128 32-bit registers, according to CTRL2[RFFN]. This table can be configured to one of the following formats (see also Legacy RX FIFO structure):

- Format A: 128 Identifier Acceptance Filters (IDAFs) — extended or standard IDs including IDE and RTR
- Format B: 256 IDAFs — standard IDs or extended 14-bit ID slices including IDE and RTR
- Format C: 512 IDAFs — standard or extended 8-bit ID slices

NOTE

A chosen format is applied to all entries of the filter table. It is not possible to mix formats within the table.

Every frame available in the Legacy RX FIFO has a corresponding Identifier Acceptance Filter Hit Indicator (IDHIT). The IDHIT can be read in the IDHIT field in the Control and Status word, as shown in the Legacy RX FIFO Structure description. The CPU can also obtain this information by accessing Legacy RX FIFO Information (RXFIR). RXFIR[IDHIT] refers to the message at the output of the Legacy FIFO, and is valid when IFLAG1[BUF5I] is set. RXFIR must be read only before clearing the flag, guaranteeing that the information refers to the correct frame within the Legacy FIFO.

The Individual Mask Registers (RXIMR*n*) individually affect up to 32 elements of the filter table, according to the value of CTRL2[RFFN]. This configuration allows very powerful filtering criteria to be defined. If MCR[IRMQ] is 0, the Legacy RX FIFO filter table is affected by Legacy RX FIFO Global Mask (RXFGMASK).

NOTE

See Table 242 for information about the difference between FD and non-FD regarding this feature.

39.3.9.1 Legacy RX FIFO in DMA operation

The receive-only Legacy FIFO can support DMA. To enable this feature, write 1 to both MCR[RFEN] and MCR[DMA]. To maintain backward compatibility with previous versions of the module that did not have the DMA feature, the reset value of MCR[DMA] is zero.

The DMA controller can read the received message by reading a message buffer structure at the Legacy FIFO output port in the 80h–8Ch address range.

NOTE

FlexCAN supports 32-bit access only for DMA transfers.

When MCR[DMA] = 1, the CPU must not access the Legacy FIFO output port address range. Before writing 1 to MCR[DMA], the CPU must service the IFLAG flags set in the Legacy RX FIFO region. Otherwise, these flags may indicate that the FIFO has data to be serviced, and mistakenly generate a DMA request. Before writing 0 to MCR[DMA], the CPU must perform a clear Legacy FIFO operation.

When at least one frame available to be read from the FIFO, IFLAG1[BUF5I] (Frames available in Legacy RX FIFO) is set. A DMA request is generated simultaneously. Upon receiving the request, the DMA controller can read the message (accessing the output of the Legacy FIFO as a message buffer). The DMA reading process must end by reading address 8Ch. This read operation:

- Clears IFLAG1[BUF5I].
- Updates the FIFO output with the next message (if the FIFO is not empty).
- Updates Legacy RX FIFO Information (RXFIR) with the attributes of the new message.

If there are more messages stored in the FIFO, IFLAG1[BUF5I] is reasserted and another DMA request is issued. Otherwise, the flag remains cleared.

IFLAG1[BUF6I] and IFLAG1[BUF7I] are not used when the DMA feature is enabled.

When FlexCAN is working with DMA, the CPU does not receive any Legacy RX FIFO interruption and must not clear the related IFLAG flags. The related IMASK bits are not used to mask the generation of DMA requests.

NOTE

See Table 242 for information about the difference between FD and non-FD regarding this feature.

39.3.9.2 Clear legacy FIFO

When MCR[RFEN] = 1, you can use the clear Legacy FIFO operation to empty Legacy FIFO contents. When the CPU writes 1 to IFLAG1[BUF0I], the clear FIFO operation occurs. This operation can only be performed in Freeze mode; FlexCAN blocks it in other modes. This operation does not clear the FIFO IFLAG flags; the CPU must service all FIFO IFLAG flags before executing the clear FIFO operation.

When Legacy RX FIFO is working with DMA, the clear FIFO operation clears IFLAG1[BUF5I], and the DMA request is canceled.

CAUTION

The clear Legacy FIFO operation does not clear IFLAG flags, except when MCR[DMA] = 1; in this case, only IFLAG1[BUF5I] is cleared.

39.3.10 Fault reaction

This situation is catastrophic, and in response to the fault reaction, FlexCAN goes into a safe state (Freeze) and drives a safe value (recessive bit) on the bus. Either CTRL2[FLT_RXN] or an input "fault reaction" signal from another module can trigger a fault reaction. FlexCAN remains in a safe state as long as CTRL2[FLT_RXN] = 1 or the "fault reaction" signal is asserted and MCR[FRZ] = 1.

A fault can occur at any time, including during frame transmission, reception, and BUS IDLE. If a fault occurs during frame transmission, FlexCAN stops the frame transmission and goes into a safe state.

If a fault occurs during frame reception, FlexCAN stops receiving the frame and goes into a safe state. If FlexCAN is in loopback or self-reception mode, the effect of the fault reaction is the same as described above.

When the fault condition is removed, FlexCAN remains in Freeze mode until the CPU writes 0 to MCR[FRZ].

39.3.11 Retransmission

If a message transmission is unsuccessful due to a reason other than a loss of arbitration or being aborted by the host, FlexCAN attempts to retransmit the message a number of times determined by the value of CTRL2[RETRY].

When a message transmission is unsuccessful, a counter is decremented. When that counter reaches zero, FlexCAN aborts the message. FlexCAN sets the IFLAG for the aborted message and an interrupt is generated (if enabled). FlexCAN sets the IFLAG and updates the CODE field of the control and status word by abort code. The host can read the IFLAG flag and CODE field to identify which message FlexCAN aborted due to retransmission attempts greater than the value of CTRL2[RETRY].

39.3.12 CAN protocol-related features

39.3.12.1 CAN FD ISO compliance

The CAN FD protocol has been improved to increase the failure-detection capability that was in the original CAN FD protocol. This original protocol is also called non-ISO CAN FD, by CAN in Automation (CiA). A three-bit stuff counter and a parity bit have been introduced in the improved CAN FD protocol, now called ISO CAN FD. The CRC calculation has also been modified. All these improvements make the ISO CAN FD protocol incompatible with the non-ISO CAN FD protocol. FlexCAN still supports non-ISO CAN FD, so it can be used during an intermediate phase, for evaluation and development purposes.

It is recommended that you configure FlexCAN with the ISO CAN FD protocol by writing 1 to CTRL2[ISOCANFDEN].

39.3.12.2 CAN FD frames

ISO 11898-1_2024 specifies the Classical Frame format compliant to ISO 11898-1:2003 (2003) and introduces the CAN Flexible Data Rate Frame format (CAN FD). The Classical Frame format allows bit rates up to one Mbit/s and payloads up to eight bytes per frame. The Flexible Data Rate Frame format allows bit rates faster than one Mbit/s and payloads longer than eight bytes per frame. FlexCAN can receive and transmit CAN FD messages interleaved with Classical CAN messages.

There are additional control bits in the CAN FD frame:

- The Extended Data Length (EDL) bit enables a longer data payload with different data length coding.
- The Bit Rate Switch (BRS) bit decides whether the bit rate is switched inside a CAN FD format frame.
- The Error State Indicator (ESI) flag is transmitted dominant by error active nodes, and recessive by error passive nodes.

There are no Remote Frames (see Remote frames) in the CAN FD format. A message configured to transmit a Remote Frame is always sent out in Classical CAN format. When an FD frame is received and matches a message buffer, the RTR bit in the receiving message buffer becomes 0. The RTR bit must be considered in classical frames only.

39.3.12.2.1 CAN FD messages

CAN FD messages may be formatted as long frames where the data field exceeds eight bytes, and may range from 12 up to 64 bytes. They can also be configured to support bit rate switching. In this case, the control field, data field, and CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and end of the frame. Messages in the classical CAN format are limited to transport a maximum payload of eight bytes at nominal rate. Figure 147 illustrates the message formats for classical and FD frames with either standard or extended ID.

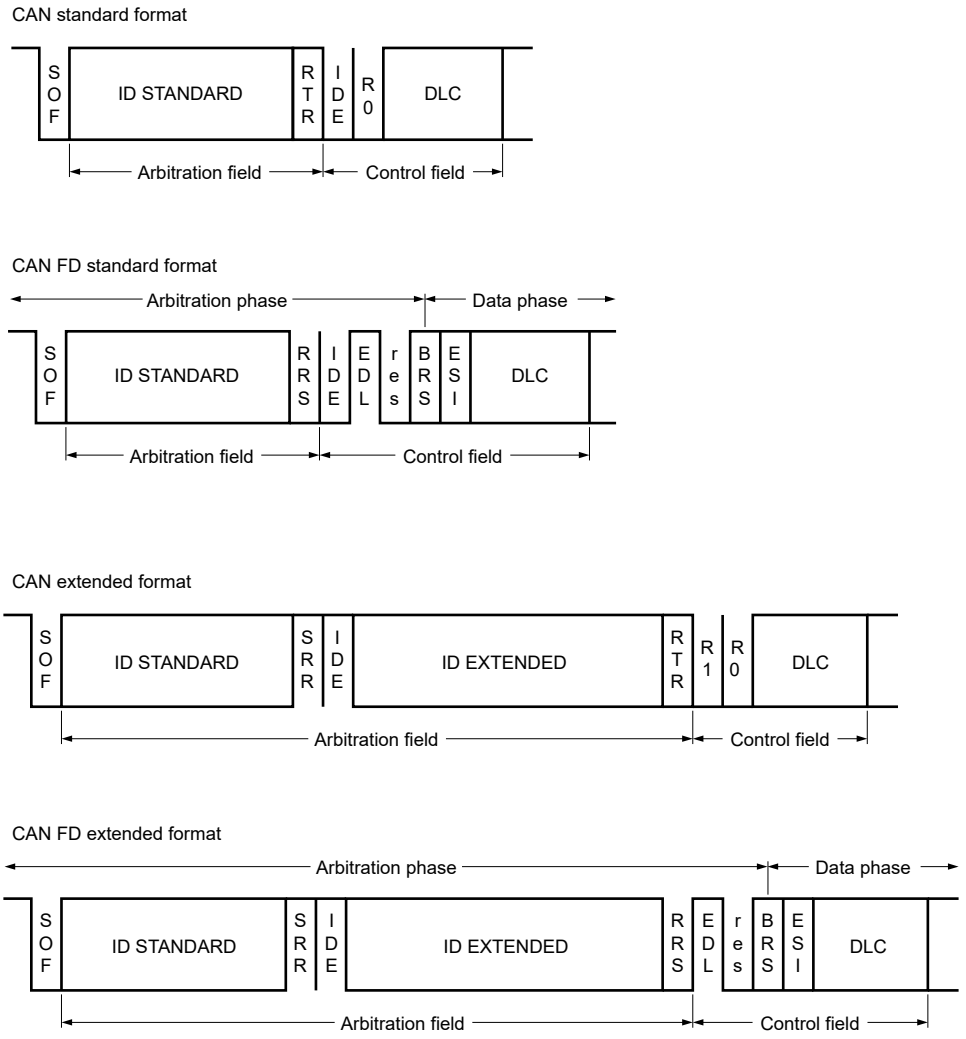


Figure 147. CAN message formats

MCR[FDEN] enables the ability to receive and transmit CAN FD messages. A recessive R0 bit in CAN frames with 11-bit identifiers, or a recessive R1 bit in CAN frames with 29-bit identifiers, is decoded as an EDL bit (not a reserved one). A recessive EDL bit identifies a CAN FD frame, and a dominant EDL bit identifies a Classical CAN frame. The BRS bit specifies whether this frame switches the bit rate in its data phase. A long frame is decoded according to the DLC field value (see DLC definition in Message buffer structure).

CAN FD messages can be transmitted with two different bit rates. The first part of a CAN FD frame, from the Start of Frame (SOF) bit until the Bit Rate Switch (BRS) bit is called the arbitration phase. This part is transmitted with the nominal bit rate based on a set of nominal CAN bit timing configuration values. The second part, from the BRS bit until the CRC Delimiter bit, is called the data phase. When this second part is transmitted, a second set of CAN data bit timing configuration values determines the data bit rate. Finally, from the CRC delimiter until the Intermission bits, the transmission returns to nominal bit rate.

39.3.12.2.2 BRS in CAN FD

In CAN FD frames with bit rate switching, the bit timing changes inside the frame at the sample point of the BRS bit if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by CAN Bit Timing (CBT). (Control 1 (CTRL1) also defines this timing for backward compatibility.) Upon detecting a recessive BRS bit, the CAN data bit timing is used as defined by CAN FD Bit Timing (FDCBT).

NOTE

If the time quantum length in nominal bit timing and in the data bit timing are not identical, a quantization error of up to one time quantum of the arbitration phase may be present as a phase error. This situation can occur after the switch from arbitration to data phase, and it lasts until the next synchronization event. The length of the time quantum should be the same in nominal and data bit timing. This configuration minimizes the chance of error frames on the CAN bus, and optimizes the clock tolerance in networks that use FD frames.

If BRS = 1 in the selected TX MB, FDCTRL[FDRATE] enables the transmission of all frames with bit rate switching. If FDCTRL[FDRATE] = 0, the transmission is performed at nominal rate regardless of the BRS bit value. FDCTRL[FDRATE] can be written at any time but takes effect only for the next message transmitted or received.

Nominal bit timing is resumed at the sample point of the CRC Delimiter bit or when an error is detected, whichever occurs first. Figure 148 describes the mechanism for entering and leaving the data phase when the BRS bit is recessive.

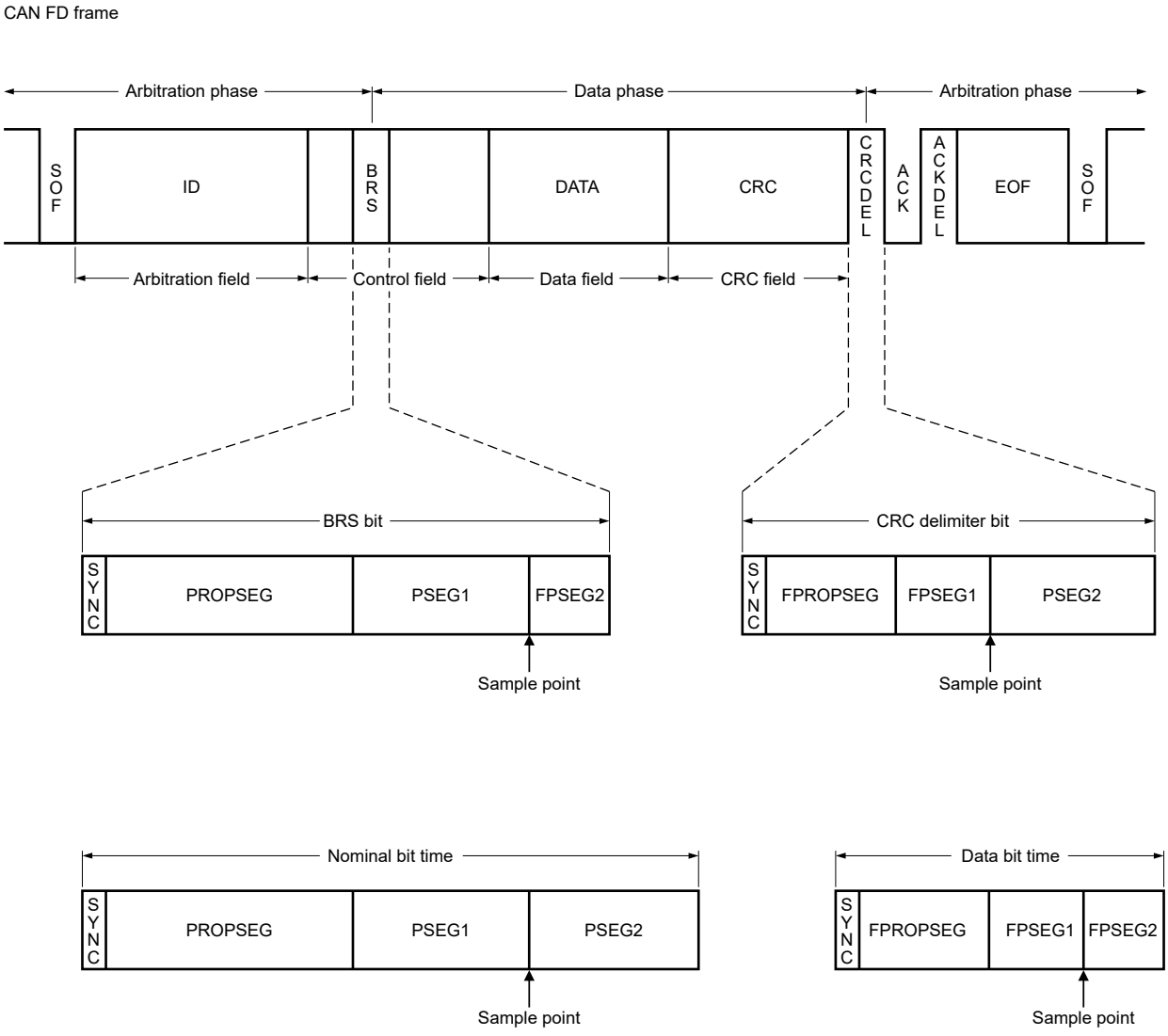


Figure 148. Bit rate switching mechanism for CAN FD messages

NOTE

In Classical CAN frames, the CRC delimiter is one recessive bit. In CAN FD frames, the CRC delimiter may consist of one or two recessive bits. FlexCAN sends only one recessive bit as the CRC delimiter. It accepts two recessive bits before the recessive-to-dominant edge that starts the acknowledge slot. As a receiver, FlexCAN sends its acknowledge bit after the first CRC delimiter bit. In CAN FD frames, FlexCAN accepts a two-bit dominant ACK slot as a valid ACK to compensate for phase shifts between the receivers.

The maximum configurable bit rate in the CAN FD data phase depends on the clock frequency of the CAN_PE subblock. For example, for a CAN_PE clock frequency of 40 MHz and the shortest configurable bit time of 5 time quanta, the bit rate in the data phase is 8 Mbit/s.

NOTE

The frequency used in this example may not be supported on this chip. It is shown only to demonstrate how the maximum configurable bit rate is calculated.

39.3.12.2.3 ESI in CAN FD

The value of the ESI field is determined by either of the following:

- The error state of the transmitter at the start of the transmission, if the frame is originated in the FlexCAN node
- The original transmitting node when FlexCAN is acting as a gateway for the message

If the transmitter is error-passive, ESI is transmitted recessive; otherwise, it is transmitted dominant. The permutations of the relationship between the written value and the transmitted value of the ESI are shown in Table 255.

Table 255. Written versus transmitted values of the ESI field

FlexCAN fault confinement status at start of frame	ESI field of TX MB	Transmitted ESI
Error active	0	0 (Error Active)
Error passive	0	1 (Error Passive)
Error active	1	1 (Error Passive)
Error passive	1	1 (Error Passive)

39.3.12.2.4 CRC calculations in CAN FD

Different CAN frame formats have different CRC polynomials. The first polynomial, CRC_15, is used for all frames in Classical CAN format. The second, CRC_17, is used for frames in CAN FD format with a data field up to 16 bytes long. The third, CRC_21, is used for frames in CAN FD format with a data field longer than 16 bytes. Each polynomial results in a Hamming distance of 6. At the start of the frame, all three CRC polynomials are calculated concurrently. The values of the EDL bit and the DLC field select the CRC sequence to be transmitted. When receiving a message, FlexCAN decodes EDL and DLC to select the adequate CRC polynomial to check for a CRC error.

In CAN FD format frames, stuff bits are included in the bit stream for CRC calculation. In Classical CAN format frames, stuff bits are not included. After the transmission of the last bit relevant to the CRC calculation, CAN FD CRC (FDCRC) stores the calculated CRC for the transmitted message. This storage is performed with adequate length for the type of message, for CAN FD and non-FD messages. Cyclic Redundancy Check (CRCR) reports a valid CRC for Classical CAN messages only.

In CAN FD format frames, the CAN bit stuffing method changes for the CRC sequence, so the stuff bits are inserted at fixed positions. When FlexCAN is transmitting a CAN FD frame, a fixed stuff bit is inserted just before the first bit of the CRC sequence. This insertion occurs even if the last bits of the preceding field do not fulfill the CAN stuff condition. Additional stuff bits are inserted after each fourth bit of the CRC sequence. The value of any fixed stuff bit is the inverse value of its preceding bit. When FlexCAN receives a CAN FD frame, it discards the fixed stuff bits from the bit stream for the CRC check. A stuff error is detected if the fixed stuff bit has the same value as its preceding bit.

39.3.12.2.5 CAN FD errors

FlexCAN detects errors in CAN FD frames the same way as in Classical CAN frames. The error counters ECR[RXERRCNT] and ECR[TXERRCNT] accumulate the counts of RX and TX errors, respectively, for both FD and non-FD frames indiscriminately. Two extra error counters, ECR[RXERRCNT_FAST] and ECR[TXERRCNT_FAST], accumulate RX and TX errors occurring in the data phase of CAN FD frames with BRS = 1 only. The rules for updating the error counters are the same for both CAN FD and non-FD frames (see Error Counter (ECR)).

These error flags report errors in both CAN FD and non-FD frames:

- ESR1[BIT1ERR]
- ESR1[BIT0ERR]
- ESR1[ACKERR]
- ESR1[CRCERR]
- ESR1[FRMERR]
- ESR1[STFERR]

If CTRL1[ERRMSK] = 1, they also generate the ERRINT interrupt.

These additional error flags indicate the occurrence of errors in the data phase of CAN FD frames with BRS = 1:

- ESR1[BIT1ERR_FAST]
- ESR1[BIT0ERR_FAST]
- ESR1[CRCERR_FAST]
- ESR1[FRMERR_FAST]
- ESR1[STFERR_FAST]

No ACKERR is detected in the data phase of a CAN FD frame. Fault confinement status reported in ESR1[FLTCONF] is the same for both CAN FD and Classical CAN frames, and is based on ECR[RXERRCNT] and ECR[TXERRCNT] only. Information in ECR[RXERRCNT_FAST] and ECR[TXERRCNT_FAST] may be considered as status to help detect the error nature related to the bit rate value.

When FlexCAN detects an error while transmitting or receiving a CAN FD message in the data phase, it immediately switches:

- Back to the arbitration phase, and
- Back to the nominal rate to start an error flag.

39.3.12.2.6 CAN FD synchronization

Resynchronization and hard synchronization occur in CAN FD frames in the same way as in Classical CAN ones. A hard synchronization is also performed at the recessive-to-dominant edge from EDL to R0 in CAN FD format frames. FlexCAN does not resynchronize when transmitting in the CAN FD data phase.

39.3.12.3 Transceiver delay compensation

The CAN FD protocol allows the transmission and reception of data at a higher bit rate than the nominal rate used in the arbitration phase, when BRS = 1 in the message. This feature enables the use of rates up to 8 Mbit/s.

During the data phase of a CAN FD frame, if the transmitter cannot receive its own latest transmitted bit at the sample point of that bit, it detects a bit error. When bit rate switching is enabled (BRS = 1), the CAN bit time in the data phase can become shorter than the loop delay of the transceiver. This condition impedes the correct comparison between the transmitted bit and the received bit within the current CAN bit time interval.

The transceiver delay compensation (TDC) process defines a secondary sample point where the transmitted bit is correctly compared to the received bit to check for bit errors.

You can enable the TDC mechanism via FDCTRL[TDCEN] or ETDC[ETDCEN]. The TDC mechanism is effective only during the data phase of FD frames with BRS = 1. It has no effect on either non-FD frames or FD frames transmitted at the normal bit rate. When the transmitted message has BRS = 1, TDC is active from the sample point of the BRS bit until the sample point of the CRC Delimiter bit. When TDC is active, the real received bit is compared to the delayed transmitted bit, where the delay is calculated based on the measured transceiver loop delay.

NOTE

The transmitters using TDC disregard the value of the CRC delimiter bit. A global error at the end of the CRC field causes the receivers to send error frames that the transmitter detects during Acknowledge or End of Frame.

For every transmitted FD frame with BRS = 1, the transition from the recessive EDL bit to the dominant R0 bit triggers the delay measurement (as shown in Figure 149). The loop delay is measured in Protocol Engine (PE) clock periods (CANCLK, see Protocol timing), from the transmitted EDL-R0 edge to the received EDL-R0 edge. The measured loop delay time added to an offset value specified in FDCTRL[TDCOFF] or ETDC[ETDCOFF] determines the position of the secondary sample point. FDCTRL[TDCVAL] or ETDC[ETDCVAL] stores the result of this calculation. The TDCVAL and ETDCVAL value saturates at its maximum value of 63 CANCLK and 255 CANCLK when the delay measurement is too long.

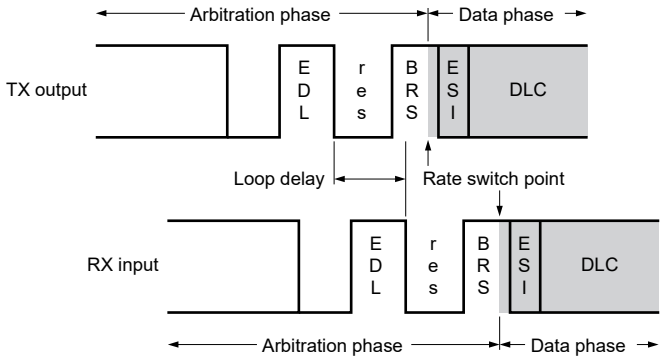


Figure 149. Transceiver loop delay measurement

The measured loop delay is not enough to define the secondary sample point, because it relates to the CAN bit edges. The transceiver delay compensation offset FDCTRL[TDCOFF] or ETDC[ETDCOFF] is used to shift the secondary sample point to an intermediate point inside the bit time, far away from its edges. The value of FDCTRL[TDCOFF] or ETDC[ETDCOFF] cannot be larger than the CAN bit duration in the data phase.

If the secondary sample point is set very near the CAN bit edge (SYNC field), problems may occur during the bit sampling in the data phase. For the TDC to work reliably, the offset must use optimal settings. To ensure that bit sampling is performed in the best region, configure the TDC offset as shown in this equation:

$$Offset = (FPSEG1 + FPROPSEG + 2) \times (FPRESDIV + 1)$$
or
$$Offset = (DTSEG1 + 2) \times (EDPRESDIV + 1), \textit{ if ETDCEN}$$

Equation 18. TDC offset calculation

Figure 150 shows the SSP position when these settings are used.

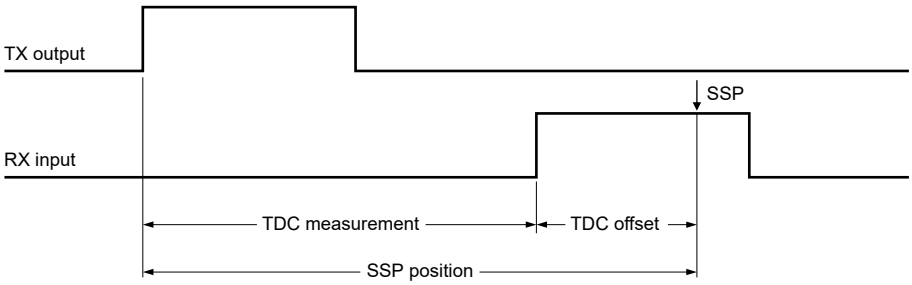


Figure 150. SSP position with optimal values

Alternatively, if CTRL2[BTE] and ETDC[ETDCEN] are 1, you can write 1 to ETDC[TDMDIS] to disable the transceiver delay measurement. In this case, only ETDC[ETDCOFF] defines the SSP position. Figure 151 shows the secondary sample point position when the transceiver delay measurement is disabled.

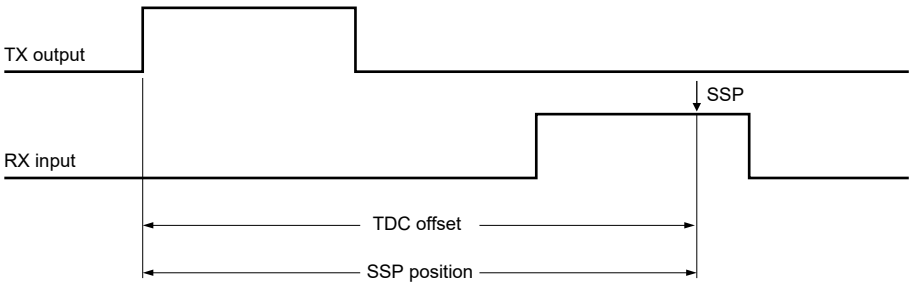


Figure 151. SSP position with TDC measurement disabled by ETDCEN and TDMDIS

During the data phase of CAN FD frames with bit rate switching enabled, at the onset of every TX CAN bit:

- The transmitted TX bit value is temporarily stored in a buffer.
- A time countdown based on FDCTRL[TDCVAL] or ETDC[ETDCVAL] is started. This countdown ends with the comparison of the received RX bit (delayed by the external loop delay plus the specified offset) to the stored TX bit.

If a bit error is detected at the secondary sample point, FlexCAN issues an error flag to the CAN bus at the next sample point.

During the arbitration phase, delay compensation is always disabled. During the data phase, the TDC mechanism of FlexCAN can compensate a maximum delay of 3 CAN bit times − 2 Tq. Beyond this limit, the FDCTRL[TDCFAIL] or ETDC[ETDCFAIL] flag is set. The flag indicates when the TDC mechanism is out of range and is unable to compensate the transceiver loop delay.

39.3.12.4 Remote frames

A remote frame is a special type of frame. You can program a message buffer to be a remote request frame by configuring the message buffer as Transmit with the RTR = 1. After the remote request frame is transmitted successfully, the message buffer becomes a receive message buffer, with the same ID as before.

When FlexCAN receives a remote request frame, the frame can be treated in different ways, depending on remote request storing (CTRL2[RRS]) and RX FIFO Enable (MCR[RFEN]):

- If RRS = 0, the ID of the frame is compared to the IDs of the transmit message buffers with the CODE field 1010b. If a matching ID exists, this message buffer frame is transmitted. If the matching message buffer has the RTR = 1, FlexCAN transmits a remote frame as a response. The received remote request frame is not stored in a receive buffer. It is only used to trigger a transmission of a frame in response.

  The mask registers are not used in remote frame matching, and all ID bits (except RTR) of the incoming received frame should match. If a remote request frame is received and matches a message buffer, this message buffer immediately enters the internal arbitration process. However, it is considered a normal TX message buffer, with no higher priority. The data length of this frame is independent of the DLC field in the remote frame that initiated its transmission.

- If CTRL2[RRS] = 1, the ID of the frame is compared to the IDs of the receive message buffers with the CODE field 0100b, 0010b, or 0110b. If a matching ID exists, this message buffer stores the remote frame in the same fashion as a data frame. No automatic remote response frame is generated. The mask registers are used in the matching process.
- If MCR[RFEN] = 1, FlexCAN does not generate an automatic response for remote request frames that match the Legacy FIFO filtering criteria. If the remote frame matches one of the target IDs, it is stored in the Legacy FIFO and presented to the CPU.

  For filtering formats A and B (see Legacy RX FIFO structure), it is possible to select whether remote frames are accepted or not. For format C, remote frames are always accepted if they match the ID. Remote request frames are considered as normal frames. They generate a Legacy FIFO overflow when a successful reception occurs and the Legacy FIFO is already full.
- If ERFCR[ERFEN] = 1, FlexCAN does not generate an automatic response for remote request frames that match the Enhanced RX FIFO filtering criteria. Remote Request Frames are considered normal frames. They generate an Enhanced RX FIFO overflow when a successful reception occurs and the enhanced RX FIFO is already full.

NOTE

There is no remote frame in the CAN FD format. A fixed dominant RRS bit replaces the RTR bit. FlexCAN receives and transmits remote frames in the Classical CAN format.

39.3.12.5 Overload frames

When a dominant bit is detected on the CAN bus in these locations, FlexCAN transmits overload frames:

- The first or second bit of Intermission.
- The seventh bit (last) of End of Frame field (RX frames).
- The eighth bit (last) of Error Frame Delimiter or Overload Frame Delimiter.

39.3.12.6 Message buffer timestamp

The value of the free-running timer is sampled at the beginning of the Identifier field on the CAN bus. This value is stored at the end of move-in in the TIME_STAMP field of a message buffer, providing network behavior regarding time.

When CTRL2[TIMER_SRC] = 1, an external time tick continuously clocks the free-running timer.

When CTRL2[TIMER_SRC] = 0, the FlexCAN bit clock clocks the free-running timer, which defines the baud rate on the CAN bus. During a message transmission or reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate.

The free-running timer is not incremented during Disable, Doze, Stop, and Freeze modes. It can be reset upon a specific frame reception, enabling network time synchronization. See CTRL1[TSYN].

Alternatively, by configuring CTRL2[MBTSBASE], the timestamp of the message buffer can capture the lower or higher 16 bits of the high-resolution dedicated counter. See the chip-specific FlexCAN timestamp implementation.

39.3.12.7 High-resolution timestamp

The high-resolution timestamp (HR_TIME_STAMP) uses a dedicated timer with a 32-bit counter operating in free-running mode. CTRL2[TSTAMPCAP] enables the high-resolution timestamp. When this field is not zero, the dedicated 32-bit counter value is captured during a valid CAN frame and stored in an HR_TIME_STAMP*n* register.

Each HR_TIME_STAMP*n* corresponds to a specific message buffer. For example, HR_TIME_STAMP0 stores the 32-bit timestamp associated with message buffer 0. HR_TIME_STAMP1 stores the 32-bit timestamp associated with message buffer 1, and so on.

The counter value is captured according to CTRL2[TSTAMPCAP]. For classical CAN frames, the capture points can be the start of frame bit or the point in time a CAN frame is considered valid. This valid point is the seventh bit of end of frame for transmission and the sixth bit of end of frame for reception.

For CAN FD frames, the capture points can be:

- The start of frame

- The point in time a CAN FD frame is considered valid
- The res bit of a CAN FD frame

The 16-bit timestamp of the message buffer can be configured to capture the lower or higher 16 bits of the high-resolution timer. This configuration is made by CTRL2[MBTSBASE]. See the chip-specific FlexCAN timestamp implementation.

39.3.12.8 Protocol timing

Figure 152 shows the structure of the clock generation circuitry that feeds the CAN Protocol Engine (PE) submodule. The clock source field CTRL1[CLKSRC] defines whether the internal clock is connected to the output of a crystal oscillator (oscillator clock) or to the peripheral clock. To guarantee reliable operation, select the clock source when the module is in Disable mode (MCR[MDIS] = 1).

NOTE

To identify the proper clock source, see the clock distribution chapter (module clocks table).

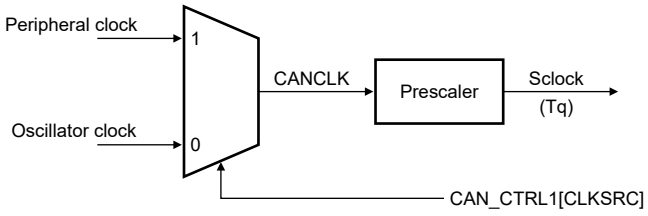


Figure 152. CAN engine clocking scheme

Select the oscillator clock whenever a tight tolerance (up to 0.1%) is required in the CAN bus timing. The crystal oscillator clock has better jitter performance than the peripheral clock.

39.3.12.8.1 Bit timing configuration

FlexCAN supports various means to configure bit timing parameters required by the CAN protocol. Control 1 (CTRL1) has various fields to control bit timing parameters:

- CTRL1[PRESDIV]
- CTRL1[PROPSEG]
- CTRL1[PSEG1]
- CTRL1[PSEG2]
- CTRL1[RJW]

CAN Bit Timing (CBT) extends the range of the CAN bit timing variables in CTRL1. Enhanced Data Phase CAN Bit Timing (EDCBT) provides a second set of CAN bit timing variables to be applied at the data phase of CAN FD frames with the Bit Rate Switch (BRS) = 1.

Enhanced Nominal CAN Bit Timing (ENCBT) extends the range of CAN bit timing variables in CBT. Enhanced Nominal CAN Bit Timing (ENCBT) extends the range of CAN bit timing variables in FDCBT. When using ENCBT and EDCBT, you must program the nominal bit timing and data phase serial clock (Sclock) dividers in Enhanced CAN Bit Timing Prescalers (EPRS).

NOTE

When the CAN FD feature is enabled, always write 1 to CBT[BTF] or CTRL2[BTE] and specify the CAN bit timing variables in CBT or ENCBT. See CAN Bit Timing (CBT) or Enhanced Nominal CAN Bit Timing (ENCBT).

CTRL1[PRESDIV], and its extended range CBT[EPRESDIV] (or EPRS[ENPRESDIV]) and FDCBT[FPRESDIV] (or EPRS[EDPRESDIV]) for the data phase bits of CAN FD messages, defines the prescaler value that generates the serial

clock (Sclock). (See Equation 19.) The period of Sclock defines the time quantum used to compose the CAN waveform. A time quantum (Tq) is the atomic unit of time managed by the CAN engine. It is the smallest time unit for all configuration values.

$$Tq = \frac{(PRESDIV + 1)}{f_{CANCLK}}$$

Equation 19. Time quantum

The bit rate, which defines the rate the CAN message is received or transmitted, is calculated with the formula:

$$CAN\ bit\ time = (Number\ of\ time\ quanta\ in\ 1\ bit\ time) \times Tq$$

$$Bit\ rate = \frac{1}{CAN\ bit\ time}$$

Equation 20. CAN bit time and baud rate

39.3.12.8.2 Bit time segments

A bit time is subdivided into three segments as shown in Figure 153. See also Figure 154 , Figure 155, and Table 256.

NOTE

For further explanation of the underlying concepts, see ISO 11898-1_2024. See also *CAN Specification Version 2.0, Part A and Part B* for bit timing.

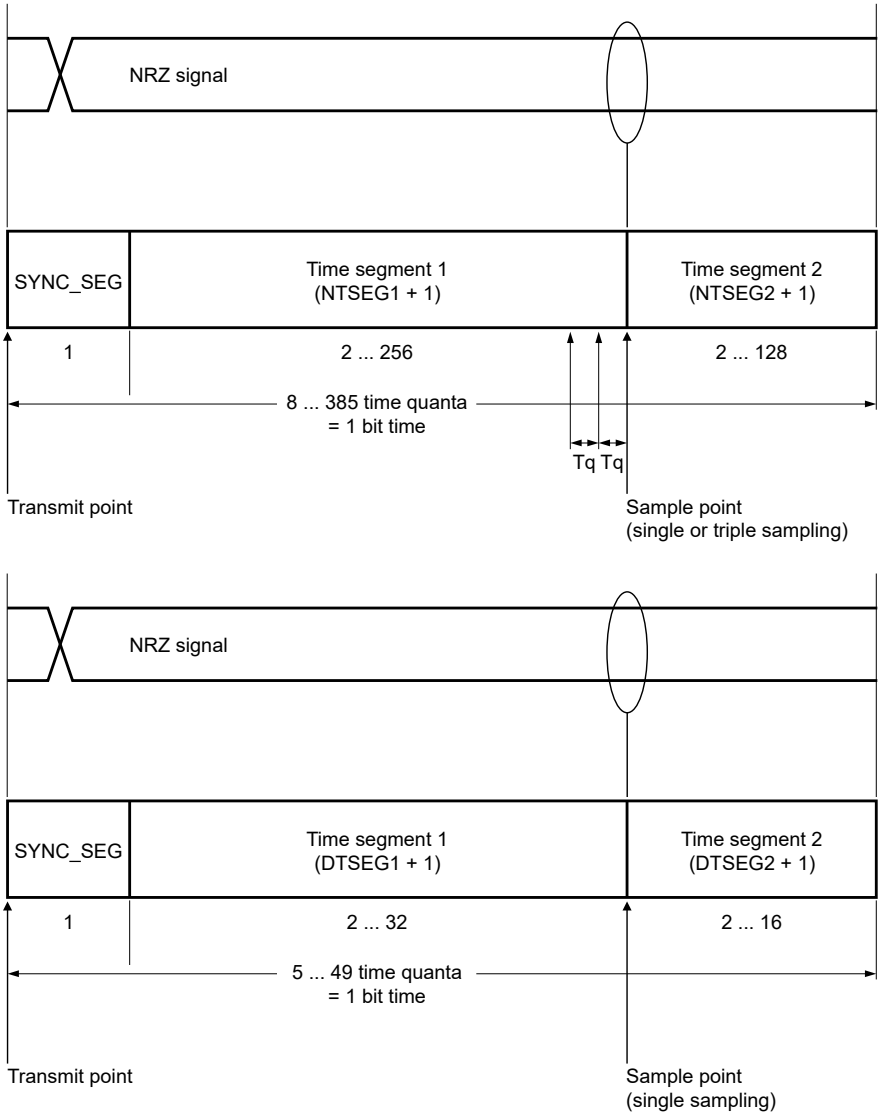


Figure 153. Segments within the bit timing (example using ENCBT and EDCBT bit timing variables)

The three bit time segments are:

- SYNC_SEG—this segment has a fixed length of one time quantum. Signal edges are expected to occur within this section.
- Time Segment 1—this segment includes the propagation segment and the phase segment 1 of the CAN standard.

  It can be programmed by configuring CTRL1[PROPSEG] and CTRL1[PSEG1] so that the sum (plus 2) is 2–16 time quanta. When CBT[BTF] = 1, FlexCAN uses CBT[EPROPSEG] and CBT[EPSEG1] so that the sum (plus 2) is 2–96 time quanta. For messages in CAN FD format with the BRS = 1, FlexCAN uses FDCBT[FPROPSEG] and FDCBT[FPSEG1] so that the sum (plus 1) is 2–39 time quanta.

  If CTRL2[BTE] = 1, FlexCAN uses ENCBT[NTSEG1] to configure time segment 1 to 2–256 time quanta. For the data phase in CAN FD messages with BRS = 1, EDCBT[DTSEG1] must be used for configuring time segment 1 to 2–32 time quanta.
- Time Segment 2—this segment represents the phase segment 2 of the CAN standard.

  It can be programmed by configuring CTRL1[PSEG2] (plus 1) to be 2–8 time quanta. When CBT[BTF] = 1, FlexCAN uses CBT[EPSEG2] so that its value (plus 1) is 2–32 time quanta. For messages in CAN FD format with the BRS = 1, FlexCAN uses

FDCBT[FPSEG2] instead, so that its value (plus 1) is 2–8 time quanta. Time segment 2 cannot be smaller than the Information Processing Time (IPT), which is 2 time quanta in FlexCAN.

If CTRL2[BTE] = 1, FlexCAN uses ENCBT[NTSEG2] to configure time segment 2 to 2–128 time quanta. For the data phase in CAN FD messages with BRS = 1, EDCBT[DTSEG2] must configure time segment 2 to 2–16 time quanta.

NOTE

The bit time defined by the above time segments must not be smaller than five time quanta. For bit time calculations, use an Information Processing Time (IPT) of two, which is the value implemented in the FlexCAN module.

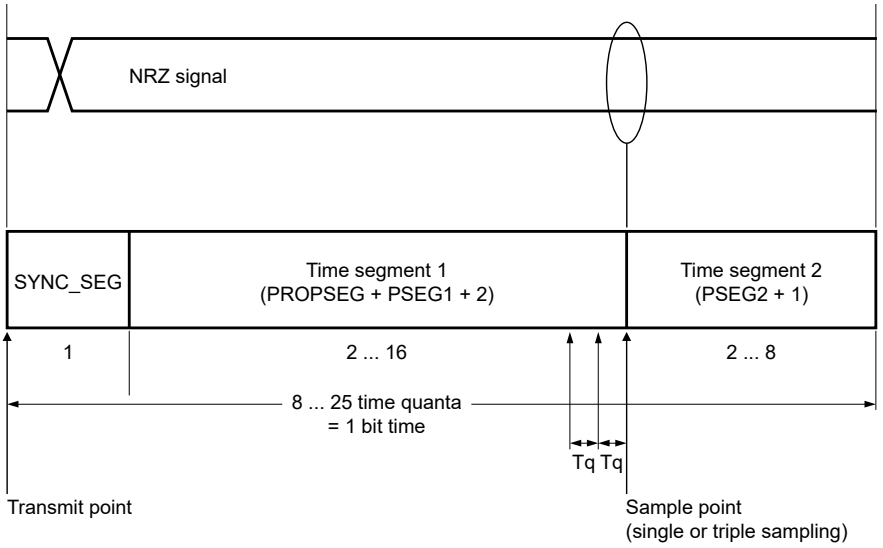


Figure 154. Segments within the bit time (example using CTRL1 bit timing variables for Classical CAN format)

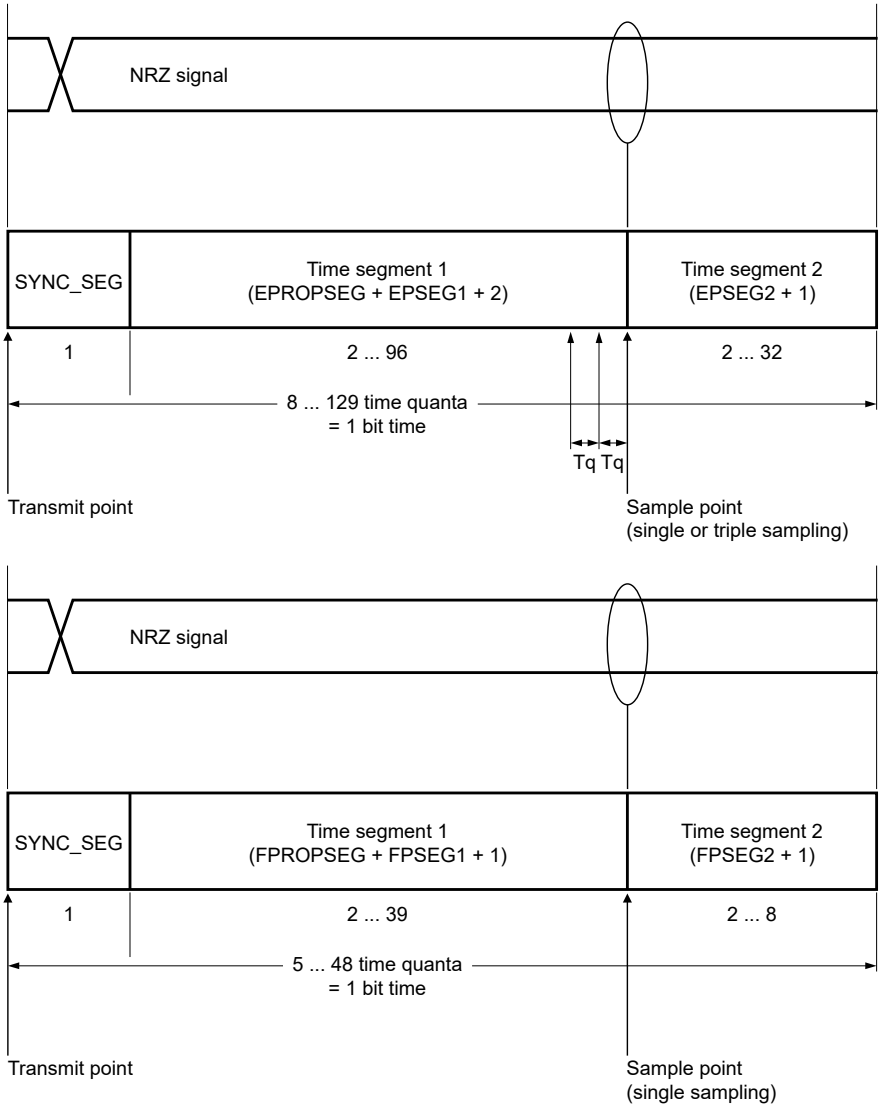


Figure 155. Segments within the bit time (example using CBT and FDCBT bit timing variables for CAN FD format)

Table 256. Time segment syntax

Syntax	Description
SYNC_SEG	Period during which the system expects transitions to occur on the bus
TSEG1	Period corresponding to the sum of PROPSEG and PSEG1
TSEG2	Period corresponding to the PSEG2 value
Transmit point	Point at which a node in Transmit mode transfers a new value to the CAN bus
Sample point	Point at which a node samples the bus. If the option of three samples per bit is selected, this point marks the position of the third sample.

Table 257 gives some examples of the CAN-compliant segment settings for Classical CAN format (Bosch CAN 2.0B) (non-FD) messages.

Table 257. Bosch CAN 2.0B standard compliant bit time segment settings

Time segment 1	Time segment 2	Resynchronization jump width
5 to 10	2	1 to 2
4 to 11	3	1 to 3
5 to 12	4	1 to 4
6 to 13	5	1 to 4
7 to 14	6	1 to 4
8 to 15	7	1 to 4
9 to 16	8	1 to 4

NOTE

You must ensure the bit time settings comply with the CAN Protocol standard ISO 11898-1_2024.

39.3.12.8.3 Calculating peripheral clocks

A CAN bit can be used as a measure of duration (for example, estimating the occurrence of a CAN bit event in a message). When a CAN bit is used in this way, the number of peripheral clocks in one CAN bit (NumClkBit) can be calculated as:

$$NumClkBit = \frac{f_{SYS}}{f_{CANCLK}} \times (PRESDIV + 1) \times (PROPSEG + PSEG1 + PSEG2 + 4)$$

Equation 21. Number of peripheral clocks per CAN bit when CTRL2[BTE] = 0

Or, if CTRL2[BTE] = 1:

$$NumClkBit = \frac{f_{SYS}}{f_{CANCLK}} \times (ENPRESDIV + 1) \times (NTSEG1 + NTSEG2 + 3)$$

Equation 22. Number of peripheral clocks per CAN bit when CTRL2[BTE] = 1

Where:

- NumClkBit is the number of peripheral clocks in one CAN bit.
- f_{CANCLK} is the Protocol Engine (PE) Clock (see Figure 152), in Hz.
- f_{SYS} is the frequency of operation of the system (CHI) clock, in Hz.
- PSEG1 is the value of CTRL1[PSEG1].
- PSEG2 is the value of CTRL1[PSEG2].
- PROPSEG is the value of CTRL1[PROPSEG].
- PRESDIV is the value in CTRL1[PRESDIV].
- ENPRESDIV is the value of EPRS[ENPRESDIV].
- NTSEG1 is the value of ENCBT[NTSEG1].
- NTSEG2 is the value of ENCBT[NTSEG2].

The formula above is also applicable to the alternative CAN bit timing variables described in:

- CAN Bit Timing (CBT)

- Enhanced Nominal CAN Bit Timing (ENCBT)
- CAN FD Bit Timing (FDCBT)
- Enhanced Nominal CAN Bit Timing (ENCBT)

For example, 180 CAN bits = (180 × NumClkBit) peripheral clock periods.

39.3.12.9 Arbitration and matching timing

During normal reception and transmission, the matching, arbitration, move-in, and move-out processes are executed during certain time windows inside the CAN frame. These windows are shown in the following figures.

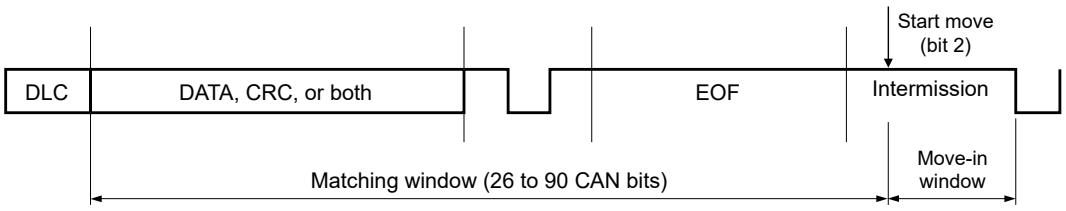


Figure 156. Matching and move-in time windows

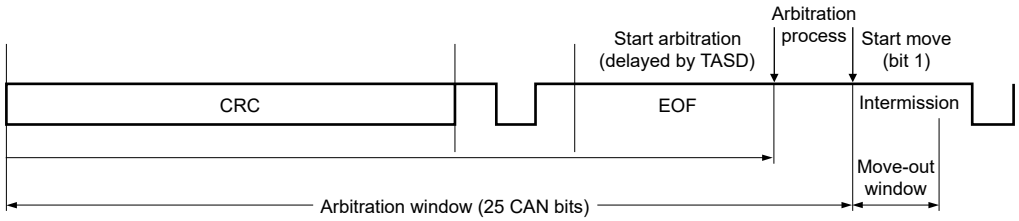


Figure 157. Arbitration and move-out time windows

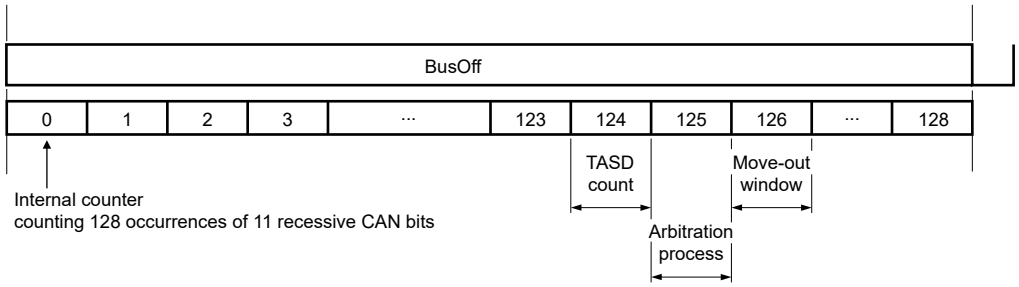


Figure 158. Arbitration at the end of bus off and move-out time windows

NOTE

In these figures, the matching and arbitration timing do not consider delays caused by concurrent memory access due to the CPU or other internal FlexCAN subblocks.

39.3.12.10 TX arbitration start delay

TX Arbitration Start Delay (CTRL2[TASD]) indicates the number of CAN bits that FlexCAN uses to delay the TX arbitration process starting point from the first bit of the CRC field of the current frame. This variable can be written only in Freeze mode; FlexCAN blocks it in other modes.

The ability of the CPU to reconfigure message buffers for transmission after the end of the internal arbitration process impacts transmission performance. In the arbitration process, FlexCAN finds the winner MB for transmission (see Arbitration process). If

the arbitration ends too early (before the first bit of the Intermission field) the CPU may reconfigure some TX message buffers. It is possible that the winning message buffer is no longer the best candidate to be transmitted.

TASD can optimize the transmission performance by defining the arbitration start point, as shown in Figure 159, based on factors such as:

- Peripheral-to-oscillator clock ratio
- CAN bit timing variables that determine the CAN bit rate
- Number of message buffers in use by the matching and arbitration processes

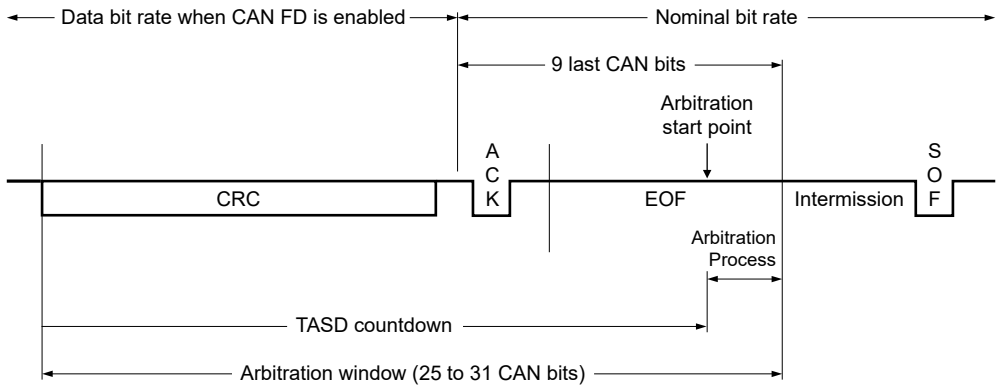


Figure 159. Optimal TX arbitration start point

The duration of an arbitration process, in terms of CAN bits, is:

- Directly proportional to the number of available message buffers
- Directly proportional to the CAN bit rate
- Inversely proportional to the peripheral clock frequency

The optimal arbitration timing occurs when the last message buffer is scanned immediately before the first bit of the Intermission field of a CAN frame. For instance, if the following are true:

- There are few message buffers.
- The peripheral-to-oscillator clock ratio is high.
- The CAN baud rate is low.

Then the arbitration can be placed closer to the end of the frame, adding more delay to its starting point, and vice versa.

If CTRL2[TASD] = 0, the arbitration start is not delayed, and more time is reserved for arbitration. Alternatively, if CTRL2[TASD] is close to 24, the CPU can configure a TX message buffer later, and less time is reserved for arbitration. If too little time is reserved for arbitration, FlexCAN may not be able to find a winner MB in time. The transmitted arbitration winner may not have the best chance to win the bus arbitration against external nodes on the CAN bus.

The optimal TASD value can be calculated as follows:

For CAN FD frames and (MAXMB + 1) ≤ NMB_{END}

$$TASD = 31 - \frac{2 * (MAXMB + 1) + 4}{CPCB_N}$$

For CAN FD frames and (MAXMB + 1) > NMB_{END}

$$TASD = 22 - \frac{2 * (MAXMB + 1) - NMB_{END}}{CPCB_F}$$

For non-FD frames

$$TASD = 25 - \frac{2 * (MAXMB + 1) + 4}{CPCB}$$

Equation 23. Optimal value for TASD

Where:

$$NMB_{END} = \frac{(9 \times CPCB_N) - 4}{2}$$

$$BITRATE_N = \left(\frac{f_{CANCLK}}{[1 + (EPSEG1 + 1) + (EPSEG2 + 1) + (EPROPSEG + 1)] \times (EPRESDIV + 1)} \right)$$

$$BITRATE_F = \left(\frac{f_{CANCLK}}{[1 + (FPSEG1 + 1) + (FPSEG2 + 1) + FPROPSEG] \times (FPRESDIV + 1)} \right)$$

$$CPCB_N = \frac{f_{SYS}}{BITRATE_N}$$

$$CPCB_F = \frac{f_{SYS}}{BITRATE_F}$$

$$CPCB = CPCB_N$$

Equation 24. Variables used in TASD calculation

- MAXMB is the value in MCR[MAXMB].
- NMB_{END} is the number of message buffers that the arbitration process can scan during the last nine CAN bits at the end of a frame. (See Equation 24.)
- $BITRATE_N$ is the CAN bit rate in bits per second calculated by the nominal CAN bit time variables.
- $BITRATE_F$ is the CAN bit rate in bits per second calculated by the data CAN bit time variables.
- $CPCB_N$ is the number of peripheral clocks per CAN bit in nominal bit rate for CAN FD frames.
- $CPCB_F$ is the number of peripheral clocks per CAN bit in data bit rate for CAN FD frames.
- CPCB is the number of peripheral clocks per CAN bit for non-FD frames.
- f_{CANCLK} is the oscillator clock, in Hz.
- f_{SYS} is the peripheral clock, in Hz.
- EPSEG1 is the value in CBT[EPSEG1] (CTRL1[PSEG1] can also be used).
- EPSEG2 is the value in CBT[EPSEG2] (CTRL1[PSEG2] can also be used).
- EPROPSEG is the value in CBT[EPROPSEG] (CTRL1[PROPSEG] can also be used).
- EPRESDIV is the value in CBT[EPRESDIV] (CTRL1[PRESDIV] can also be used).
- FPSEG1 is the value in FDCBT[FPSEG1].

- FPSEG2 is the value in FDCBT[FPSEG2].
- FPROPSEG is the value in FDCBT[FPROPSEG].
- FPRESDIV is the value in FDCBT[FPRESDIV].
- NTSEG1 is the value in ENCBT[NTSEG1].
- NTSEG2 is the value in ENCBT[NTSEG2].
- ENPRESDIV is the value in EPRS[ENPRESDIV].
- DTSEG1 is the value in EDCBT[DTSEG1].
- DTSEG2 is the value in EDCBT[DTSEG2].
- EDPRESDIV is the value in EPRS[EDPRESDIV].

If CTRL2[BTE] = 1, then:

$$BITRATE_N = \frac{f_{CANCLK}}{[1 + (NTSEG1 + 1) + (NTSEG2 + 1)] \times (ENPRESDIV + 1)}$$

Equation 25. Nominal baud rate when CTRL2[BTE] = 1

$$BITRATE_F = \frac{f_{CANCLK}}{[1 + (DTSEG1 + 1) + (DTSEG2 + 1)] \times (EDPRESDIV + 1)}$$

Equation 26. Fast baud rate when CTRL2[BTE] = 1

See also Protocol timing for more details.

39.3.12.10.1 TASD configuration examples

The following tables show the TASD value calculated for some configuration cases.

Case 1:

- Clock ratio = 2:1 (for example, peripheral clock 80 MHz and oscillator clock 40 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 258. TASD values in Case 1

Number of message buffers	TASD value	Maximum bit rate in data phase (MBd)
16	24	Invalid
32	24	8.0
64	23	8.0
96	22	8.0

Case 2:

- Clock ratio = 1:1 (for example, peripheral clock 40 MHz and oscillator clock 40 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 259. TASD values in Case 2

Number of message buffers	TASD value	Maximum bit rate in data phase (MBd)
16	24	Invalid

Table continues on the next page...

Table 259. TASD values in Case 2 (continued)

Number of message buffers	TASD value	Maximum bit rate in data phase (MBd)
32	23	6.67
54	22	5.0
64	21	3.33
96	20	1.6

Case 3:

- Clock ratio = 2:1 (for example, peripheral clock 40 MHz and oscillator clock 20 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 260. TASD values in Case 3

Number of message buffers	TASD value	Maximum bit rate in data phase (MBd)
16	24	Invalid
32	23	4.0
54	22	4.0
64	21	3.33
96	20	1.54

39.3.13 Clocks

The following table describes the clock sources for FlexCAN. See the Clock Controller Module (CCM) for clock setting, configuration, and gating information.

Table 261. FlexCAN clocks

Clock name	Description
MODULE_CLK (system_clk)	Peripheral clock
MODULE_CLK_CHI (host_clock)	Control Host Interface (CHI) clock
MODULE_CLK_PE (protocol_engine_clock)	Protocol Engine (PE) clock
MODULE_CLK_PE_NOGATE (protocol_engine_clock_nogate)	Protocol Engine clock (no gating)
MODULE_CLK_S (system_clock_nogate)	Peripheral access clock

39.3.13.1 Clock domains and restrictions

FlexCAN has two clock domains asynchronous to each other:

- The bus domain feeds the Control Host Interface (CHI) submodule.
- The oscillator domain feeds the CAN Protocol Engine (PE) submodule.

When CTRL1[CLKSRC] = 1, synchronous operation occurs because both domains are connected to the peripheral clock. This setting creates a 1:1 ratio between the peripheral and oscillator clocks.

When the two domains are connected to clocks with different frequencies or phases, the frequency relationship between the two clock domains is restricted. In asynchronous operation, the bus domain clock frequency must always be greater than the oscillator domain clock frequency.

NOTE

Asynchronous operation with a 1:1 ratio between peripheral and oscillator clocks is not allowed.

When performing matching and arbitration, FlexCAN must scan the whole message buffer memory during the time slot of one CAN frame, comprised of a number of CAN bits. To provide sufficient time for the scan, observe the following requirements:

- The peripheral clock frequency cannot be less than the oscillator clock frequency.
- There must be a minimum number of peripheral clocks per CAN bit, as specified in Table 262.

Table 262. Minimum number of peripheral clocks per CAN bit for Classical CAN format

Number of message buffers	Value of MCR[RFEN]	Value of ERFCR[ERFEN]	Minimum number of peripheral clocks per CAN bit
16	0	0	16
32	0	0	16
64	0	0	25
96	0	0	37
16	1	0	16
32	1	0	17
64	1	0	30
96	1	0	42
16	0	1	16
32	0	1	19
64	0	1	31
96	0	1	42

For classical frame format, the minimum number of peripheral clocks per CAN bit specified in Table 262 determines the minimum peripheral clock frequency for a given number of message buffers and for an expected CAN bit rate. The CAN bit rate depends on the number of time quanta in a CAN bit. This number can be defined by adjusting one or more of the bit timing values contained in:

- Control 1 (CTRL1)
- CAN Bit Timing (CBT)
- Enhanced Nominal CAN Bit Timing (ENCBT)

The time quantum (Tq) is defined in Protocol timing. The minimum number of time quanta per CAN bit must be eight, so the oscillator clock frequency should be at least eight times the CAN bit rate.

39.3.13.1.1 Clock restrictions for CAN FD

For CAN FD frame format, some constraints must be satisfied. The equation below calculates the number of peripheral clocks per CAN bit in nominal bit rate (NumClkNomBit).

$$NumClkNomBit = \frac{f_{SYS}}{f_{CANCLK}} \times (PRESDIV + 1) \times (PROPSEG + PSEG1 + PSEG2 + 4)$$

$$= \frac{f_{SYS}}{NomBitRate}$$

Equation 27. Number of peripheral clocks per nominal CAN bit

Where PRESDIV, PSEG1, and PSEG2 are CAN bit time values in Control 1 (CTRL1). Alternatively, EPRESDIV, EPSEG1, and EPSEG2 values in CAN Bit Timing (CBT) or the values of EPRS[ENPRESDIV], ENCBT[NTSEG1], and ENCBT[NTSEG2] can be used instead. NumClkNomBit can also be calculated as a function of the expected nominal bit rate used in the arbitration phase (NomBitRate), as shown in the equation above.

The number of CAN bits in the data phase of an FD frame with BRS = 1 (fast CAN bits) depends on the number of data bytes in the payload. The number of fast CAN bits (NumOfFastBits) can be determined in Table 263. Having fewer data bytes means having fewer fast CAN bits. It also means that less time is available for FlexCAN to scan the whole message buffer memory during the internal matching and arbitration processes.

Table 263. Number of fast CAN bits in a CAN FD frame

Minimum number of data bytes	DLC field	NumOfFastBits
0	0h	21
1	1h	29
2	2h	37
3	3h	45
4	4h	53
5	5h	61
6	6h	69
7	7h	77
8	8h	85
12	9h	117
16	Ah	149
20	Bh	186
24	Ch	218
32	Dh	282
48	Eh	410
64	Fh	538

The critical part of a CAN FD frame is during the data phase, where the CAN bit rate is faster than in the arbitration phase. The minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) can be calculated to guarantee that enough time is available for FlexCAN to scan the message buffer memory during reception and transmission. The equation below calculates this constraint.

$$MinNumClkFastBit_A = \frac{(8.5 \times MaxNumOfMb) + [ERFEN \times (2 \times NFE + 4)] + 64 - (9 \times NumClkNomBit)}{NumOfFastBits}$$

Equation 28. Minimum number of peripheral clocks per fast CAN bit for FlexCAN scan process

Where MaxNumOfMb is the maximum number of available message buffers defined in MCR[MAXMB]. NFE and ERFEN are the fields defined in Enhanced RX FIFO Control (ERFCR).

The clock-domain-crossing circuit between the CHI and PE subblocks also imposes a minimum number of peripheral clocks per fast CAN bit. This minimum is required for the handshake mechanism to work properly without losing status information through the interface, as shown in the equation below.

$$MinNumClkFastBit_B = 3 \times \left(1 + \frac{f_{SYS}}{f_{CANCLK}}\right)$$

Equation 29. Minimum number of peripheral clocks per fast CAN bit for FlexCAN clock domain interface

Therefore, the larger of the two values calculated above determines the minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit).

$$MinNumClkFastBit = Maximum\ (MinNumClkFastBitA, MinNumClkFastBitB)$$

Equation 30. Minimum number of peripheral clocks per fast CAN bit

Then, the maximum CAN bit rate in the data phase of CAN FD frames (DataBitRateMAX) can be calculated as below.

$$DataBitrate_{MAX} = \frac{f_{CANCLK}}{ROUNDUP\left(\frac{MinNumClkFastBit \times f_{CANCLK}}{f_{SYS}}\right)}$$

Equation 31. Maximum achievable baud rate for data phase

These factors affect the maximum data bit rate attainable by FlexCAN in CAN FD mode:

- The peripheral and oscillator clock frequencies
- The maximum number of message buffers
- The expected nominal bit rate

Also, the data bit rate depends on the minimum payload size of FD frames used in a given application.

To illustrate how the configuration of FlexCAN variables affects the CAN FD bit rate, consider this application example:

- The peripheral clock frequency is set to 50 MHz
- The oscillator clock frequency is set to 40 MHz

1. Considering the nominal bit rate as 1 Mbit/s, the number of peripheral clocks per CAN bit in nominal bit rate is calculated as below.

   $$NumClkNomBit = \frac{50 \times 10^6}{1 \times 10^6} = 50$$

   Equation 32. Calculation example for number of peripheral clocks per nominal CAN bit

2. The number of fast CAN bits (NumOfFastBits) is determined in Table 263. For example, if the minimum payload in FD frames is 8 bytes, there are 85 CAN bits in the data phase.
3. Assuming the maximum number of message buffers is 96, and Enhanced RX FIFO is disabled, the minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) can be calculated.

$$MinNumClkFastBit_A = \frac{(8.5 \times 96) + 64 - (9 \times 50)}{85} = 5.06$$

Equation 33. Calculation example for number of peripheral clocks per fast CAN bit for FlexCAN scan process

$$MinNumClkFastBit_B = 3 \times \left(1 + \frac{50}{40}\right) = 6.75$$

Equation 34. Calculation example for number of peripheral clocks per fast CAN bit for FlexCAN clock domain interface

$$MinNumClkFastBit = Maximum\ (\ 5.06, 6.75\ ) = 6.75$$

Equation 35. Calculation example for number of peripheral clocks per fast CAN bit

4. The maximum CAN bit rate in the data phase can finally be found.

$$DataBitRate_{MAX} = \frac{40 \times 10^6}{ROUNDUP\left(\frac{6.75 \times 40 \times 10^6}{50 \times 10^6}\right)} = 6.667\ Mbps$$

Equation 36. Calculation example for maximum achievable baud rate

Even though the oscillator clock frequency (40 MHz) is adequate to generate a data rate of 8 Mbit/s in CAN FD mode, the specific FlexCAN configuration limits this rate to 6.667 Mbit/s. This limitation is mainly due to the low peripheral clock frequency that imposes the MinNumClkFastBitB bound.

Table 264 shows the maximum data rate for CAN FD with Enhanced RX FIFO disabled according to clock frequencies, payload size, and number of available message buffers. For some cases, if the number of available message buffers is reduced, FlexCAN can then achieve a data rate up to 8 Mbit/s.

Table 264. Maximum CAN bit rate in data phase on CAN FD frames with Enhanced RX FIFO disabled

Peripheral clock frequency (MHz)	Minimum payload size	Number of available message buffers	Maximum data rate (Mbit/s)
40	8	94	6.667
40	8	114	>5.0
40	12	>117	6.667
40	12	128	5.714
50	12–64	128	6.667
60	8	126	8.0
60	12	128	8.0
67	6	128	8.0
80	3	128	8.0
100	0	128	8.0

39.3.14 Reset

You can reset FlexCAN in the following ways:

- Chip-level hard reset, which resets all memory-mapped registers asynchronously.
- Soft reset:
  - MCR[SOFTRST], which resets some of the memory-mapped registers synchronously. To see which registers soft reset affects, see Table 267.
  - The chip level soft reset input is not supported, use the MCR[SOFTRST] to trigger a soft reset.

Soft reset is synchronous and must follow an internal request-and-acknowledge procedure across clock domains. Therefore, it may take some time to propagate its effects fully. MCR[SOFTRST] remains 1 when soft reset is pending, so software can poll this field to identify when the reset has completed. Soft reset cannot be applied when clocks are shut down in a low-power mode. The low-power mode should be exited and the clocks resumed before applying soft reset.

The clock source should be selected when the module is in Disable mode (see CTRL1[CLKSRC]). After the clock source is selected and the module is enabled (MCR[MDIS] becomes 0), FlexCAN automatically enters Freeze mode. In Freeze mode:

1. FlexCAN is unsynchronized to the CAN bus.
2. MCR[HALT] and MCR[FRZ] become 1.
3. The internal state machines are disabled.
4. MCR[FRZACK] and MCR[NOTRDY] become 1.

The TX pin is in the recessive state and FlexCAN does not initiate any transmission or reception of CAN frames. Reset does not affect the message buffers and the RX Individual Mask registers, so they are not automatically initialized.

39.3.15 Interrupts

FlexCAN has many interrupt sources:

- Interrupts due to message buffers
- Interrupts due to interrupts combined via an OR operator from:
  - Message buffers
  - Bus Off
  - Bus Off Done
  - Error
  - Error Fast (errors detected in the data phase of CAN FD format messages with BRS = 1)
  - Wake Up
  - TX Warning
  - RX Warning

If its corresponding IMASK bit is 1, each message buffer can be an interrupt source. There is no distinction between TX and RX interrupts for a particular buffer, under the assumption that the buffer is initialized for either transmission or reception. Each buffer has an assigned flag bit in the IFLAG registers. When the corresponding buffer completes a successful transfer, the flag is set. When the CPU writes 1 to it, the flag is cleared (unless another interrupt is generated at the same time).

NOTE

The CPU must clear only the bit causing the current interrupt. For this reason, do not use bit manipulation instructions (BSET) to clear interrupt flags. These instructions may cause accidental clearing of interrupt flags which are set after entering the current interrupt handler.

If the Legacy RX FIFO is enabled (MCR[RFEN] = 1) and DMA is disabled (MCR[DMA] = 0), the interrupts corresponding to message buffers 0–7 have different meanings.

- Bit 7 of Interrupt Flags 1 (IFLAG1) becomes the Legacy FIFO Overflow flag
- Bit 6 becomes the Legacy FIFO Warning flag

- Bit 5 becomes the Frames Available in Legacy FIFO flag
- Bits 4–0 are unused.

See Interrupt Flags 1 (IFLAG1) for more information.

If both Legacy RX FIFO and DMA are enabled (MCR[RFEN] = 1 and MCR[DMA] = 1), FlexCAN does not generate any Legacy FIFO interrupt. Bit 5 of IFLAG1 still indicates Frames Available in Legacy FIFO and generates a DMA request. Bits 7, 6, and 4–0 are unused.

CAUTION

Legacy FIFO cannot be enabled when CAN FD is enabled.

When multiple message buffer interrupt sources are combined via an OR operator into a single interrupt, the interrupt is generated when any associated message buffer (or FIFO, if applicable) generates an interrupt. In this case, the CPU must read the IFLAG registers to determine which message buffer or FIFO source caused the interrupt.

These interrupt sources generate interrupts like the message buffer interrupt sources, and can be read from Error and Status 1 (ESR1):

- Bus Off
- Bus Off Done
- Error
- Error Fast
- Wake Up
- TX Warning
- RX Warning

The Bus Off, Error, TX Warning, and RX Warning interrupt masks are located in Control 1 (CTRL1); the wake-up interrupt mask is located in Module Configuration (MCR).

39.3.16 Bus interface

CPU access to FlexCAN registers is subject to the following rules:

- Unrestricted read and write access to supervisor registers results in an access error. In Table 267, supervisor registers are registers identified with an S, and registers that are identified with S/U that are in Supervisor Mode
- Read and write access to implemented reserved address space results in an access error.
- Write access to positions whose bits are all currently read-only results in an access error. If at least one of the bits is not read-only, no access error is issued. Write permission to specific positions or some of their bits can change depending on the mode of operation or transitory state. See register and field descriptions for details.
- Read and write access to unimplemented address space results in an access error.
- Read and write access to RAM-located positions during Low-Power mode results in an access error.
- The RXIMR memory region can be considered as general-purpose memory and available for access via these methods:
  - If you write 0 to MCR[IRMQ], the individual masks (RXIMR) are disabled. In this case, the RXIMR memory region is considered general-purpose memory.
  - If MCR[MAXMB] is programmed with a value smaller than the available number of message buffers, the unused memory space can be used as general-purpose RAM space. Reserved words within RAM cannot be used. For example, suppose the RAM in FlexCAN can support up to 16 message buffers, CTRL2[RFFN] = 0h, and MCR[MAXMB] = 0.
    - In this case, the maximum number of message buffers becomes one.
    - The RAM starts at 0080h, and the space 0080h–008Fh is used by the one message buffer.

- The memory space 0090h–017Fh is available.
- The space 0180h–087Fh is reserved.
- The space 0880h–0883h is used by the one individual mask and the available memory in the mask register space is 0884h–08BFh.
- In the space from 08C0h–09DFh, there are reserved words for internal use which cannot be used as general-purpose RAM.

As a rule, free memory space for general purpose depends only on MCR[MAXMB].

— If MCR[FDEN] = 1, general-purpose memory can be used only outside Freeze mode.

Table 265. Access permissions

Supervisor access	0				1		
Supervisor mode MCR[SUPV] = 1	0			1	Any		
Modes of operation	Normal	Freeze	Low-power	Any	Normal	Freeze	Low-power
MCR	Bus error	Bus error	Bus error	Bus error	Read and write	Read and write	Read and write
CTRL1	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
TIMER	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
TCR	Bus error	Bus error	Bus error	Bus error	Bus error	Bus error	Bus error
RXMGMASK[1]	Bus error for write operation	Read and write	Bus error for write operation	Bus error	Bus error for write operation	Read and write	Bus error for write operation
RX14MASK[1]	Bus error for write operation	Read and write	Bus error for write operation	Bus error	Bus error for write operation	Read and write	Bus error for write operation
RX15MASK[1]	Bus error for write operation	Read and write	Bus error for write operation	Bus error	Bus error for write operation	Read and write	Bus error for write operation
ECR	Bus error for write operation	Read and write	Bus error for write operation	Bus error	Bus error for write operation	Read and write	Bus error for write operation
ESR1	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
IMASK2	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
IMASK1	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
IFLAG2	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write

Table continues on the next page...

Table 265. Access permissions (continued)

Supervisor access	0			1			
Supervisor mode MCR[SUPV] = 1	0			1	Any		
Modes of operation	Normal	Freeze	Low-power	Any	Normal	Freeze	Low-power
IFLAG1	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
CTRL2	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
ESR2	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
CRCR	Bus error for write operation	Bus error for write operation	Bus error for write operation	Bus error	Bus error for write operation	Bus error for write operation	Bus error for write operation
RXFGMASK[1]	Bus error for write operation	Read and write	Bus error for write operation	Bus error	Bus error for write operation	Read and write	Bus error for write operation
RXFIR[1]	Bus error for write operation	Bus error for write operation	Bus error for write operation	Bus error	Bus error for write operation	Bus error for write operation	Bus error for write operation
CBT	Bus error for write operation	Read and write	Bus error for write operation	Bus error	Bus error for write operation	Read and write	Bus error for write operation
DBG1	Bus error for write operation	Bus error for write operation	Bus error for write operation	Bus error for write operation	Bus error for write operation	Bus error for write operation	Bus error for write operation
DBG2	Bus error for write operation	Bus error for write operation	Bus error for write operation	Bus error for write operation	Bus error for write operation	Bus error for write operation	Bus error for write operation
IMASK3	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
IFLAG3	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
MB[1 2]	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
Legacy FIFO header[1]	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
Legacy FIFO reserved space[1]	Bus error	Bus error	Bus error	Bus error	Bus error	Bus error	Bus error
Legacy FIFO filters[1]	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write

Table continues on the next page...

Table 265. Access permissions (continued)

Supervisor access	0			1			
Supervisor mode MCR[SUPV] = 1	0			1	Any		
Modes of operation	Normal	Freeze	Low-power	Any	Normal	Freeze	Low-power
RXIMR[1]	Bus error	Read and write	Bus error	Bus error	Bus error	Read and write	Bus error
MECR	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
ERRIAR	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
ERRIDPR	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
ERRIPPR	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
RERRAR [3]	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
RERRDR[3]	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
RERRSYNR[3]	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
ERRSR	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
EPRS	Bus error for write operation	Read and write	Bus error for write operation	Bus error	Bus error for write operation	Read and write	Bus error for write operation
ENCBT	Bus error for write operation	Read and write	Bus error for write operation	Bus error	Bus error for write operation	Read and write	Bus error for write operation
EDCBT	Bus error for write operation	Read and write	Bus error for write operation	Bus error	Bus error for write operation	Read and write	Bus error for write operation
ETDC	Bus error for write operation	Read and write	Bus error for write operation	Bus error	Bus error for write operation	Read and write	Bus error for write operation
FDCTRL	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
FDCBT	Read and write[3]	Read and write	Read and write[3]	Bus error	Read and write[3]	Read and write	Read and write[3]

Table continues on the next page...

Table 265. Access permissions (continued)

Supervisor access	0			1			
Supervisor mode MCR[SUPV] = 1	0			1	Any		
Modes of operation	Normal	Freeze	Low-power	Any	Normal	Freeze	Low-power
FDCRC	Bus error for write operation	Bus error for write operation	Bus error for write operation	Bus error	Bus error for write operation	Bus error for write operation	Bus error for write operation
ERFCR	Read and write[3]	Read and write	Read and write[3]	Bus error	Read and write[3]	Read and write	Read and write[3]
ERFIER	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
ERFSR	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
HR_TIME_STAMP	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
Enhanced Rx FIFO header[4]	Bus error for write operation	Bus error for write operation	Bus error for write operation	Bus error	Bus error for write operation	Bus error for write operation	Bus error for write operation
Enhanced Rx FIFO reserved space[4]	Bus error	Bus error	Bus error	Bus error	Bus error	Bus error	Bus error
ERFFEL	Bus error for write operation	Read and write	Bus error for write operation	Bus error	Bus error for write operation	Read and write	Bus error for write operation
General purpose RAM[1]	Read and write	Read and write	Read and write	Bus error	Read and write	Read and write	Read and write
Reserved space (used)[1]	Bus error	Bus error	Bus error	Bus error	Bus error	Bus error	Bus error
Reserved space (empty)[1]	Bus error	Bus error	Bus error	Bus error	Bus error	Bus error	Bus error

1. Access in low power is only possible if RAM_clk and MODULE_CLK are enabled.
2. If MCR[RFEN] = 1, see Legacy FIFO access rules in the rows below.
3. Write operation has no effect.
4. If MCR[RFEN] = 1, do not access Enhanced RX FIFO.

39.3.17 Detection and correction of memory errors

To update the parity bits in memory properly, all FlexCAN memory must be initialized before starting its operation. CTRL2[WRMFRZ] grants write access to all memory positions that require initialization, from 080h–ADFh and from C20h–31FFh. You must also initialize these registers:

- RX Message Buffers Global Mask (RXMGMASK)
- Receive 14 Mask (RX14MASK)
- Receive 15 Mask (RX15MASK)
- Legacy RX FIFO Global Mask (RXFGMASK)

MCR[RFEN] and ERFCR[ERFEN] must not be 1 during memory initialization.

FlexCAN supports detection and correction of errors in memory read accesses. Each byte of FlexCAN memory is associated with five parity bits that ensure a Hamming distance of 4. The error correction mechanism ensures that in this 13-bit word, errors in one bit can be corrected (correctable errors). Errors in two bits can be detected but not corrected (uncorrectable errors). Errors in more than two bits may not be detected. For uncorrectable errors, the error correction logic does not change the corrupted data. When a read access is performed, the parity bits are used to calculate a syndrome, which indicates the error in each byte.

When an all-zeroes or an all-ones read occurs, FlexCAN detects an uncorrectable error. See Error Report Syndrome (RERRSYNR).

Memory errors are indicated to the host via status register (Error Status (ERRSR)) and bus transfer errors. Memory errors are reported via these report registers:

- Error Report Address (RERRAR)
- Error Report Data (RERRDR)
- Error Report Syndrome (RERRSYNR)

MECR[ECCDIS] determines whether the error detection and correction mechanism can be activated. When disabled, updates on indications and reporting registers are stopped. To ensure that memory has consistent parity bits associated with the data, the parity bits are still calculated and written with data in memory write operations.

To avoid accidentally changing the critical error correction configuration, follow this protocol to enable the update of Memory Error Control (MECR):

1. Write 1 to CTRL2[ECRWRE]. (By default, CTRL2[ECRWRE] is 0 and MECR[ECRWRDIS] is 1.)
2. Write 0 to MECR[ECRWRDIS].
3. All writes to Memory Error Control (MECR) must keep MECR[ECRWRDIS] = 0.
4. After configuration is done, lock MECR by writing 1 to MECR[ECRWRDIS] or writing 0 to CTRL2[ECRWRE].

39.3.17.1 Sources of memory access

These major sources (or requestors) can access FlexCAN memory:

- Host (CPU). The largest word accessed is 32-bit.
- FlexCAN internal processes:
  - RX matching
  - TX arbitration
  - Move-in on reception
  - Move-out on transmission

  The largest word accessed is 64-bit.

The source of access determines the way that uncorrectable errors are indicated and reported.

39.3.17.2 Error indication

These flags indicate memory errors:

- ERRSR[HANCEIF]
- ERRSR[FANCEIF]
- ERRSR[CEIF]

Uncorrectable errors detected in memory reads requested by the host are indicated separately from the errors detected in requests by FlexCAN internal processes. When correctable errors are detected, FlexCAN makes no distinction of the source of the access. There are three independent flags for these three cases, and each flag raises an interrupt unless a mask in Memory Error Control (MECR) masks it. If both uncorrectable and correctable errors are found in different bytes in the same read operation, both flags are set.

An uncorrectable error detected in host access is also indicated as a bus transfer error. A bus wait request may be asserted to extend the memory transaction to the moment the report registers are updated. This indication cannot be masked. If ERRSR[HANCEIF] is not masked, the same uncorrectable error raises a bus transfer error and an interrupt request.

Each indication flag has one overrun flag in Error Status (ERRSR). The overrun flags do not request interrupts. Overrun flags for uncorrectable errors indicate that other errors of the same nature were detected after the current error being treated. Overrun flags for correctable errors indicate that other errors of the same nature were detected before the current error being treated.

The recommended handling sequence for error indication is:

1. Get error report information from report registers.
2. Use this information to take proper measures in the application.
3. Clear the ERRSR[HANCEIF], ERRSR[FANCEIF], and ERRSR[CEIF] flags.
4. If the overrun flag is active:
    a. Alert the application that at least one error could not be managed.
    b. Clear the overrun flag.

FlexCAN internal processes can access memory in transactions larger than 32 bits. For the indication, this kind of access is considered a consecutive sequence of 32-bit accesses. If errors are found in two or more 32-bit words, the interrupt and overrun flags are set simultaneously.

39.3.17.3 Error reporting

The error report registers provide detailed information about the address read, raw data, and syndrome read with error. The flags described in Error indication indicate these report registers:

- Error Report Address (RERRAR)
- Error Report Data (RERRDR)
- Error Report Syndrome (RERRSYNR)

The address, data, and syndrome registers are updated simultaneously with the error flags, according to these rules:

1. If either of the uncorrectable error flags is set, the report registers are not updated. The previous uncorrectable error reporting is preserved.
2. Otherwise, either no error flag is set or only the correctable error flag is set. The report registers are updated according to the new error, or according to the most severe new error, if uncorrectable and correctable errors are simultaneously detected.

Reporting of errors detected in accesses larger than 32 bits follows the rules described in Error indication and in the description of Error Report Address (RERRAR).

The addresses reported in RERRAR and defined in ERRIAR are not the same as the addresses listed in the module memory map. The relation between the reported addresses and the respective ones in the module memory map is shown in the description of Error Injection Address (ERRIAR).

Addresses reported when reading memory portions organized as FIFOs refer to the address of the specific entry accessed in the FIFO, not to the FIFO base address. Such memory portions include:

- Legacy RX FIFO Structure
- Enhanced RX FIFO Structure
- Legacy RX FIFO Information (RXFIR)

To ensure coherence of error report registers, disable the report update by writing 1 to MECR[RERRDIS] before reading the report registers.

39.3.17.4 Error response

Correctable errors have no consequence to FlexCAN operation because the host or FlexCAN internal processes corrects affected data before its use.

For host-initiated reads, an uncorrectable error may affect the host, but does not affect FlexCAN operation.

Uncorrectable errors detected on memory read operations requested by FlexCAN internal processes may result in incorrect operation depending on the state of MECR[NCEFAFRZ], as follows:

- During reception (either matching or move-in process), when an uncorrectable error occurs:
  - An incorrect destination may be selected to store the incoming frame.
  - A corrupted frame may be stored in the correct destination.
  - Both of these events may occur.

  If MECR[NCEFAFRZ] = 1, FlexCAN stops operation automatically and enters Freeze mode to prevent corrupted data from being treated as valid by FlexCAN internal processes. If MECR[NCEFAFRZ] = 0, FlexCAN continues working, and a corrupted frame is received.
- During arbitration, when an uncorrectable error occurs:
  - A non-highest-priority TX message buffer may be mistakenly selected for transmission.
  - Its data may be corrupted.

  If MECR[NCEFAFRZ] = 1, FlexCAN stops operation automatically and enters Freeze mode before starting the move-out. If MECR[NCEFAFRZ] = 0, FlexCAN proceeds to move-out with a corrupted frame that will be transmitted on the CAN bus.
- During move-out, when an uncorrectable error occurs, a corrupted frame is copied from the selected TX MB that won the arbitration to the TX SMB for transmission. If MECR[NCEFAFRZ] = 1, FlexCAN stops operation automatically and enters Freeze mode before starting the transmission. When MECR[NCEFAFRZ] = 0, the corrupted frame is transferred from the TX SMB to the Protocol Engine (PE) subblock and is transmitted on the CAN bus.
- An uncorrectable error can also be detected beyond move-out, when TX data is read from TX SMB (buffer located in RAM) to be transferred to the PE subblock for transmission. In this case, a frame with corrupted ID or data is transmitted on the CAN bus.

  To prevent the external nodes from successfully receiving the frame, FlexCAN inverts all bits in the CRC field (CRC sequence plus CRC delimiter). Also, it transmits an error flag just after CRC delimiter due to self-detecting a Bit1 error and a form error due to the CRC field inversion. When MECR[NCEFAFRZ] = 1, FlexCAN stops operation automatically and enters Freeze mode just after the error frame. When MECR[NCEFAFRZ] = 0, FlexCAN may attempt to retransmit the same frame, as long as no other higher-priority TX MB is configured for transmission after.

  If the uncorrectable error persists, FlexCAN eventually reaches the Bus Off state due to consecutive error detections. ECR[TXERRCNT] is updated every time FlexCAN inverts the CRC field, causing errors as described above.

When MECR[NCEFAFRZ] = 1 and FlexCAN enters Freeze mode, only the CPU can cause FlexCAN to exit Freeze mode and resume Normal mode. MECR[NCEFAFRZ] becoming 1 is the only way to prevent corrupted frames from being transmitted on the CAN bus to the move-out internal process.

The error report registers can provide information to the application for customized management of these situations.

39.3.17.5 Error injection

These error injection registers are used to inject errors in memory reads to force errors and update the indication and reporting registers:

- Error Injection Address (ERRIAR)
- Error Injection Data Pattern (ERRIDPR)
- Error Injection Parity Pattern (ERRIPPR)

The relation between the error injection addresses and the corresponding addresses in the module memory map is shown in Error Injection Address (ERRIAR).

The injection is done by flipping the data and parity bits corresponding to the bits set to 1 in ERRIDPR and ERRIPPR. You can select injection specifically for memory accesses requested by the host or by FlexCAN internal processes.

For accesses larger than 32 bits, MECR[EXTERRIE] extends the injection pattern, replicating it in 32-bit words to fill the width of the access.

NOTE

It is possible, but very unlikely, for error injection to correct a bit with error. This correction does not raise the error flags and reports as expected.

To ensure coherence among error injection registers and avoid spurious error injections, you must clear MECR[HAERRIE] and MECR[FAERRIE] when configuring the memory injection registers.

39.4 External signal descriptions

FlexCAN has two I/O signals connected to the external chip pins. These signals are summarized in Table 266 and described in the following subsections.

Table 266. FlexCAN signal descriptions

Signal	Description	I/O
CAN RX	CAN receive pin	Input
CAN TX	CAN transmit pin	Output

39.4.1 CAN RX

This pin is the receive pin from the CAN bus transceiver. Logic level 0 represents its dominant state. Logic level 1 represents its recessive state.

39.4.2 CAN TX

This pin is the transmit pin to the CAN bus transceiver. Logic level 0 represents its dominant state. Logic level 1 represents its recessive state.

39.5 Initialization and application information

39.5.1 FlexCAN initialization sequence

For any configuration change or initialization, you must put FlexCAN into Freeze mode (see Freeze mode). The module must be initialized after every reset. FlexCAN memory must be initialized before switching to functional mode.

The following is a generic initialization sequence applicable to FlexCAN:

1. Initialize Module Configuration (MCR).
    a. Enable the individual filtering per message buffer and reception queue features by writing 1 to MCR[IRMQ].
    b. Enable the warning interrupts by writing 1 to MCR[WRNEN].
    c. If required, disable frame self-reception by writing 1 to MCR[SRXDIS].
    d. Enable the Legacy RX FIFO by writing 1 to MCR[RFEN] or enable the Enhanced RX FIFO by writing 1 to ERFCR[ERFEN].
    e. If Legacy RX FIFO or Enhanced RX FIFO is enabled and DMA is required, write 1 to MCR[DMA].
    f. Enable the abort mechanism by writing 1 to MCR[AEN].

g. Enable the local priority feature by writing 1 to MCR[LPRIOEN].

2. Initialize Control 1 (CTRL1) and CAN FD Bit Timing (FDCBT). Optionally initialize CAN Bit Timing (CBT).
   a. Determine the bit timing parameters: CTRL1[PROPSEG], CTRL1[PSEG1], CTRL1[PSEG2], and CTRL1[RJW].
   b. Optionally determine the bit timing parameters: CBT[EPROPSEG], CBT[EPSEG1], CBT[EPSEG2], and CBT[ERJW].
   c. Determine the CAN FD bit timing parameters: FDCBT[FPROPSEG], FDCBT[FPSEG1], FDCBT[FPSEG2], and FDCBT[FRJW].
   d. Determine the bit rate by programming CTRL1[PRESDIV] and optionally programming CBT[EPRESDIV].
   e. Determine the CAN FD bit rate by programming FDCBT[FPRESDIV].
   f. Determine the internal arbitration mode by programming CTRL1[LBUF].
3. If Error Code Correction (ECC) is enabled, you must initialize all FlexCAN memory. See Detection and correction of memory errors.
4. Initialize the message buffers. (See Message buffer structure for message buffer details.)
   a. The control and status word of all message buffers must be initialized.
   b. If RX FIFO is enabled, the ID filter table must be initialized.
   c. Other entries in each message buffer should be initialized as required.
5. Initialize Receive Individual Mask (RXIMR0 - RXIMR95).
6. Write 1 to required interrupt mask bits in:
   - IMASK registers (for all message buffer interrupts)
   - Module Configuration (MCR) (for wake-up interrupt)
   - Control 1 (CTRL1) and Control 2 (CTRL2) (for Bus Off and Error interrupts)
7. Write 0 to MCR[HALT].

After the last step listed above, FlexCAN attempts to synchronize to the CAN bus.

39.6 Memory map and register definition

This section describes the registers and data structures in FlexCAN. The base address of the module depends on the particular memory map of the chip.

39.6.1 FlexCAN memory mapping

The address space occupied by FlexCAN has 128 bytes for registers starting at the module base address, followed by embedded RAM starting at address offset 0080h.

Each individual register is identified by its complete name and corresponding mnemonic. The access type can be Supervisor (S) or Unrestricted (U). Most registers can be configured to have either Supervisor or Unrestricted access by programming MCR[SUPV]. These registers are identified as S/U in the Access column of Table 267.

NOTE

An invalid register access results in a bus error. Invalid accesses include reading a write-only register, writing a read-only register, and accessing an invalid address.

NOTE

To update the parity bits in memory properly, all FlexCAN memory must be initialized before reading registers which are implemented in memory. CTRL2[WRMFRZ] grants write access to all memory positions that require initialization, from 080h–ADFh and from C20h–31FFh. You must also initialize RX Message Buffers Global Mask (RXMGMASK), Receive 14 Mask (RX14MASK), Receive 15 Mask (RX15MASK) and Legacy RX FIFO Global Mask (RXFGMASK). MCR[RFEN] and ERFCR[ERFEN] must not be 1 during memory initialization.

Table 267. Register access and reset information

Register	Access type	Affected by hard reset	Affected by soft reset
Module Configuration (MCR)	S	Yes	Yes
Control 1 (CTRL1)	S/U	Yes	No
Free-Running Timer (TIMER)	S/U	Yes	Yes
RX Message buffers Global Mask (RXMGMASK)	S/U	No	No
RX Buffer 14 Mask (RX14MASK)	S/U	No	No
RX Buffer 15 Mask (RX15MASK)	S/U	No	No
Error Counter (ECR)	S/U	Yes	Yes
Error and Status 1 (ESR1)	S/U	Yes	Yes
Interrupt Masks 2 (IMASK2)	S/U	Yes	Yes
Interrupt Masks 1 (IMASK1)	S/U	Yes	Yes
Interrupt Flags 2 (IFLAG2)	S/U	Yes	Yes
Interrupt Flags 1 (IFLAG1)	S/U	Yes	Yes
Control 2 (CTRL2)	S/U	Yes	No
Error and Status 2 (ESR2)	S/U	Yes	Yes
Cyclic Redundancy Check (CRCR)	S/U	Yes	Yes
RX FIFO Global Mask (RXFGMASK)	S/U	No	No
RX FIFO Information (RXFIR)	S/U	No	No
CAN Bit Timing (CBT)	S/U	Yes	No
Interrupt Masks 3 (IMASK3)	S/U	Yes	Yes
Interrupt Flags 3 (IFLAG3)	S/U	Yes	Yes
Message buffers	S/U	No	No

Table continues on the next page...

Table 267. Register access and reset information (continued)

Register	Access type	Affected by hard reset	Affected by soft reset
RX Individual Masks	S/U	No	No
Memory Error Control (MECR)	S/U	Yes	Yes
Error Injection Address (ERRIAR)	S/U	Yes	Yes
Error Injection Data Pattern (ERRIDPR)	S/U	Yes	Yes
Error Injection Parity Pattern (ERRIPPR)	S/U	Yes	Yes
Error Report Address (RERRAR)	S/U	Yes	Yes
Error Report Data (RERRDR)	S/U	Yes	Yes
Error Report Syndrome (RERRSYNR)	S/U	Yes	Yes
Error Status (ERRSR)	S/U	Yes	Yes
Enhanced CAN Bit Timing Prescalers (EPRS)	S/U	Yes	No
Enhanced Nominal CAN Bit Timing (ENCBT)	S/U	Yes	No
Enhanced Data Phase CAN bit Timing (EDCBT)	S/U	Yes	No
Enhanced Transceiver Delay Compensation (ETDC)	S/U	Yes	No
CAN FD Control (FDCTRL)	S/U	Yes	No
CAN FD Bit Timing (FDCBT)	S/U	Yes	No
CAN FD CRC (FDCRC)	S/U	Yes	Yes
Enhanced RX FIFO Control (ERFCR)	S/U	Yes	Yes
Enhanced RX FIFO Interrupt Enable (ERFIER)	S/U	Yes	Yes
Enhanced RX FIFO Status (ERFSR)	S/U	Yes	Yes
High-Resolution Timestamp (HR_TIME_STAMP)	S/U	No	No
Enhanced RX FIFO	S/U	No	No
Enhanced RX FIFO Filter Element (ERFFEL)	S/U	No	No

FlexCAN can store CAN messages for transmission and reception using message buffers and RX FIFO structures.

39.6.2 FlexCAN register descriptions

The following table shows the FlexCAN memory map.

The address range from offset 80h–67Fh allocates the ninety-six 128-bit message buffers. The memory maps for the message buffers are in FlexCAN message buffer memory map.

The address range from offset 2000h–204Ch allocates the Enhanced RX FIFO output, and the address range from offset 2050h–263Ch allocates the rest of Enhanced RX FIFO 19 elements. The memory map for the Enhanced RX FIFO is in Enhanced RX FIFO structure.

39.6.2.1 FlexCAN memory map

CAN_FD1 base address: 443A_0000h

CAN_FD2 base address: 425B_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Module Configuration (MCR)	32	RW	5980_040F
4	Control 1 (CTRL1)	32	RW	0000_0000
8	Free-Running Timer (TIMER)	32	RW	0000_0000
10	RX Message Buffers Global Mask (RXMGMASK)	32	RW	See section
14	Receive 14 Mask (RX14MASK)	32	RW	See section
18	Receive 15 Mask (RX15MASK)	32	RW	See section
1C	Error Counter (ECR)	32	RW	0000_0000
20	Error and Status 1 (ESR1)	32	RW	0000_0000
24	Interrupt Masks 2 (IMASK2)	32	RW	0000_0000
28	Interrupt Masks 1 (IMASK1)	32	RW	0000_0000
2C	Interrupt Flags 2 (IFLAG2)	32	RW	0000_0000
30	Interrupt Flags 1 (IFLAG1)	32	RW	0000_0000
34	Control 2 (CTRL2)	32	RW	0060_001C
38	Error and Status 2 (ESR2)	32	R	0000_1000
44	Cyclic Redundancy Check (CRCR)	32	R	0000_0000
48	Legacy RX FIFO Global Mask (RXFGMASK)	32	RW	See section
4C	Legacy RX FIFO Information (RXFIR)	32	R	See section
50	CAN Bit Timing (CBT)	32	RW	0000_0000
6C	Interrupt Masks 3 (IMASK3)	32	RW	0000_0000
74	Interrupt Flags 3 (IFLAG3)	32	RW	0000_0000
78	External Timer (ET)	32	R	0000_0000
7C	Fault Confinement Interrupt Enable (FLTCONF_IE)	32	RW	0000_0000
880 - 9FC	Receive Individual Mask (RXIMR0 - RXIMR95)	32	RW	See section
AE0	Memory Error Control (MECR)	32	RW	800C_0080
AE4	Error Injection Address (ERRIAR)	32	RW	0000_0000
AE8	Error Injection Data Pattern (ERRIDPR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
AEC	Error Injection Parity Pattern (ERRIPPR)	32	RW	0000_0000
AF0	Error Report Address (RERRAR)	32	R	0000_0000
AF4	Error Report Data (RERRDR)	32	R	0000_0000
AF8	Error Report Syndrome (RERRSYNR)	32	R	0000_0000
AFC	Error Status (ERRSR)	32	RW	0000_0000
BF0	Enhanced CAN Bit Timing Prescalers (EPRS)	32	RW	0000_0000
BF4	Enhanced Nominal CAN Bit Timing (ENCBT)	32	RW	0000_0000
BF8	Enhanced Data Phase CAN Bit Timing (EDCBT)	32	RW	0000_0000
BFC	Enhanced Transceiver Delay Compensation (ETDC)	32	RW	0000_0000
C00	CAN FD Control (FDCTRL)	32	RW	8000_0100
C04	CAN FD Bit Timing (FDCBT)	32	RW	0000_0000
C08	CAN FD CRC (FDCRC)	32	R	0000_0000
C0C	Enhanced RX FIFO Control (ERFCR)	32	RW	0000_0000
C10	Enhanced RX FIFO Interrupt Enable (ERFIER)	32	RW	0000_0000
C14	Enhanced RX FIFO Status (ERFSR)	32	RW	0000_0000
C30 - DAC	High-Resolution Timestamp (HR_TIME_STAMP0 - HR_TIME_STAMP95)	32	RW	See section
3000 - 31FC	Enhanced RX FIFO Filter Element (ERFFEL0 - ERFFEL127)	32	RW	See section

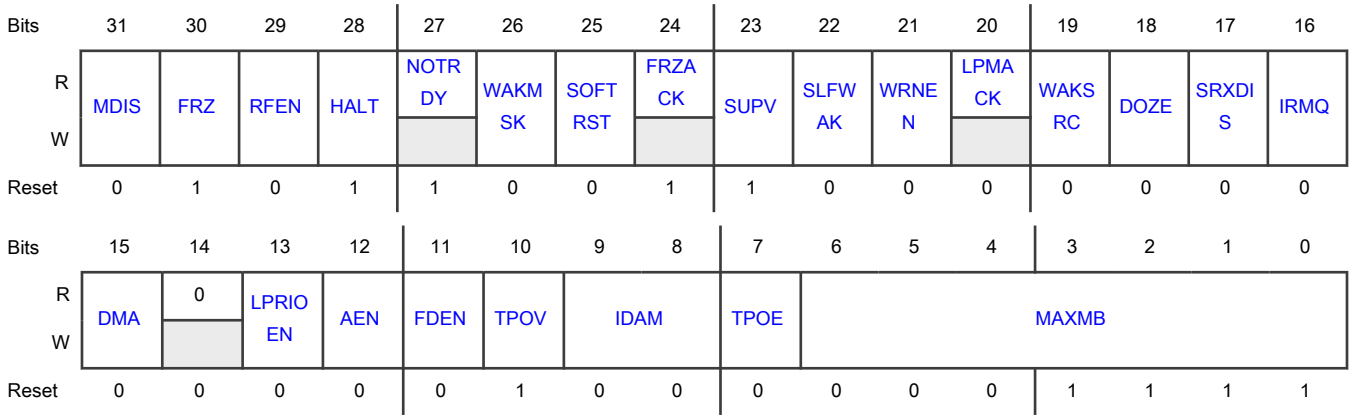
39.6.2.2 Module Configuration (MCR)

Defines global system configurations, including the module operation modes and the maximum message buffer configuration.

Offset

Register	Offset
MCR	0h

Diagram



Fields

Field	Description
31 MDIS	Module Disable Disables FlexCAN. When disabled, FlexCAN disables the clocks to the CAN Protocol Engine and Controller Host Interface submodules. Soft reset does not affect this field. 0 - Enable 1 - Disable
30 FRZ	Freeze Enable Specifies FlexCAN behavior when MCR[HALT] = 1 or when Debug mode is requested at chip level. When this field becomes 1, FlexCAN can enter Freeze mode. Writing 0 to this field causes FlexCAN to exit from Freeze mode. The chip writes 1 to this field when a noncorrectable error is detected (MECR[NCEFAFRZ] becomes 1). 0 - Disable 1 - Enable
29 RFEN	Legacy RX FIFO Enable Enables the Legacy RX FIFO feature. When this field is 1, message buffers 0–5 cannot be used for normal reception and transmission. The corresponding memory region (80h–DCh) is used by the FIFO engine and additional message buffers (up to 32, depending on CTRL2[RFFN]). These message buffers are used as Legacy RX FIFO ID filter table elements. This field also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in Table 262. This field can be written in Freeze mode only; the module blocks it in other modes. **NOTE** When CAN FD operation is enabled (see MCR[FDEN]), you cannot write 1 to this field. **NOTE** This field must not be 1 if ERFCR[ERFEN] = 1. 0 - Disable 1 - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
28 HALT	Halt FlexCAN Puts FlexCAN into Freeze mode. The CPU should write 0 to this field after initializing the message buffers and Control 1 (CTRL1) and Control 2 (CTRL2). FlexCAN performs no reception or transmission before this field becomes 0. Freeze mode cannot be entered when FlexCAN is in a low-power mode. The module writes 1 to this field when a noncorrectable error is detected and MECR[NCEFAFRZ] = 1. 0 - No request 1 - Enter Freeze mode, if MCR[FRZ] = 1.
27 NOTRDY	FlexCAN Not Ready Indicates whether FlexCAN is in Disable mode, Doze mode, Stop mode or Freeze mode. When FlexCAN has exited these modes, this field becomes 0. Soft reset does not affect this field. 0 - FlexCAN is in Normal mode, Listen-Only mode, or Loopback mode. 1 - FlexCAN is in Disable mode, Doze mode, Stop mode, or Freeze mode.
26 WAKMSK	Wake-up Interrupt Mask Enables the wake-up interrupt generation under the Self Wake-Up mechanism. 0 - Disabled 1 - Enabled
25 SOFTRST	Soft Reset Resets internal state machines of FlexCAN and some memory-mapped registers. The CPU can write 1 to this field directly. This field also becomes 1 when global soft reset is requested at the chip level. Because soft reset is synchronous and must follow a request-and-acknowledge procedure across clock domains, it may take some time to propagate its effect fully. When reset is pending, this field remains 1; it automatically becomes 0 when reset completes. You can poll this field to know when the soft reset has completed. Soft reset cannot be applied when clocks are shut down in a low-power mode. Transfer the module out of the low-power mode before applying soft reset. Soft reset does not affect this field. **NOTE** This field becomes 0 within 2 CAN bits after assertion of this bit. 0 - No reset 1 - Soft reset affects reset registers
24 FRZACK	Freeze Mode Acknowledge Indicates whether FlexCAN is in Freeze mode and its prescaler is stopped. The Freeze mode request cannot be granted until current transmission or reception processes have finished. Therefore, you can poll this field to know when FlexCAN has entered Freeze mode. If the Freeze mode request is negated, this field becomes 0 after the FlexCAN prescaler is running again. If Freeze mode is requested when FlexCAN is in a low-power mode, this field becomes 1 only when the low-power mode is exited. See Freeze mode. Soft reset does not affect this field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field becomes 1 within 180 CAN bits if current transmission or reception is ongoing for classical frames and 730 CAN bits if current transmission or reception is ongoing for FD frames after the low-power mode requested by CPU. This field becomes 0 within 2 CAN bits after the Freeze mode request removal (see Protocol timing). 0 - Not in Freeze mode, prescaler running. 1 - In Freeze mode, prescaler stopped.
23 SUPV	Supervisor Mode Configures FlexCAN to be either in Supervisor or User mode. The registers that this field affects are marked as S/U in the Access type column of Table 267. The affected registers start with Supervisor access allowance only. In User mode, affected registers allow both Supervisor and Unrestricted accesses. In Supervisor mode, affected registers allow only Supervisor access. Unrestricted access behaves as though the access is performed on an unimplemented register location. This field can be written only in Freeze mode; the module blocks it in other modes. 0 - User mode 1 - Supervisor mode
22 SLFWAK	Self-Wake-Up Feature Enables the self-wake-up feature when FlexCAN is in a low-power mode other than Disable mode. When this feature is enabled, the FlexCAN module monitors the bus for a wake-up event (that is, a recessive-to-dominant transition). If a wake-up event is detected during Doze mode, FlexCAN requests to resume its clocks. If enabled to do so, it also generates a wake-up interrupt to the CPU. If a wake-up event is detected during Stop mode, FlexCAN generates, if enabled to do so, a wake-up interrupt to the CPU. The CPU can exit Stop mode globally and FlexCAN can request to resume the clocks. When FlexCAN is in a low-power mode other than Disable mode, this field cannot be written, as the module blocks it. 0 - Disable 1 - Enable
21 WRNEN	Warning Interrupt Enable Enables the generation of the flags ESR1[TWRNINT] and ESR1[RWRNINT]. When this field is 1, TWRNINT and RWRNINT flags are set when the respective error counter transitions from less than 96 to greater than or equal to 96. When this field is 0, the TWRNINT and RWRNINT flags are always zero, independent of the values of the error counters. No warning interrupt is generated. This field can be written in Freeze mode only; the module blocks it in other modes. 0 - Disable 1 - Enable
20	Low-Power Mode Acknowledge

Table continues on the next page...

Table continued from the previous page...

Field	Description
LPMACK	Indicates whether FlexCAN is in a low-power mode (Disable mode, Doze mode, Stop mode). A low-power mode cannot be entered until all current transmission and reception processes have finished. The CPU can poll this field to know when FlexCAN has entered low-power mode. Soft reset does not affect this field. **NOTE** This field becomes 1 within 180 CAN bits if current transmission or reception is ongoing for classical frames and 730 CAN bits if current transmission or reception is ongoing for FD frames after the low-power mode requested by CPU. This field becomes 0 within 2 CAN bits after the low-power mode request removal (see Protocol timing). 0 - Not in a low-power mode 1 - In a low-power mode
19 WAKSRC	Wake-Up Source Determines whether the integrated low-pass filter is applied to the RX input that FlexCAN uses to detect recessive-to-dominant edges on the CAN bus. This filter can protect the RX CAN input from spurious wake-up. This field can be written only in Freeze mode. The module blocks it in other modes. 0 - No filter applied 1 - Filter applied
18 DOZE	Doze Mode Enable Determines whether FlexCAN can enter low-power mode when Doze mode is requested at chip level. This field is automatically reset when FlexCAN wakes from Doze mode upon detecting activity on the CAN bus (self wake-up enabled). 0 - Disable 1 - Enable
17 SRXDIS	Self-Reception Disable Determines whether FlexCAN can receive frames transmitted by itself. If 1, frames transmitted by the module are not stored in any MB, regardless of whether the MB is programmed with an ID that matches the transmitted frame. No interrupt flag or interrupt signal is generated due to the frame reception. This field can be written only in Freeze mode; the module blocks it in other modes. 0 - Enable 1 - Disable
16 IRMQ	Individual RX Masking and Queue Enable Indicates whether RX matching process is based on individual masking and queue, or based on a masking scheme with RX Message Buffers Global Mask (RXMGMASK), Receive 14 Mask (RX14MASK), Receive 15 Mask (RX15MASK), and Legacy RX FIFO Global Mask (RXFGMASK). When this field is disabled, for backward compatibility with legacy applications, reading the Control and Status word locks the MB even if it is empty. This field can be written in Freeze mode only. The module blocks it in other modes. 0 - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable
15 DMA	DMA Enable Enables DMA. The DMA feature can only be used in Legacy RX FIFO or Enhanced RX FIFO, so MCR[RFEN] or ERFCR[ERFEN] must be 1. When DMA and RFEN are 1, IFLAG1[BUF5I] generates the DMA request, and no RX FIFO interrupt is generated. This field can be written in Freeze mode only; the module blocks it in other modes. 0 - Disable 1 - Enable
14 —	Reserved
13 LPRIOEN	Local Priority Enable Enables the local priority feature. It is used to expand the ID used during the arbitration process. With this expanded ID concept, the arbitration process is done based on the full 32-bit word. However, the actual transmitted ID is 11 bits for standard frames and 29 bits for extended frames. This field can be written only in Freeze mode; the module blocks it in other modes. This bit is provided for backward compatibility with legacy applications. 0 - Disable 1 - Enable
12 AEN	Abort Enable Enables the TX abort mechanism. This mechanism guarantees a safe procedure for aborting a pending transmission, so that no frame is sent in the CAN bus without notification. This field can be written only in Freeze mode; the module blocks it in other modes. **NOTE** When this field is 1, only use the abort mechanism (see Transmission abort mechanism) to update message buffers configured for transmission. **CAUTION** Writing the abort code into RX message buffers can cause unpredictable results when this field is 1. 0 - Disabled 1 - Enabled
11 FDEN	CAN FD Operation Enable Enables the CAN with flexible data rate (CAN FD) operation. This field can be written in Freeze mode only. FlexCAN can receive and transmit messages in CAN 2.0 format. If this field is enabled, FlexCAN can also receive and transmit messages in CAN FD format. FlexCAN can transmit FD frame format according to ISO 11898-1_2024.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE If the value of this field is 1, the Legacy RX FIFO Enable (MCR[RFEN]) field cannot be 1. 0 - Disable 1 - Enable
10 TPOV	TX Pin Override Value Indicates forced value on the TX pin. This bit is affected by soft reset. 0 - TX pin is forced to be dominant 1 - TX pin is forced to be recessive
9-8 IDAM	ID Acceptance Mode Identifies the format of the Legacy RX FIFO ID filter table elements. This field configures all elements of the table at the same time; they are all the same format. See Legacy RX FIFO structure. This field can be written only in Freeze mode; the module blocks it in other modes. 00 - Format A: One full ID (standard and extended) per ID filter table element. 01 - Format B: Two full standard IDs or two partial 14-bit (standard and extended) IDs per ID filter table element. 10 - Format C: Four partial 8-bit standard IDs per ID filter table element. 11 - Format D: All frames rejected.
7 TPOE	TX Pin Override Enable Enables forcing of the TX pin to recessive or dominant. This bit is affected by soft reset. **NOTE** This bit must not be enabled during the functional mode. 0 - TX pin forcing is disabled 1 - TX pin forcing is enabled
6-0 MAXMB	Number of the Last Message Buffer Defines the number of the last message buffer that takes part in the matching and arbitration processes. The reset value (0Fh) is equivalent to a 16-MB configuration. This field can be written only in Freeze mode; the module blocks it in other modes. **NOTE** You must write a value smaller than or equal to the number of available message buffers to this field, as described in FlexCAN memory partition for CAN FD. Additionally, the MAXMB value must consider the region of message buffers occupied by Legacy RX FIFO and its ID filters table space defined by CTRL2[RFFN]. MAXMB also impacts the definition of the minimum number of peripheral clocks per CAN bit, as described in Table 262.

39.6.2.3 Control 1 (CTRL1)

Contains specific FlexCAN control features related to the CAN bus. These features include bit rate, programmable sampling point within an RX bit, Loopback mode, Listen-Only mode, Bus Off recovery behavior, and interrupt enabling (Bus-Off, Error, Warning). It also determines the division factor for the clock prescaler.

The CAN bit timing variables (CTRL1[PRESDIV], CTRL1[PROPSEG], CTRL1[PSEG1], CTRL1[PSEG2], and CTRL1[RJW]) can also be configured in CAN Bit Timing (CBT), which extends the range of all these variables. If CBT[BTF] = 1, CTRL1[PRESDIV], CTRL1[PROPSEG], CTRL1[PSEG1], CTRL1[PSEG2], and CTRL1[RJW] become read-only.

If CTRL2[BTE] = 1, CTRL1[PRESDIV], CTRL1[PROPSEG], CTRL1[PSEG1], CTRL1[PSEG2], and CTRL1[RJW] are not used by the module. Instead, these fields are read as zero, and a write operation to them has no effect.

NOTE

When the CAN FD feature is enabled, do not use CTRL1[PRESDIV], CTRL1[PROPSEG], CTRL1[PSEG1], CTRL1[PSEG2], and CTRL1[RJW] for CAN bit timing. Instead use CBT[EPRESDIV], CBT[ERJW], CBT[EPSEG1], CBT[EPSEG2], and CBT[EPROPSEG].

The CAN bit variables in CTRL1 and in CBT are stored in the same internal register.

Soft reset does not affect the contents of this register.

Offset

Register	Offset
CTRL1	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PRESDIV								RJW		PSEG1		PSEG2			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BOFF MSK	ERRM SK	CLKS RC	LPB	TWRN MSK	RWRN MSK	0	ROM	SMP	BOFF REC	TSYN	LBUF	LOM	PROPSEG		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 PRESDIV	Prescaler Division Factor Determines the ratio between the PE clock frequency and the serial clock (Sclock) frequency. The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency. The maximum value of this field is FFh, which gives a minimum Sclock frequency equal to the PE clock frequency divided by 256. See Protocol timing for more information. This field can be written only in Freeze mode; the module blocks it in other modes.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Sclock frequency = PE clock frequency ÷ (PRESDIV + 1).
23-22 RJW	Resync Jump Width Defines the maximum number of time quanta that one resynchronization can change a bit time. One time quantum is equal to one Sclock period. The valid programmable values are 0–3. See Protocol timing for more information. This field can be written only in Freeze mode; the module blocks it in other modes. Resync Jump Width = RJW + 1.
21-19 PSEG1	Phase Segment 1 Defines the length of phase segment 1 in the bit time. The valid programmable values are 0–7. See Protocol timing for more information. This field can be written only in Freeze mode; the module blocks it in other modes. Phase Buffer Segment 1 = (PSEG1 + 1) × Time Quanta.
18-16 PSEG2	Phase Segment 2 Defines the length of phase segment 2 in the bit time. The valid programmable values are 1–7. See Protocol timing for more information. This field can be written only in Freeze mode; the module blocks it in other modes. Phase Buffer Segment 2 = (PSEG2 + 1) × Time Quanta.
15 BOFFMSK	Bus Off Interrupt Mask Provides a mask for the Bus Off interrupt ESR1[BOFFINT]. 0 - Interrupt disabled 1 - Interrupt enabled
14 ERRMSK	Error Interrupt Mask Provides a mask for the Error interrupt ESR1[ERRINT]. 0 - Interrupt disabled 1 - Interrupt enabled
13 CLKSRC	CAN Engine Clock Source Selects the clock source to the CAN Protocol Engine (PE) to be either the peripheral clock or the oscillator clock. When the oscillator clock is selected, the oscillator clock frequency must be lower than the bus clock. The selected clock is fed to the prescaler to generate the serial clock (Sclock). To guarantee reliable operation, this field can be written only in Disable mode. The module blocks it in other modes. See Protocol timing. Ensure the protocol engine clock tolerance according to the CAN Protocol standard ISO 11898-1_2024. To identify the proper clock source, see the clock distribution chapter (module clocks table). If this field is 0, you can further select the peripheral clock source according to the Clock Root section of CCM chapter. 0 - Peripheral clock

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Bus clock
12 LPB	Loopback Mode Configures FlexCAN to operate in Loopback mode. In this mode, FlexCAN performs an internal loopback that can be used for self-test operation. The bit stream output of the transmitter is fed back internally to the receiver input. The RX CAN input pin is ignored and the TX CAN output goes to the recessive state (logic 1). FlexCAN behaves as it normally does when transmitting, and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field. It generates an internal acknowledge bit to ensure proper reception of its own message. Both transmit and receive interrupts are generated. This field can be written only in Freeze mode; the module blocks it in other modes. **NOTE** In this mode, MCR[SRXDIS] cannot become 1, because it would impede the self-reception of a transmitted message. **NOTE** FDCTRL[TDCEN] and MCR[ETDCEN] must be 0 when this field is 1. 0 - Disabled 1 - Enabled
11 TWRNMSK	TX Warning Interrupt Mask Provides a mask for the TX Warning interrupt associated with the ESR1[TWRNINT] flag. When MCR[WRNEN] = 0, this field is read as 0. This field can be written only if MCR[WRNEN] = 1. 0 - Disabled 1 - Enabled
10 RWRNMSK	RX Warning Interrupt Mask Provides a mask for the RX Warning interrupt associated with the ESR1[RWRNINT] flag. When MCR[WRNEN] = 0, this field is read as 0. This field can be written only if MCR[WRNEN] = 1. 0 - Disabled 1 - Enabled
9 —	Reserved
8 ROM	Restricted Operation Mode Enables restricted operation mode. In restricted operation mode, FlexCAN is able to receive and acknowledge CAN messages, but it cannot send error frames and overload frames. If an error or overload condition is detected, FlexCAN treats it as a protocol exception and enters an integrating state. In restricted operation mode, FlexCAN does not increment or decrement the error counters. Besides,

Table continues on the next page...

Table continued from the previous page...

Field	Description
	FlexCAN is able to transmit time reference messages of potential time controllers in a network, but other types of frames must not be configured to be transmitted. **NOTE** This field can be written in freeze mode only. 0 - Restricted operation is disabled 1 - Restricted operation is enabled
7 SMP	CAN Bit Sampling Determines the sampling mode of CAN bits at the RX input. This field can be written in Freeze mode only; the module blocks it in other modes. **NOTE** For proper operation, to write 1 to this field, you must guarantee a minimum value of two time quanta in CTRL1[PSEG1] (or CBT[EPSEG1]).This bit cannot become 1 when CAN FD is enabled (MCR[FDEN] = 1). 0 - One sample is used to determine the bit value. 1 - Three samples are used to determine the value of the received bit: the regular one (sample point) and two preceding samples. A majority rule is used.
6 BOFFREC	Bus Off Recovery Determines how FlexCAN recovers from Bus Off state. If 0, automatic recovering from Bus Off state occurs according to the CAN Specification 2.0B. If 1, automatic recovering from Bus Off is disabled. The module remains in Bus Off state until you write 1 to this field. If this field becomes 0 before 128 sequences of 11 recessive bits are detected on the CAN bus, Bus Off recovery happens as if this field had never become 1. If this field becomes 0 after 128 sequences of 11 recessive bits occurred, FlexCAN resynchronizes to the bus. It waits for 11 recessive bits before joining the bus. After this field becomes 0, it can become 1 again during Bus Off, but it will only be effective the next time the module enters Bus Off. If this field becomes 0 when the module is in Bus Off, writing 1 to this field is not effective for the current Bus Off recovery. **NOTE** See Bus Off in the CAN protocol standard ISO 11898-1_2024 for details. 0 - Enabled 1 - Disabled
5 TSYN	Timer Sync Enables a mechanism that resets the free-running timer each time a message is received in message buffer 0. This feature provides the means to synchronize multiple FlexCAN stations with a special "SYNC" message (that is, global network time). If MCR[RFEN] = 1 (Legacy RX FIFO enabled), the first available message buffer, according to CTRL2[RFFN], is used for timer synchronization instead of MB0. This field can be written in Freeze mode only; the module blocks it in other modes.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Disable 1 - Enable
4 LBUF	Lowest Buffer Transmitted First Determines the ordering mechanism for message buffer transmission. When 1, MCR[LPRIOEN] does not affect the priority arbitration. This field can be written in Freeze mode only; the module blocks it in other modes. 0 - Buffer with highest priority is transmitted first. 1 - Lowest number buffer is transmitted first.
3 LOM	Listen-Only Mode Configures FlexCAN to operate in Listen-Only mode. In this mode, transmission is disabled, all error counters described in Error Counter (ECR) are frozen, and the module operates in CAN Error Passive mode. Only messages acknowledged by another CAN station are received. If FlexCAN detects an unacknowledged message, it flags a BIT0 error without changing the receive error counter (ECR[RXERRCNT]), as if it is trying to acknowledge the message. FlexCAN acknowledges Listen-Only mode when ESR1[FLTCONF] = 1, indicating the Error Passive state. There can be some delay between the Listen-Only mode request and its acknowledgment. This field can be written in Freeze mode only; the module blocks it in other modes. 0 - Listen-Only mode is deactivated. 1 - FlexCAN module operates in Listen-Only mode.
2-0 PROPSEG	Propagation Segment Defines the length of the propagation segment in the bit time. The valid programmable values are 0–7. This field can be written only in Freeze mode; the module blocks it in other modes. Propagation segment time = (PROPSEG + 1) × Time Quanta. One Time Quantum = one Sclock period.

39.6.2.4 Free-Running Timer (TIMER)

Represents a 16-bit free-running counter that the CPU can read and write. The timer starts from 0h after reset, counts linearly to FFFFh, and wraps around.

When CTRL2[TIMER_SRC] = 1, an external time tick continuously increments the timer. The time tick must be synchronous to the peripheral clock, with a minimum pulse width of one clock cycle.

When CTRL2[TIMER_SRC] = 0, the CAN bit clock increments the timer, which defines the baud rate on the CAN bus. During a message transmission or reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate. The timer is not incremented during Disable, Doze, Stop and Freeze modes.

The timer value is captured when the second bit of the identifier field of any frame is on the CAN bus. This captured value is written into the timestamp entry in a message buffer after a successful reception or transmission of a message.

If CTRL1[TSYN] = 1, the timer is reset whenever a message is received in the first available message buffer, according to CTRL2[RFFN].

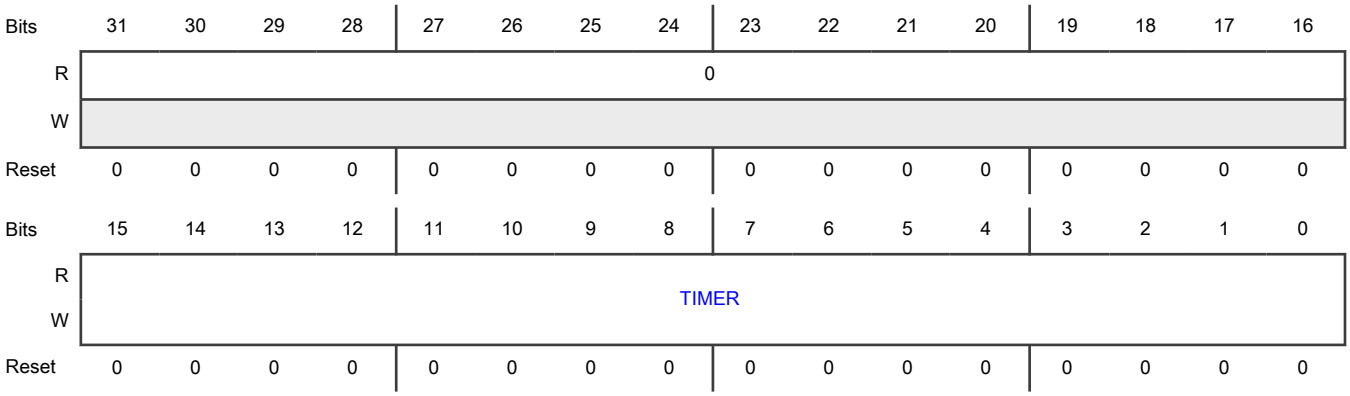
The CPU can write to this register anytime. However, if the write occurs simultaneously with the timer being reset by a reception in the first message buffer, the write value is discarded.

Reading this register affects the message buffer unlocking procedure (see Message buffer lock mechanism).

Offset

Register	Offset
TIMER	8h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 TIMER	Timer Value Contains the free-running counter value.

39.6.2.5 RX Message Buffers Global Mask (RXMGMASK)

Masks the filter bits of all RX message buffers, excluding MB14 and MB15, which have individual mask registers.

This register is located in RAM.

RXMGMASK is provided for legacy application support.

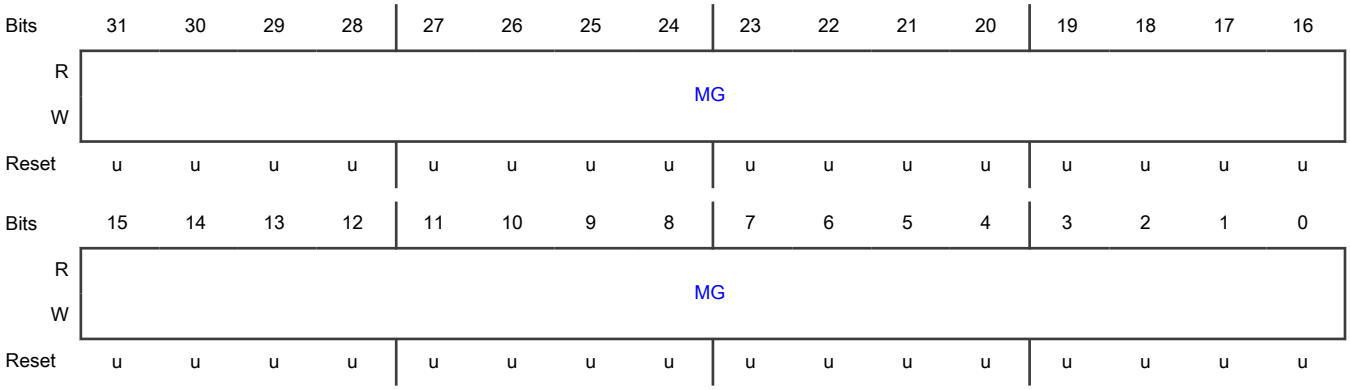
- When MCR[IRMQ] is 0, RXMGMASK is always in effect. The bits in RXMGMASK[MG] mask the MB filter bits.
- When MCR[IRMQ] is 1, RXMGMASK has no effect. The bits in RXMGMASK[MG] do not mask the MB filter bits.

This register can only be written in Freeze mode; the module blocks it in other modes.

Offset

Register	Offset
RXMGMASK	10h

Diagram



Fields

Field	Description
31-0 MG	Global Mask for RX Message Buffers Masks the message buffer filter bits. The alignment with the ID word of the message buffer is imperfect. The two most significant MG bits affect the fields RTR and IDE, which are located in the Control and Status word of the MB. The following table shows which MG bits mask each MB filter field.

SMB[RTR] [1]	CTRL2[RRS]	CTRL2[EACEN]	Message buffer filter fields			
			MB[RTR]	MB[IDE]	MB[ID]	Reserved
0	-	0	Note [2]	Note [3]	MG[28:0]	MG[31:29]
0	-	1	MG[31]	MG[30]	MG[28:0]	MG[29]
1	0	-	-	-	-	MG[31:0]
1	1	0	-	-	MG[28:0]	MG[31:29]
1	1	1	MG[31]	MG[30]	MG[28:0]	MG[29]

1. RTR bit of the incoming frame. It is saved into an auxiliary MB called RX serial message buffer (RX SMB).
2. If CTRL2[EACEN] is 0, the RTR bit of MB is never compared with the RTR bit of the incoming frame.
3. If CTRL2[EACEN] is 0, the IDE bit of MB is always compared with the IDE bit of the incoming frame.

0b - The corresponding bit in the filter is "don't care."

1b - The corresponding bit in the filter is checked.

39.6.2.6 Receive 14 Mask (RX14MASK)

Masks the filter fields of MB14.

This register is located in RAM.

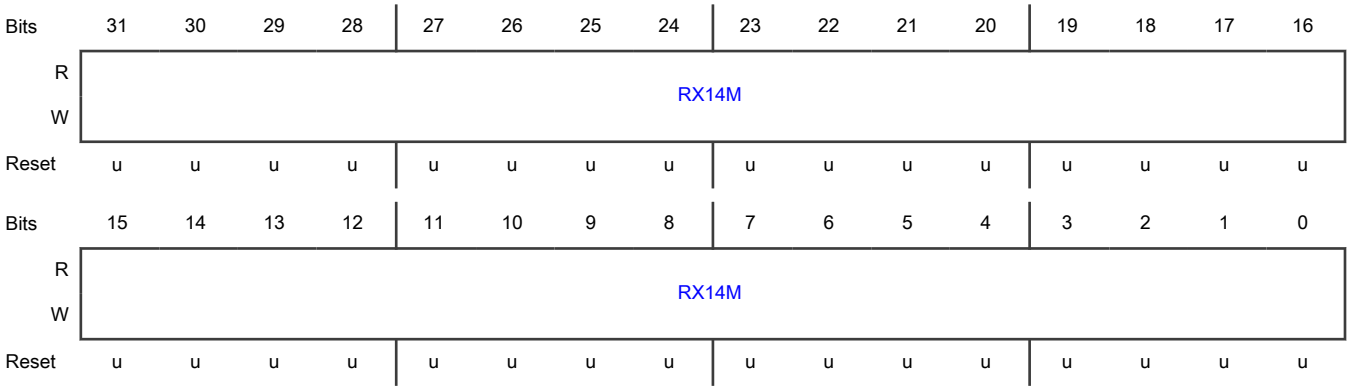
RX14MASK is provided for legacy application support. When MCR[IRMQ] = 1, RX14MASK has no effect.

This register can only be programmed when the module is in Freeze mode; the module blocks it in other modes.

Offset

Register	Offset
RX14MASK	14h

Diagram



Fields

Field	Description
31-0 RX14M	RX Buffer 14 Mask Bits Masks the corresponding MB14 filter field in the same way that RX Message Buffers Global Mask (RXMGMASK) masks the filters of the other message buffers. 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

39.6.2.7 Receive 15 Mask (RX15MASK)

Masks the filter fields of MB15.

This register is located in RAM.

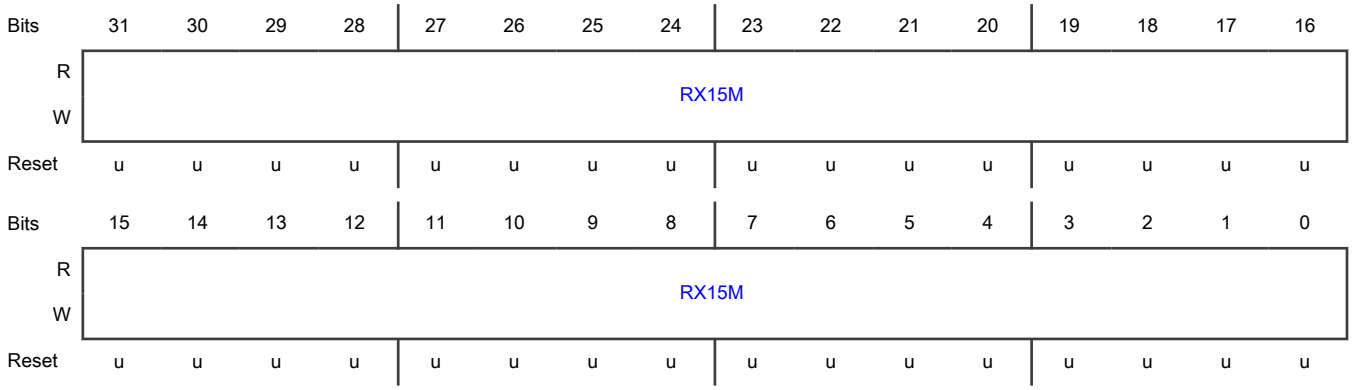
RX15MASK is provided for legacy application support. When MCR[IRMQ] = 1, RX15MASK has no effect.

This register can be programmed only when the module is in Freeze mode; the module blocks it in other modes.

Offset

Register	Offset
RX15MASK	18h

Diagram



Fields

Field	Description
31-0 RX15M	RX Buffer 15 Mask Bits Masks the corresponding MB15 filter field in the same way that RX Message Buffers Global Mask (RXMGMASK) masks the filters of other message buffers. 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

39.6.2.8 Error Counter (ECR)

Contains error counters for received and transmitted messages.

TXERRCNT and RXERRCNT consider all errors in both CAN FD and non-FD message formats. TXERRCNT_FAST and RXERRCNT_FAST count only the errors that occur in the data phase of CAN FD frames that have BRS = 1.

The Fault Confinement state (ESR1[FLTCONF]) is updated based on TXERRCNT and RXERRCNT counters only. The rules for increasing and decreasing these counters are described in the CAN protocol and are entirely implemented in FlexCAN.

The basic rules for FlexCAN bus state transitions are:

- If the value of TXERRCNT or RXERRCNT becomes greater than or equal to 128, ESR1[FLTCONF] is updated to reflect Error Passive state.
- If the state of FlexCAN is Error Passive, and TXERRCNT or RXERRCNT decrements to a value less than 128 when the other already satisfies this condition, ESR1[FLTCONF] is updated to reflect Error Active state.
- If the value of TXERRCNT becomes greater than 255, ESR1[FLTCONF] is updated to reflect Bus Off state, and an interrupt may be issued. The value of TXERRCNT is then reset to zero.
- If FlexCAN is in Bus Off, TXERRCNT is cascaded with another internal counter to count the occurrences of 11 consecutive recessive bits on the bus. TXERRCNT is reset to zero. It counts in a manner where the internal counter counts 11 such bits and then wraps around when incrementing the TXERRCNT. When TXERRCNT reaches the value of 128, ESR1[FLTCONF] is updated to Error Active, and both error counters are reset to zero. Upon any instance of a dominant bit following a stream of less than 11 consecutive recessive bits, the internal counter resets itself to zero without affecting the TXERRCNT value. The TXERRCNT_FAST counter is frozen during Bus Off.
- If only one node is operating during system startup, its TXERRCNT increases upon each attempted message transmission, as a result of acknowledge errors (indicated by ESR1[ACKERR]). After the transition to Error Passive state, TXERRCNT no longer increments upon acknowledge errors. The chip never goes into the Bus Off state.

- If RXERRCNT increases to a value greater than 127, it is not incremented further, even if more errors are detected when being a receiver. At the next successful message reception, the counter is set to a value between 119 and 127 to return to the Error Active state.
- TXERRCNT_FAST and RXERRCNT_FAST error counter values increment and decrement based on errors detected only in the data phase of CAN FD frames that have BRS = 1. These counters follow the same increment and decrement rules as TXERRCNT and RXERRCNT. These counters do not wrap around and get stuck at their maximum value (255). They stop counting and keep their values frozen when FlexCAN is in the Bus Off state. They are reset when FlexCAN leaves the Bus Off state and resume counting after FlexCAN returns to the Error Active state.

NOTE

See Fault confinement in the CAN Protocol standard ISO 11898-1_2024 for details.

Offset

Register	Offset
ECR	1Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXERRCNT_FAST								TXERRCNT_FAST							
W	0								0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXERRCNT								TXERRCNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 RXERRCNT_FAST	Receive Error Counter for Fast Bits Counts errors detected in the data phase of received CAN FD messages that have BRS = 1. This field is read-only except in Freeze mode, when the CPU can write an 8-bit zero value only.
23-16 TXERRCNT_FAST	Transmit Error Counter for Fast Bits Counts errors detected in the data phase of transmitted CAN FD messages that have BRS = 1. This field is read-only except in Freeze mode, when the CPU can write an 8-bit zero value only.
15-8 RXERRCNT	Receive Error Counter Counts all errors detected in received messages. This field is read-only except in Freeze mode, when the CPU can write to it.
7-0	Transmit Error Counter

Table continues on the next page...

Table continued from the previous page...

Field	Description
TXERRCNT	Counts all errors detected in transmitted messages. This field is read-only except in Freeze mode, when the CPU can write to it.

39.6.2.9 Error and Status 1 (ESR1)

Reports various error conditions detected in the reception and transmission of a CAN frame. This register provides status information about the chip, and is the source of some interrupts to the CPU. The reported error conditions are:

NOTE

Reading host can clear these fields.

- Errors detected in CAN frames of any format:
  - BIT1ERR
  - BIT0ERR
  - ACKERR
  - CRCERR
  - FRMERR
  - STFERR
- Errors detected in the data phase of CAN FD frames with the BRS bit set only:
  - BIT1ERR_FAST
  - BIT0ERR_FAST
  - CRCERR_FAST
  - FRMERR_FAST
  - STFERR_FAST

One or more error flags may report an error detected in a single CAN frame. To account for more error events occurring in subsequent frames when the CPU does not attempt to read this register, error reporting is cumulative.

Status flags:

- TXWRN
- RXWRN
- IDLE
- TX
- FLTCONF
- RX
- SYNCH

Interrupt flags:

- BOFFINT
- BOFFDONEINT
- ERRINT
- ERRINT_FAST

- WAKINT
- TWRNINT
- RWRNINT

The CPU should follow this procedure when servicing interrupt requests generated by these flags:

1. Read this register to capture all error condition and status flags. This action clears the respective flags that were set since the last read access.
2. Write 1 to clear the interrupt flag that triggered the interrupt request.
3. Write 1 to clear the ERROVR flag, if it is set.

Starting from all error flags cleared, a first error event sets either ERRINT or ERRINT_FAST (provided the corresponding mask bit is 1). If other error events in subsequent frames occur before the CPU serves the interrupt request, the ERROVR flag is set to indicate that errors from different frames have accumulated.

Table 268. CAN bus status

SYNCH	IDLE	TX	RX	FlexCAN state
0	0	0	0	Not synchronized to CAN bus
1	1	X	X	Idle
1	0	1	0	Transmitting
1	0	0	1	Receiving

NOTE

See Fault confinement in the CAN Protocol standard ISO 11898-1_2024 for details.

Offset

Register	Offset
ESR1	20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BIT1ERR...	BIT0ERR...	0	CRCERR_...	FRMERR_...	STFERR_...	0	0	PTA	ATP	ERROVR	ERRINT_...	BOFFDON...	SYNCH	TWRNINT	RWRNINT
W									W1C	W1C	W1C	W1C	W1C		W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BIT1ERR	BIT0ERR	ACKERR	CRCERR	FRMERR	STFERR	TXWRN	RXWRN	IDLE	TX	FLTCONF		RX	BOFFINT	ERRINT	WAKINT
W														W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 BIT1ERR_FAST	Fast Bit1 Error Flag Indicates when an inconsistency occurs between the transmitted and the received bit in the data phase of CAN FD frames that have BRS = 1. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - At least one bit transmitted as recessive is received as dominant.
30 BIT0ERR_FAST	Fast Bit0 Error Flag Indicates when an inconsistency occurs between the transmitted and the received bit in the data phase of CAN FD frames that have BRS = 1. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - At least one bit transmitted as dominant is received as recessive.
29 —	Reserved
28 CRCERR_FAST	Fast Cyclic Redundancy Check Error Flag Indicates that the receiver node has detected a CRC error in the CRC field of CAN FD frames that have BRS = 1. This error means that the calculated CRC is different from the received CRC. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - A CRC error occurred since last read of this register.
27 FRMERR_FAST	Fast Form Error Flag Indicates whether the receiver node has detected a form error in the data phase of CAN FD frames that have BRS = 1. This error means that a fixed-form bit field contains at least one illegal bit. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - A form error occurred since last read of this register.
26 STFERR_FAST	Fast Stuffing Error Flag Indicates that a stuffing error has been detected in the data phase of CAN FD frames that have BRS = 1. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - A stuffing error occurred since last read of this register.
25 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
24 —	Reserved
23 PTA	Passive to Active State If this field is 1, controller is transitioned from passive to active error state. If this field is 0, controller is not transitioned from passive to active error state. Interrupt signals are asserted if enabled from FLTCONF_IE when the controller is transitioned from passive to active error state. 0 - Does not transition 1 - Transitions
22 ATP	Active to Passive State If this field is 1, controller is transitioned from active to passive error state. If this field is 0, controller is not transitioned from active to passive error state. Interrupt signals are asserted if enabled from FLTCONF_IE when the controller is transitioned from active to passive error state. 0 - Does not transition 1 - Transitions
21 ERROVR	Error Overrun Flag Indicates that an error condition occurred when any error flag is already set. 0 - No overrun 1 - Overrun
20 ERRINT_FAST	Fast Error Interrupt Flag Indicates that at least one error flag detected in the data phase of CAN FD frames that have BRS = 1 (BIT1ERR_FAST, BIT0ERR_FAST, CRCERR_FAST, FRMERR_FAST, or STFERR_FAST) is set. If CTRL2[ERRMSK_FAST] = 1, an interrupt is generated to the CPU. 0 - No such occurrence. 1 - Error flag set in the data phase of CAN FD frames that have BRS = 1.
19 BOFFDONEINT	Bus Off Done Interrupt Flag Indicates whether ECR[TXERRCNT] has finished counting 128 occurrences of 11 consecutive recessive bits on the CAN bus and is ready to leave Bus Off. If CTRL2[BOFFDONEMSK] = 1, an interrupt is generated to the CPU. 0 - No such occurrence 1 - FlexCAN module has completed Bus Off process.
18 SYNCH	CAN Synchronization Status Flag Indicates whether FlexCAN is synchronized to the CAN bus and able to participate in the communication process. FlexCAN sets and clears this flag. See the table in Error and Status 1 (ESR1). 0 - Not synchronized

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Synchronized
17 TWRNINT	TX Warning Interrupt Flag Indicates whether TX error counter changed from less than 96 to greater than or equal to 96. If MCR[WRNEN] = 1, this flag is set when the TXWRN flag transitions from 0 to 1, meaning that the TX error counters reached 96. If CTRL1[TWRNMSK] = 1, an interrupt is sent to the CPU. When MCR[WRNEN] = 0, this flag is masked. The CPU must clear this flag before writing 0 to MCR[WRNEN]. Otherwise, this flag is set when MCR[WRNEN] = 1 again. Writing 0 has no effect. This flag is not generated when in the Bus Off state. This flag is not updated during Freeze mode. 0 - No such occurrence 1 - TX error counter changed from less than 96 to greater than or equal to 96.
16 RWRNINT	RX Warning Interrupt Flag Indicates whether the RX error counter changed from less than 96 to greater than or equal to 96. If MCR[WRNEN] = 1, this flag is set when the RXWRN flag transitions from 0 to 1, meaning that the RX error counters reached 96. If CTRL1[RWRNMSK] = 1, an interrupt is sent to the CPU. When MCR[WRNEN] = 0, this flag is masked. The CPU must clear this flag before writing 0 to MCR[WRNEN]. Otherwise, this flag is set when MCR[WRNEN] = 1 again. Writing 0 has no effect. This flag is not updated during Freeze mode. 0 - No such occurrence 1 - RX error counter changed from less than 96 to greater than or equal to 96.
15 BIT1ERR	Bit1 Error Flag Indicates when an inconsistency occurs between the transmitted and the received bit in a non-CAN FD message or in the arbitration or data phase of a CAN FD message. **NOTE** A transmitter does not set this flag for an arbitration field or ACK slot. It is not set for a node sending an error passive flag that detects dominant bits. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - At least one bit sent as recessive is received as dominant.
14 BIT0ERR	Bit0 Error Flag Indicates when an inconsistency occurs between the transmitted and the received bit in a non-CAN FD message or in the arbitration or data phase of a CAN FD message. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - At least one bit sent as dominant is received as recessive.
13	Acknowledge Error Flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
ACKERR	Indicates whether the transmitter node has detected an acknowledge error. This error means that a dominant bit has not been detected during the ACK SLOT. After a read operation, the field's value clears to 0. 0 - No error 1 - Error occurred since last read of this register.
12 CRCERR	Cyclic Redundancy Check Error Flag Indicates whether the receiver node has detected a cyclic redundancy check (CRC) error either in a non-FD message or in the arbitration or data phase of a frame in CAN FD format. This error means that the calculated CRC is different from the received. After a read operation, the field's value clears to 0. 0 - No error 1 - Error occurred since last read of this register.
11 FRMERR	Form Error Flag Indicates whether a form error has been detected in a non-FD message or in the arbitration or data phase of an FD message by the receiver node. This error means that a fixed-form field contains at least one illegal bit. After a read operation, the field's value clears to 0. 0 - No error 1 - Error occurred since last read of this register.
10 STFERR	Stuffing Error Flag Indicates that a stuffing error has been detected in a non-FD message or in the arbitration or data phase of an FD message by the receiver node. After a read operation, the field's value clears to 0. 0 - No error 1 - Error occurred since last read of this register.
9 TXWRN	TX Error Warning Flag Indicates when repetitive errors occur during message transmission. Only the value of ECR[TXERRCNT] affects this flag. This flag is not updated during Freeze mode. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - TXERRCNT is 96 or greater.
8 RXWRN	RX Error Warning Flag Indicates when repetitive errors occur during message reception. Only the value of ECR[RXERRCNT] affects this flag. This flag is not updated during Freeze mode. After a read operation, the field's value clears to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - No such occurrence. 1 - RXERRCNT is greater than or equal to 96.
7 IDLE	Idle Indicates whether CAN bus is in IDLE state. See Table 268. 0 - Not IDLE 1 - IDLE
6 TX	FlexCAN In Transmission Indicates whether FlexCAN is transmitting a message. See Table 268. 0 - Not transmitting 1 - Transmitting
5-4 FLTCONF	Fault Confinement State Indicates the confinement state of FlexCAN. If CTRL1[LOM] = 1, after a delay that depends on the CAN bit timing, this field indicates Error Passive. The same delay affects the way that this field reflects an update to ECR register by the CPU. It may be necessary to wait up to one CAN bit time for coherence to be restored. Soft reset affects this field, but if CTRL1[LOM] = 1, its reset value lasts for only one CAN bit. After that time, this field reports Error Passive. 00 - Error Active 01 - Error Passive 1x - Bus Off
3 RX	FlexCAN in Reception Flag Indicates whether FlexCAN is receiving a message. See the table in Error and Status 1 (ESR1). 0 - Not receiving 1 - Receiving
2 BOFFINT	Bus Off Interrupt Flag Indicates whether FlexCAN has entered Bus Off state. If CTRL1[BOFFMSK]) = 1, an interrupt is generated to the CPU. Writing 0 to this field has no effect. 0 - No such occurrence. 1 - FlexCAN module entered Bus Off state.
1 ERRINT	Error Interrupt Flag Indicates that at least one of the error flags (ESR1[BIT1ERR], ESR1[BIT0ERR], ESR1[ACKERR], ESR1[CRCERR], ESR1[FRMERR], or ESR1[STFERR]) is set. If the corresponding mask CTRL1[ERRMSK] = 1, an interrupt is generated to the CPU. Writing 0 to this field has no effect.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - No such occurrence. 1 - Indicates setting of any error flag in the Error and Status register.
0 WAKINT	Wake-up Interrupt Flag Generates an interrupt to the CPU when a recessive-to-dominant transition is detected on the CAN bus and MCR[WAKMSK] = 1. This field applies when FlexCAN is in low-power mode during Self Wake-up: • Doze mode • Stop mode When MCR[SLFWAK] = 0, this flag is masked. The CPU must clear this flag before disabling the field. Otherwise, it is set when MCR[SLFWAK] becomes 1 again. Writing 0 has no effect. 0 - No such occurrence. 1 - Indicates that a recessive-to-dominant transition was received on the CAN bus.

39.6.2.10 Interrupt Masks 2 (IMASK2)

Masks interrupt flags. This register allows any of the 32 message buffer interrupts to be enabled or disabled for MB63–MB32. It contains one interrupt mask bit per buffer. This configuration allows the CPU to determine which buffer generates an interrupt after a successful transmission or reception when the corresponding Interrupt Flags 2 (IFLAG2) flag is set.

Offset

Register	Offset
IMASK2	24h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BUF63TO32M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF63TO32M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BUF63TO32M	Buffer MBi Mask Masks the corresponding FlexCAN message buffer interrupt for MB63–MB32. **NOTE** If the corresponding Interrupt Flags 2 (IFLAG2) flag is set, writing 1 or 0 to a field in IMASK2 can set or clear an interrupt request. 0b - The corresponding buffer interrupt is disabled. 1b - The corresponding buffer interrupt is enabled.

39.6.2.11 Interrupt Masks 1 (IMASK1)

Masks interrupt flags. This register allows any of the 32 message buffer interrupts to be enabled or disabled for MB31–MB0. It contains one interrupt mask bit per buffer. This configuration allows the CPU to determine which buffer generates an interrupt after a successful transmission or reception when the corresponding Interrupt Flags 1 (IFLAG1) flag is set.

Offset

Register	Offset
IMASK1	28h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BUF31TO0M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF31TO0M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BUF31TO0M	Buffer MBi Mask Enables or disables the corresponding FlexCAN message buffer interrupt for MB31–MB0.

Table continues on the next page...

Field	Description
	NOTE If the corresponding Interrupt Flags 1 (IFLAG1) flag is set, writing 1 or 0 to a field in IMASK1 can set or clear an interrupt request. 0b - The corresponding buffer interrupt is disabled. 1b - The corresponding buffer interrupt is enabled.

39.6.2.12 Interrupt Flags 2 (IFLAG2)

Contains the flags for the 32 message buffer interrupts for MB63–MB32. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the respective flag in this register. If the corresponding Interrupt Masks 2 (IMASK2) bit is set, an interrupt is generated.

Before updating MCR[MAXMB], the CPU must service the IFLAG2 flags whose MB value is greater than the MAXMB to be updated. Otherwise, those flags remain set and are inconsistent with the number of message buffers available.

Offset

Register	Offset
IFLAG2	2Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF63TO32I															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF63TO32I															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BUF63TO32I	Buffer MBi Interrupt Flags the corresponding FlexCAN message buffer interrupt for MB63–MB32. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.

39.6.2.13 Interrupt Flags 1 (IFLAG1)

Contains the flags for the 32 message buffer interrupts for MB31–MB0. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG1 bit. If the corresponding Interrupt Masks 1 (IMASK1) bit is set, an interrupt is generated. There is an exception when DMA for Legacy RX FIFO is enabled, as described below.

The BUF7I–BUF5I flags also represent Legacy FIFO interrupts when the Legacy RX FIFO is enabled. When MCR[RFEN] is 1 and MCR[DMA] is 0, the function of the eight least significant interrupt flags changes:

- BUF7I, BUF6I, and BUF5I indicate operating conditions of the Legacy FIFO.
- BUF4I–BUF1I fields are reserved.
- BUF0I empties the Legacy FIFO.

Before writing 1 to MCR[RFEN], the CPU must service the IFLAG flags set in the Legacy RX FIFO region; see Legacy RX FIFO. Otherwise, these IFLAG flags mistakenly show the related message buffers now belonging to Legacy FIFO as having contents to be serviced. When MCR[RFEN] is 0, the Legacy FIFO flags must be cleared. The same care must be taken when a CTRL2[RFFN] value is selected, extending Legacy RX FIFO filters beyond MB7. For example, when RFFN is 8h, Legacy RX FIFO filters occupy the MB23–MB0 range, and related IFLAG flags must be cleared.

When MCR[RFEN] and MCR[DMA] are 1 (DMA feature for Legacy RX FIFO enabled), the function of the eight least significant interrupt flags (BUF7I–BUF0I) changes to support DMA operation. BUF7I, BUF6I, and BUF4I–BUF1I are not used.

BUF5I indicates the operating condition of the Legacy FIFO, and BUF0I empties the Legacy FIFO. Moreover, BUF5I does not generate a CPU interrupt, but it does generate a DMA request. IMASK1 bits in the Legacy RX FIFO region are not considered when bit MCR[DMA] = 1. In addition, the CPU must not clear the BUF5I flag when DMA is enabled. Before writing 1 to MCR[DMA], the CPU must service the IFLAG flags set in the Legacy RX FIFO region. When MCR[DMA] is 0, the Legacy FIFO must be empty. Legacy FIFO must be disabled when MCR[FDEN] = 1.

Before updating MCR[MAXMB], the CPU must service the IFLAG1 flags whose MB value is greater than the MCR[MAXMB] to be updated. Otherwise, those flags remain set and are inconsistent with the number of message buffers available.

Offset

Register	Offset
IFLAG1	30h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF31TO8I															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF31TO8I								BUF7I	BUF6I	BUF5I	BUF4TO1I				BUF0I
W	W1C								W1C	W1C	W1C	W1C				W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 BUF31TO8I	Buffer MBi Interrupt Flags the corresponding FlexCAN message buffer interrupt for MB31–MB8. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.
7 BUF7I	Buffer MB7 Interrupt or Legacy RX FIFO Overflow Flags the interrupt for MB7 when MCR[RFEN] = 0 (Legacy RX FIFO disabled). **NOTE** When a CPU write changes the value of MCR[RFEN], FlexCAN clears this flag. When MCR[RFEN] = 1, this flag represents a Legacy RX FIFO overflow. In this case, the flag indicates that a message was lost because the Legacy RX FIFO is full. When the Legacy RX FIFO is full and a message buffer captures the message, this flag is not set. 0 - No occurrence of MB7 completing transmission or reception, or no FIFO overflow. 1 - MB7 completed transmission or reception, or FIFO overflow.
6 BUF6I	Buffer MB6 Interrupt or Legacy RX FIFO Warning Flags the interrupt for MB6 when MCR[RFEN] = 0 (Legacy RX FIFO disabled). **NOTE** When a CPU write changes the value of MCR[RFEN], FlexCAN clears this flag. When MCR[RFEN] = 1, this flag represents a Legacy RX FIFO warning. In this case, the flag indicates when the number of unread messages within the Legacy RX FIFO is increased to five from four due to the reception of a new message. In other words, the Legacy RX FIFO is almost full. If this flag is cleared when there are more than four unread messages, it does not set again until the number of unread messages in the Legacy RX FIFO decreases to four or fewer. 0 - No occurrence of MB6 completing transmission or reception, or FIFO not almost full. 1 - MB6 completed transmission or reception, or FIFO almost full.
5 BUF5I	Buffer MB5 Interrupt or Frames available in Legacy RX FIFO Flags the interrupt for MB5 when MCR[RFEN] = 0 (Legacy RX FIFO disabled). **NOTE** When a CPU write changes the value of MCR[RFEN], FlexCAN clears this flag. When MCR[RFEN] = 1, the BUF5I flag represents frames available in Legacy RX FIFO. In this case, the flag indicates that at least one frame is available to be read from the Legacy RX FIFO. When MCR[DMA] = 1, this flag generates a DMA request. The CPU must not clear this field by writing 1 to BUF5I. 0 - No occurrence of completed transmission or reception, or no frames available

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - MB5 completed transmission or reception, or frames available
4-1 BUF4TO1I	Buffer MBi Interrupt or Reserved Flags the interrupts for MB4–MB1 when MCR[RFEN] = 0 (Legacy RX FIFO disabled). **NOTE** When a CPU write changes the value of MCR[RFEN], FlexCAN clears these flags. When MCR[RFEN] = 1, the BUF4TO1I flags are reserved. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.
0 BUF0I	Buffer MB0 Interrupt or Clear Legacy FIFO bit Flags the interrupt for MB0 when MCR[RFEN] = 0 (Legacy RX FIFO disabled). If MCR[RFEN] = 1, this field is used to trigger the clear Legacy FIFO operation. This operation empties the Legacy FIFO contents. Before performing this operation, the CPU must service all Legacy FIFO-related IFLAG flags. When MCR[DMA] = 1, this operation also clears the BUF5I flag, aborting the DMA request. The clear Legacy FIFO operation occurs when the CPU writes 1 to BUF0I. This operation is only allowed in Freeze mode; the module blocks it in other conditions. 0 - MB0 has no occurrence of successfully completed transmission or reception. 1 - MB0 has successfully completed transmission or reception.

39.6.2.14 Control 2 (CTRL2)

Complements Control 1 (CTRL1), providing control bits for memory write-access in Freeze mode. This register extends Legacy FIFO filter quantity, and adjusts the operation of internal FlexCAN processes such as matching and arbitration.

Soft reset does not affect the contents of this register.

Table 269 shows how the value of CTRL2[RFFN] determines the Legacy RX FIFO filter structure.

Table 269. Possible Legacy RX FIFO filter structures

RFFN[3:0]	Number of Legacy RX FIFO filter elements	Message buffers occupied by Legacy RX FIFO and ID filter table	Remaining available message buffers	Legacy RX FIFO ID filter table elements affected by RX individual masks	Legacy RX FIFO ID filter table elements affected by Legacy RX FIFO global mask
0h	8	MB 0–7	MB 8–95	Elements 0–7	None
1h	16	MB 0–9	MB 10–95	Elements 0–9	Elements 10–15
2h	24	MB 0–11	MB 12–95	Elements 0–11	Elements 12–23
3h	32	MB 0–13	MB 14–95	Elements 0–13	Elements 14–31
4h	40	MB 0–15	MB 16–95	Elements 0–15	Elements 16–39

Table continues on the next page...

Table 269. Possible Legacy RX FIFO filter structures (continued)

RFFN[3:0]	Number of Legacy RX FIFO filter elements	Message buffers occupied by Legacy RX FIFO and ID filter table	Remaining available message buffers	Legacy RX FIFO ID filter table elements affected by RX individual masks	Legacy RX FIFO ID filter table elements affected by Legacy RX FIFO global mask
5h	48	MB 0–17	MB 18–95	Elements 0–17	Elements 18–47
6h	56	MB 0–19	MB 20–95	Elements 0–19	Elements 20–55
7h	64	MB 0–21	MB 22–95	Elements 0–21	Elements 22–63
8h	72	MB 0–23	MB 24–95	Elements 0–23	Elements 24–71
9h	80	MB 0–25	MB 26–95	Elements 0–25	Elements 26–79
Ah	88	MB 0–27	MB 28–95	Elements 0–27	Elements 28–87
Bh	96	MB 0–29	MB 30–95	Elements 0–29	Elements 30–95
Ch	104	MB 0–31	MB 32–95	Elements 0–31	Elements 32–103
Dh	112	MB 0–33	MB 34–95	Elements 0–31	Elements 32–111
Eh	120	MB 0–35	MB 36–95	Elements 0–31	Elements 32–119
Fh	128	MB 0–37	MB 38–95	Elements 0–31	Elements 32–127

Offset

Register	Offset
CTRL2	34h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERRMSK_...	BOFFDON...	ECRWRE	WRMFRZ	RFFN				TASD					MRP	RRS	EACEN
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TIMER_S...	PREXCEN	BTE	ISOCANF...	EDFLTDIS	FLT_RXN	MBTSBASE		TSTAMPCAP		0	RETRY			0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0

Fields

Field	Description
31	Error Interrupt Mask for Errors Detected in the Data Phase of Fast CAN FD Frames Enables the ESR1[ERRINT_FAST] interrupt.

Table continues on the next page...

Table continued from the previous page...

Field	Description
ERRMSK_FAST	0 - Disable 1 - Enable
30 BOFFDONEMSK	Bus Off Done Interrupt Mask Enables the Bus Off Done interrupt, ESR1[BOFFDONEINT]. 0 - Disable 1 - Enable
29 ECRWRE	Error Correction Configuration Register Write Enable Enables updates for Memory Error Control (MECR). If the protocol described in Detection and correction of memory errors is not followed, this field is automatically 0. 0 - Disable 1 - Enable
28 WRMFRZ	Write Access to Memory in Freeze Mode Enables unrestricted write access to FlexCAN memory in Freeze mode. When this field is 0, write access restrictions are maintained. This field can only be written in Freeze mode, and has no effect out of Freeze mode. **NOTE** Do not write 1 to MCR[RFEN] during FlexCAN memory initialization. 0 - Disable 1 - Enable
27-24 RFFN	Number of Legacy Receive FIFO Filters Defines the number of Receive Legacy FIFO filters, as shown in Table 269. The chip determines the maximum selectable number of filters. Do not program this field with values that cause the number of message buffers occupied by Legacy RX FIFO and Legacy RX FIFO ID Filter to exceed MCR[MAXMB]. MCR[MAXMB] defines the number of message buffers present. This field can only be written in Freeze mode; the module blocks it in other modes. Each group of eight filters occupies a memory space equivalent to two message buffers. The more filters are implemented, the fewer message buffers are available. The Legacy RX FIFO occupies the memory space originally reserved for MB5–MB0. This field should be programmed with a value corresponding to a number of filters less than the number of available memory words. The number of available words can be calculated as follows: (SETUP_MB - 6) × 4 Where SETUP_MB is the smaller of the parameter NUMBER_OF_MB and MCR[MAXMB]. The number of remaining message buffers available is: (SETUP_MB - 8) - (RFFN × 2)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If the number of Legacy RX FIFO filters programmed through RFFN exceeds the SETUP_MB value (memory space available), the exceeding ones are not functional. **NOTE** • The number of the last remaining available message buffers is the smaller of (NUMBER_OF_MB - 1) and MCR[MAXMB]. • If RX Individual Mask registers are not enabled, the Legacy RX FIFO Global Mask affects all Legacy RX FIFO filters.
23-19 TASD	Transmission Arbitration Start Delay Indicates by how many CAN bits the transmission arbitration process start point can be delayed from the first bit of CRC field on CAN bus. See TX arbitration start delay for details. This field can be written only in Freeze mode; the module blocks it in other modes.
18 MRP	Message Buffers Reception Priority Sets the priority for the matching process. This field can be written only in Freeze mode; the module blocks it in other modes. 0 - Matching starts from Legacy RX FIFO or Enhanced RX FIFO and continues on message buffers. 1 - Matching starts from message buffers and continues on Legacy RX FIFO or Enhanced RX FIFO.
17 RRS	Remote Request Storing Determines what the module does with a remote request. The remote request frame is submitted to a matching process. If this field is 1, the frame is stored in the corresponding message buffer in the same fashion as a data frame. No automatic remote response frame is generated. If this field is 0, an automatic remote response frame is generated if a message buffer with CODE = 1010b is found with the same ID. You can only write to this field in Freeze mode. The module blocks it in other modes. 0 - Generated 1 - Stored
16 EACEN	Entire Frame Arbitration Field Comparison Enable for RX Message Buffers Controls the comparison of IDE and RTR fields within RX message buffer filters with their corresponding bits in the incoming frame by the matching process. If enabled, the IDE and RTR fields of the RX message buffer are compared to their corresponding bits within the incoming frame (mask bits apply). If disabled, the IDE field of the RX message buffer filter is always compared and RTR is never compared despite mask bits. This field does not affect matching for Legacy RX FIFO or Enhanced RX FIFO. You can only write to this field in Freeze mode; the module blocks it in other modes. 0 - Disable 1 - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
15 TIMER_SRC	Timer Source Selects the time tick source used for incrementing the free-running timer counter. If CAN bit clock is selected, it defines the baud rate on the CAN bus. If External time tick is selected, the period can be adjusted to match the baud rate on the CAN bus. It can also be adjusted to a different value as required. See the chip-specific section for details about the external time tick. You can only write to this field in Freeze mode. 0 - CAN bit clock 1 - External time tick
14 PREXCEN	Protocol Exception Enable Enables the protocol exception feature. You can only write to this field in Freeze mode. **NOTE** See Protocol exception event in the CAN Protocol standard ISO 11898-1_2024 for details. 0 - Disabled 1 - Enabled
13 BTE	Bit Timing Expansion Enable Enables the use of Enhanced CAN Bit Timing Prescalers (EPRS), Enhanced Data Phase CAN Bit Timing (EDCBT), and Enhanced Nominal CAN Bit Timing (ENCBT) to configure the CAN bit timing segments, instead of using the bit timing fields of CAN Bit Timing (CBT), CAN FD Bit Timing (FDCBT), and Control 1 (CTRL1). If this field is 1: • CTRL1[PRESDIV], CTRL1[PROPSEG], CTRL1[PSEG1], CTRL1[PSEG2], and CTRL1[RJW] are read as zero. A write operation to these fields has no effect. • CBT[EPRESDIV], CBT[ERJW], CBT[EPROPSEG], CBT[EPSEG1], and CBT[EPSEG2], and the corresponding fields in CAN FD Bit Timing (FDCBT), are read as zero. A write operation to these fields has no effect. • ETDC[ETDCOFF], ETDC[ETDCEN], ETDC[ETDCFAIL], and ETDC[ETDCVAL] are used by FlexCAN instead of FDCTRL[TDCOFF], FDCTRL[TDCEN], FDCTRL[TDCFAIL], and FDCTRL[TDCVAL]. These fields are read as zero, and a write operation to them has no effect. • ETDC[TDMDIS] can be used to disable transceiver delay measurement. 0 - Disable 1 - Enable
12 ISOCANFDEN	ISO CAN FD Enable Enables the CAN FD protocol according to ISO specification ISO 11898-1_2024 (see CAN FD ISO compliance). When disabled, FlexCAN operates using the non-ISO CAN FD protocol.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can only write to this field in Freeze mode. **NOTE** FlexCAN is able to transmit FD frame format according to CAN Protocol standard ISO 11898-1_2024. 0 - Disable 1 - Enable
11 EDFLTDIS	Edge Filter Disable Disables the edge filter used during the Bus Integration state. When the Edge Filter is enabled, two consecutive nominal time quanta with dominant bus states are required to detect an edge that causes synchronization. When synchronization occurs, the counting of the sequence of 11 consecutive recessive bits is restarted. The edge filter prevents dominant pulses that are shorter than a nominal bit time (present during the data phase of an FD frame) from being mistaken for an idle condition. You can only write to this field in Freeze mode. **NOTE** See Bus Integration state in the CAN Protocol standard ISO 11898-1_2024 for details. 0 - Enabled 1 - Disabled
10 FLT_RXN	Fault reaction Indicates fault reaction during the functional mode. Setting this bit makes the TX pin to recessive state. 0 - Fault reaction is disabled 1 - Fault reaction is enabled
9-8 MBTSBASE	Message Buffer Timestamp Base Selects the timebase used for capturing the 16-bit TIME_STAMP field of the message buffer register. This field can be written in Freeze Mode only. 00 - TIMER 01 - Lower 16 bits of high-resolution timer 10 - Upper 16 bits of high-resolution timer 11 - Reserved
7-6 TSTAMPCAP	Timestamp Capture Point Configures the point in time when a 32-bit timebase is captured during a CAN frame. This base is stored in the high-resolution timestamp register (HR_TIME_STAMP). For classical CAN frames, capture points can be the start-of-frame bit or the point when CAN frame is considered valid. This point is the seventh bit of the end-of-frame for transmission and the sixth bit of the

Table continues on the next page...

Table continued from the previous page...

Field	Description
	end-of-frame for reception. For CAN FD frames, the high-resolution timestamp can be captured at the start of frame, when a CAN FD frame is considered valid, or the res bit. You can only write to this field in Freeze mode. 00 - Disabled 01 - End of the CAN frame 10 - Start of the CAN frame 11 - Start of frame for classical CAN frames; res bit for CAN FD frames
5 —	Reserved
4-2 RETRY	Number of Retransmission Requests The message is discarded in case of a re-transmission counter exhausted. **NOTE** You can only write to this field in Freeze mode. 000 - No retransmission 001 - Count of re-transmission attempts 010 - Count of re-transmission attempts 011 - Count of re-transmission attempts 100 - Count of re-transmission attempts 101 - Count of re-transmission attempts 110 - Count of re-transmission attempts 111 - Unlimited number of retransmission
1-0 —	Reserved

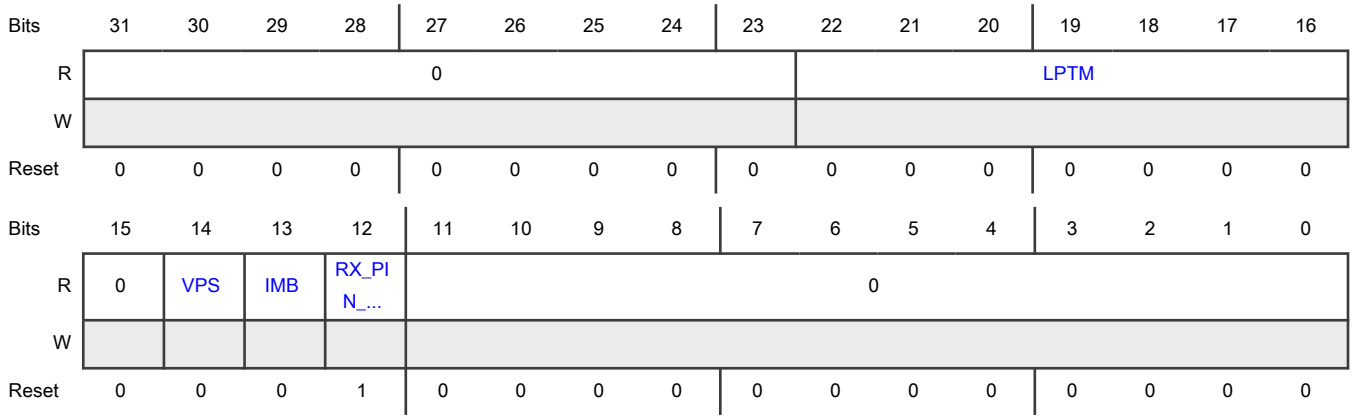
39.6.2.15 Error and Status 2 (ESR2)

Reports general status information.

Offset

Register	Offset
ESR2	38h

Diagram



Fields

Field	Description
31-23 —	Reserved
22-16 LPTM	Lowest Priority TX Message Buffer Indicates the lowest number inactive message buffer when ESR2[VPS] = 1 (see ESR2[IMB]). If no message buffer is inactive, the message buffer indicated depends on the value of CTRL1[LBUF]. If CTRL1[LBUF] = 0, the message buffer indicated is the one with the greatest arbitration value (see Highest-priority message buffer first). If CTRL1[LBUF] = 1, the message buffer indicated is the active TX message buffer with the highest number. If a TX message buffer is being transmitted, it is not considered in the LPTM calculation. If ESR2[IMB] is not 0 and a frame is transmitted successfully, the value of LPTM is updated with its message buffer number.
15 —	Reserved
14 VPS	Valid Priority Status Indicates whether the contents of ESR2[IMB] and ESR2[LPTM] are valid. It becomes 1 upon every complete TX arbitration process, unless the CPU writes to the Control and Status word of a message buffer already scanned. In other words, it is behind the TX Arbitration Pointer, during the TX arbitration process. If there is no inactive message buffer and only one TX message buffer that is being transmitted, this field remains 0. This field becomes 0 upon the start of every TX arbitration process or upon a write to the Control and Status word of any message buffer. **NOTE** No CPU write to the Control and Status of a message buffer that the abort mechanism blocks affects this field. When MCR[AEN] = 1, the abort code write to the Control and Status of an MB being transmitted (pending abort) is blocked. Any write attempt to a TX MB with its IFLAG flag set is also blocked. 0 - Invalid 1 - Valid

Table continues on the next page...

Table continued from the previous page...

Field	Description
13 IMB	Inactive Message Buffer Indicates whether any message buffer is inactive (CODE field is either 1000b or 0b) when ESR2[VPS] = 1. This field becomes 1 when: • A lowest-priority TX message buffer (ESR2[LPTM]) is found and it is inactive during arbitration. • This field is not 1, and a frame is transmitted successfully. This field always becomes 0 at the start of arbitration (see Arbitration process). If a message buffer is successfully transmitted and this field is 0 (no inactive message buffer), ESR2[VPS] and this field both become 1. The index related to the MB transmitted is loaded into ESR2[LPTM]. In this case, the value of ESR2[LPTM] is the number of the first inactive message buffer. 0 - Message buffer indicated by ESR2[LPTM] is not inactive. 1 - At least one message buffer is inactive.
12 RX_PIN_ST	RX Pin Status Indicates the current state of the RX pin. This bit is affected by soft reset. **NOTE** IP takes a few clock cycles to propagate the value of RX pin to this status bit. 0 - RX pin is in the dominant state 1 - RX pin is in a recessive state
11-0 —	Reserved

39.6.2.16 Cyclic Redundancy Check (CRCR)

Provides information about the CRC of transmitted messages for non-FD messages. This register only reports the 15 low-order bits of CRC calculations for messages in CAN FD format that require either 17 or 21 bits. For CAN FD format frames, you must use the CAN FD CRC (FDCRC). This register is updated at the same time that the TX interrupt flag is set.

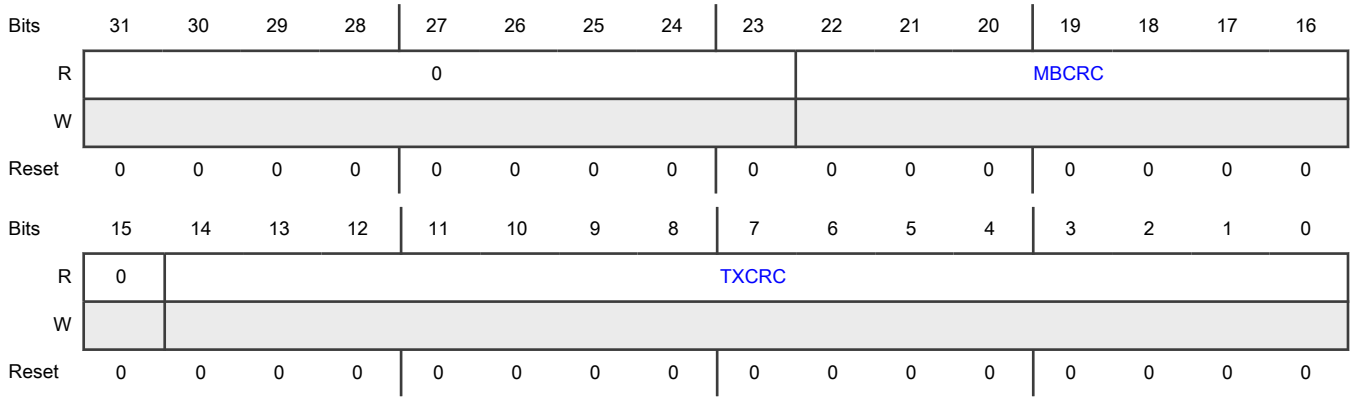
NOTE

See CRC sequence calculation in the CAN Protocol standard ISO 11898-1_2024 for details.

Offset

Register	Offset
CRCR	44h

Diagram



Fields

Field	Description
31-23 —	Reserved
22-16 MBCRC	CRC Message Buffer Indicates the number of the message buffer corresponding to the value in CRCR[TXCRC].
15 —	Reserved
14-0 TXCRC	Transmitted CRC value Indicates the CRC value of the last transmitted message for non-FD frames. For FD frames, CRC value is reported in CAN FD CRC (FDCRC).

39.6.2.17 Legacy RX FIFO Global Mask (RXFGMASK)

Masks the Legacy RX FIFO ID filter table elements that do not have a corresponding RXIMR according to CTRL2[RFFN], when Legacy RX FIFO is enabled.

This register is located in RAM.

You can only write to this register in Freeze mode; the module blocks it in other modes.

The following table shows how the FGM bits correspond to each IDAF field.

Table 270. Correspondence of Legacy RX FIFO global mask bits to IDF fields

Legacy RX FIFO ID filter table elements format (MCR[IDAM])	Identifier acceptance filter fields					
	RTR	IDE	RXIDA	RXIDB [1]	RXIDC [2]	Reserved
A	FGM[31]	FGM[30]	FGM[29:1]	—	—	FGM[0]
B	FGM[31], FGM[15]	FGM[30], FGM[14]	—	FGM[29:16], FGM[13:0]		—

Table continues on the next page...

Table 270. Correspondence of Legacy RX FIFO global mask bits to IDF fields (continued)

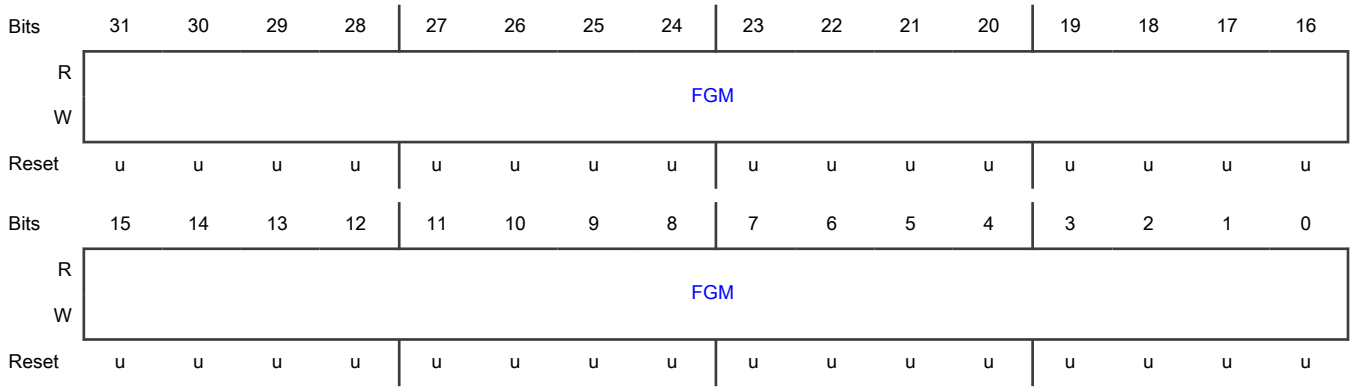
Legacy RX FIFO ID filter table elements format (MCR[IDAM])	Identifier acceptance filter fields					
	RTR	IDE	RXIDA	RXIDB [1]	RXIDC [2]	Reserved
C	—	—		—	FGM[31:24], FGM[23:16], FGM[15:8], FGM[7:0]	

1. If MCR[IDAM] is equivalent to format B, only the 14 most significant bits of the identifier of the incoming frame are compared with the Legacy RX FIFO filter.
2. If MCR[IDAM] is equivalent to format C, only the eight most significant bits of the identifier of the incoming frame are compared with the Legacy RX FIFO filter.

Offset

Register	Offset
RXFGMASK	48h

Diagram



Fields

Field	Description
31-0 FGM	Legacy RX FIFO Global Mask Bits Masks the ID filter table elements bits in a perfect alignment. 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

39.6.2.18 Legacy RX FIFO Information (RXFIR)

Provides information about Legacy RX FIFO.

This register is the port through which the CPU accesses the output of the Legacy RXFIR FIFO located in RAM. FlexCAN writes to the Legacy RXFIR FIFO when a new message is moved into the Legacy RX FIFO. Also, its output is updated whenever the output of the Legacy RX FIFO is updated with the next message. See Legacy RX FIFO for instructions on reading this register.

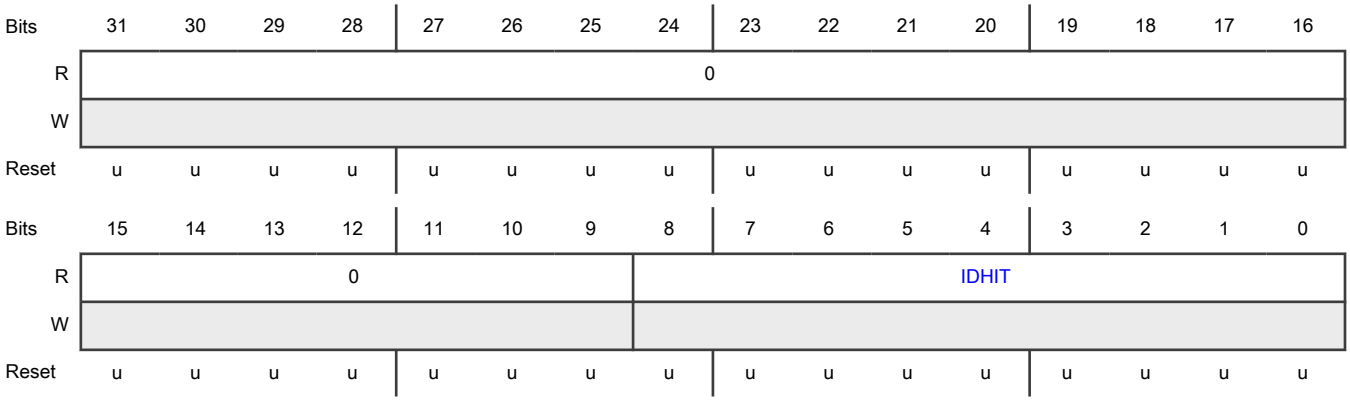
NOTE

RXFIR can be written only during memory initialization, due to the error code correction (ECC) feature. In every other case, this register is read-only.

Offset

Register	Offset
RXFIR	4Ch

Diagram



Fields

Field	Description
31-9 —	Reserved
8-0 IDHIT	Identifier Acceptance Filter Hit Indicator Indicates which Identifier Acceptance filter that the received message hit in the output of the Legacy RX FIFO. If multiple filters match the incoming message ID, the first matching IDAF found (lowest number) by the matching process is indicated. This field is valid only when IFLAG1[BUF5I] is set.

39.6.2.19 CAN Bit Timing (CBT)

Provides an alternative way to store the CAN bit timing variables described in Control 1 (CTRL1). EPRESDIV, EPROPSEG, EPSEG1, EPSEG2, and ERJW are extended versions of CTRL1[PRESDIV], CTRL1[PROPSEG], CTRL1[PSEG1], CTRL1[PSEG2], and CTRL1[RJW] respectively.

NOTE

The CAN bit variables in CTRL1 and in CBT are stored in the same register.

CBT[BTF] selects the use of the timing variables defined in this register.

When the CAN FD feature is enabled (MCR[FDEN] = 1), always write 1 to CBT[BTF].

Soft reset does not affect the contents of this register.

NOTE

Ensure that bit time settings and protocol engine tolerance are in compliance with the CAN Protocol standard ISO 11898-1_2024.

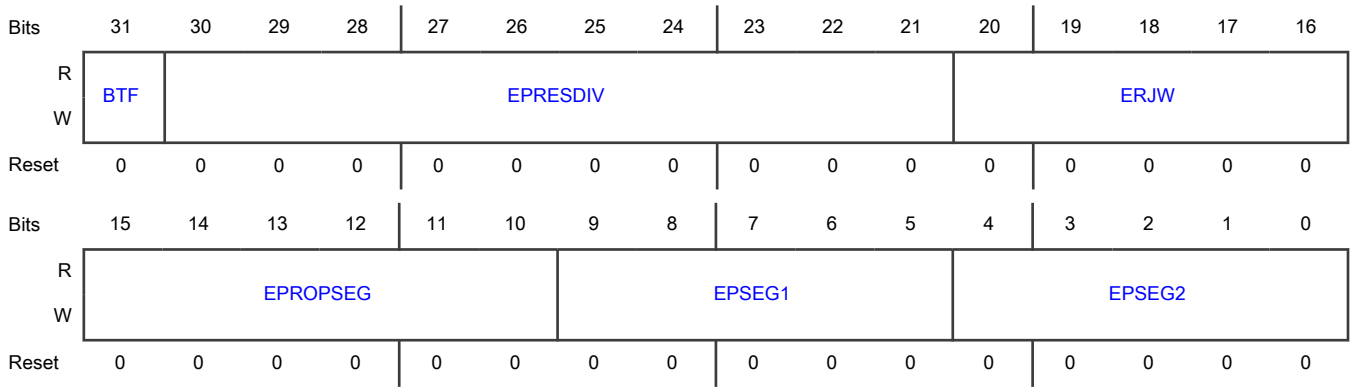
NOTE

If CTRL2[BTE] = 1, EPRESDIV, ERJW, EPROPSEG, EPSEG1, and EPSEG2 are read as zero. A write operation to them has no effect.

Offset

Register	Offset
CBT	50h

Diagram



Fields

Field	Description
31 BTF	Bit Timing Format Enable Enables the use of extended CAN bit timing fields EPRESDIV, EPROPSEG, EPSEG1, EPSEG2, and ERJW. These fields replace the CAN bit timing variables defined in Control 1 (CTRL1). This field can be written in Freeze mode only. 0 - Disable 1 - Enable
30-21 EPRESDIV	Extended Prescaler Division Factor Defines the ratio between the PE clock frequency and the serial clock (Sclock) frequency when CBT[BTF] = 1, otherwise it has no effect. It extends the CTRL1[PRESDIV] value range. The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency (see Protocol timing). This field can be written only in Freeze mode; the module blocks it in other modes. Sclock frequency = PE clock frequency ÷ (EPRESDIV + 1)
20-16	Extended Resync Jump Width

Table continues on the next page...

Table continued from the previous page...

Field	Description
ERJW	Defines the maximum number of time quanta that one resynchronization can change a bit time when CBT[BTF] = 1. Otherwise, it has no effect. It extends the CTRL1[RJW] value range. This field can be written only in Freeze mode; the module blocks it in other modes. Resync Jump Width = ERJW + 1. One Time Quantum = one Sclock period.
15-10 EPROPSEG	Extended Propagation Segment Defines the length of the propagation segment in the bit time when CBT[BTF] = 1, otherwise it has no effect. It extends the CTRL1[PROPSEG] value range. This field can be written only in Freeze mode; the module blocks it in other modes. Propagation Segment Time = (EPROPSEG + 1) × Time Quanta. One Time Quantum = one Sclock period.
9-5 EPSEG1	Extended Phase Segment 1 Defines the length of phase segment 1 in the bit time when CBT[BTF] = 1, otherwise it has no effect. It extends the CTRL1[PSEG1] value range. This field can be written only in Freeze mode; the module blocks it in other modes. Phase Buffer Segment 1 = (EPSEG1 + 1) × Time Quanta. One Time Quantum = one Sclock period.
4-0 EPSEG2	Extended Phase Segment 2 Defines the length of phase segment 2 in the bit time when CBT[BTF] = 1, otherwise it has no effect. It extends the CTRL1[PSEG2] value range. This field can be written only in Freeze mode; the module blocks it in other modes. Phase Buffer Segment 1 = (EPSEG2 + 1) × Time Quanta. One Time Quantum = one Sclock period.

39.6.2.20 Interrupt Masks 3 (IMASK3)

Enables or disables any number of the 32 message buffer interrupts for MB95–MB64. It contains one interrupt mask bit per buffer. This configuration allows the CPU to determine which buffer generates an interrupt after a successful transmission or reception when the corresponding IFLAG3 flag is set.

Offset

Register	Offset
IMASK3	6Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BUF95TO64M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF95TO64M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BUF95TO64M	Buffer MBi Mask Enables or disables the corresponding FlexCAN message buffer interrupt for MB95–MB64. When CAN FD is enabled, the MB range is defined in accordance with FDCTRL[MBDSRn]. **NOTE** If the corresponding IFLAG3 flag is set, writing 1 or 0 to a field in IMASK3 can set or clear an interrupt request. 0b - The corresponding buffer interrupt is disabled. 1b - The corresponding buffer interrupt is enabled.

39.6.2.21 Interrupt Flags 3 (IFLAG3)

Defines the flags for the 32 message buffer interrupts for MB95–MB64. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG3 flag. If the corresponding IMASK3 bit is 1, an interrupt is generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect.

Before updating MCR[MAXMB], the CPU must service the IFLAG3 flags whose MB value is greater than the MAXMB to be updated. Otherwise, they remain set and are inconsistent with the number of message buffers available.

Offset

Register	Offset
IFLAG3	74h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF95TO64															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF95TO64															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BUF95TO64	Buffer MBi Interrupt Flags the corresponding FlexCAN message buffer interrupt for MB95–MB64.When CAN FD is enabled, the MB range is defined in accordance with FDCTRL[MBDSRn]. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.

39.6.2.22 External Timer (ET)

This register captures the value of external timer.

Offset

Register	Offset
ET	78h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TIMER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TIMER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TIMER	Timer Indicates the current value of the external timer. This field is affected by soft reset.

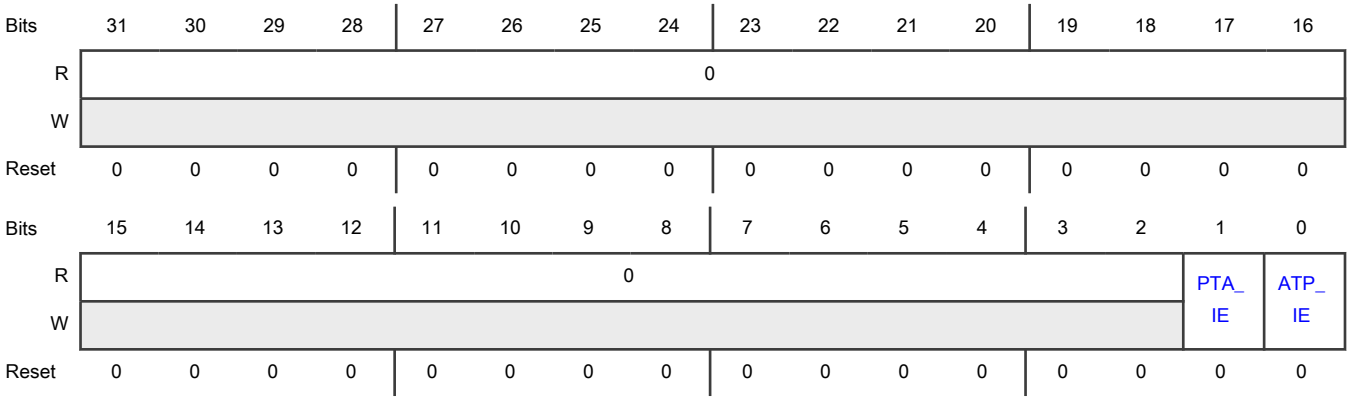
39.6.2.23 Fault Confinement Interrupt Enable (FLTCONF_IE)

Interrupt enables for active to passive and passive to active state transitions.

Offset

Register	Offset
FLTCONF_IE	7Ch

Diagram



Fields

Field	Description
31-2 —	Reserved
1 PTA_IE	Passive to Active Interrupt Enable Enables passive to active error state interrupt. When this field is 1, passive to active error state interrupt is enabled. When this field is 0, passive to active error state interrupt is not enabled. 0 - Disable 1 - Enable
0 ATP_IE	Active to Passive Interrupt Enable Enables active to passive error state interrupt. When this field is 1, active to passive error state interrupt is enabled. When this field is 0, active to passive error state interrupt is not enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Disable 1 - Enable

39.6.2.24 Receive Individual Mask (RXIMR0 - RXIMR95)

Stores the acceptance masks for ID filtering in RX message buffers and the Legacy RX FIFO.

When the Legacy RX FIFO is disabled (MCR[RFEN] = 0), an individual mask is provided for each available RX message buffer on a one-to-one correspondence. When the Legacy RX FIFO is enabled (MCR[RFEN] = 1), an individual mask is provided for each Legacy RX FIFO ID filter table element on a one-to-one correspondence. This correspondence depends on the setting of CTRL2[RFFN] (see Legacy RX FIFO).

RXIMR0 stores the individual mask associated with either MB0 or ID filter table element 0. RXIMR1 stores the individual mask associated with either MB1 or ID filter table element 1, and so on.

The CPU can only access the RXIMR registers when the module is in Freeze mode; otherwise, the module blocks them. Reset does not affect these registers. They are located in RAM and must be explicitly initialized prior to any reception.

It is possible for the RXIMR memory region to be accessed as general-purpose memory. See Bus interface for more information.

Offset

For n = 0 to 95:

Register	Offset
RXIMRn	880h + (n × 4h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MI															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MI															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Description
31-0 MI	Individual Mask Bits

Table continues on the next page...

Field	Description
	Masks the corresponding bit in both the message buffer filter and Legacy RX FIFO ID filter table element in distinct ways. For message buffer filters, see RX Message Buffers Global Mask (RXMGMASK). For Legacy RX FIFO ID filter table elements, see Legacy RX FIFO Global Mask (RXFGMASK). 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

39.6.2.25 Memory Error Control (MECR)

Contains control bits for memory error detection and correction (ECC).

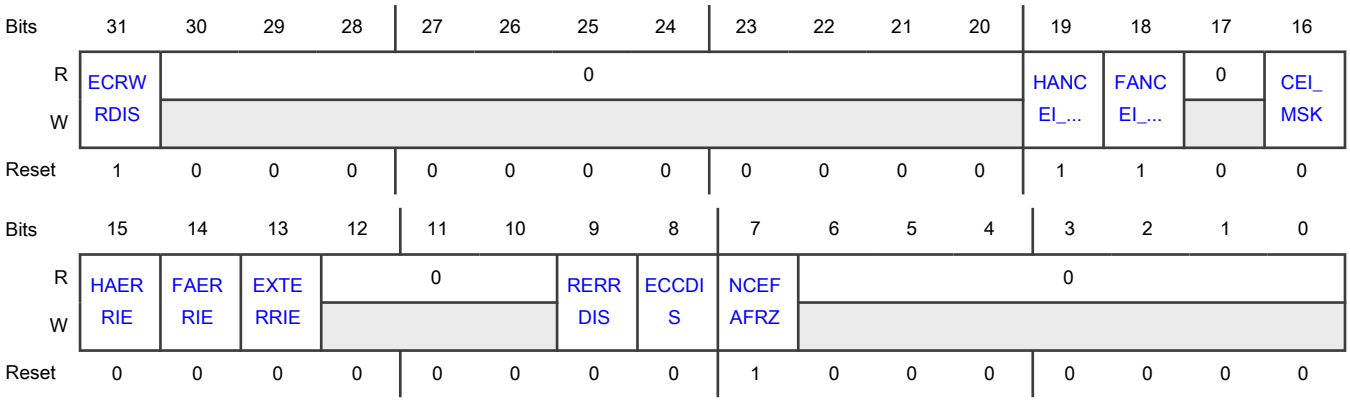
NOTE

When CTRL2[ECRWRE] = 0, writes to this register are blocked, except for MECR[ECRWRDIS].

Offset

Register	Offset
MECR	AE0h

Diagram



Fields

Field	Description
31 ECRWRDIS	Error Configuration Register Write Disable Disables writes on this register. This field automatically becomes 1 (disabled) when CTRL2[ECRWRE] is 1. The protocol described in Detection and correction of memory errors must be followed. 0 - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Disable
30-20 —	Reserved
19 HANCEI_MSK	Host Access with Noncorrectable Errors Interrupt Mask Enables the interrupt for noncorrectable errors detected in memory reads issued by the host (CPU). 0 - Disable 1 - Enable
18 FANCEI_MSK	FlexCAN Access with Noncorrectable Errors Interrupt Mask Enables the interrupt for noncorrectable errors detected in memory reads issued by FlexCAN internal processes. 0 - Disable 1 - Enable
17 —	Reserved
16 CEI_MSK	Correctable Errors Interrupt Mask Enables the interrupt for correctable errors detected in memory reads issued by the host or FlexCAN internal processes. 0 - Disable 1 - Enable
15 HAERRIE	Host Access Error Injection Enable Enables the injection of errors only in memory reads issued by the host (CPU). 0 - Disable 1 - Enable
14 FAERRIE	FlexCAN Access Error Injection Enable Enables the injection of errors only in memory reads issued by FlexCAN internal processes. 0 - Disable 1 - Enable
13 EXTERRIE	Extended Error Injection Enable Extends the error injection on 32-bit memory accesses to the complementary 32-bit word. This feature uses the same 32-bit error injection data and parity words used for 64-bit memory accesses performed by internal FlexCAN processes. See Error Injection Data Pattern (ERRIDPR) and Error Injection Parity Pattern (ERRIPPR). 0 - Disable. Apply error injection only to the 32-bit word.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable. Apply error injection to the 64-bit word.
12-10 —	Reserved
9 RERRDIS	Error Report Disable Disables the update of the error report registers. The update of error-related flags and the generation of bus transfer errors are still active. **NOTE** When reading the report registers, this field must be 1 to ensure coherence on the consecutive register reads. 0 - Enable 1 - Disable
8 ECCDIS	Error Correction Disable Completely disables the memory detection and correction mechanism. Besides disabling the error report mechanism, it also stops the update of the error-related flags and the generation of bus transfer errors. The parity bits continue to be calculated and written into memory on write transactions. 0 - Enable 1 - Disable
7 NCEFAFRZ	Noncorrectable Errors in FlexCAN Access Put Chip in Freeze Mode Determines whether to put FlexCAN into Freeze mode when a noncorrectable error is detected in a memory read performed by FlexCAN internal processes. In this case, entering Freeze mode prevents FlexCAN internal processes from treating corrupted data as valid. See Freeze mode for more information. 0 - Normal operation 1 - Freeze mode
6-0 —	Reserved

39.6.2.26 Error Injection Address (ERRIAR)

Contains the address where error is to be injected.

Use the following table to convert from the memory map address to the location in the physical FlexCAN RAM. Where pairs of values are provided, the first is the address for MCR[FDEN] = 0, the second is for MCR[FDEN] = 1.

Table 271. Error injection address for classical CAN format

RAM contents	Injection address	Memory map
FlexCAN registers	Not mapped	—

Table continues on the next page...

Table 271. Error injection address for classical CAN format (continued)

RAM contents	Injection address	Memory map
Message buffers	0000h	0080h
RXIMRs	600h	0880h
RXFIR_0	780h	0A80h
RXFIR_1	784h	0A84h
RXFIR_2	788h	0A88h
RXFIR_3	78Ch	0A8Ch
RXFIR_4	790h	0A90h
RXFIR_5	794h	0A94h
Reserved	—	0A98h
RXMGMASK	7A0h	0AA0h
RXFGMASK	7A4h	0AA4h
RX14MASK	7A8h	0AA8h
RX15MASK	7ACh	0AACh
Tx_SMB	7B0h	0AB0h / 0F28h
Rx_SMB0	7C0h / 7F8h	0AC0h / 0F70h
Rx_SMB1	7D0h / 840h	0AD0h / 0FB8h
Rx_SMB0_TIME_STAMP	888h	0C20h
Rx_SMB1_TIME_STAMP	88Ch	0C24h
HR_TIME_STAMP	890h	0C30h
Enhanced RX FIFO	A10h	2000h
ERFFEL	1050h	3000h

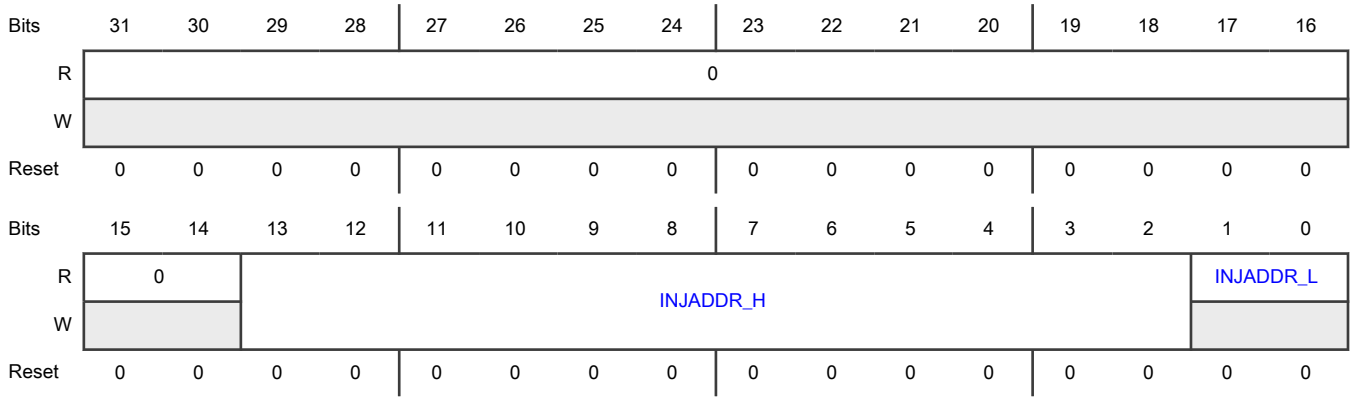
NOTE

For RXFIR, Enhanced RX FIFO, and HR_TIME_STAMP addresses, you must inject ECC errors in host access only.

Offset

Register	Offset
ERRIAR	AE4h

Diagram



Fields

Field	Description
31-14 —	Reserved
13-2 INJADDR_H	Error Injection Address High Defines the 12 most significant bits of the physical RAM address where error is to be injected (see table above).
1-0 INJADDR_L	Error Injection Address Low Defines the two least significant bits of the physical RAM address where error is to be injected. These bits ensure that the address is on a 32-bit boundary.

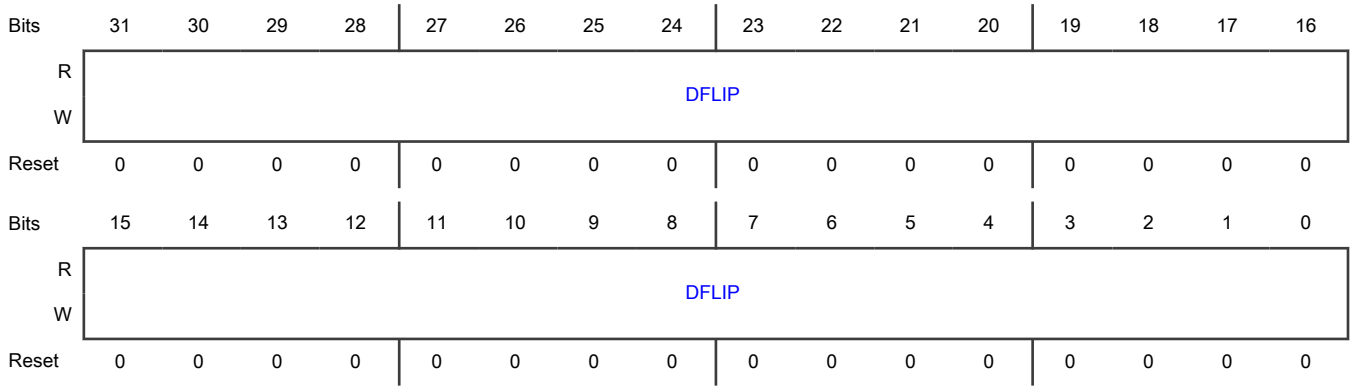
39.6.2.27 Error Injection Data Pattern (ERRIDPR)

Contains the error pattern to be injected in the data word read from memory.

Offset

Register	Offset
ERRIDPR	AE8h

Diagram



Fields

Field	Description
31-0 DFLIP	Data Flip Pattern Contains the flip pattern. Bits set to 1 in the flip pattern cause the corresponding data bit in the word read from memory to invert.

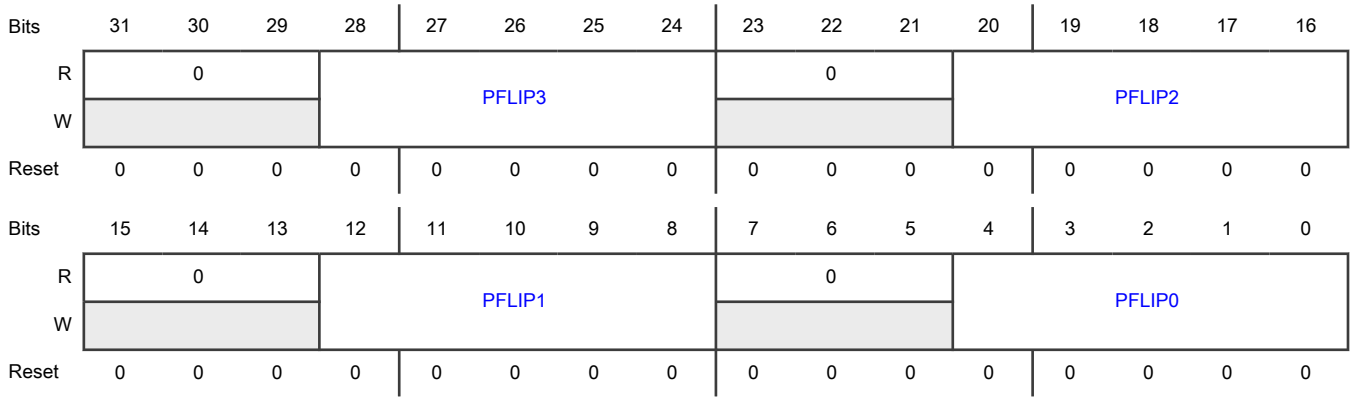
39.6.2.28 Error Injection Parity Pattern (ERRIPPR)

Contains the error pattern to be injected in parity bits read from memory with the data word. Bits set to 1 in the flip pattern cause the corresponding parity bit in the word read from memory to invert.

Offset

Register	Offset
ERRIPPR	AECh

Diagram



Fields

Field	Description
31-29 —	Reserved
28-24 PFLIP3	Parity Flip Pattern for Byte 3 (Most Significant) Contains the error injection pattern for Byte 3.
23-21 —	Reserved
20-16 PFLIP2	Parity Flip Pattern for Byte 2 Contains the error injection pattern for Byte 2.
15-13 —	Reserved
12-8 PFLIP1	Parity Flip Pattern for Byte 1 Contains the error injection pattern for Byte 1.
7-5 —	Reserved
4-0 PFLIP0	Parity Flip Pattern for Byte 0 (Least Significant) Contains the error injection pattern for Byte 0.

39.6.2.29 Error Report Address (RERRAR)

Reports the address used for an access operation in which an error (correctable or noncorrectable) was detected. Also reports the identification of the source of that access.

This address is always reported using a 32-bit alignment. Non-aligned accesses (ERRADDR[1:0] nonzero) are reported with the address aligned, and data is reported in Error Report Data (RERRDR) shifted accordingly. When errors are detected in accesses larger than 32-bit (as performed by FlexCAN internal processes), the address of the 32-bit word where the error was detected is reported. For errors detected in more than one 32-bit word, only the least significant address is reported.

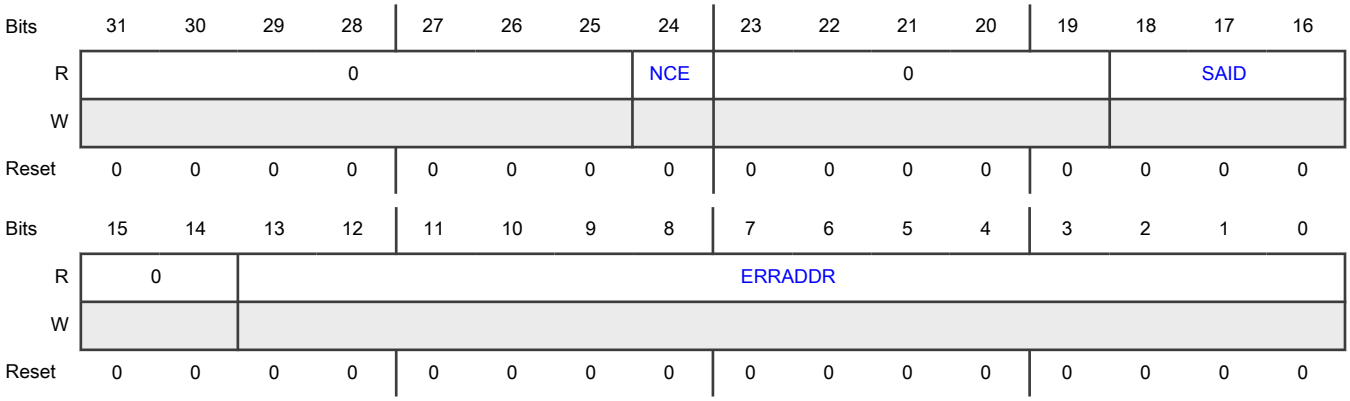
Table 272. Source of memory access

SAID[2:0]	Error during...
0	Move-out FlexCAN access
1	Move-in
2	TX arbitration
3	RX matching
4	Move-out host access
5–7	Reserved

Offset

Register	Offset
RERRAR	AF0h

Diagram



Fields

Field	Description
31-25 —	Reserved
24 NCE	Noncorrectable Error Indicates that the report is due to a noncorrectable error. 0 - Reporting a correctable error 1 - Reporting a noncorrectable error
23-19 —	Reserved
18-16 SAID	SAID SAID[2] — Identification of the requester of the memory read request: • 0 = Requested by FlexCAN internal processes • 1 = Requested by host (CPU) SAID[1] — Details of FlexCAN operation: • 0 = Move • 1 = Scanning SAID[0] — Operation that requested the memory read: • 0 = Transmission

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• 1 = Reception For more information, see Table 272.
15-14 —	Reserved
13-0 ERRADDR	Address Where Error Detected See description of Error Injection Address (ERRIAR).

39.6.2.30 Error Report Data (RERRDR)

Reports the raw data (unmodified by the correction performed by ECC logic) read from memory with error. The value reported does not represent the transient values of the BUSY bit (see Table 278) when reading a message buffer.

Offset

Register	Offset
RERRDR	AF4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RDATA	Raw Data Word Read from Memory with Error

39.6.2.31 Error Report Syndrome (RERRSYNR)

Contains the syndrome detected in a memory read with error. It also reports the bytes which were read in this 32-bit read transaction.

Each SYND*n* field indicates the type of error and which bit in byte (*n*) the error affects. SYND3 corresponds to the most significant byte in the data word read from memory; SYND0 corresponds to the least significant.

Table 273. Syndrome definition

SYND*n* (hex)	Type	Bit affected
00	—	None (no error)
01	Code	0
02	Code	1
04	Code	2
07	Data	5
08	Code	3
0E	Data	7
10	Code	4
13	Data	2
15	Data	6
16	Data	1
19	Data	3
1A	Data	4
1C	Data	0
06	—	All-zeroes noncorrectable error
1F	—	All-ones noncorrectable error
All others	—	Noncorrectable error

Each BE*n* field indicates which byte in the 32-bit word reported was effectively read. The syndrome bits are calculated for all bytes, including the non-read ones. Errors detected in non-read bytes are indicated (see Error indication) and reported (see Error reporting).

Offset

Register	Offset
RERRSYNR	AF8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BE3	0			SYND3				BE2	0			SYND2			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BE1	0			SYND1				BE0	0			SYND0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 BE3	Byte Enabled for Byte 3 (Most Significant) 0 - Byte was not read. 1 - Byte was read.
30-29 —	Reserved
28-24 SYND3	Error Syndrome for Byte 3 (Most Significant) See Table 273.
23 BE2	Byte Enabled for Byte 2 0 - Byte was not read. 1 - Byte was read.
22-21 —	Reserved
20-16 SYND2	Error Syndrome for Byte 2 See Table 273.
15 BE1	Byte Enabled for Byte 1 0 - Byte was not read. 1 - Byte was read.
14-13 —	Reserved
12-8 SYND1	Error Syndrome for Byte 1 See Table 273.

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 BE0	Byte Enabled for Byte 0 (Least Significant) 0 - Byte was not read. 1 - Byte was read.
6-5 —	Reserved
4-0 SYND0	Error Syndrome for Byte 0 (Least Significant) See Table 273.

39.6.2.32 Error Status (ERRSR)

Contains the status bits of the error correction and detection operations. These flags can be cleared by writing 1 to them. Writing 0 has no effect.

Offset

Register	Offset
ERRSR	AFCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												HANCEIF	FANCEIF	0	CEIF
W													W1C	W1C		W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												HANCEIOF	FANCEIOF	0	CEIOF
W													W1C	W1C		W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-20 —	Reserved
19	Host Access with Noncorrectable Error Interrupt Flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
HANCEIF	Indicates that a noncorrectable error was detected in a memory read initiated by host. A bus transfer error is asserted for that access. If MECR[HANCEI_MSK] = 1, the interrupt is 1. 0 - No errors detected 1 - Error detected
18 FANCEIF	FlexCAN Access with Noncorrectable Error Interrupt Flag Indicates that a noncorrectable error was detected in a memory read initiated by FlexCAN internal processes. If MECR[FANCEI_MSK] = 1, the interrupt is 1. 0 - No errors detected 1 - Error detected
17 —	Reserved
16 CEIF	Correctable Error Interrupt Flag Indicates that a correctable error was detected in a memory read. If MECR[CEI_MSK] = 1, the interrupt is 1. 0 - No errors detected 1 - Error detected
15-4 —	Reserved
3 HANCEIOF	Host Access with Noncorrectable Error Interrupt Overrun Flag Indicates that a noncorrectable error was detected in a memory read initiated by host when ERRSR[HANCEIF] was set. No interrupt is associated with this flag. See Error indication. 0 - No errors detected 1 - Error detected
2 FANCEIOF	FlexCAN Access with Noncorrectable Error Interrupt Overrun Flag Indicates that a noncorrectable error was detected in a memory read initiated by FlexCAN internal processes when ERRSR[FANCEIF] was set. No interrupt is associated with this flag. See Error indication. 0 - No errors detected 1 - Error detected
1 —	Reserved
0 CEIOF	Correctable Error Interrupt Overrun Flag Indicates that a correctable error was detected in a memory read when ERRSR[CEIF] was set. No interrupt is associated with this flag. See Error indication. 0 - No errors detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Error detected

39.6.2.33 Enhanced CAN Bit Timing Prescalers (EPRS)

Defines the CAN bit timing prescalers for the nominal phase and data phase when CTRL2[BTE] = 1.

Used by the module only if CTRL2[BTE] = 1; otherwise, a write operation has no effect and all fields are read as zero.

This register can be written only in Freeze mode; the module blocks it in other modes.

Soft reset does not affect the contents of this register.

Offset

Register	Offset
EPRS	BF0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								EDPRESDIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ENPRESDIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-26 —	Reserved
25-16 EDPRESDIV	Extended Data Phase Prescaler Division Factor Defines the ratio between the PE clock frequency and the Sclock frequency in the data phase of a CAN FD message when CTRL2[BTE] = 1. The Sclock period defines the time quantum of the CAN FD protocol for the data bit rate. Sclock frequency = PE clock frequency ÷ (EDPRESDIV + 1)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE To minimize errors when processing FD frames, use the same value for this field and for EPRS[ENPRESDIV]. See the first note in CAN FD frames for details.
15-10 —	Reserved
9-0 ENPRESDIV	Extended Nominal Prescaler Division Factor Defines the ratio between the PE clock frequency and the Sclock frequency when CTRL2[BTE] = 1. Otherwise, it reads as 0 and a write operation has no effect. The Sclock period defines the time quantum of the CAN protocol in the nominal phase. For the reset value, the Sclock frequency is equal to the PE clock frequency (see Protocol timing). Sclock frequency = PE clock frequency ÷ (ENPRESDIV + 1)

39.6.2.34 Enhanced Nominal CAN Bit Timing (ENCBT)

Provides an alternative way to store the CAN bit timing variables described in Control 1 (CTRL1) and CAN Bit Timing (CBT), to get higher CAN bit timing resolution.

This register is used by the module only if CTRL2[BTE] = 1. Otherwise, a write operation has no effect and all fields are read as zero.

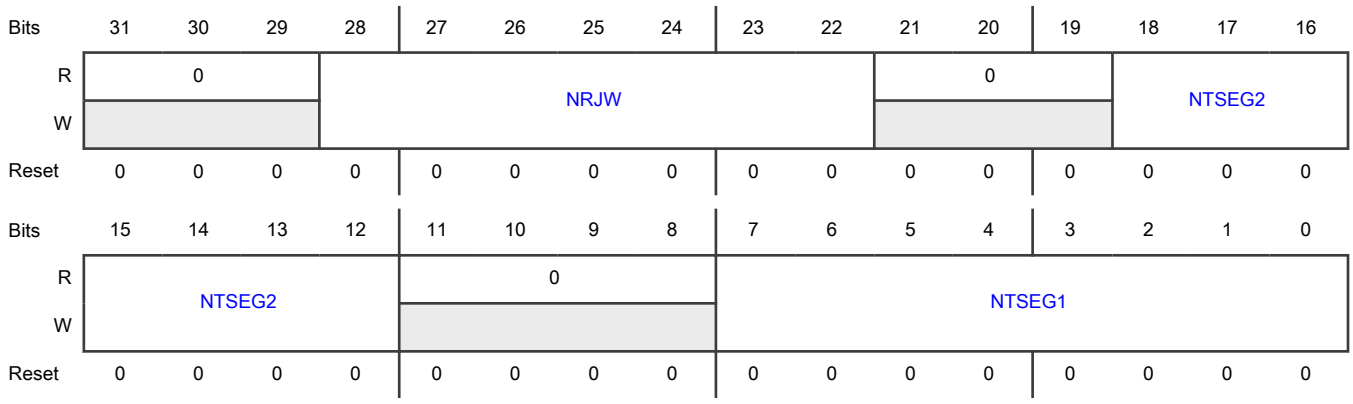
Soft reset does not affect the contents of this register.

This register can be written only in Freeze mode; the module blocks it in other modes.

Offset

Register	Offset
ENCBT	BF4h

Diagram



Fields

Field	Description
31-29 —	Reserved
28-22 NRJW	Nominal Resynchronization Jump Width Defines the maximum number of time quanta that one resynchronization can change a nominal bit time when CTRL2[BTE] = 1. Otherwise, it has no effect. One time quantum = one Sclock period Nominal Resync Jump Width = NRJW + 1
21-19 —	Reserved
18-12 NTSEG2	Nominal Time Segment 2 Defines the length of Time Segment 2 in the nominal bit time when CTRL2[BTE] = 1. Otherwise, it has no effect. Nominal Time Segment 2 = (NTSEG2 + 1) × Time Quanta One time quantum = one Sclock period
11-8 —	Reserved
7-0 NTSEG1	Nominal Time Segment 1 Defines the length of Time Segment 1 in the bit time when CTRL2[BTE] = 1. Otherwise, it has no effect. Nominal Time Segment 1 = (NTSEG1 + 1) × Time Quanta One time quantum = one Sclock period

39.6.2.35 Enhanced Data Phase CAN Bit Timing (EDCBT)

Provides an alternative way to store the data phase CAN bit timing variables described in CAN FD Bit Timing (FDCBT) to achieve higher CAN bit timing resolution.

This register is used by the module only if CTRL2[BTE] = 1; otherwise, a write operation has no effect and all fields are read as zero.

Soft reset does not affect the contents of this register.

This register can be written only in Freeze mode; the module blocks it in other modes.

NOTE

Ensure that bit time settings and protocol engine tolerance are in compliance with the CAN Protocol standard ISO 11898-1_2024.

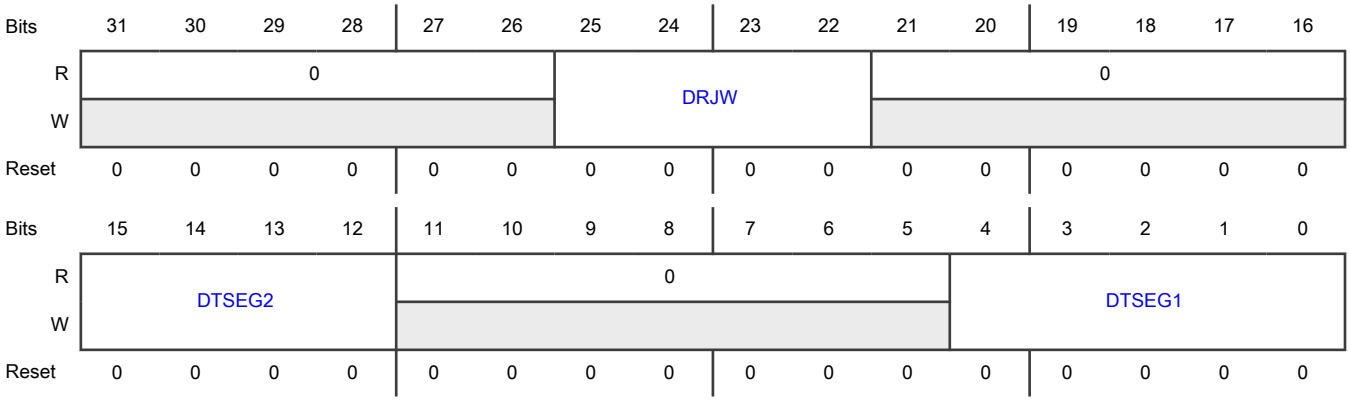
NOTE

DTSEG1 must be at least two time quanta.

Offset

Register	Offset
EDCBT	BF8h

Diagram



Fields

Field	Description
31-26 —	Reserved
25-22 DRJW	Data Phase Resynchronization Jump Width Defines the maximum number of time quanta that one resynchronization can change a data phase bit time when CTRL2[BTE] = 1. Otherwise, it has no effect. Data Phase Resync Jump Width = DRJW + 1.
21-16 —	Reserved
15-12 DTSEG2	Data Phase Time Segment 2 Defines the length of time segment 2 in the data phase bit time when CTRL2[BTE] = 1. Otherwise, it has no effect. Data Phase Time Segment 2 = (DTSEG2 + 1) × Time Quanta. One Time Quantum = one Sclock period.
11-5 —	Reserved
4-0 DTSEG1	Data Phase Segment 1 Defines the length of time segment 1 in the data phase bit time when CTRL2[BTE] = 1. Otherwise, it has no effect.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Data Phase Time Segment 1 = (NTSEG1 + 1) × Time Quanta. One Time Quantum = one Sclock period.

39.6.2.36 Enhanced Transceiver Delay Compensation (ETDC)

Contains extended versions of FDCTRL[TDCOFF] and FDCTRL[TDCVAL]. This register is used by the module only if CTRL2[BTE] = 1. Otherwise, a write operation has no effect and all fields are read as zero.

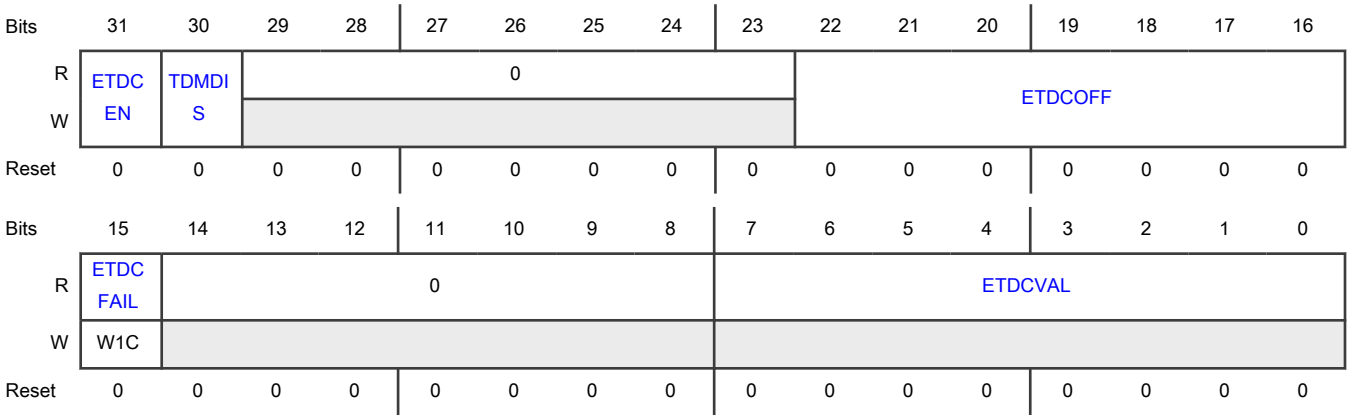
NOTE

See Transmitter delay compensation in the CAN Protocol standard ISO 11898-1_2024 for details.

Offset

Register	Offset
ETDC	BFCh

Diagram



Fields

Field	Description
31 ETDCEN	Transceiver Delay Compensation Enable Enables the TDC feature. It can be written in Freeze mode only. **NOTE** See Transmitter delay compensation in the CAN Protocol standard ISO 11898-1_2024 for details. **NOTE** TDC must be disabled when the Loop Back Mode is enabled. See CTRL1[LPB].

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Disable 1 - Enable
30 TDMDIS	Transceiver Delay Measurement Disable Disables the transceiver delay measurement. When the TDC measurement is disabled, only ETDC[ETDCOFF] determines the secondary sample point position. If TCD measurement is enabled, the sum of the transceiver delay measurement plus the enhanced TDC offset determines the secondary sample point position. Soft reset does not affect this field. **NOTE** This bit can be enabled only if CTRL2[BTE] = 1. 0 - Enable 1 - Disable
29-23 —	Reserved
22-16 ETDCOFF	Enhanced Transceiver Delay Compensation Offset Contains the offset value to be added to the loop delay of the measured transceiver. This value defines the position of the delayed comparison point when bit rate switching is active. See Transceiver delay compensation for details on how the loop delay measurement is performed. This field can be written in Freeze mode only. Its value can be defined in protocol engine (PE) clock periods (CANCLK, see Protocol timing for more details). It must be smaller than the CAN bit duration in the data bit rate for proper operation. Do not write 0 to this field. **NOTE** If CTRL2[BTE] becomes 1 after a chip-level hard reset, this field is read as 1h.
15 ETDCFAIL	Transceiver Delay Compensation Fail Indicates whether the transceiver delay compensation (TDC) mechanism is out of range. In this case, it is unable to compensate the loop delay of the transceiver and successfully compare the delayed received bits to the transmitted ones. (See Transceiver delay compensation.) This field becomes 0 the first time FlexCAN detects the out of range condition. 0 - In range 1 - Out of range
14-8 —	Reserved
7-0	Enhanced Transceiver Delay Compensation Value

Table continues on the next page...

Table continued from the previous page...

Field	Description
ETDCVAL	Contains ETDC[ETDCOFF] added to the measured value of the transceiver loop delay in the latest transmitted CAN FD frame, with BRS = 1. The module only updates this field when ETDC[ETDCEN] = 1. Soft reset affects this field. **NOTE** If ETDC[TDMDIS] = 1, this field stores ETDC[ETDCOFF] only.

39.6.2.37 CAN FD Control (FDCTRL)

Contains control bits for CAN FD operation. It also defines the data size of message buffers allocated in different partitions of RAM (memory blocks) as described in Table 274.

When an 8-byte payload is selected:

- Block R0 allocates MB0–MB31.
- Block R1 allocates MB32–MB63.
- Block R2 allocates MB64–MB95.

When a payload larger than eight bytes is selected, the maximum number of message buffers in a block is limited as described below.

Table 274. Number of message buffers

Payload size	Maximum number of message buffers per RAM block
8 bytes	32
16 bytes	21
32 bytes	12
64 bytes	7

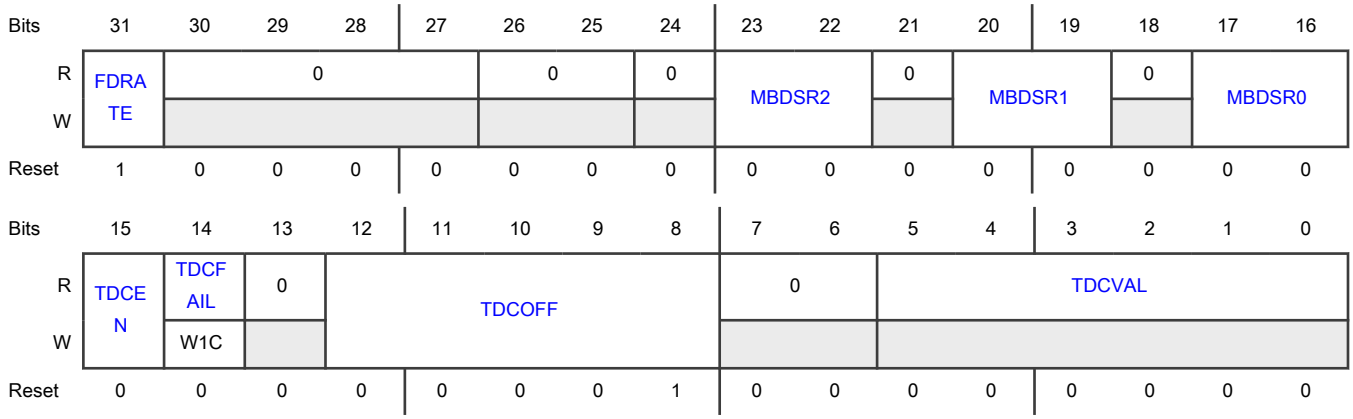
One memory block fits exactly 32 message buffers with an 8-byte payload. For other possible payload sizes, empty memory may exist between the last message buffer in a block and the beginning of the next block. This empty memory corresponds to less than one message buffer, and must not be used.

Soft reset does not affect the contents of this register.

Offset

Register	Offset
FDCTRL	C00h

Diagram



Fields

Field	Description
31 FDRATE	Bit Rate Switch Enable Enables the effect of the Bit Rate Switch (BRS bit) during the data phase of TX messages. When 1, if BRS = 1 in the TX message buffer, frames are transmitted with bit rate switching. When 0, frames are transmitted at a nominal rate, and the BRS bit in the TX MB has no effect. The CPU can write to this field at any time. However, its effect becomes active only under one of these conditions: • The CAN bus is in the Wait for Bus Idle state. • The CAN bus is in the Bus Idle state. • The CAN bus is in the Bus Off state. • The current frame under reception or transmission reaches the interframe space. By writing 0 to FDCTRL[FDRATE], the CPU can force all bits in CAN FD messages to be transmitted at nominal bit rate. This transmission occurs regardless of the value in the BRS bit of the TX message buffers. 0 - Disable 1 - Enable
30-27 —	Reserved
26-25 —	Reserved
24 —	Reserved
23-22 MBDSR2	Message Buffer Data Size for Region 2 Selects the data size per message buffer for region R2 of message buffers allocated in RAM. This field can be written in Freeze mode only.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	00 - 8 bytes 01 - 16 bytes 10 - 32 bytes 11 - 64 bytes
21 —	Reserved
20-19 MBDSR1	Message Buffer Data Size for Region 1 Selects the data size per message buffer for region R1 of message buffers allocated in RAM. This field can be written in Freeze mode only. 00 - 8 bytes 01 - 16 bytes 10 - 32 bytes 11 - 64 bytes
18 —	Reserved
17-16 MBDSR0	Message Buffer Data Size for Region 0 Selects the data size per message buffer for region R0 of message buffers allocated in RAM. This field can be written in Freeze mode only. 00 - 8 bytes 01 - 16 bytes 10 - 32 bytes 11 - 64 bytes
15 TDCEN	Transceiver Delay Compensation Enable Enables the TDC feature. It can be written in Freeze mode only. See Transmitter delay compensation in the CAN Protocol standard ISO 11898-1_2024 for details. TDC must be disabled when Loopback mode is enabled (see CTRL1[LPB]). **NOTE** If CTRL2[BTE] = 1, this field is read as 0 and a write operation has no effect. 0 - Disable 1 - Enable
14	Transceiver Delay Compensation Fail

Table continues on the next page...

Table continued from the previous page...

Field	Description
TDCFAIL	Indicates whether the Transceiver Delay Compensation (TDC) mechanism is out of range. In this case, the mechanism cannot compensate for the loop delay of the transceiver and successfully compare the delayed received bits to the transmitted ones (see Transceiver delay compensation). The first time that FlexCAN detects the out-of-range condition, this field becomes 1. **NOTE** If CTRL2[BTE] = 1, this field is read as 0 and a write operation has no effect. 0 - In range 1 - Out of range
13 —	Reserved
12-8 TDCOFF	Transceiver Delay Compensation Offset Contains the offset value to be added to the loop delay of the measured transceiver. This value defines the position of the delayed comparison point when bit rate switching is active. See Transceiver delay compensation for details about loop delay measurement. This field can be written in Freeze mode only. Its value can be defined in Protocol Engine Clock periods (CANCLK, see Protocol timing for more details). The value must be smaller than the CAN bit duration in the data bit rate for proper operation. **NOTE** If CTRL2[BTE] = 1, TDCOFF is read as 0 and a write operation has no effect. Do not write 0 to this field.
7-6 —	Reserved
5-0 TDCVAL	Transceiver Delay Compensation Value Contains the value of the transceiver loop delay measured from the transmitted EDL-to-R0 transition edge to the respective received one added to FDCTRL[TDCOFF]. This value is an integer multiple of the Protocol Engine Clock period (CANCLK). If CTRL2[BTE] = 1, this field is read as 0. See Protocol timing for details on the loop delay measurement.

39.6.2.38 CAN FD Bit Timing (FDCBT)

Stores the CAN bit timing variables used in the data phase of CAN FD messages when the FDCTRL[FDRATE] = 1, compatible with the CAN FD specification. Fields in this register define:

- The time quantum duration
- The number of time quanta per CAN bit
- The sample point position for the data bit rate portion of a CAN FD message with BRS = 1

Soft reset does not affect the contents of this register.

The sum of the Fast Propagation Segment (FPROPSEG) and Fast Phase Segment 1 (FPSEG1) must be at least two time quanta.

Ensure bit time settings and protocol engine tolerance are in compliance with the CAN Protocol standard ISO 11898-1_2024.

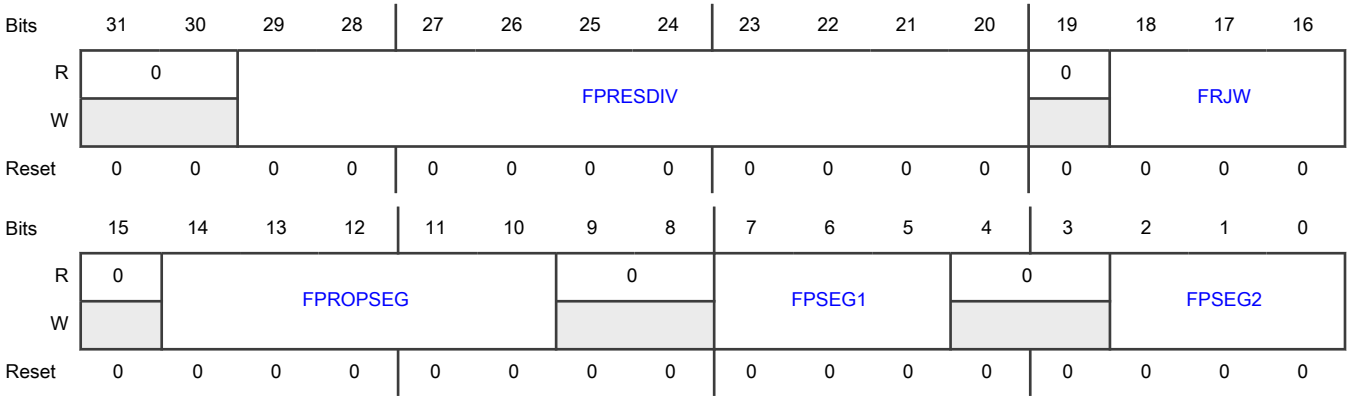
NOTE

If CTRL2[BTE] = 1, this register is read as zero and a write operation has no effect.

Offset

Register	Offset
FDCBT	C04h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-20 FPRESDIV	Fast Prescaler Division Factor Defines the ratio between the PE clock frequency and the serial clock (Sclock) frequency in the data bit rate portion of a CAN FD message with BRS = 1. The Sclock period defines the time quantum of the CAN FD protocol for the data bit rate. This field can be written only in Freeze mode; the module blocks it in other modes. Sclock frequency = PE clock frequency ÷ (FPRESDIV + 1). **NOTE** To minimize errors when processing FD frames, use the same value for this field and for CTRL1[PRESDIV] or CBT[EPRESDIV]. See the first note in CAN FD frames for details.
19 —	Reserved
18-16	Fast Resync Jump Width

Table continues on the next page...

Table continued from the previous page...

Field	Description
FRJW	Defines the maximum number of time quanta that one resynchronization can change a bit time in the data bit rate portion of a CAN FD message with BRS = 1. This field can be written only in Freeze mode; the module blocks it in other modes. Resync Jump Width = FSJW + 1. One Time Quantum = one Sclock period.
15 —	Reserved
14-10 FPROPSEG	Fast Propagation Segment Defines the length of the propagation segment in the bit time in the data bit rate portion of a CAN FD message with BRS = 1. This field can be written only in Freeze mode; the module blocks it in other modes. Propagation Segment Time = FPROPSEG × Time Quanta. One Time Quantum = one Sclock period.
9-8 —	Reserved
7-5 FPSEG1	Fast Phase Segment 1 Defines the length of phase segment 1 in the bit time in the data bit rate portion of a CAN FD message with BRS = 1. This field can be written only in Freeze mode; the module blocks it in other modes. Phase Segment 1 = (FPSEG1 + 1) × Time Quanta. One Time Quantum = one Sclock period.
4-3 —	Reserved
2-0 FPSEG2	Fast Phase Segment 2 Defines the length of phase segment 2 in the data bit rate portion of a CAN FD message with BRS = 1. This field can be written only in Freeze mode; the module blocks it in other modes. Phase Segment 2 = (FPSEG2 + 1) × Time Quanta. One Time Quantum = one Sclock period.

39.6.2.39 CAN FD CRC (FDCRC)

Provides information about the cyclic redundancy check (CRC) of transmitted messages.

FlexCAN uses different CRC polynomials for different frame formats.

- The CRC_15 polynomial is used for all frames in CAN format.
- The CRC_17 polynomial is used for frames in CAN FD format with a DATA FIELD up to 16 bytes.
- The CRC_21 polynomial is used for frames in CAN FD format with a DATA FIELD longer than 16 bytes.

Each polynomial shown below results in a Hamming distance of 6. This register is updated at the same time that the TX Interrupt flag is set.

$$CRC_15 = C599h: \ (x^{15} + x^{14} + x^{10} + x^{8} + x^{7} + x^{4} + x^{3} + 1)$$

$$CRC_17 = 3685Bh: \ (x^{17} + x^{16} + x^{14} + x^{13} + x^{11} + x^{6} + x^{4} + x^{3} + x^{1} + 1)$$

$$CRC_21 = 302899h: \ (x^{21} + x^{20} + x^{13} + x^{11} + x^{7} + x^{4} + x^{3} + 1)$$

Equation 37. CRC polynomial used on CAN frame

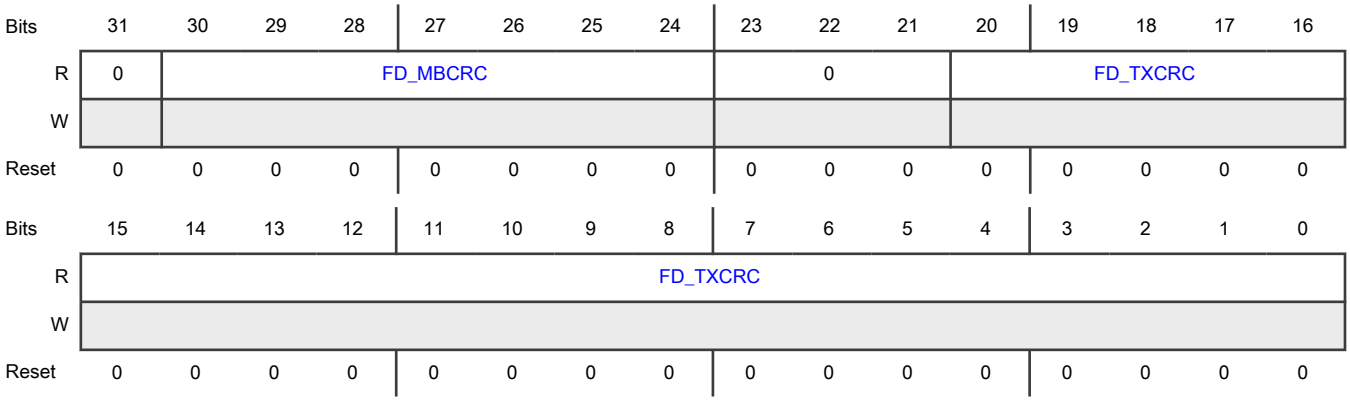
NOTE

See CRC sequence calculation in the CAN Protocol standard ISO 11898-1_2024 for details.

Offset

Register	Offset
FDCRC	C08h

Diagram



Fields

Field	Description
31 —	Reserved
30-24 FD_MBCRC	CRC Message Buffer Number for FD_TXCRC Indicates the number of the message buffer corresponding to the value in FDCRC[FD_TXCRC], for both FD and non-FD frames. It reports the same information as in CRCR[MBCRC].
23-21	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
20-0 FD_TXCRC	Extended Transmitted CRC value Contains the CRC value calculated over the most recent transmitted message. Different CRC polynomials are used for different frame formats. For CRC_15 and CRC_17, the six most significant bits and the four most significant bits are reported as zeroes, respectively. For CRC_15, this field has the same content as Cyclic Redundancy Check (CRCR).

39.6.2.40 Enhanced RX FIFO Control (ERFCR)

Defines the Enhanced RX FIFO configuration.

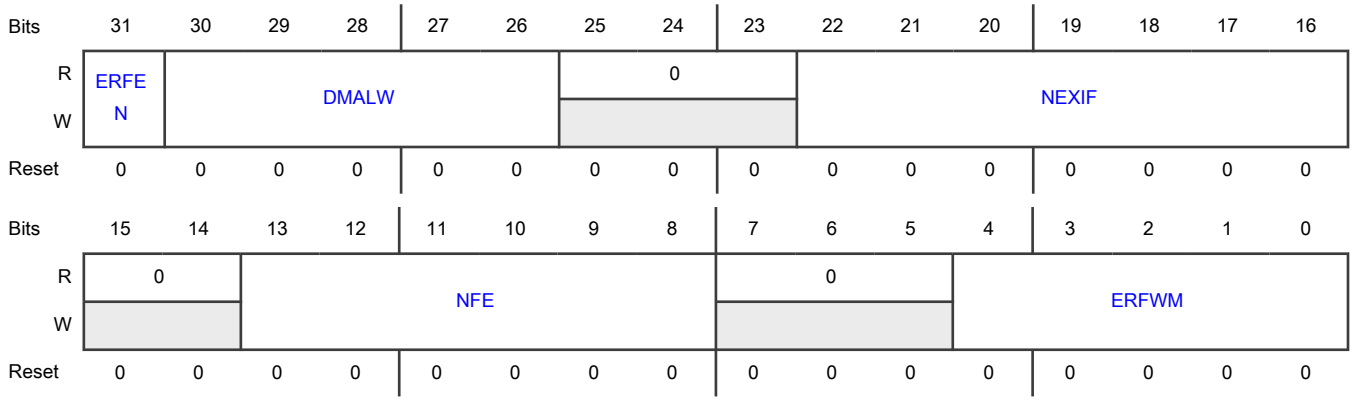
This register can be written only in Freeze mode.

Soft reset does not affect any of the contents of this register.

Offset

Register	Offset
ERFCR	C0Ch

Diagram



Fields

Field	Description
31 ERFEN	Enhanced RX FIFO enable Enables the Enhanced RX FIFO.

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>NOTE
If MCR[RFEN] = 1, do not write 1 to this field.

0 - Disable

1 - Enable</td></tr>
<tr><td>30-26

DMALW</td><td>DMA Last Word

Defines the last DMA address for each Enhanced RX FIFO element.

This table shows the number of elements and the last address for each Enhanced RX FIFO element according to the value of DMALW.

<table>
<tr><th>DMALW</th><th>Number of 32-bit words transferred</th><th>Last FIFO address</th></tr>
<tr><td>0</td><td>1</td><td>2000h</td></tr>
<tr><td>1</td><td>2</td><td>2004h</td></tr>
<tr><td>2</td><td>3</td><td>2008h</td></tr>
<tr><td>3</td><td>4</td><td>200Ch</td></tr>
<tr><td>4</td><td>5</td><td>2010h</td></tr>
<tr><td>5</td><td>6</td><td>2014h</td></tr>
<tr><td>6</td><td>7</td><td>2018h</td></tr>
<tr><td>7</td><td>8</td><td>201Ch</td></tr>
<tr><td>8</td><td>9</td><td>2020h</td></tr>
<tr><td>9</td><td>10</td><td>2024h</td></tr>
<tr><td>10</td><td>11</td><td>2028h</td></tr>
<tr><td>11</td><td>12</td><td>202Ch</td></tr>
<tr><td>12</td><td>13</td><td>2030h</td></tr>
<tr><td>13</td><td>14</td><td>2034h</td></tr>
<tr><td>14</td><td>15</td><td>2038h</td></tr>
<tr><td>15</td><td>16</td><td>203Ch</td></tr>
<tr><td>16</td><td>17</td><td>2040h</td></tr>
<tr><td>17</td><td>18</td><td>2044h</td></tr>
<tr><td>18</td><td>19</td><td>2048h</td></tr>
<tr><td>19</td><td>20</td><td>204Ch</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>NOTE
Undefined DMALW values in the table are reserved and must not be used.</td></tr>
<tr><td>25-23
—</td><td>Reserved</td></tr>
<tr><td>22-16
NEXIF</td><td>Number of Extended ID Filter Elements
Defines the number of extended ID filter elements used during the Enhanced RX FIFO matching process.
The value of this field must be less than or equal to NFE + 1.
The number of standard ID filter elements is 2 × (NFE - NEXIF + 1).
This table shows the number of extended ID filters and standard ID filters available for Enhanced RX FIFO if all filter elements are used.
<table>
<tr><th>NEXIF</th><th>NFE</th><th>Number of Extended ID filter elements</th><th>Number of Standard ID filter elements</th></tr>
<tr><td>0</td><td>63</td><td>0</td><td>128</td></tr>
<tr><td>1</td><td>63</td><td>1</td><td>126</td></tr>
<tr><td>2</td><td>63</td><td>2</td><td>124</td></tr>
<tr><td>3</td><td>63</td><td>3</td><td>122</td></tr>
<tr><td>4</td><td>63</td><td>4</td><td>120</td></tr>
<tr><td>5</td><td>63</td><td>5</td><td>118</td></tr>
<tr><td>6</td><td>63</td><td>6</td><td>116</td></tr>
<tr><td>7</td><td>63</td><td>7</td><td>114</td></tr>
<tr><td>8</td><td>63</td><td>8</td><td>112</td></tr>
<tr><td>9</td><td>63</td><td>9</td><td>110</td></tr>
<tr><td>10</td><td>63</td><td>10</td><td>108</td></tr>
<tr><td>11</td><td>63</td><td>11</td><td>106</td></tr>
<tr><td>12</td><td>63</td><td>12</td><td>104</td></tr>
<tr><td>13</td><td>63</td><td>13</td><td>102</td></tr>
<tr><td>14</td><td>63</td><td>14</td><td>100</td></tr>
<tr><td>15</td><td>63</td><td>15</td><td>98</td></tr>
<tr><td>16</td><td>63</td><td>16</td><td>96</td></tr>
<tr><td>17</td><td>63</td><td>17</td><td>94</td></tr>
<tr><td>18</td><td>63</td><td>18</td><td>92</td></tr>
</table></td></tr>
</table>

Table continued from the previous page...

Field	Description			
	NEXIF	NFE	Number of Extended ID filter elements	Number of Standard ID filter elements
	19	63	19	90
	20	63	20	88
	21	63	21	86
	22	63	22	84
	23	63	23	82
	24	63	24	80
	25	63	25	78
	26	63	26	76
	27	63	27	74
	28	63	28	72
	29	63	29	70
	30	63	30	68
	31	63	31	66
	32	63	32	64
	33	63	33	62
	34	63	34	60
	35	63	35	58
	36	63	36	56
	37	63	37	54
	38	63	38	52
	39	63	39	50
	40	63	40	48
	41	63	41	46
	42	63	42	44
	43	63	43	42
	44	63	44	40
	45	63	45	38
	46	63	46	36
	47	63	47	34

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>NEXIF</th><th>NFE</th><th>Number of Extended ID filter elements</th><th>Number of Standard ID filter elements</th></tr>
<tr><td>48</td><td>63</td><td>48</td><td>32</td></tr>
<tr><td>49</td><td>63</td><td>49</td><td>30</td></tr>
<tr><td>50</td><td>63</td><td>50</td><td>28</td></tr>
<tr><td>51</td><td>63</td><td>51</td><td>26</td></tr>
<tr><td>52</td><td>63</td><td>52</td><td>24</td></tr>
<tr><td>53</td><td>63</td><td>53</td><td>22</td></tr>
<tr><td>54</td><td>63</td><td>54</td><td>20</td></tr>
<tr><td>55</td><td>63</td><td>55</td><td>18</td></tr>
<tr><td>56</td><td>63</td><td>56</td><td>16</td></tr>
<tr><td>57</td><td>63</td><td>57</td><td>14</td></tr>
<tr><td>58</td><td>63</td><td>58</td><td>12</td></tr>
<tr><td>59</td><td>63</td><td>59</td><td>10</td></tr>
<tr><td>60</td><td>63</td><td>60</td><td>8</td></tr>
<tr><td>61</td><td>63</td><td>61</td><td>6</td></tr>
<tr><td>62</td><td>63</td><td>62</td><td>4</td></tr>
<tr><td>63</td><td>63</td><td>63</td><td>2</td></tr>
<tr><td>64</td><td>63</td><td>64</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>15-14
—</td><td>Reserved</td></tr>
<tr><td>13-8
NFE</td><td>Number of Enhanced RX FIFO Filter Elements
Defines the total number of filter elements used during the enhanced RX FIFO matching process according to the table.
<table>
<tr><th>NFE</th><th>Maximum number of extended ID filter elements (NEXIF = NFE + 1)</th><th>Maximum number of standard ID filter elements (NEXIF = 0)</th></tr>
<tr><td>0</td><td>1</td><td>2</td></tr>
<tr><td>1</td><td>2</td><td>4</td></tr>
<tr><td>2</td><td>3</td><td>6</td></tr>
<tr><td>3</td><td>4</td><td>8</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Description

NFE	Maximum number of extended ID filter elements (NEXIF = NFE + 1)	Maximum number of standard ID filter elements (NEXIF = 0)
4	5	10
5	6	12
6	7	14
7	8	16
8	9	18
9	10	20
10	11	22
11	12	24
12	13	26
13	14	28
14	15	30
15	16	32
16	17	34
17	18	36
18	19	38
19	20	40
20	21	42
21	22	44
22	23	46
23	24	48
24	25	50
25	26	52
26	27	54
27	28	56
28	29	58
29	30	60
30	31	62
31	32	64
32	33	66

Table continues on the next page...

Table continued from the previous page...

Field	Description		
	NFE	Maximum number of extended ID filter elements (NEXIF = NFE + 1)	Maximum number of standard ID filter elements (NEXIF = 0)
	33	34	68
	34	35	70
	35	36	72
	36	37	74
	37	38	76
	38	39	78
	39	40	80
	40	41	82
	41	42	84
	42	43	86
	43	44	88
	44	45	90
	45	46	92
	46	47	94
	47	48	96
	48	49	98
	49	50	100
	50	51	102
	51	52	104
	52	53	106
	53	54	108
	54	55	110
	55	56	112
	56	57	114
	57	58	116
	58	59	118
	59	60	120
	60	61	122
	61	62	124

Table continues on the next page...

Table continued from the previous page...

Field	Description
	<table><tr><th>NFE</th><th>Maximum number of extended ID filter elements (NEXIF = NFE + 1)</th><th>Maximum number of standard ID filter elements (NEXIF = 0)</th></tr><tr><td>62</td><td>63</td><td>126</td></tr><tr><td>63</td><td>64</td><td>128</td></tr></table>
7-5 —	Reserved
4-0 ERFWM	Enhanced RX FIFO Watermark Defines the minimum number of CAN messages stored in the Enhanced RX FIFO. When that number is reached, ERFSR[ERFWMI] becomes 1. Minimum number of CAN messages = ERFWM + 1. **NOTE** If MCR[DMA] = 1, write 0h to this field.

39.6.2.41 Enhanced RX FIFO Interrupt Enable (ERFIER)

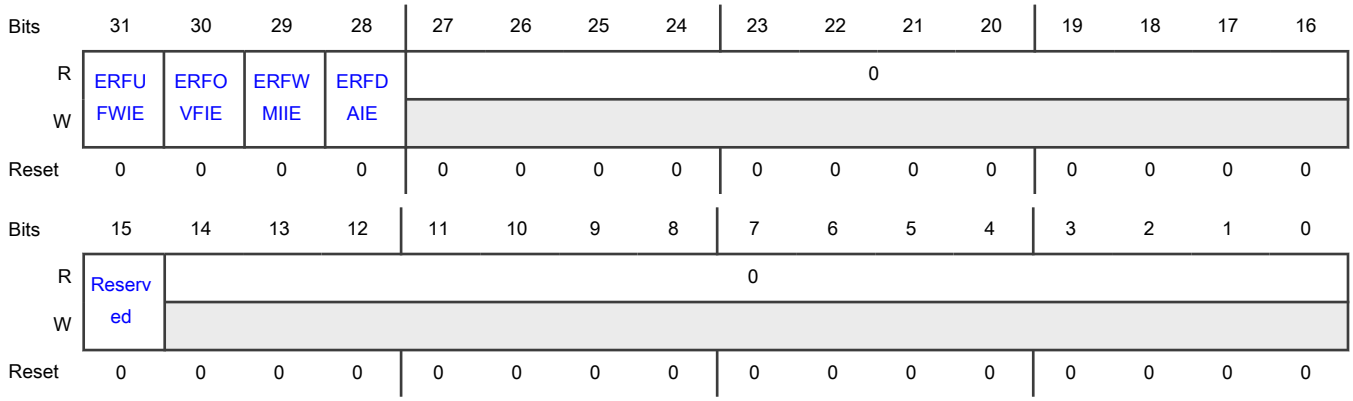
Contains the interrupt enables for the Enhanced RX FIFO.

Soft reset does not affect this register.

Offset

Register	Offset
ERFIER	C10h

Diagram



Fields

Field	Description
31 ERFUFWIE	Enhanced RX FIFO Underflow Interrupt Enable Enables interrupt for ERFSR[ERFUFW]. 0 - Disable 1 - Enable
30 ERFOVFIE	Enhanced RX FIFO Overflow Interrupt Enable Enables interrupt for ERFSR[ERFOVF]. 0 - Disable 1 - Enable
29 ERFWMIIE	Enhanced RX FIFO Watermark Indication Interrupt Enable Enables interrupt for ERFSR[ERFWMI]. 0 - Disable 1 - Enable
28 ERFDAIE	Enhanced RX FIFO Data Available Interrupt Enable Enables interrupt for ERFSR[ERFDA]. 0 - Disable 1 - Enable
27-16 —	Reserved
15 —	Reserved
14-0 —	Reserved

39.6.2.42 Enhanced RX FIFO Status (ERFSR)

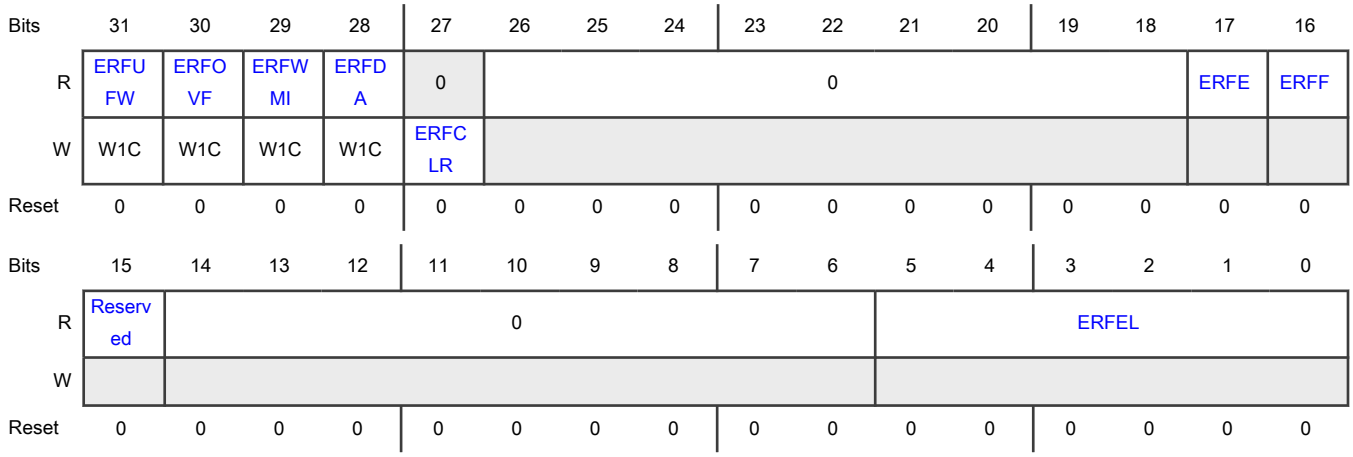
Contains the status fields of the Enhanced RX FIFO including error indications and a clear FIFO field.

Soft reset does not affect this register.

Offset

Register	Offset
ERFSR	C14h

Diagram



Fields

Field	Description
31 ERFUFW	Enhanced RX FIFO Underflow Flag Indicates whether an underflow condition occurred in the enhanced RX FIFO. If ERFIER[ERFUFWIE] = 1, this field generates an interrupt. 0 - No such occurrence 1 - Underflow
30 ERFOVF	Enhanced RX FIFO Overflow Flag Indicates whether an overflow condition occurred in the Enhanced RX FIFO. If ERFIER[ERFOVFIE] = 1, this field generates an interrupt. 0 - No such occurrence 1 - Overflow
29 ERFWMI	Enhanced RX FIFO Watermark Indication Flag Indicates whether the number of messages available in the Enhanced RX FIFO is greater than the watermark defined in ERFCR[ERFWM]. If ERFIER[ERFWMIIE] = 1, this field generates an interrupt. 0 - No such occurrence 1 - Number of messages in FIFO is greater than the watermark
28 ERFDA	Enhanced RX FIFO Data Available Flag Indicates whether there is at least one message stored in the ERX FIFO. If ERFIER[ERFDAIE] = 1, this field generates an interrupt. 0 - No such occurrence 1 - At least one message stored in Enhanced RX FIFO

Table continues on the next page...

Table continued from the previous page...

Field	Description
27 ERFCLR	Enhanced RX FIFO Clear Writing 1 to this field during Freeze mode resets the internal FIFO pointers, but the FIFO content stored in RAM is not changed. Writing to this field outside Freeze mode, or writing 0 to this field, has no effect. 0 - No effect 1 - Clear enhanced RX FIFO content
26-18 —	Reserved
17 ERFE	Enhanced RX FIFO Empty Flag Indicates whether Enhanced RX FIFO is empty. 0 - Not empty 1 - Empty
16 ERFF	Enhanced RX FIFO Full Flag Indicates whether enhanced RX FIFO is full. 0 - Not full 1 - Full
15 —	Reserved
14-6 —	Reserved
5-0 ERFEL	Enhanced RX FIFO Elements Indicates the number of CAN messages stored in the Enhanced RX FIFO.

39.6.2.43 High-Resolution Timestamp (HR_TIME_STAMP0 - HR_TIME_STAMP95)

Stores a copy of a 32-bit timer at the start or end of a CAN frame.

HR_TIME_STAMP0 stores the 32-bit timestamp associated with MB0, HR_TIME_STAMP1 stores the 32-bit timestamp associated with MB1, and so on.

Reset does not affect these registers.

NOTE

Do not write to these registers outside Freeze mode.

Table 275. High-Resolution Timestamp register operation

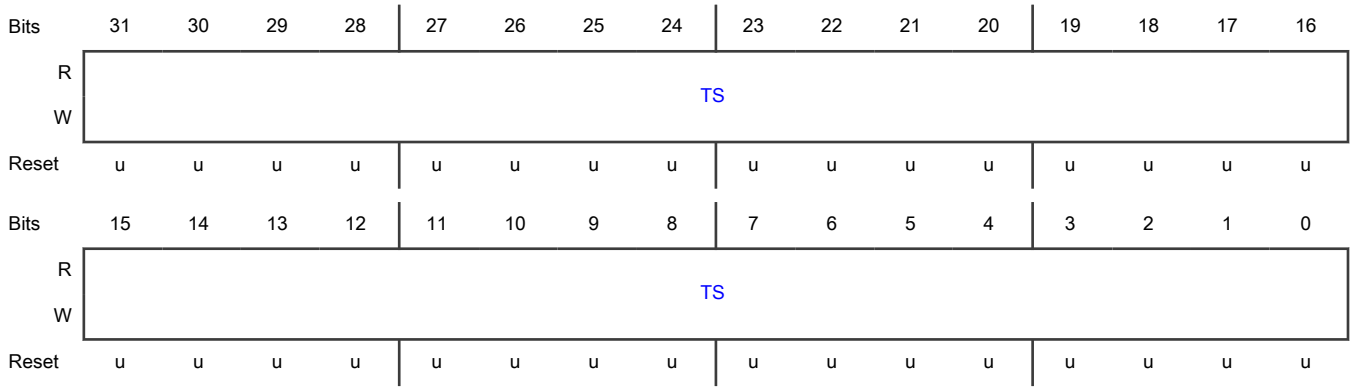
TSTAMPCAP	Captured timebase	Capture point
b00	None	None
b01	32 bits of high-resolution on-chip timer	Seventh bit of the end-of-frame field for transmission and sixth bit of the end-of-frame field for reception.
b10	32 bits of high-resolution on-chip timer	Start of frame
b11	32 bits of high-resolution on-chip timer	Start of frame for classical CAN frame format and res bit for CAN FD frame format

Offset

For n = 0 to 95:

Register	Offset
HR_TIME_STAMPn	C30h + (n × 4h)

Diagram



Fields

Field	Description
31-0 TS	High-Resolution Timestamp Captures the copy of the timestamp of corresponding message buffer. This field always captures a 32-bit timer value.

39.6.2.44 Enhanced RX FIFO Filter Element (ERFFEL0 - ERFFEL127)

Stores the filter elements of the Enhanced RX FIFO.

For standard ID filtering, each ERFFEL register stores one filter element. For extended ID filtering, each pair of ERFFEL registers stores one filter element.

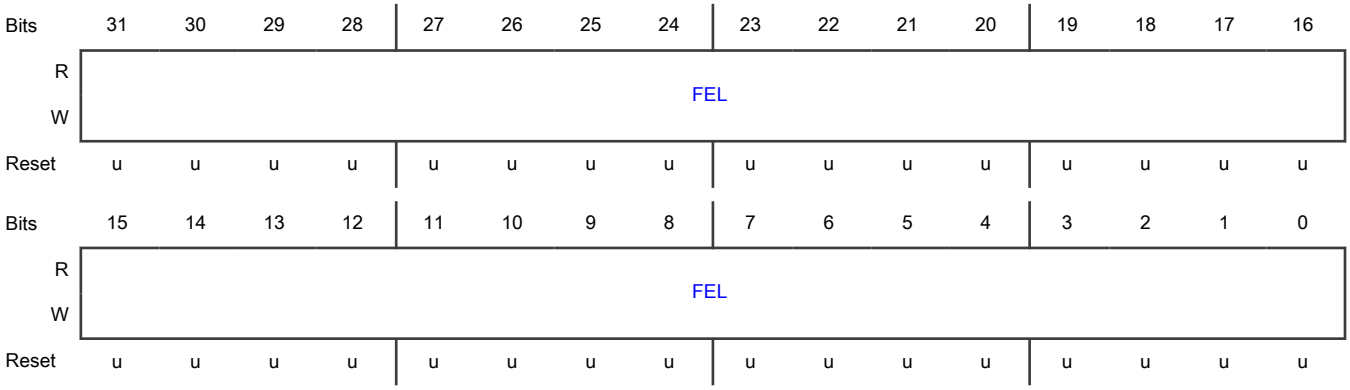
ERFFEL registers can be written only in Freeze mode; otherwise, the module blocks them. Reset does not affect these registers. They are located in RAM and must be explicitly initialized prior to any reception.

Offset

For n = 0 to 127:

Register	Offset
ERFFELn	3000h + (n × 4h)

Diagram



Fields

Field	Description
31-0 FEL	Filter Element Bits Stores filter elements. Each filter element is used during the match process. If the matching criteria are met, a message is stored in the Enhanced RX FIFO.

39.6.3 Message buffer structure

The message buffer structure used by FlexCAN is represented in the following figure. Both extended (29-bit identifier) and standard (11-bit identifier) frames used in the CAN specification (Version 2.0 Part B) are represented. Each individual message buffer is 16, 24, 40, or 72 bytes, depending on the quantity of data bytes allocated for the message payload: 8, 16, 32, or 64 data bytes, respectively.

The memory area 80h–67Fh is used by the message buffers. When CAN FD is enabled, the exact address for each message buffer depends on the size of its payload. See FlexCAN memory partition for CAN FD.

Table 276. Message buffer structure example with 64-byte payload

	31	30	29	28	27 – 24	23	22	21	20	19 – 18	17 – 16	15 – 8	7 – 0
0h	EDL	BRS	ESI		CODE		SRR	IDE	RTR	DLC		TIMESTAMP	
4h	PRIO			ID (standard/extended)						ID (extended)			
8h	Data byte 0				Data byte 1							Data byte 2	Data byte 3
Ch	Data byte 4				Data byte 5							Data byte 6	Data byte 7
10h	Data byte 8				Data byte 9							Data byte 10	Data byte 11

Table continues on the next page...

Table 276. Message buffer structure example with 64-byte payload (continued)

14h	Data byte 12	Data byte 13	Data byte 14	Data byte 15
18h	Data byte 16	Data byte 17	Data byte 18	Data byte 19
1Ch	Data byte 20	Data byte 21	Data byte 22	Data byte 23
20h	Data byte 24	Data byte 25	Data byte 26	Data byte 27
24h	Data byte 28	Data byte 29	Data byte 30	Data byte 31
28h	Data byte 32	Data byte 33	Data byte 34	Data byte 35
2Ch	Data byte 36	Data byte 37	Data byte 38	Data byte 39
30h	Data byte 40	Data byte 41	Data byte 42	Data byte 43
34h	Data byte 44	Data byte 45	Data byte 46	Data byte 47
38h	Data byte 48	Data byte 49	Data byte 50	Data byte 51
3Ch	Data byte 52	Data byte 53	Data byte 54	Data byte 55
40h	Data byte 56	Data byte 57	Data byte 58	Data byte 59
44h	Data byte 60	Data byte 61	Data byte 62	Data byte 63

		= Unimplemented or reserved

Table 277. Field descriptions

Mnemonic	Field	Description
EDL	Extended Data Length	Distinguishes between CAN format and CAN FD format frames. EDL must not be 1 for message buffers configured to RANSWER with code field 1010b (see Table 278).
BRS	Bit Rate Switch	Defines whether the bit rate is switched inside a CAN FD format frame.
ESI	Error State Indicator	Indicates whether the transmitting node is error-active or error-passive.
CODE	Message Buffer Code	Can be accessed (read or write) by the CPU and by the FlexCAN module itself, as part of the message buffer matching and arbitration process. The encoding is shown in Table 278 and Table 279. See Functional description.

Table 278. Message buffer code for RX buffers

CODE description	RX code BEFORE RX new frame	SRV[1]	RX code AFTER successful reception[2]	RRS[3]	Comment
0000b: INACTIVE. Message buffer is not active.	INACTIVE	—	—	—	Message buffer does not participate in the matching process.
0100b: EMPTY. Message buffer is active and empty.	EMPTY	—	FULL	—	When a frame is received successfully (after Move-in), CODE is automatically updated to FULL.

Table continues on the next page...

Table 278. Message buffer code for RX buffers (continued)

CODE description	RX code BEFORE RX new frame	SRV[1]	RX code AFTER successful reception[2]	RRS[3]	Comment
0010b: FULL. Message buffer is full.	FULL	Yes	FULL	—	The act of reading the Control and Status word followed by unlocking the message buffer (SRV) does not make CODE return to EMPTY. It remains FULL. If a new frame is moved to the message buffer after the message buffer is serviced, the code remains FULL. See Matching process for matching details related to FULL code.
		No	OVERRUN	—	If the message buffer is FULL and a new frame is moved to this message buffer before the CPU services it, CODE is automatically updated to OVERRUN. See Matching process for details about overrun behavior.
0110b: OVERRUN. Message buffer is being overwritten into a full buffer.	OVERRUN	Yes	FULL	—	If CODE indicates OVERRUN and the CPU has serviced the message buffer, when a new frame is moved to the message buffer, CODE returns to FULL.
		No	OVERRUN	—	If CODE already indicates OVERRUN, and another new frame must be moved, the message buffer is overwritten again, and CODE remains OVERRUN. See Matching process for details about overrun behavior.
1010b: RANSWER[4]. A frame was configured to recognize a Remote	RANSWER	—	TANSWER (1110b)	0	A Remote Answer was configured to recognize a Remote Request

Table continues on the next page...

Table 278. Message buffer code for RX buffers (continued)

CODE description	RX code BEFORE RX new frame	SRV[1]	RX code AFTER successful reception[2]	RRS[3]	Comment
Request frame and transmit a Response frame in return.[5]					frame received. After that, a message buffer is set to transmit a response frame. CODE is automatically changed to TANSWER (1110b). See Matching process for details. If CTRL2[RRS] = 0, transmit a response frame when a remote request frame with the same ID is received.
		—	—	1	This code is ignored during matching and arbitration process. See Matching process for details.
CODE[0] = 1: BUSY. FlexCAN is updating the contents of the message buffer. The CPU must not access the message buffer.	BUSY[6]	—	FULL	—	Indicates that the message buffer is being updated. It automatically becomes 0 and does not interfere with the next CODE.
		—	OVERRUN	—	

1. SRV: Serviced message buffer. Message buffer was read and unlocked by reading TIMER or another message buffer.
2. A frame is considered a successful reception after the frame is moved to a message buffer (move-in process). See Move-in.
3. Remote Request Stored field. See Control 2 (CTRL2).
4. Code 1010b is not considered TX and a message buffer with this code should not be aborted.
5. Code 1010b must be used in message buffers configured in CAN FD format, with EDL = 1.
6. For TX message buffers, the BUSY bit should be ignored upon read, except when MCR[AEN] = 1. If this field is 1, the corresponding message buffer does not participate in the matching process.

Table 279. Message buffer code for TX buffers

CODE Description	TX Code BEFORE TX frame	MB RTR	TX Code AFTER successful transmission	Comment
1000b: INACTIVE. Message buffer is not active.	INACTIVE	—	—	Message buffer does not participate in arbitration process.
1001b: ABORT. Message buffer is aborted.	ABORT	—	—	Message buffer does not participate in arbitration process.
1100b: DATA. Message buffer is a TX data frame (MB RTR must be 0).	DATA	0	INACTIVE	Transmit data frame unconditionally once. After transmission, the message

Table continues on the next page...

Table 279. Message buffer code for TX buffers (continued)

CODE Description	TX Code BEFORE TX frame	MB RTR	TX Code AFTER successful transmission	Comment
				buffer automatically returns to the INACTIVE state.
1100b: REMOTE. Message buffer is a Transmit Remote Request frame (MB RTR must be 1).	REMOTE	1	EMPTY	Transmit remote request frame unconditionally once. After transmission, the message buffer automatically becomes an RX Empty message buffer with the same ID.
1110b: TANSWER. Message buffer is a Transmit Response frame from an incoming Remote Request frame.	TANSWER	—	RANSWER	This intermediate code is automatically written to the message buffer by the CHI as a result of a match to a Remote Request frame. The Remote Response frame is transmitted unconditionally once, then the code automatically returns to RANSWER (1010b). The CPU can also write this code with the same effect. The Remote Response frame can be a data frame or another remote request frame, depending on the value of RTR. See Matching process and Arbitration process for details.

Table 280. RX and TX message buffer field descriptions

Mnemonic	Field	Description
SRR	Substitute Remote Request	Fixed recessive bit, used only in extended format. Write 1 to SRR for transmission (TX Buffers). SRR is stored with the value received on the CAN bus for RX receiving buffers. It can be received as either recessive or dominant. If FlexCAN receives this bit as dominant, it is interpreted as an arbitration loss. 1: Recessive value is compulsory for transmission in extended format frames. 0: Dominant is not a valid value for transmission in extended format frames.
IDE	ID Extended Bit	Identifies whether the frame format is standard or extended. 1: Frame format is extended 0: Frame format is standard

Table continues on the next page...

Table 280. RX and TX message buffer field descriptions (continued)

Mnemonic	Field	Description
RTR	Remote Transmission Request	Affects the behavior of remote frames and is part of the reception filter. See Table 278, Table 279, and CTRL2[RRS]. If FlexCAN transmits this field as 1 (recessive) and receives it as 0 (dominant), it is interpreted as an arbitration loss. If this field is transmitted as 0 (dominant) and it is received as 1 (recessive), FlexCAN treats it as a bit error. If the value received matches the value transmitted, it is considered a successful bit transmission. 1: If message buffer is TX, indicates that the current message buffer may have a Remote Request frame to be transmitted. If the message buffer is RX, incoming remote request frames may be stored. 0: Indicates that the current message buffer has a Data frame to be transmitted. In an RX message buffer, it may be considered in matching processes. **NOTE** When configuring CAN FD frames, this field must be 0.
DLC	Data Length Code	Indicates the length (in bytes) of the RX or TX data, which is located in offset 8h–Fh of the message buffer space (see Table 276). In reception, this field is written by FlexCAN, copied from the DLC field of the received frame. In transmission, this field is written by the CPU and corresponds to the DLC field value of the frame to be transmitted. When RTR = 1, the frame to be transmitted is a remote frame and does not include the data field, regardless of the DLC field (see Table 282).
TIMESTAMP	Free-Running Counter Timestamp	Provides a copy of the Free-Running Timer, captured for TX and RX frames when the beginning of the Identifier field appears on the CAN bus. See Table 281 for Timestamp operation.
PRIO	Local priority	Used only when MCR[LPRIOEN] = 1, and only makes sense for transmit message buffers. These bits are not transmitted. They are appended to the regular ID to define the transmission priority. See Arbitration process.
ID	Frame Identifier	In standard frame format, only the 11 most significant bits (28 to 18) are used for frame identification in both receive and transmit cases. The 18 least significant bits are ignored. In extended frame format, all bits are used for frame identification in both receive and transmit cases.
DATA BYTE 0–63	Data Field	Up to 64 bytes can be used for a data frame, depending on the size of payload selected for the message buffers. For RX frames, the data is stored as it is received from the CAN bus. DATA BYTE (*n*) is valid only if *n* is less than DLC, as shown in Table 282.

Table 281. Timestamp operation

TSTAMPCAP	MBTSBASE	TIMER_SOURCE	Captured timebase	Capture point
b00	bxx	0	CAN_TIMER incremented by CAN bit clock	Second bit of identifier field
b00	bxx	1	CAN_TIMER incremented by on-chip timer clock	Second bit of identifier field
bxx	b00	0	CAN_TIMER incremented by CAN bit clock	Second bit of identifier field
bxx	b00	1	CAN_TIMER incremented by on-chip timer clock	Second bit of identifier field
b01	b01	x	Lower 16 bits of high-resolution on-chip timer	Seventh bit of the end of frame field for transmission and sixth bit of the end of frame field for reception
b01	b10	x	Upper 16 bits of high-resolution on-chip timer	Seventh bit of the end of frame field for transmission and sixth bit of the end of frame field for reception
b10	b01	x	Lower 16 bits of high-resolution on-chip timer	Start of frame
b10	b10	x	Upper 16 bits of high-resolution on-chip timer	Start of frame
b11	b01	x	Lower 16 bits of high-resolution on-chip timer	Start of frame for classical CAN frame format and res bit for CAN FD frame format
b11	b10	x	Upper 16 bits of high-resolution on-chip timer	Start of frame for classical CAN frame format and res bit for CAN FD frame format

Table 282. DATA BYTE validity

DLC	Valid data bytes
0	None
1	DATA BYTE 0
2	DATA BYTE 0–1
3	DATA BYTE 0–2
4	DATA BYTE 0–3
5	DATA BYTE 0–4
6	DATA BYTE 0–5
7	DATA BYTE 0–6
8	DATA BYTE 0–7

Table continues on the next page...

Table 282. DATA BYTE validity (continued)

DLC	Valid data bytes
9	DATA BYTE 0–11
10	DATA BYTE 0–15
11	DATA BYTE 0–19
12	DATA BYTE 0–23
13	DATA BYTE 0–31
14	DATA BYTE 0–47
15	DATA BYTE 0–63

39.6.4 FlexCAN memory partition for CAN FD

When CAN FD is enabled, FlexCAN RAM can be partitioned into blocks of 512 bytes each. Each block can accommodate a number of message buffers depending on the configuration provided by FDCTRL[MBDSR*n*] as shown in Table 283.

Table 283. RAM partition

RAM block	Number of MBs with 8 bytes (default range)	Size control field in FDCTRL	Number of MBs of different sizes, per block
0	0 to 31	MBDSR0	MBDSR0 = 00, 32 MBs with 8-byte payload MBDSR0 = 01, 21 MBs with 16-byte payload MBDSR0 = 10, 12 MBs with 32-byte payload MBDSR0 = 11, 7 MBs with 64-byte payload
1	32 to 63	MBDSR1	MBDSR1 = 00, 32 MBs with 8-byte payload MBDSR1 = 01, 21 MBs with 16-byte payload MBDSR1 = 10, 12 MBs with 32-byte payload MBDSR1 = 11, 7 MBs with 64-byte payload
2	64 to 95	MBDSR2	MBDSR2 = 00, 32 MBs with 8-byte payload MBDSR2 = 01, 21 MBs with 16-byte payload MBDSR2 = 10, 12 MBs with 32-byte payload MBDSR2 = 11, 7 MBs with 64-byte payload

Payload sizes of 16, 32, or 64 bytes may be configured in some or all of RAM blocks. In those cases, the total number of MBs and their respective number order may differ from the default configuration of 8 bytes. Consider an example where:

- Block0 is configured to an 8-byte payload
- Block1 is configured to a 16-byte payload
- Block2 is configured to 32-byte payload

In this case, Table 284 indicates how the message buffers are arranged in RAM.

Table 284. RAM partition example

RAM block	Payload size	Number of MBs in the RAM block	Message buffer range
0	FDCTRL[MBDSR0] = 00, 8-byte payload	32	0 to 31
1	FDCTRL[MBDSR1] = 01, 16-byte payload	21	32 to 52
2	FDCTRL[MBDSR2] = 10, 32-byte payload	12	53 to 64

39.6.5 FlexCAN message buffer memory map

The FlexCAN memory buffers are allocated in memory according to the tables below.

Table 285. 8-byte message buffers

Address offset (hex)	MBDSR = b00 8-byte payload
0080	MB0
0090	MB1
00A0	MB2
00B0	MB3
00C0	MB4
00D0	MB5
00E0	MB6
00F0	MB7
0100	MB8
0110	MB9
0120	MB10
0130	MB11
0140	MB12
0150	MB13
0160	MB14
0170	MB15
0180	MB16
0190	MB17
01A0	MB18
01B0	MB19
01C0	MB20

Table continues on the next page...

Table 285. 8-byte message buffers (continued)

Address offset (hex)	MBDSR = b00 8-byte payload
01D0	MB21
01E0	MB22
01F0	MB23
0200	MB24
0210	MB25
0220	MB26
0230	MB27
0240	MB28
0250	MB29
0260	MB30
0270	MB31
0280	MB32
0290	MB33
02A0	MB34
02B0	MB35
02C0	MB36
02D0	MB37
02E0	MB38
02F0	MB39
0300	MB40
0310	MB41
0320	MB42
0330	MB43
0340	MB44
0350	MB45
0360	MB46
0370	MB47
0380	MB48
0390	MB49
03A0	MB50
03B0	MB51

Table continues on the next page...

Table 285. 8-byte message buffers (continued)

Address offset (hex)	MBDSR = b00 8-byte payload
03C0	MB52
03D0	MB53
03E0	MB54
03F0	MB55
0400	MB56
0410	MB57
0420	MB58
0430	MB59
0440	MB60
0450	MB61
0460	MB62
0470	MB63
0480	MB64
0490	MB65
04A0	MB66
04B0	MB67
04C0	MB68
04D0	MB69
04E0	MB70
04F0	MB71
0500	MB72
0510	MB73
0520	MB74
0530	MB75
0540	MB76
0550	MB77
0560	MB78
0570	MB79
0580	MB80
0590	MB81
05A0	MB82

Table continues on the next page...

Table 285. 8-byte message buffers (continued)

Address offset (hex)	MBDSR = b00 8-byte payload
05B0	MB83
05C0	MB84
05D0	MB85
05E0	MB86
05F0	MB87
0600	MB88
0610	MB89
0620	MB90
0630	MB91
0640	MB92
0650	MB93
0660	MB94
0670	MB95

Table 286. 16-byte message buffers

Address offset (hex)	MBDSR = b01 16-byte payload
0080	MB0
0098	MB1
00B0	MB2
00C8	MB3
00E0	MB4
00F8	MB5
0110	MB6
0128	MB7
0140	MB8
0158	MB9
0170	MB10
0188	MB11
01A0	MB12
01B8	MB13

Table continues on the next page...

Table 286. 16-byte message buffers (continued)

Address offset (hex)	MBDSR = b01 16-byte payload
01D0	MB14
01E8	MB15
0200	MB16
0218	MB17
0230	MB18
0248	MB19
0260	MB20
0280	MB21
0298	MB22
02B0	MB23
02C8	MB24
02E0	MB25
02F8	MB26
0310	MB27
0328	MB28
0340	MB29
0358	MB30
0370	MB31
0388	MB32
03A0	MB33
03B8	MB34
03D0	MB35
03E8	MB36
0400	MB37
0418	MB38
0430	MB39
0448	MB40
0460	MB41
0480	MB42
0498	MB43
04B0	MB44

Table continues on the next page...

Table 286. 16-byte message buffers (continued)

Address offset (hex)	MBDSR = b01 16-byte payload
04C8	MB45
04E0	MB46
04F8	MB47
0510	MB48
0528	MB49
0540	MB50
0558	MB51
0570	MB52
0588	MB53
05A0	MB54
05B8	MB55
05D0	MB56
05E8	MB57
0600	MB58
0618	MB59
0630	MB60
0648	MB61
0660	MB62

Table 287. 32-byte message buffers

Address offset (hex)	MBDSR = b10 32-byte payload
0080	MB0
00A8	MB1
00D0	MB2
00F8	MB3
0120	MB4
0148	MB5
0170	MB6
0198	MB7
01C0	MB8

Table continues on the next page...

Table 287. 32-byte message buffers (continued)

Address offset (hex)	MBDSR = b10 32-byte payload
01E8	MB9
0210	MB10
0238	MB11
0280	MB12
02A8	MB13
02D0	MB14
02F8	MB15
0320	MB16
0348	MB17
0370	MB18
0398	MB19
03C0	MB20
03E8	MB21
0410	MB22
0438	MB23
0480	MB24
04A8	MB25
04D0	MB26
04F8	MB27
0520	MB28
0548	MB29
0570	MB30
0598	MB31
05C0	MB32
05E8	MB33
0610	MB34
0638	MB35

Table 288. 64-byte message buffers

Address offset (hex)	MBDSR = b11 64-byte payload
0080	MB0
00C8	MB1
0110	MB2
0158	MB3
01A0	MB4
01E8	MB5
0230	MB6
0280	MB7
02C8	MB8
0310	MB9
0358	MB10
03A0	MB11
03E8	MB12
0430	MB13
0480	MB14
04C8	MB15
0510	MB16
0558	MB17
05A0	MB18
05E8	MB19
0630	MB20

39.6.6 Legacy RX FIFO structure

When MCR[RFEN] = 1, the memory area 80h–DCh (which is normally occupied by MBs 0–5) is used by the reception Legacy FIFO engine.

The region 80h–8Ch contains the output of the Legacy RX FIFO, which the CPU must read as a message buffer. This output contains the oldest message that has been received but not yet read. The region 90h–DCh is reserved for internal use of the Legacy RX FIFO engine.

An additional memory area, which starts at E0h and may extend up to 2DCh (normally occupied by MBs 6–37) depending on the value of CTRL2[RFFN], contains the ID filter table (configurable from 8 to 128 table elements) that specifies filtering criteria for accepting frames into the Legacy RX FIFO.

Out of reset, the ID filter table flexible memory area defaults to E0h and extends only to FCh, which corresponds to MBs 6 to 7 for RFFN = 0, for backward compatibility with previous versions of FlexCAN.

The following shows the Legacy RX FIFO data structure.

Table 289. Legacy RX FIFO structure

	31 28	24 23	22	21	20	19 18 17 16	15 8	7 0
80h	IDHIT		SRR	IDE	RTR	DLC	TIMESTAMP	
84h		ID standard				ID extended		
88h	Data byte 0		Data byte 1				Data byte 2	Data byte 3
8Ch	Data byte 4		Data byte 5				Data byte 6	Data byte 7
90h–DCh	Reserved							
E0h	ID filter table element 0							
E4h	ID filter table element 1							
E8h–2D4h	ID filter table elements 2 to 125							
2D8h	ID filter table element 126							
2DCh	ID filter table element 127							
		= Unimplemented or reserved						

Each ID filter table element occupies an entire 32-bit word. One, two, or four Identifier Acceptance Filters (IDAFs) can compound each element, depending on MCR[IDAM]. The following tables show the IDAF indexing.

Table 290 shows the three different formats of the ID table elements. All elements of the table must have the same format. See Legacy RX FIFO for more information.

Table 290. ID table structure

Format	31	30	29 24	23 16	15	14	13 8	7 1	0
A	RTR	IDE	RXIDA (standard = 29–19, extended = 29–1)						
B	RTR	IDE	RXIDB_0 (standard = 29–19, extended = 29–16)		RTR	IDE	RXIDB_1 (standard = 13–3, extended = 13–0)		
C	RXIDC_0 (std and ext = 31–24)		RXIDC_1 (std and ext = 23–16)		RXIDC_2 (std and ext = 15–8)		RXIDC_3 (std and ext = 7–0)		
	= Unimplemented or Reserved								

Table 291. Field descriptions

Mnemonic	Field	Description
RTR	Remote Frame	Specifies whether remote frames are accepted into the Legacy FIFO if they match the target ID. 1: Remote frames can be accepted and data frames are rejected.

Table continues on the next page...

Table 291. Field descriptions (continued)

Mnemonic	Field	Description
		0: Remote frames are rejected and data frames can be accepted.
IDE	Extended Frame	Specifies whether extended or standard frames are accepted into the Legacy FIFO if they match the target ID. 1: Extended frames can be accepted and standard frames are rejected. 0: Extended frames are rejected and standard frames can be accepted.
RXIDA	RX Frame Identifier (Format A)	Specifies an ID to be used as acceptance criteria for the Legacy FIFO. In the standard frame format, only the 11 most significant bits (29 to 19) are used for frame identification. In the extended frame format, all bits are used.
RXIDB_0, RXIDB_1	RX Frame Identifier (Format B)	Specifies an ID to be used as acceptance criteria for the Legacy FIFO. In the standard frame format, the 11 most significant bits (a full standard ID) (29 to 19 and 13 to 3) are used for frame identification. In the extended frame format, all 14 bits of the field are compared to the 14 most significant bits of the received ID.
RXIDC_0, RXIDC_1, RXIDC_2, RXIDC_3	RX Frame Identifier (Format C)	Specifies an ID to be used as acceptance criteria for the Legacy FIFO. In both standard and extended frame formats, all 8 bits of the field are compared to the 8 most significant bits of the received ID.
IDHIT	Identifier Acceptance Filter Hit Indicator	Indicates which identifier acceptance filter the received message in the output of the Legacy RX FIFO hit. See Legacy RX FIFO for more information.

39.6.7 Enhanced RX FIFO structure

When ERFCR[ERFEN] = 1, the Enhanced RX FIFO is enabled. The region 2000h–204Ch contains the output of the Enhanced RX FIFO, which the CPU must read as a message buffer. This output contains the oldest message that has been received but not yet read.

Table 292. Enhanced RX FIFO structure

	31	30	29	28 … 24	23	22	21	20	19	18	17	16	15 … 8	7	6 … 0
2000h	EDL	BRS	ESI	Reserved		SRR	IDE	RTR	DLC				TIMESTAMP LEGACY		
2004h	Reserved			ID (standard/extended)									ID (extended)		
2008h	Data byte 0					Data byte 1							Data byte 2		Data byte 3
200Ch	Data byte 4					Data byte 5							Data byte 6		Data byte 7
2010h	Data byte 8					Data byte 9							Data byte 10		Data byte 11
2014h	Data byte 12					Data byte 13							Data byte 14		Data byte 15
2018h	Data byte 16					Data byte 17							Data byte 18		Data byte 19
201Ch	Data byte 20					Data byte 21							Data byte 22		Data byte 23
2020h	Data byte 24					Data byte 25							Data byte 26		Data byte 27
2024h	Data byte 28					Data byte 29							Data byte 30		Data byte 31
2028h	Data byte 32					Data byte 33							Data byte 34		Data byte 35

Table continues on the next page...

Table 292. Enhanced RX FIFO structure (continued)

202Ch	Data byte 36	Data byte 37	Data byte 38	Data byte 39
2030h	Data byte 40	Data byte 41	Data byte 42	Data byte 43
2034h	Data byte 44	Data byte 45	Data byte 46	Data byte 47
2038h	Data byte 48	Data byte 49	Data byte 50	Data byte 51
203Ch	Data byte 52	Data byte 53	Data byte 54	Data byte 55
2040h	Data byte 56	Data byte 57	Data byte 58	Data byte 59
2044h	Data byte 60	Data byte 61	Data byte 62	Data byte 63
IH_OFF	Reserved			ID HIT
TS_OFF	HR TIMESTAMP			
2050h	19 Enhanced FIFO Elements (Reserved)			
...				
263Ch				

NOTE

ID HIT offset and high-resolution timestamp offset change dynamically according to data length code (DLC) as shown in Table 293.

Table 293. ID HIT offset and high-resolution timestamp offset

Data Length Code (DLC)	ID HIT offset (IH_OFF)	High-resolution timestamp offset (TS_OFF)
0	2008h	200Ch
1–4	200Ch	2010h
5–8	2010h	2014h
9	2014h	2018h
10	2018h	201Ch
11	201Ch	2020h
12	2020h	2024h
13	2028h	202Ch
14	2038h	203Ch
15	2048h	204Ch

Table 294. Field descriptions

Mnemonic	Field	Description
EDL	Extended Data Length	Distinguishes between classical CAN format and CAN FD format frames. 0: Classical CAN frame format 1: CAN FD frame format

Table continues on the next page...

Table 294. Field descriptions (continued)

Mnemonic	Field	Description
BRS	Bit Rate Switch	Defines whether the bit rate is switched inside a CAN FD format frame. 0: Bit rate is not switched in a CAN FD frame. 1: Bit rate is switched in a CAN FD frame.
ESI	Error State Indicator	Indicates whether the transmitting node is error-active or error-passive. This field is meaningful only if EDL = 1. 0: Error-active 1: Error-passive
SRR	Substitute Remote Request	Fixed recessive bit, used only in extended format. Transmitting nodes always send it as recessive and receiving nodes can receive it as either recessive or dominant. If FlexCAN receives this bit as dominant, it is interpreted as an arbitration loss.
IDE	ID Extended Bit	Identifies whether the frame format is standard or extended. 0: Standard 1: Extended
RTR	Remote Frame	Identifies whether the current frame is a data frame or a remote request. 0: Data frame 1: Remote request
DLC	Data Length Code	Defines the number of bytes in the data field of a CAN frame (Data byte 0 to Data byte 63). When RTR = 1, the frame is a remote request and does not include the data field, regardless of the DLC field. See Table 282 for more details.
LEGACY TIMESTAMP	16-bit Timestamp	Provides a copy of the Free-Running Timer, captured during the CAN frame. See Table 281 for details about legacy timestamp operation.
ID	Frame Identifier	In base frame format, only the 11 most significant bits are used for frame identification. The 18 least significant bits are ignored. In extended frame format, all bits are used for frame identification.
DATA BYTE 0–63	Data Field	Up to 64 bytes can be stored in the data field.
IDHIT	Identifier Acceptance Filter Hit Indicator	Indicates which Enhanced RX FIFO Filter Element (ERFFEL0 - ERFFEL127) the received message in the output of the Enhanced RX FIFO hit. For each filter region, standard-ID filter space, and extended-ID filter space, there is an independent index starting from zero. Table 295 shows how FlexCAN writes IDHIT according to each filter element.
HR TIMESTAMP	High-resolution Timestamp	32-bit timebase captured during the CAN frame. When CTRL2[TSTAMPCAP] is not zero, a 32-bit timebase is captured from a dedicated on-chip timer which operates in free-running mode. See CTRL2[TSTAMPCAP] for details about capture point configuration of the high-resolution timestamp.

Table 295. IDHIT for Enhanced RX FIFO

Enhanced RX FIFO filter element - ERFFEL	IDHIT value	Filter element type
ERFFEL0	0	Extended-ID
ERFFEL1	1	Extended-ID
. . .	. . .	Extended-ID
ERFFEL(m-1)	m-1	Extended-ID
ERFFEL(m)	0	Standard-ID
ERFFEL(m+1)	1	Standard-ID
. . .	. . .	Standard-ID
ERFFEL(2n-m+1)	2x(n-m)+1	Standard-ID

NOTE

Where m = NEXIF and n = NFE. If NEXIF = 0, only standard-ID filter elements exist. If NEXIF > NFE, only extended-ID filter elements exist.

39.7 Glossary

Active message buffer — A message buffer is active if it can participate in the current matching or arbitration process.

Bus interface unit — FlexCAN submodule responsible for the interface to the CPU.

Bus off — See CAN Specification.[4]

CAN — Controller Area Network, a serial communication protocol defined in CAN Specification[4] and ISO International Standard.[5]

CHI — Controller-host interface, a FlexCAN submodule responsible for message buffer matching and arbitration algorithms.

CRC — Cyclic redundancy check

Dominant bit — A dominant bit wins the arbitration on the CAN bus. It is transmitted as 0.

Doze mode — A system low-power mode where the CPU bus remains active and a global Doze mode request is sent to all peripherals asking them to enter low power mode.

Error active — See CAN Specification.[4]

Error frame — See CAN Specification.[4]

Error passive — See CAN Specification.[4]

[4] Controller Area Network - CAN Specification Version 2.0 Part A, Part B, Robert Bosch GmbH, 1991.
[5] ISO International Standard - ISO 11898 First Edition 1993 Road Vehicles - Interchange of Digital Information - Controller Area Network (CAN) for high-speed Communication.

Form error	See CAN Specification.[4]
Hard reset	A reset coming from an external pin and/or following power-on. It resets everything.
Idle	See CAN Specification.[4]
Information processing time	See CAN Specification.[4]
Intermission	See CAN Specification.[4]
Matching elements	Data used in the matching process, such as ID, filter, mask, etc.
MB	See Message buffer.
Message buffer	Internal FlexCAN data structure containing bytes received from, or to be transmitted to, the CAN line, as well as information about this data.
Overload frame	See CAN Specification.[4]
Phase buffer segment	See CAN Specification.[4]
Recessive bit	A recessive bit loses the arbitration on the CAN bus. It is transmitted as 1.
Sclock	Serial clock. Obtained by dividing the clock feeding the CAN engine (oscillator or bus clock) by a prescaler factor. The Sclock period defines the time quantum for CAN protocol timing.
Soft reset	Global reset typically used by peripherals to reinitialize some of its registers, but not all of them.
Stop mode	A system low-power mode in which all chip clocks are stopped for maximum power savings.
Stuffing error	See CAN Specification.[4]
Time quantum	This is equal to the Sclock period. It is the minimum time period used to compose the CAN protocol bit timing.

Chapter 40
Flexible I/O (FlexIO)

40.1 Chip-specific FLEXIO information

Table 296. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

This chip includes two instances of FLEXIO.

FLEXIO_TRIGGER_IN[3:0] are tied to 0.

FLEXIO_TRIGGER_OUT[7:0] are timer trigger outputs. They are routed to ANAMIX TRG_MUX (for controlling ADC triggering). See TRIGMUX for more details.

Table 297. ADC Trigger Inputs - FLEXIO

13	FLEXIO1	Trigger from FlexIO1
14	FLEXIO2	Trigger from FlexIO2

40.2 Overview

FLEXIO is a highly configurable module that provides:

- Emulation of various serial or parallel communication protocols.
- Flexible 16-bit timers with support for various trigger, reset, enable, and disable conditions.
- Programmable logic blocks that allow the implementation of digital logic functions on-chip and configurable interaction of internal and external modules.
- Programmable state machine for offloading basic system control functions from the CPU.

40.2.1 Block diagram

The following diagram provides a high-level overview of the FLEXIO timer and shifter configuration.

FLEXIO uses shifters, timers, and external triggers to shift data into or out of FLEXIO. As shown in the block diagram, timers control the timing of this data shift. You can configure the timers to use generic timer functions, external triggers, or various other conditions to determine the control.

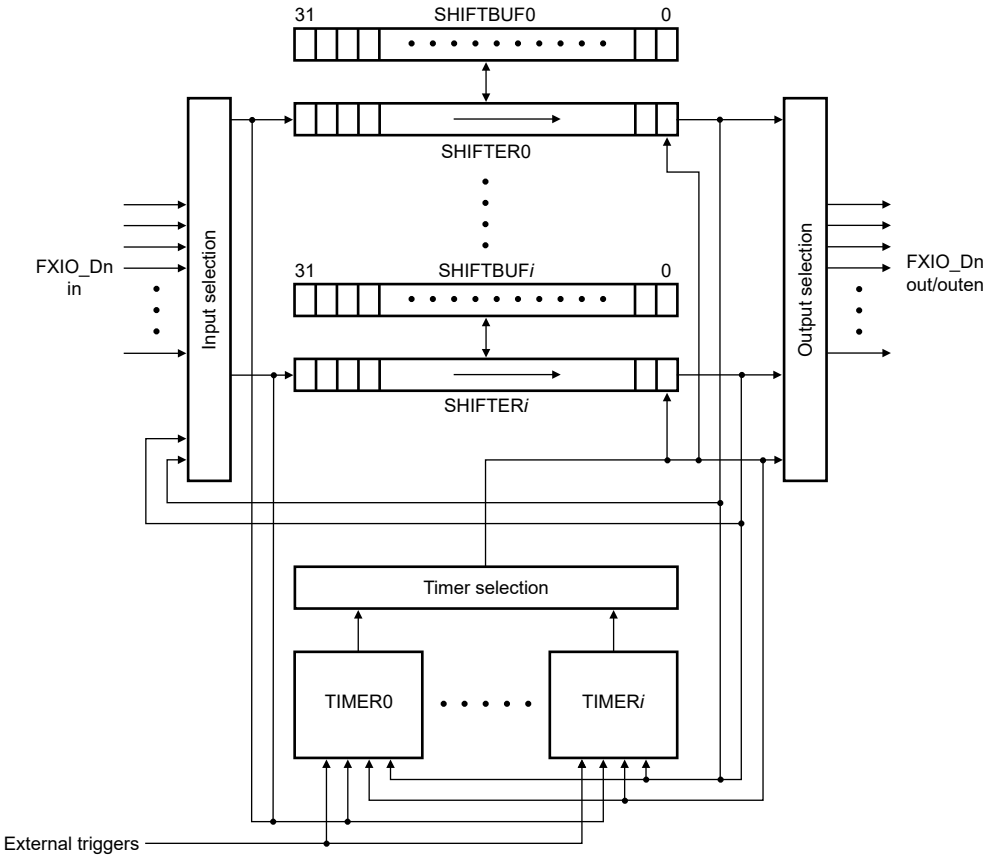


Figure 160. Block diagram

40.2.2 Features

- Array of 32-bit shift registers with transmit, receive, data match, logic, and state modes:
  - Double-buffered shifter operation for continuous data transfer
  - Shifter concatenation to support large transfer sizes
  - Automatic start and stop bit generation
  - 1, 2, 4, 8, 16, or 32 multi-bit shift widths for parallel interface support
  - Interrupt, DMA, or polled transmit and receive operation
- Highly flexible 16-bit timers with support for various internal or external triggers, reset, enable, and disable conditions:
  - Programmable baud rates independent of bus clock frequency, with support for asynchronous operation during Stop mode
  - Programmable logic mode for integrating external digital logic functions on-chip, or combining pin, shifter, or timer functions to generate complex outputs
  - Programmable state machine for offloading basic system control functions from CPU, with support for up to eight states, eight outputs, and three selectable inputs per state
- Integrated general-purpose I/O registers and pin rising or falling edge interrupts to simplify software support
- Support for a wide range of protocols, including but not limited to:
  - UART

— I2C

— SPI

— I2S

— Camera IF

— Motorola 68K or Intel 8080 bus

— PWM or waveform generation

— Input-capture (pulse-edge interval measurement), such as SENT

40.3 Functional description

40.3.1 Shifter operation

Shifters are responsible for buffering and shifting data into or out of FLEXIO. The timer assigned to the shifter controls the timing of shift, load, and store events via SHIFTCTL*n*[TIMSEL]. Shifters are designed to support either DMA, interrupt, or polled operations. The following figure provides a detailed view of the shifter microarchitecture.

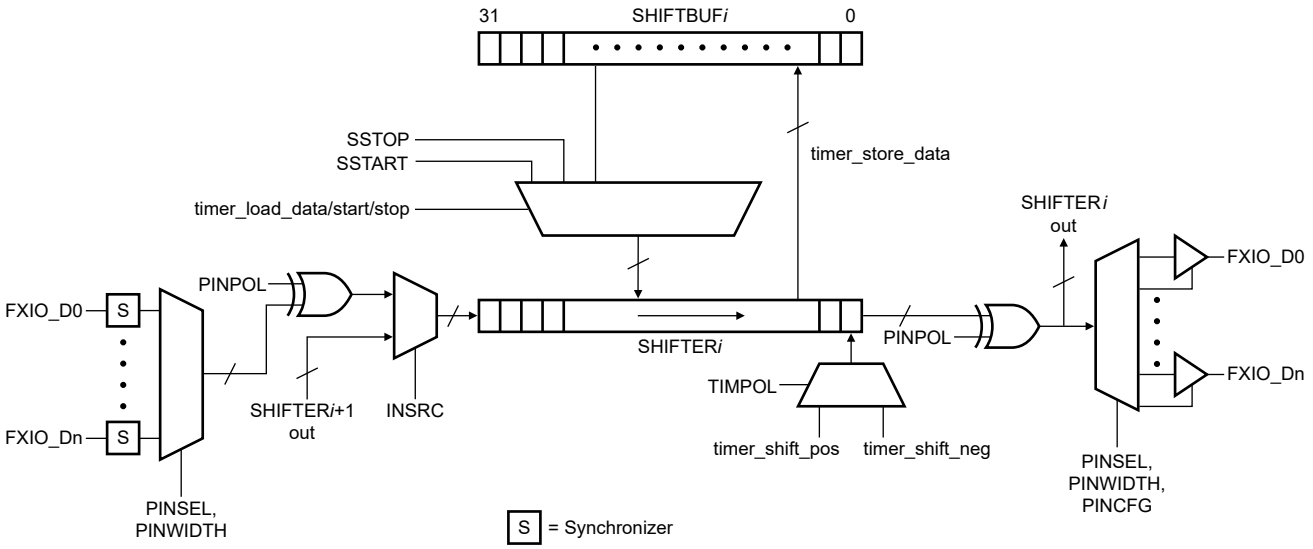


Figure 161. Shifter microarchitecture

40.3.1.1 Transmit mode

In Transmit mode (SHIFTCTL*n*[SMOD] = 010b), the shifter loads data from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) and shifts data out when the assigned timer signals a load event. An optional start and stop bit can be automatically loaded before or after SHIFTBUF register data by configuring either SHIFTCFG[SSTART] and TIMCFG[TSTART], or SHIFTCFG[SSTOP] and TIMCFG[TSTOP] in the shifter and timer.

NOTE

If a stop bit is enabled, the shifter immediately loads a stop bit when it is initially configured for Transmit mode.

The shifter status flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when data has either been loaded from the SHIFTBUF register into the shifter or when the shifter is initially configured for Transmit mode. To clear the flag, write 1 or write new data to SHIFTBUF. In Transmit mode, write any value to the SHIFTBUF register to clear the corresponding shifter status flag, which is cleared regardless of what is writing to the register (DMA or interrupt), or the state of the DMA or interrupt enables. See the functional description of SHIFTSTAT[SSF] for information on how the flag is set and cleared for each mode.

The shifter error flag (SHIFTERR[SEF]) and any enabled interrupts are set when an attempt to load data from an empty SHIFTBUF register occurs (buffer underrun). Clear the flag by writing 1.

40.3.1.2 Receive mode

When the assigned timer signals a store event in Receive mode (SHIFTCTL*n*[SMOD] = 001b), the shifter shifts and stores data in Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). You can check for a start and stop bit before or after the shifter data is sampled by configuring either SHIFTCFG[SSTART] and TIMCFG[TSTART], or SHIFTCFG[SSTOP] and TIMCFG[TSTOP] in the shifter and timer.

The shifter status flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when data is stored in the SHIFTBUF register from the shifter. To clear the flag, write 1 to or read the data from SHIFTBUF. Any read of the SHIFTBUF register clears the corresponding shifter status flag when the shifter is in Receive mode. The flag is cleared regardless of what is reading the register (DMA or interrupt) or the state of the DMA or interrupt enables. See the functional description of SHIFTSTAT[SSF] for information on how the flag is set or cleared for each mode.

The shifter error flag (SHIFTERR[SEF]) and any enabled interrupts are set either when an attempt to store data into a full SHIFTBUF register occurs (buffer overrun) or when a mismatch occurs on a start or stop bit check. Write 1 to clear the flag.

40.3.1.3 Match Store mode

In Match Store mode (SHIFTCTL*n*[SMOD] = 100b), the shifter shifts data in, checks for a match result, and stores matched data in Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) when the assigned timer signals a store event. By configuring either SHIFTCFG[SSTART], TIMCFG[TSTART], and SHIFTCFG[SSTOP], or TIMCFG[TSTOP] in the shifter and timer, you can check for a start and stop bit before or after the shifter data is sampled. You can compare up to 16 bits of data using SHIFTBUF[31:16] to configure the data to be matched and SHIFTBUF[15:0] to mask the match result.

The shifter status flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when a match occurs and the matched data is stored in the SHIFTBUF register from the shifter. To clear the flag, read the matched data from the SHIFTBUF register or write 1 to the flag. Any read of the SHIFTBUF register clears the corresponding shifter status flag when the shifter is configured in Match Store mode. The flag is cleared regardless of what is reading the register (DMA or interrupt) or the state of the DMA or interrupt enables. See the functional description for SHIFTSTAT[SSF] to know how the flag is set or cleared for each mode.

The shifter error flag (SHIFTERR[SEF]) and any enabled interrupts are set when an attempt to store matched data into a full SHIFTBUF register occurs (buffer overrun), or when a mismatch occurs on a start or stop bit check. Write 1 to clear the flag.

40.3.1.4 Match Continuous mode

In Match Continuous mode (SHIFTCTL*n*[SMOD] = 101b), the shifter shifts data in and continuously checks for a match result whenever a shift event is signaled by the assigned timer. You can compare up to 16 bits of data using SHIFTBUF[31:16] to configure the data to be matched and SHIFTBUF[15:0] to mask the match result.

The shifter status flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when a match occurs. The flag clears automatically as soon as no match exists between the shifter data and Shifter Buffer (SHIFTBUF0 - SHIFTBUF7).

You cannot clear the flag by reading the SHIFTBUF register.

The shifter error flag (SHIFTERR[SEF]) and any enabled interrupts are set when a match occurs. To clear the flag, write 1 or perform a read from the SHIFTBUF register.

40.3.1.5 State mode

State mode enables you to implement any state machine with up to eight states, eight outputs, and three selectable inputs per state. This feature allows basic control functions to be offloaded from the CPU, which can remain in a low-power mode.

In State mode (SHIFTCTL*n*[SMOD] = 110b), when the shifter is selected by the current state pointer (SHIFTSTATE[STATE]), use the SHIFTBUF register to drive the output and compute the next state values. The following figure provides a detailed view of the shifter microarchitecture when configured for State mode.

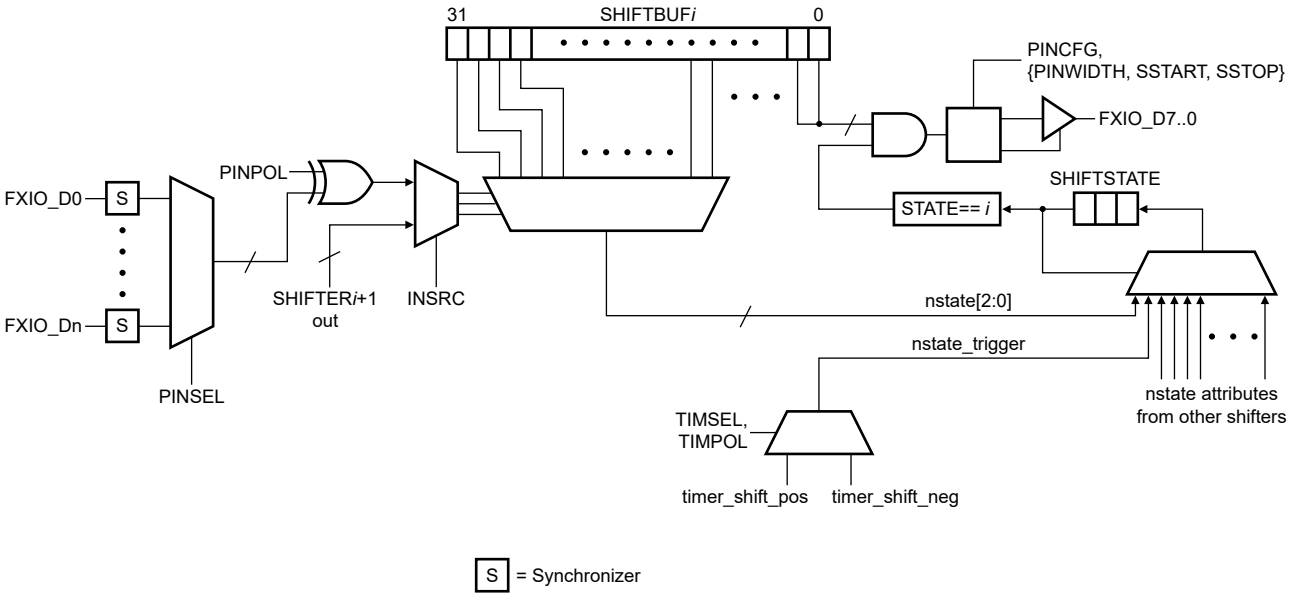


Figure 162. State microarchitecture

When the current state pointer selects a specific shifter (shifter *n*), output pins FXIO_D[7:0] are driven by SHIFTBUF*n*[31:24]; the configuration is defined by SHIFTCTL*n*[PINCFG]. Write 1 to Shifter Configuration (SHIFTCFG0 - SHIFTCFG7) {PWIDTH[3:0],SSTOP[1:0],SSTART[1:0]} to disable the output drive on pins FXIO_D[7:0] for state machine applications that require less than eight output pins.

Use the three input pins selected by SHIFTCTL*n*[PINSEL] and SHIFTBUF*n*[23:0] to compute the next state value.

NOTE

Each state can use a different set of three input pins.

The following table shows how the next state value is computed when the current state pointer is pointing to shifter *n*.

Table 298. Next state computation for SHIFTSTATE[STATE] = *n*

FXIO_D[PINSEL + 2]	FXIO_D[PINSEL + 1]	FXIO_D[PINSEL]	Next state value
0	0	0	SHIFTBUF*n*[2:0]
0	0	1	SHIFTBUF*n*[5:3]
0	1	0	SHIFTBUF*n*[8:6]
0	1	1	SHIFTBUF*n*[11:9]
...	...	...	...
1	1	1	SHIFTBUF*n*[23:21]

NOTE

You can configure other shifters and timers to drive the input pins of a given state, allowing you to create complex combinations of shifters and timers as needed. For example, the output of a shifter configured for Logic mode can be used to drive a state machine input.

The next state transition is triggered using the timer output selected by SHIFTCTL*n*[TIMSEL], with polarity controlled by SHIFTCTL*n*[TIMPOL].

NOTE

Each state can use a different timer to trigger each next state transition, allowing various internal or external trigger sources and clocking configurations to be used. See Timer section for more information.

The current state pointer defaults to shifter 0 at reset; however, you can write to select a different shifter for the initial state. If the current state pointer selects a shifter that is not configured for State mode, then outputs are not driven and the next state transition is never triggered.

The shifter status flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when the shifter is selected by the current state pointer. The flag is cleared when the current state pointer is updated to a different shifter.

40.3.1.6 Logic mode

Logic mode enables you to implement a small amount of programmable digital logic within a FLEXIO shifter.

In Logic mode (SHIFTCTL*n*[SMOD] = 111b), use Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) to implement a 5-input, 32-bit programmable logic lookup table. The following figure provides a detailed view of shifter microarchitecture when configured for Logic mode.

Use the SHIFTBUF register to configure the lookup table for the four pin inputs. You can also use SHIFTER*n* to configure a feedback or delayed pin source as the fifth input to the lookup table.

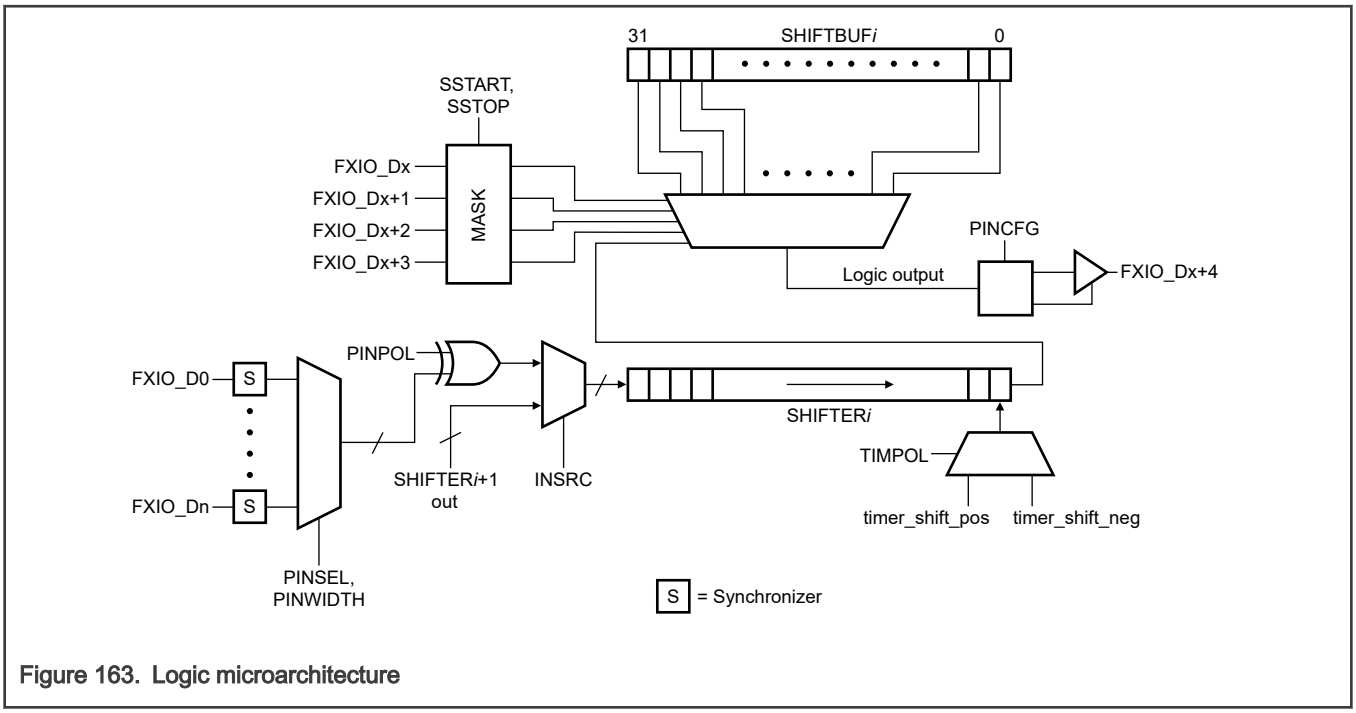


Figure 163. Logic microarchitecture

The lookup table is driven using four pin inputs (maskable using SHIFTCFG[SSTOP] and SHIFTCFG[SSTART]), plus one input from the internal shifter. It can be configured to drive an output pin using SHIFTCTL[PINCFG]. Pin inputs and outputs are fixed for each logic lookup table and are not selectable. The following table lists the logic output value selected by the lookup table for shifter *n*.

Table 299. Logic lookup table for shifter *n*

SHIFTER*n*[0]	FXIO_D[*x* + 3][1]	FXIO_D[*x* + 2]	FXIO_D[*x* + 1]	FXIO_D[*x*]	Logic output to FXIO_D[*x* + 4]
0	0	0	0	0	SHIFTBUF*n*[0]
0	0	0	0	1	SHIFTBUF*n*[1]

Table continues on the next page...

Table 299. Logic lookup table for shifter *n* (continued)

SHIFTER*n*[0]	FXIO_D[*x* + 3][1]	FXIO_D[*x* + 2]	FXIO_D[*x* + 1]	FXIO_D[*x*]	Logic output to FXIO_D[*x* + 4]
0	0	0	1	0	SHIFTBUF*n*[2]
0	0	0	1	1	SHIFTBUF*n*[3]
...	...	...	...	...	...
1	1	1	1	1	SHIFTBUF*n*[31]

1. for shifters *n* = 0...3, *x* = *n*

   for shifter *n* = 4...7, *x* = *n* + 4

To minimize output glitches, use SHIFTCFG*n*[SSTOP] and SHIFTCFG*n*[SSTART] to mask unused input pins. When these fields are 1, {SSTOP[1:0] and SSTART[1:0]} mask FXIO_D[*x* + 3]...FXIO_D[*x*] inputs respectively so that any transitions on these pins do not cause the logic output to glitch.

NOTE

You can configure other shifters and timers to drive the input pins of a given lookup table, allowing you to concatenate lookup tables or create complex combinations of shifters and timers as needed.

SHIFTCFG[PWIDTH] controls the number of delay stages introduced by the internal shifter input (SHIFTER*n*[0]). For example, when configured for a 1-bit shift (SHIFTCFG[PWIDTH] = 0), the internal shifter introduces a 32-shift clock delay before passing its input (selected by SHIFTCTL[PINSEL]) to the lookup table. When configured for a 32-bit shift (SHIFTCFG[PWIDTH] = 16...31), the internal shifter introduces a 1-shift clock delay to its input.

The shifter status flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set whenever the output pin allocated to the logic lookup table has a value of 1 after being synchronized with the FLEXIO clock. The flag clears when the output pin has a value of 0. This also allows SHIFTSTAT[SSF] to be used as a trigger to a timer if needed.

The shifter error flag (SHIFTERR[SEF]) and any enabled interrupts are set when the output pin allocated to the logic lookup table is asserted. Clear the flag by writing 1 to it.

The Logic mode input pins, including pins driven by other shifters and timers, are synchronized with the FLEXIO functional clock before they are input to the programmable logic lookup table.

40.3.2 Timer operation

The FLEXIO 16-bit timers control the loading, shifting, and storing of the shift registers. The counters load the contents of the compare register and decrement down to zero on the FLEXIO clock. The counters can perform generic timer functions such as generating a clock, select output, or a PWM waveform. You can configure these timers to perform any of the following functions:

- Enable in response to a trigger, pin, or shifter condition.
- Decrement always or only on a trigger or pin edge.
- Reset in response to a trigger or pin condition.
- Disable on a trigger or pin condition or on a timer compare.

Timers can optionally include a start condition and a stop condition.

Although each timer operates independently, you can configure a timer to enable or disable at the same time as the previous timer (for example, timer 1 can enable or disable at the same time as timer 0) and a timer output can be used to trigger any other timer. The trigger used by each timer is configured independently as a timer output, shifter status flag, pin input, or an external trigger input. The trigger configuration is separate from pin configuration; you can perform it to configure input, output data, or output enable. See the chip-specific FLEXIO information for information on external trigger connections.

You must configure Timer Configuration (TIMCFG0 - TIMCFG7) before writing 1 to TIMCTL*n*[TIMOD].

40.3.2.1 Timer 8-bit Baud Counter mode

In 8-bit Baud Counter mode, the 16-bit counter is divided into two 8-bit counters. The lower 8 bits are used to configure the baud rate of the shift clock and the upper 8 bits are used to configure the number of shift clock edges in the transfer. When the lower 8 bits decrement to zero, the timer output is toggled and the lower 8 bits reload from the compare register. The upper 8 bits decrement when the lower 8 bits become zero and decrement.

NOTE

A timer reset event in 8-bit Baud Counter mode only resets the lower 8-bit counter. The upper 8-bit counter is not affected and can decrement if the timer reset is configured to update the state of the timer output, which toggles as a result of the timer reset event.

A timer compare event occurs when the upper 8 bits equal zero and decrement. The timer status flag is set on a timer compare event.

40.3.2.2 Timer 8-bit High PWM mode

In 8-bit High PWM mode, the 16-bit counter is divided into two 8-bit counters. The lower 8 bits are used to configure the timer output high period and the upper 8 bits are used to configure the timer output low period. The lower 8 bits decrement when the output is high. When the lower 8 bits become zero and decrement, the timer output is cleared and the lower 8 bits are reloaded from the compare register. The upper 8 bits decrement when the output is low. When the upper 8 bits become zero and decrement, the timer output is set and the upper 8 bits are reloaded from the compare register.

A timer compare event occurs when the upper 8 bits become zero and decrement. The timer status flag is set on a timer compare event.

40.3.2.3 Timer 16-bit Counter mode

In 16-bit Counter mode, you can use the 16-bit counter to configure either the baud rate of the shift clock (for example, TIMDEC[1:0] ≠ 10 or 11) or the number of shift clock edges in the transfer (for example, TIMDEC[1:0] = 10 or 11). When the 16-bit counter equals zero and decrements, the timer output toggles and the counter reloads from the compare register.

A timer compare event occurs when the 16-bit counter equals zero and decrements. The timer status flag is set on a timer compare event.

40.3.2.4 Timer 16-bit Counter Disable mode

In 16-bit Counter Disable mode, the 16-bit counter can be used to configure either the baud rate of the shift clock (for example, TIMDEC[1:0] ≠ 10 or 11) or the number of shift clock edges in the transfer (for example, TIMDEC[1:0] = 10 or 11). When the 16-bit counter equals zero and decrements, the timer output toggles and the counter reloads from the compare register.

A timer compare event occurs when the 16-bit counter equals zero and decrements. The timer status flag is set on a timer disable event.

40.3.2.5 Timer 8-bit Word Counter mode

In 8-bit Word Counter mode, the 16-bit counter is divided into two 8-bit counters. The lower 8 bits are used to configure the number of shift clock edges in each word and the upper 8 bits are used to configure the number of words in the transfer. When the lower 8 bits decrement to zero, the timer output is toggled and the lower 8 bits reload from the compare register. The upper 8 bits only decrement when the lower 8 bits become zero and decrement.

A timer compare event occurs when the lower 8 bits become zero and decrement. The timer status flag is set when the upper 8 bits become zero and decrement.

40.3.2.6 Timer 8-bit Low PWM mode

In 8-bit Low PWM mode, the 16-bit counter is divided into two 8-bit counters. The lower 8 bits are used to configure the timer output low period and the upper 8 bits are used to configure the timer output high period. The lower 8 bits decrement when the output is low. When the lower 8 bits become zero and decrement, the timer output is set and the lower 8 bits are reloaded from the compare

register. The upper 8 bits decrement when the output is high. When the upper 8 bits become zero and decrement, the timer output is cleared and the upper 8 bits are reloaded from the compare register.

A timer compare event occurs when the upper 8 bits become zero and decrement. The timer status flag is set on a timer compare event.

40.3.2.7 Timer enable and start functions

The following events occur when you configure TIMCTL*n*[TIMOD] for the desired mode and the condition configured by the timer enable (TIMCFG*n*[TIMENA]) is detected. When TIMCTL*n*[ONETIM] is 1, the timer status flag must be clear to generate a timer enable event; otherwise, the timer enable event is blocked. You can use this to enforce software intervention after each timer iteration:

- The timer counter loads the current value of the compare register and starts decrementing, as configured by TIMCFG*n*[TIMDEC].
- The timer output may update to its initial state depending on the configuration of TIMCFG*n*[TIMOUT]. Shifters that are controlled by this timer do not see this as a rising edge on the timer shift clock.
- Transmit shifters controlled by this timer either output their start bit value or load the shift register from the shift buffer and output the first bit, as configured by SHIFTCFG*n*[SSTART].

If the timer start bit is enabled, the timer counter reloads with the compare register on the first rising edge of the shift clock after the timer starts decrementing. If there is no falling edge on the shift clock before the first rising edge (for example, when TIMCFG*n*[TIMOUT] = 1), a shifter that is configured to shift on the falling edge and load on the first shift does not load correctly.

40.3.2.8 Timer decrement and reset functions

The timer generates the timer output and timer shift clock depending on the fields, TIMCTL*n*[TIMOD] and TIMCFG*n*[TIMDEC]. The shifter clock is either equal to the timer output (when TIMCFG*n*[TIMDEC] ≠ 10 or 11) or equal to the decrement clock (when TIMCFG*n*[TIMDEC] = 10 or 11). If you configure TIMCFG*n*[TIMDEC] to decrement from a pin or trigger, the timer decrements on both rising and falling edges.

If a timer is configured to decrement on the FLEXIO functional clock divided by 16 or 256 (when TIMCFG*n*[TIMDEC] = 100 or 101), then a common prescaler that is shared by all timers is used to generate the two divide ratios. This prescaler is reset when all timers are either idle or configured not to use the prescaler (TIMCFG*n*[TIMDEC] ≠ 100 or 101).

If you configure the timer to reset as determined by TIMCFG*n*[TIMRST], then the timer counter loads the current value of the compare register again. You can configure the timer output and timer shift clock to update on timer reset, as configured by TIMCFG*n*[TIMOUT]. If the time output toggles as a result of the timer reset, this can result in a timer shift clock edge. In 8-bit Baud Counter mode, this also decrements the upper 8 bits of the counter.

In general, when the timer counter decrements to zero, a timer compare event is triggered. The timer compare event causes:

- The timer counter to load the contents of the timer compare register.
- The timer output to toggle.
- Any configured transmit shift registers to load.
- Any configured receive shift registers to store.

Depending on the timer mode, the timer status flag may also be set.

40.3.2.9 Timer disable and stop functions

When the timer is configured to add a stop bit on each compare, the following additional events occur:

- Transmit shifters controlled by this timer output their stop bit value (if configured by SHIFTCFG*n*[SSTOP]).
- Receive shifters controlled by this timer store the contents of the shift register in their shift buffer, as configured by SHIFTCFG*n*[SSTOP].

- The timer counter reloads the current value of the compare register on the first rising edge of the shifter clock after the compare.

If you configure the timer to insert a stop bit on each compare, you must configure the transmit shifters to load on the first shift.

When the condition configured by timer disable (TIMCFG*n*[TIMDIS]) is detected, the following events occur:

- Timer counter reloads the current value of the compare register and starts decrementing as configured by TIMCFG*n*[TIMDEC].
- Timer output clears. Shifters that are controlled by this timer do not see this as a falling edge on the timer shift clock, but can generate a shift event if the timer shift clock otherwise generates one.
- Transmit shifters controlled by this timer output their stop bit value (if configured by SHIFTCFG*n*[SSTOP]).
- Receive shifters controlled by this timer store the contents of the shift register in their shift buffer, as configured by SHIFTCFG*n*[SSTOP].

If the timer stop bit is enabled, the timer counter continues decrementing until the next rising edge of the shift clock is detected, at which point it finishes decrementing. Although the timer output is forced low during the stop bit, the timer shift clock can toggle during the stop bit. The timer output does not generate shift events during the stop bit.

A timer enable condition can be detected in the same cycle as a timer disable condition (if timer stop bit is disabled), or on the first rising edge of the shift clock after the disable condition (if stop bit is enabled). When TIMCTL*n*[ONETIM] is 1, the timer status flag must be clear before the next timer enable condition is detected. When the timer is in the stop state condition, receive shift registers with stop bit enabled store the contents of the shift register into the shift buffer and verify the state of the input data on the configured shift edge. If there is no configured edge between the timer disable and the next rising edge of the shift clock, then the final store and verify do not occur.

40.3.3 Pin operation

The pin configuration for each timer and shifter can be set to use any FLEXIO pin with either polarity. You can configure each timer and shifter as an input, output data, output enable, or bidirectional output. A pin configured for output enable can be used as an open drain (with inverted polarity because the output enable assertion causes logic zero to be output on the pin) or to control the enable on the bidirectional output. You can configure any timer or shifter to control the output enable for a pin where the bidirectional output data is driven by another timer or shifter.

When more than one shifter or timer is configured to use the same pin for output data, then the output data to the pin is combined using an OR function. This also applies if more than one timer or shifter is configured to use the same pin for output enable.

40.3.3.1 Parallel interface

You can configure shifters to use multiple FLEXIO pins simultaneously by using SHIFTCFG*n*[PWIDTH], which is used to configure the following settings of a shifter:

1. Number of bits shifted per shift clock.
2. Number of pins driven by the shifter per shift clock (only on shifters supporting parallel transmit—that is, SHIFTER0 and SHIFTER4.)
3. Number of pins sampled by the shifter per shift clock (only on shifters supporting parallel receive—that is, SHIFTER3 and SHIFTER7.)

When configured for parallel shift, either 4, 8, 16, or 32 bits can be shifted on every shift clock. If an adjacent shifter is selected as the input source (SHIFTCFG*n*[INSRC] = 1), the least significant 4, 8, 16, or 32 bits from the adjacent shifter are sampled on each shift clock.

For shifters supporting parallel receive (SHIFTER3, SHIFTER7), you can configure the shifter to sample multiple pins (INSRC = 0), with PWIDTH and PINSEL selecting the pins as FXIO_D[PINSEL+PWIDTH]:FXIO_D[PINSEL].

NOTE

If PWIDTH is less than the number of bits being shifted on each shift clock, then the most significant bits are masked with 0. For example, if PINSEL = 7 and PWIDTH = 6, then SHIFTER[31:24] samples {0,0,FXIO_D[12:7]} on each shift clock.

For shifters supporting parallel transmit (SHIFTER0, SHIFTER4), you can configure the shifter to drive multiple pins using SHIFTCTL*n*[PINCFG], with PWIDTH and PINSEL selecting the pins as follows: FXIO_D[PINSEL+PWIDTH]:FXIO_D[PINSEL].

NOTE

If PWIDTH is less than the number of bits being shifted on each shift clock, then the most significant pins are not driven. For example, if PINSEL = 7 and PWIDTH = 6, then SHIFTER[5:0] drives only FXIO_D[12:7] on each shift clock.

40.3.3.2 Pin synchronization

When you configure a pin as an input (this includes a timer trigger configured as a pin input), the input signal is first synchronized with the FLEXIO clock before a timer or shifter could use the signal. This introduces a small latency of 0.5–1.5 FLEXIO clock cycles when using an external pin input to generate an output or control a shifter. This sets the maximum setup time at 1.5 FLEXIO clock cycles.

If an input is used by more than one timer or shifter, then the synchronization occurs once to ensure any edge is seen on the same cycle by all timers and shifters using that input.

NOTE

FLEXIO pins are also connected internally. Configuring a FLEXIO shifter or timer to output data on an unused pin makes an internal connection that allows other shifters and timers to use this pin as an input. This allows a shifter output to trigger a timer or a timer output to be shifted into a shifter. This path is also synchronized with the FLEXIO clock and therefore incurs a one-cycle latency.

When using a pin input as a timer trigger, timer clock, or shifter data input, the following synchronization delays occur:

- 0.5–1.5 FLEXIO clock cycles for an external pin
- One FLEXIO clock cycle for an internally driven pin

See Application information for timing considerations such as output valid time and input setup time for specific applications (SPI controller, SPI target, I2C controller, I2S controller, and I2S target).

40.3.3.3 Pin override

You can change the state of any FLEXIO pin at any time. Pin Output Enable (PINOUTE) configures any pin as an output and drives that pin with the value in Pin Output Data (PINOUTD).

Alias registers for PINOUTE and data registers also exist. Writing a logic 1 to an alias register updates the corresponding register fields in both PINOUTE and PINOUTD as follows:

- Pin Output Disable (PINOUTDIS) clears Pin Output Enable (PINOUTE) and Pin Output Data (PINOUTD).
- Pin Output Clear (PINOUTCLR) sets Pin Output Enable (PINOUTE) and clears Pin Output Data (PINOUTD).
- Pin Output Set (PINOUTSET) sets Pin Output Enable (PINOUTE) and Pin Output Data (PINOUTD).
- Pin Output Toggle (PINOUTTOG) sets Pin Output Enable (PINOUTE) and toggles Pin Output Data (PINOUTD).

40.3.3.4 Pin interrupt

You can read the state of any FLEXIO pin at any time and also configure any pin to set a status flag when either a rising or falling edge is detected on that pin. Additionally, you can configure the pin status flag to generate an interrupt.

40.3.4 Low-power modes

FLEXIO remains functional during low-power modes, if CTRL[DOZEN] is 0 and the FLEXIO functional clock remains enabled.

The exception to this is in LLS mode. In this case, CTRL[DOZEN] is ignored and FLEXIO waits for all timers to complete any pending operation before acknowledging LLS mode entry.

40.3.5 Debug mode

FLEXIO remains functional in Debug mode, provided the value of CTRL[DBGE] is 1.

40.3.6 Clocking

Table 300. FLEXIO clocks

Clock	Description
Functional clock	Is asynchronous to the bus clock and can remain enabled in low-power modes. You must enable the FLEXIO functional clock before accessing any of the FLEXIO registers. Provided the FLEXIO functional clock is at least equal to the bus clock, you can configure CTRL[FASTACC] to support fast register accesses.
Bus clock	Is used only for bus accesses to the control and configuration registers.

40.3.7 Reset

Table 301. FLEXIO reset types

Reset	Description
Chip reset	Resets the FLEXIO logic and registers to their default states on chip reset.
Software reset	Resets, using CTRL[SWRST], all logic and registers to their default states, except for the Control register.

40.3.8 Interrupts and DMA requests

The following table shows the status flags that generate the FLEXIO interrupt and DMA requests.

Table 302. FLEXIO interrupts and DMA requests

Flag	Description	Interrupt	DMA request	Low-power wake-up
SHIFTSTAT[SSF]	Shifter status flag	Y	Y	Y
SHIFTERR[SEF]	Shifter error flag	Y	N	Y
TIMSTAT[TSF]	Timer status flag	Y	Y	Y
PINSTAT[PSF]	Pin status flag	Y	N	Y
TRGSTAT[ETSF]	External trigger status flag	Y	N	Y

40.3.9 Peripheral triggers

The connection between FLEXIO peripheral triggers and other peripherals is device-specific.

40.3.9.1 Output triggers

Each FLEXIO timer generates an output trigger equal to the timer output. The output trigger is not affected by the timer pin polarity configuration.

40.3.9.2 Input trigger

FLEXIO supports multiple external trigger inputs that can be used to trigger one or more FLEXIO timers. If a rising edge is detected on an external trigger when FLEXIO is enabled, then the external trigger status flag is set. The external triggers are synchronized to the FLEXIO functional clock and must assert for at least two cycles of the FLEXIO functional clock to be sampled correctly.

40.4 External signals

Table 303. External signals

Signal	Description	Direction
FXIO_D*n* (*n* = 0...31)	Bidirectional FLEXIO shifter and timer pin	Input or output

40.5 Initialization

Perform the following procedure to initialize FLEXIO registers:

1. Enable FLEXIO by writing 1 to CTRL[FLEXEN].
2. Configure shift registers for the given application. It is recommended to write to Shifter Configuration (SHIFTCFG0 - SHIFTCFG7) before writing to the corresponding register, Shifter Control (SHIFTCTL0 - SHIFTCTL7).
3. Configure timer registers for the given application. It is recommended to write to Timer Compare (TIMCMP0 - TIMCMP7) and Timer Configuration (TIMCFG0 - TIMCFG7) before writing to the corresponding register, Timer Control (TIMCTL0 - TIMCTL7).
4. Enable interrupts and/or DMA requests, as appropriate, for the given application.
5. Write transmit data to initiate a transfer (depending on the given application).

40.6 Application information

This section provides examples for a variety of FLEXIO module applications. See FLEXIO register descriptions for more information.

40.6.1 UART transmit

UART transmit can be supported using one timer, one shifter, and one pin (two pins, if supporting CTS). The start and stop bit insertion is handled automatically, and multiple transfers are supported using the DMA controller. The timer status flag is used to indicate when the stop bit of each word is transmitted.

Break and idle characters require software intervention. Before transmitting a break or idle character, you must modify SHIFTCFG*n*[SSTART] and SHIFTCFG*n*[SSTOP] to transmit the required state, and the data to transmit must equal FFh or 00h. Supporting a second stop bit requires the stop bit to be inserted into the data stream using software (and increasing the number of bits to transmit). When performing byte writes to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) (or Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) for transmitting MSB first), the rest of the register remains unaltered. This allows an address mark bit or additional stop bit to remain undisturbed.

NOTE

FLEXIO does not support automatic insertion of parity bits.

Table 304. UART transmit configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0032h	Configure start bit of 0 and stop bit of 1.

Table continues on the next page...

Table 304. UART transmit configuration (continued)

Register	Value	Configuration
SHIFTCTL*n*	0003_0002h	Configure transmit using timer 0 on the positive edge of clock with output data on pin 0. You can configure the PINPOL field to invert output data, or support open-drain by writing 1h to the PINPOL and PINCFG fields.
TIMCMP*n*	0000_0F01h	Configure 8-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of bits × 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0000_2222h	Configure start bit, stop bit, enable on trigger asserted and disable on compare. You can support CTS by configuring the TIMENA field as 3h.
TIMCTL*n*	01C0_0001h	Configure the dual 8-bit counter using the shifter 0 status flag as an inverted internal trigger source. To support CTS, configure the PINSEL (for pin 1) and PINPOL fields as 1h.
SHIFTBUF*n*	Data to transmit	Transmit data can be written to SHIFTBUF[7:0] to initiate an 8-bit transfer. Use the shifter status flag to indicate when data can be written using an interrupt or a DMA request. Write to SHIFTBUFBBS[7:0] instead to support MSB first transfer.

The following table shows an alternative configuration that supports slower baud rates. This configuration requires two timers.

Table 305. UART transmit configuration for slow baud rate

Register	Value	Configuration
SHIFTCFG*n*	0000_0032h	Configure start bit of 0 and stop bit of 1.
SHIFTCTL*n*	0003_0002h	Configure transmit using timer 0 on the positive edge of clock with output data on pin 0. Invert output data by writing 1 to the PINPOL field. Support open-drain by configuring the PINPOL and PINCFG fields as 1h.
TIMCMP*n*	0000_000Fh	Configure for 8-bit transfer, and configure TIMCMP[15:0] as (number of bits × 2) - 1.
TIMCFG*n*	0030_2622h	Configure start bit, stop bit, enable on trigger rising edge, decrement on trigger and disable on compare.

Table continues on the next page...

Table 305. UART transmit configuration for slow baud rate (continued)

Register	Value	Configuration
TIMCTL*n*	0740_0003h	Configure the 16-bit counter using the timer 1 output as an internal trigger source.
TIMCMP(*n* + 1)	0000_0001h	Configure baud rate of divide by 4 of the FLEXIO clock, and configure TIMCMP[15:0] as (baud rate divider ÷ 2) - 1.
TIMCFG(*n* + 1)	0000_1200h	Configure enable on trigger asserted and disable on timer 0 disable. You can configure the TIMEN field as 3h to support CTS.
TIMCTL(*n* + 1)	01C0_0003h	Configure the 16-bit counter using the shifter 0 status flag as an inverted internal trigger source. You can support CTS by configuring the PINSEL (for pin 1) and PINPOL fields as 1h.
SHIFTBUF*n*	Data to transmit	Transmit data can be written to SHIFTBUF[7:0] to initiate an 8-bit transfer. Use the shifter status flag to indicate when data can be written using an interrupt or a DMA request. Write to SHIFTBUFBBS[7:0] instead to support MSB first transfer.

40.6.2 UART receive

UART receive can be supported using one timer, one shifter, and one pin (two timers and two pins, if supporting RTS). The start and stop bit verification is handled automatically and multiple transfers are supported using the DMA controller. The timer status flag is used to indicate when the stop bit of each word is received.

FLEXIO does not support triple voting of the received data, which is sampled only once in the middle of each bit. You can use a timer to implement a glitch filter on the incoming data and a different timer to detect an idle line of programmable length. Break characters cause the error flag to set, and the shifter buffer register returns 00h.

NOTE

FLEXIO does not support automatic verification of parity bits.

Table 306. UART receiver configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0032h	Configure start bit of 0 and stop bit of 1.
SHIFTCTL*n*	0080_0001h	Configure receive using timer 0 on the negative edge of clock with input data on pin 0. You can invert input data by writing 1 to the PINPOL field.
TIMCMP*n*	0000_0F01h	Configure 8-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set

Table continues on the next page...

Table 306. UART receiver configuration (continued)

Register	Value	Configuration
		TIMCMP[15:8] as (number of bits × 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0204_2422h	Configure start bit, stop bit, enable on pin positive edge and disable on compare. Enable resynchronization to received data with TIMOUT = 2h and TIMRST = 4h.
TIMCTL*n*	0000_0081h	Configure the dual 8-bit counter using the inverted pin 0 input.
SHIFTBUF*n*	Data to receive	You can read received data from SHIFTBUFBYS[7:0]. Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from SHIFTBUFBIS[7:0] instead to support MSB first transfer.

The UART receiver with RTS configuration uses a second timer to generate the RTS output. RTS asserts when the start bit is detected and negates when the data is read from the shifter buffer register. If no start bit is detected when the RTS is asserted, the received data is ignored.

Table 307. UART receiver with RTS configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0032h	Configure start bit of 0 and stop bit of 1.
SHIFTCTL*n*	0080_0001h	Configure receive using timer 0 on the negative edge of clock with input data on pin 0. Invert input data by writing 1 to the PINPOL field.
TIMCMP*n*	0000_0F01h	Configure 8-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of bits × 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0204_2522h	Configure start bit, stop bit, enable on pin positive edge with trigger asserted and disable on compare. Enable resynchronization to received data with TIMOUT = 2h and TIMRST = 4h.
TIMCTL*n*	02C0_0081h	Configure dual 8-bit counter using the inverted pin 0 input. Trigger is internal using the inverted pin 1 input.
TIMCMP(*n* + 1)	0000_FFFFh	Never compare.

Table continues on the next page...

Table 307. UART receiver with RTS configuration (continued)

Register	Value	Configuration
TIMCFG(*n* + 1)	0030_6100h	Enable on timer *n* enable and disable on the trigger falling edge. Decrement on trigger to ensure no compare.
TIMCTL(*n* + 1)	0143_0003h	Configure 16-bit counter and output on pin 1. Trigger is internal using the shifter 0 flag.
SHIFTBUF*n*	Data to receive	You can read received data using SHIFTBUFBYS[7:0]. Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from SHIFTBUFBIS[7:0] instead to support MSB first transfer.

40.6.3 SPI controller

SPI Controller mode can be supported using two timers, two shifters, and four pins. Using the DMA controller, either CPHA = 0 or CPHA = 1 and transfers can be supported. For CPHA = 1, the chip select can remain asserted for multiple transfers and the timer status flag can be used to indicate the end of the transfer.

The stop bit is used to guarantee a minimum of one clock cycle between the target chip select negating and before the next transfer. To initiate each transfer, either the core or DMA writes to the transmit buffer.

NOTE

Because of synchronization delays, the setup time for the serial input data is 1.5 FLEXIO clock cycles. This means the maximum baud rate is divide by 4 of the FLEXIO clock frequency.

Table 308. SPI controller (CPHA = 0) configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0000h	Start and stop bit disabled.
SHIFTCTL*n*	0083_0002h	Configure transmit using timer 0 on the negative edge of clock with output data on pin 0.
SHIFTCFG(*n* + 1)	0000_0000h	Start and stop bit disabled.
SHIFTCTL(*n* + 1)	0000_0101h	Configure receive using timer 0 on the positive edge of clock with input data on pin 1.
TIMCMP*n*	0000_3F01h	Configure 32-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of bits × 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0100_2222h	Configure start bit, stop bit, enable on trigger high and disable on compare; initial clock state is logic 0.

Table continues on the next page...

Table 308. SPI controller (CPHA = 0) configuration (continued)

Register	Value	Configuration
TIMCTL*n*	01C3_0201h	Configure dual 8-bit counter using the pin 2 output (shift clock), with shifter 0 flag as the inverted trigger. Write 1 to the PINPOL field to invert the output shift clock.
TIMCMP(*n* + 1)	0000_FFFFh	Never compare.
TIMCFG(*n* + 1)	0000_1100h	Enable when timer 0 is enabled and disable when timer 0 is disabled.
TIMCTL(*n* + 1)	0003_0383h	Configure 16-bit counter (never compare) using the inverted pin 3 output as target select.
SHIFTBUF*n*	Data to transmit	You can write transmit data to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be written using interrupt or DMA request. Write to Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.

Table 309. SPI controller (CPHA = 1) configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0021h	Start bit loads data on first shift.
SHIFTCTL*n*	0003_0002h	Configure transmit using timer 0 on the positive edge of clock with output data on pin 0.
SHIFTCFG(*n* + 1)	0000_0000h	Start and stop bit disabled.
SHIFTCTL(*n* + 1)	0080_0101h	Configure receive using timer 0 on the negative edge of clock with input data on pin 1.
TIMCMP*n*	0000_3F01h	Configure 32-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of bits x 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.

Table continues on the next page...

Table 309. SPI controller (CPHA = 1) configuration (continued)

Register	Value	Configuration
TIMCFG*n*	0100_2222h	Configure start bit, stop bit, enable on trigger high and disable on compare; initial clock state is logic 0.
TIMCTL*n*	01C3_0201h	Configure dual 8-bit counter using pin 2 output (shift clock), with the shifter 0 flag as the inverted trigger. Write 1 to the PINPOL field to invert the output shift clock, and set the TIMDIS field as 3 to keep target select asserted for as long as there is data in the transmit buffer.
TIMCMP(*n* + 1)	0000_FFFFh	Never compare.
TIMCFG(*n* + 1)	0000_1100h	Enable when timer 0 is enabled and disable when timer 0 is disabled.
TIMCTL(*n* + 1)	0003_0383h	Configure 16-bit counter (never compare) using inverted pin 3 output (as target select).
SHIFTBUF*n*	Data to transmit	Transmit data can be written to SHIFTBUF. Use the shifter status flag to indicate when data can be written using interrupt or DMA request. Write to Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.

40.6.4 SPI target

SPI Target mode can be supported using one timer, two shifters, and four pins. Either CPHA = 0 or CPHA = 1 can be supported and transfers can be supported using the DMA controller. For CPHA = 1, the select can remain asserted for multiple transfers and the timer status flag can be used to indicate the end of the transfer.

You must write the transmit data to the transmit buffer register before the external target select asserts; otherwise, the shifter error flag is set.

NOTE

Because of synchronization delays, the output valid time for the serial output data is 2.5 FLEXIO clock cycles. This means the maximum baud rate is divide by 6 of the FLEXIO clock frequency.

Table 310. SPI target (CPHA = 0) configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0000h	Start and stop bit disabled.
SHIFTCTL*n*	0083_0002h	Configure transmit using timer 0 on the falling edge of shift clock with output data on pin 0.
SHIFTCFG(*n* + 1)	0000_0000h	Start and stop bit disabled.
SHIFTCTL(*n* + 1)	0000_0101h	Configure receive using timer 0 on the rising edge of shift clock with input data on pin 1.
TIMCMP*n*	0000_003Fh	Configure 32-bit transfer. Set TIMCMP[15:0] as (number of bits × 2) - 1.
TIMCFG*n*	0120_0600h	Configure enable on trigger rising edge. Initial clock state is logic 0 and decrements on pin input.
TIMCTL*n*	06C0_0203h	Configure 16-bit counter using pin 2 input (shift clock), with pin 3 input (target select) as the inverted trigger.
SHIFTBUF*n*	Data to transmit	Transmit data can be written to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be written using interrupt or DMA request. Write to Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.

Table 311. SPI target (CPHA = 1) configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0001h	Shifter configured to load on first shift and stop bit disabled.
SHIFTCTL*n*	0003_0002h	Configure transmit using timer 0 on rising edge of shift clock with output data on pin 0.
SHIFTCFG(*n* + 1)	0000_0000h	Start and stop bit disabled.

Table continues on the next page...

Table 311. SPI target (CPHA = 1) configuration (continued)

Register	Value	Configuration
SHIFTCTL(*n* + 1)	0080_0101h	Configure receive using timer 0 on falling edge of shift clock with input data on pin 1.
TIMCMP*n*	0000_003Fh	Configure 32-bit transfer. Set TIMCMP[15:0] as (number of bits × 2) - 1).
TIMCFG*n*	0120_6602h	Configure start bit, enable on trigger rising edge, disable on trigger falling edge. Initial clock state is logic 0 and decrements on pin input.
TIMCTL*n*	06C0_0203h	Configure 16-bit counter using pin 2 input (shift clock), with pin 3 input (target select) as the inverted trigger.
SHIFTBUF*n*	Data to transmit	Transmit data can be written to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be written using interrupt or DMA request. Write to Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7) instead to support MSB first transfer.

40.6.5 I2C controller

I2C Controller mode can be supported using two timers, two shifters, and two pins. One timer is used to generate the SCL output and another one is used to control the shifters. The two shifters that are used to transmit and receive for every word, when receiving the transmitter, must transmit FFh to 3-state the output. FLEXIO inserts a stop bit after every word to generate and verify the ACK or NACK. FLEXIO waits for the first write to the transmit data buffer before enabling SCL generation. Data transfers can be supported using the DMA controller and the shifter error flag sets on transmit underrun or receive overflow.

The first timer generates the bit clock for the entire packet (start to repeated start or stop condition), so you must program the compare register with the total number of clock edges in the packet (minus one). The timer supports clock stretching using the reset counter when pin is equal to output. However, this increases both the clock high and clock low periods by at least one FLEXIO clock cycle each. The second timer uses the SCL input pin to control the transmit and receive shift registers. This enforces an SDA data hold time by an extra two FLEXIO clock cycles.

Both the transmit and receive shifters must be serviced for each word in the transfer. The transmit shifter must transmit FFh when receiving, and the receive shifter returns the data present on the SDA pin. The transmit shifter loads one additional word on the last falling edge of the SCL pin. When generating a stop condition or a repeated start condition, this word must be 00h and FFh, respectively. During the last word of a controller-receiver transfer, you must set the transmit SHIFTCFG*n*[SSTOP] field to generate a NACK.

The receive shift register asserts an error interrupt if a NACK is detected, but you are responsible for generating the stop or repeated start condition. If a NACK is detected during controller-transmit, the interrupt routine must immediately write 00h (when generating a stop condition) or FFh (when generating a repeated start condition) to the transmit shifter register. You must wait for the next rising edge on SCL before disabling both timers. The transmit shifter must be disabled after the setup delay for a repeated start or stop condition.

NOTE

Because of synchronization delays, the data valid time for the transmit output is two FLEXIO clock cycles. This means the maximum baud rate is divide by 6 of the FLEXIO clock frequency.

To guarantee SDA hold time, the I2C controller data valid is delayed by two cycles because the clock output is passed through a synchronizer before clocking the transmit or receive shifter. Because the SCL output is synchronous with FLEXIO clock, the synchronization delay is one cycle, and then an additional cycle is involved to generate the output.

Table 312. I2C controller configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0032h	Start bit enabled (logic 0) and stop bit enabled (logic 1).
SHIFTCTL*n*	0101_0082h	Configure transmit using timer 1 on the rising edge of clock with inverted output enable (open-drain output) on pin 0.
SHIFTCFG(*n* + 1)	0000_0020h	Start bit disabled and stop bit enabled (logic 0) for ACK or NACK detection.
SHIFTCTL(*n* + 1)	0180_0001h	Configure receive using timer 1 on the falling edge of clock with input data on pin 0.
TIMCMP*n*	0000_2501h	Configure 2 word transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of words × 18) + 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0102_2222h	Configure start bit, stop bit, enable on trigger high, disable on compare, reset if output equals pin. Initial clock state is logic 0 and is not affected by reset.
TIMCTL*n*	01C1_0101h	Configure dual 8-bit counter using pin 1 output enable (SCL open-drain), with the shifter 0 flag as the inverted trigger.
TIMCMP(*n* + 1)	0000_000Fh	Configure 8-bit transfer. Set TIMCMP[15:0] as (number of bits x 2) - 1.
TIMCFG(*n* + 1)	0020_1112h	Enable when timer 0 is enabled; disable when timer 0 is disabled. Enable start bit and stop bit at the end of each word and decrement on pin input.
TIMCTL(*n* + 1)	01C0_0183h	Configure 16-bit counter using inverted pin 1 input (SCL).

Table continues on the next page...

Table 312. I2C controller configuration (continued)

Register	Value	Configuration
SHIFTBUF*n*	Data to transmit	Transmit data can be written to SHIFTBUFBBS[7:0]. Use the shifter status flag to indicate when data can be written using interrupt or DMA request.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from SHIFTBUFBIS[7:0]. Use the shifter status flag to indicate when data can be read using interrupt or DMA request.

40.6.6 I2S controller

I2S Controller mode can be supported using two timers, two shifters, and four pins. One timer is used to generate the bit clock and control the shifters and another timer is used to generate the frame sync. FLEXIO waits for the first write to the transmit data buffer before enabling bit clock and frame sync generation. Data transfers are supported using the DMA controller and the shifter error flag sets on transmit underrun or receive overflow.

The bit clock frequency is an even integer divide of the FLEXIO clock frequency. The initial frame sync assertion occurs at the same time as the first bit clock edge. The timer uses the start bit to ensure that the frame sync is generated one clock cycle before the first output data.

NOTE

Because of synchronization delays, the setup time for the receiver input is 1.5 FLEXIO clock cycles. This means that the maximum baud rate is divide by 4 of the FLEXIO clock frequency.

Table 313. I2S controller configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0001h	Load transmit data on first shift and stop bit disabled.
SHIFTCTL*n*	0003_0002h	Configure transmit using timer 0 on the rising edge of clock with output data on pin 0.
SHIFTCFG(*n* + 1)	0000_0000h	Start and stop bit disabled.
SHIFTCTL(*n* + 1)	0080_0101h	Configure receive using timer 0 on the falling edge of clock with input data on pin 1.
TIMCMP*n*	0000_3F01h	Configure 32-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of bits × 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG*n*	0000_0202h	Configure start bit, enable on trigger high and never disable. Initial clock state is logic 1.
TIMCTL*n*	01C3_0281h	Configure dual 8-bit counter using inverted pin 2 output (bit clock), with

Table continues on the next page...

Table 313. I2S controller configuration (continued)

Register	Value	Configuration
		shifter 0 flag as the inverted trigger. Write 0 to the PINPOL field to invert the polarity of the output shift clock.
TIMCMP(*n* + 1)	0000_007Fh	Configure 32-bit transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:0] as (number of bits × baud rate divider) ÷ 1.
TIMCFG(*n* + 1)	0000_0100h	Enable when timer 0 is enabled and never disable.
TIMCTL(*n* + 1)	0003_0383h	Configure 16-bit counter using inverted pin 3 output (as frame sync). Write 0 to the PINPOL field to invert the polarity of the output frame sync.
SHIFTBUF*n*	Data to transmit	Transmit data can be written to Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7). Use the shifter status flag to indicate when data can be written using interrupt or DMA request. Write to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) instead to support LSB first transfer.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7). Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) instead to support LSB first transfer.

40.6.7 I2S target

I2S Target mode can be supported using three timers, two shifters, and four pins. For single transmit and single receive, other combinations of transmit and receive are possible.

The transmit data must be written to the transmit buffer register before the external frame sync asserts, otherwise the shifter error flag is set.

NOTE

Because of synchronization delays, the output valid time for the serial output data is 2.5 FLEXIO clock cycles. This means the maximum baud rate is divide by 6 of the FLEXIO clock frequency.

The output valid time of I2S target is maximum 2.5 cycles because there is a maximum 1.5 cycle delay on the clock synchronization, plus one cycle to output the data.

Timer 2 detects the falling edge of frame sync (start of new frame) and asserts output until the rising edge of bit clock (when the frame sync is normally sampled). Timer 0 detects the rising edge of bit clock with timer 2 output asserted and asserts output for length of frame. Timer 1 detects the falling edge of bit clock with timer 0 output asserted and controls shift registers for 32-bit transfers.

Table 314. I2S target configuration

Register	Value	Configuration
SHIFTCFG*n*	0000_0000h	Start and stop bit disabled.
SHIFTCTL*n*	0103_0002h	Configure transmit using timer 1 on the rising edge of shift clock with output data on pin 0.
SHIFTCFG(*n* + 1)	0000_0000h	Start and stop bit disabled.
SHIFTCTL(*n* + 1)	0180_0101h	Configure receive using timer 1 on the falling edge of shift clock with input data on pin 1.
TIMCMP*n*	0000_007Fh	Configure two 32-bit transfers per frame. Set TIMCMP[15:0] as (number of bits × 4) - 1.
TIMCFG*n*	0020_2500h	Configure enable on pin rising edge (inverted bit clock) with trigger high (timer 2) and disable on compare. Initial clock state is logic 1 and decrements on pin input (bit clock).
TIMCTL*n*	0B40_0203h	Configure 16-bit counter using pin 2 input (bit clock), with timer 2 output as the trigger.
TIMCMP(*n* + 1)	0000_003Fh	Configure 32-bit transfers. Set TIMCMP[15:0] as (number of bits × 2) - 1.
TIMCFG(*n* + 1)	0020_2500h	Configure enable on pin (bit clock) rising edge with trigger (timer 0) high and disable on compare. Initial clock state is logic 1 and decrement on pin input (bit clock).
TIMCTL(*n* + 1)	0340_0283h	Configure 16-bit counter using inverted pin 2 input (bit clock), with timer 0 output as the trigger.
TIMCMP(*n* + 2)	0000_0000h	Compare on zero (first edge).
TIMCFG(*n* + 2)	0020_6400h	Configure enable on inverted pin (frame sync) rising edge and disable on trigger falling edge (bit clock). Initial clock state is logic 1 and decrement on inverted pin input (frame sync).
TIMCTL(*n* + 2)	04C0_0383h	Configure 16-bit counter using inverted pin 3 input (frame sync), with pin 2 inverted input (bit clock) as the trigger.
SHIFTBUF*n*	Data to transmit	Transmit data can be written to Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7). Use the shifter status flag to indicate when data can be written

Table continues on the next page...

Table 314. I2S target configuration (continued)

Register	Value	Configuration
		using interrupt or DMA request. Write to the SHIFTBUF register instead to support LSB first transfer.
SHIFTBUF(*n* + 1)	Data to receive	Received data can be read from Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7). Use the shifter status flag to indicate when data can be read using interrupt or DMA request. Read from the SHIFTBUF register instead to support LSB first transfer.

40.6.8 SENT receiver

The SENT receiver can be supported by using one timer and one pin. The timer is configured as Input-Capture mode, and captures the counter value at the falling edge of the pin input. After the counter value is captured, the counter is automatically restarted from counter value 0. Therefore, the captured value always indicates the period between the previous falling edge and current falling edge. You can configure the CPU interrupt or DMA trigger at each capture of the counter. The CPU software performs the entire SENT frame decoding with the latest tick width adjustment.

Table 315. SENT receiver configuration

Register	Value	Configuration
TIMCMP*n*	—	Stores counter value at the falling edge of the pin.
TIMCFG*n*	0000_6000h	Timer is always enabled. It is disabled on trigger falling edge, decrement counter on FLEXIO clock.
TIMCTL*n*	0040_0007h	Single 16-bit input capture mode. The PINSEL field selects the timer pin input and output. Timer pin output disabled, internal trigger selected, select pin 0 as trigger.

40.6.9 Camera interface

Camera interface can be supported using one timer, one or more shifters, and multiple pins. Multiple transfers can be supported using the DMA controller.

The example configuration shown in the following table describes the FLEXIO configuration for interfacing with an 8-bit CMOS sensor with PCLK, VSYNC, HREF, and D[7:0] outputs. The example uses a 128-bit buffer to capture 16 pixels of image data before an interrupt or DMA transfer. You can use a bigger or smaller buffer depending on system DMA performance and FLEXIO resource usage by other applications.

NOTE

You can use additional timers to track the number of pixels per row and number of rows per frame, or HREF or VSYNC can be assigned as GPIO interrupts for software tracking.

Table 316. Camera interface configuration for 8-bit CMOS sensor

Register	Value	Configuration
SHIFTCFG*n*...*n*+2[1]	0007_0100h	Configure 8-bit parallel shift in from adjacent shifter.
SHIFTCFG*n*+3	0007_0000h	Configure 8-bit parallel shift in from pins FXIO_D[7:0] (D[7:0]).
SHIFTCTL*n*...*n*+3	0080_0001h	Configure receive using timer 0 on the negative edge of clock.
TIMCMP*n*	0000_001Fh	Configure 16 pixel (8 bits/pixel × 16 pixels = 128 bits) transfer. Set TIMCMP[15:0] as (number of pixels × 2) - 1.
TIMCFG*n*	0120_6600h	Configure enable on trigger (HREF) rising edge and disable on trigger falling edge. Initial shift clock state is logic 0 and decrement on PCLK input.
TIMCTL*n*	12C0_0803h	Configure 16-bit counter using FXIO_D[8] input (PCLK), with FXIO_D[9] input (HREF) as the inverted trigger.
SHIFTBUF*n*...*n*+3	Data to receive	Received data can be read from SHIFTBUF*n*...*n*+3. Use the shifter status flag to indicate when data can be read using interrupt or DMA request.

1. *n* = 0 or 4

40.6.10 Motorola 68K and Intel 8080 bus interface

Motorola 68K and Intel 8080 bus are two parallel interfaces commonly used by smart and asynchronous LCD controllers. With GPIO, FLEXIO can drive these interfaces using one timer and one shifter. Additional shifters can be used to support large transfers via the DMA controller.

Example configurations are provided for 1-beat and 16-beat transfers. Other lengths are also supported.

The following table provides an example of how to drive a 16-bit 68K or 8080 bus. For an 8080 bus, two GPIOs are used to drive the nCS and RS pins. For a 68K bus, an additional GPIO is required to drive the RDWR pin.

Table 317. Motorola 68K and Intel 8080 write configuration

Register	Value	Configuration
SHIFTCFG0...7	000F_0100h	Configure 16-bit parallel shift in from adjacent shifter.
SHIFTCTL0	0003_0002h	Configure transmit using timer 0 on the positive edge of clock with data output to FXIO_D[15:0].
SHIFTCTL1...7	0000_0002h	Configure transmit using timer 0 on the positive edge of clock.

Table continues on the next page...

Table 317. Motorola 68K and Intel 8080 write configuration (continued)

Register	Value	Configuration
TIMCMP0	0000_0101h (1 beat) 0000_1F01h (16 beats)	Configure 1 or 16-beat transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] as (number of beats × 2) - 1, and set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG0	0000_2200h	Configure enable on trigger high and disable on compare. Timer output high on enable.
TIMCTL0	01C3_1001h (Motorola 68K, 1 beat) 1DC3_1001h (Motorola 68K, 16 beats) 01C3_1081h (Intel 8080, 1 beat) 1DC3_1081h (Intel 8080, 16 beats)	Configure dual 8-bit counter using FXIO_D[16] output (EN pin for 68K, nWR pin for 8080), with shifter 0 (1-beat) or shifter 7 (16 beats) flag as the inverted trigger.
SHIFTBUF0...7	Data to transmit	Transmit data can be written to SHIFTBUF0 (1 beat) or SHIFTBUF0...7 (16-beats) to initiate a transfer; use the shifter status flag to indicate when data can be written using interrupt or DMA request.

Table 318. Motorola 68K and Intel 8080 read configuration

Register	Value	Configuration
SHIFTCFG0...6	000F_0100h	Configure 16-bit parallel shift in from the adjacent shifter.
SHIFTCFG7	000F_0000h	Configure 16-bit parallel shift in from pin.
SHIFTCTL0...7	0080_0001h	Configure receive using timer 0 on the negative edge of clock with data input from FXIO_D[15:0].
TIMCMP0	0000_0101h (1 beat) 0000_1F01h (16 beats)	Configure 1 or 16-beat transfer with baud rate of divide by 4 of the FLEXIO clock. Set TIMCMP[15:8] = (number of beats × 2) - 1, and set TIMCMP[7:0] = (baud rate divider ÷ 2) - 1.
TIMCFG0	0000_2220h	Configure stop_bit. Enable on trigger high and disable on compare. Timer output high on enable.
TIMCTL0	1DC3_1001h (Motorola 68K, 1 beat) 01C3_1001h (Motorola 68K, 16 beats) 1DC3_1181h (Intel 8080, 1 beat) 01C3_1181h (Intel 8080, 16 beats)	Configure dual 8-bit counter using either FXIO_D[16] output (EN pin for 68K) or FXIO_D[17] output (nRD pin for 8080), with shifter 7 flag (1-beat) or shifter 0 flag (16 beats) as the inverted trigger.

Table continues on the next page...

Table 318. Motorola 68K and Intel 8080 read configuration (continued)

Register	Value	Configuration
SHIFTBUF0...7	Data received	Received data can be read from SHIFTBUF0 (1 beat) or SHIFTBUF0...7 (16 beats). Use the shifter status flag to indicate when data can be read using interrupt or DMA request.

In general, any operation to a 68K or 8080 bus target begins with a register write cycle followed by one or more data read or write cycles. To accomplish this, perform the following procedure:

1. Configure FLEXIO with 1-beat write configuration.
2. Configure GPIO to assert the nCS and RS pins (and deassert the RDWR pin for 68K).
3. Write register index data to SHIFTBUF0[15:0].
4. Configure GPIO to deassert the RS pin (and assert the RDWR pin for 68K data read).
5. Configure FLEXIO with the desired read or write configuration (1 or 16 beats).
6. Use the shifter status flag to trigger interrupt or DMA driven data transfers to and from Shifter Buffer (SHIFTBUF0 - SHIFTBUF7).
7. Configure GPIO to deassert the nCS pin.

40.6.11 Low-power state machine

Table 319 shows an example of a hypothetical state machine to illustrate the flexibility allowed in Shifter State mode.

In this example, FLEXIO waits for the FXIO_D[2] pin to assert and then drives a complementary square wave output at a frequency of FLEXIO_CLK/131072 on the FXIO_D[1:0] pins while the comparator output is asserted. This assumes that the comparator is connected to external trigger 15. See the chip-specific FLEXIO information for actual FLEXIO trigger mappings. Throughout this operation, the CPU can be kept in a stop or VLPS mode by writing 0 to CTRL[DOZEN] and ensuring that FLEXIO_CLK is enabled. Figure 164 shows the states and transitions implemented by this example.

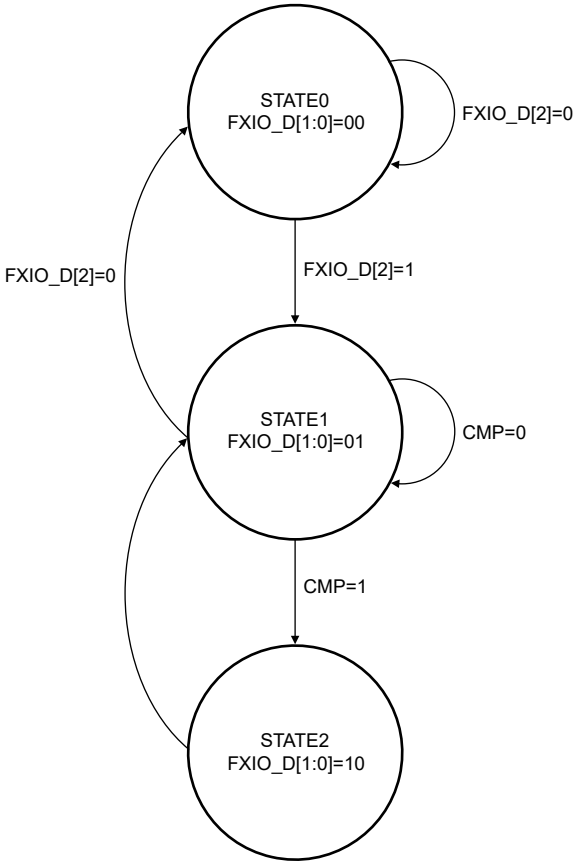


Figure 164. State diagram

Table 319. State machine configuration

Register	Value	Configuration
SHIFTCFG0...2	0000_0003h	Enable FXIO_D[1:0] as outputs.
SHIFTCTL0	0080_0206h	Configure for State mode using FXIO_D[4:2] as inputs to select next state and timer 0 output low to trigger next state.
SHIFTBUF0	0020_8208h	State0: Drive FXIO_D[1:0] = 00; transition to state0 if FXIO_D[2] = 0, state1 if FXIO_D[2] = 1.
SHIFTCTL1	0000_0206h	Configure for State mode using FXIO_D[4:2] as inputs to select next state and timer 0 output high to trigger next state.
SHIFTBUF1	0140_8408h	State1: Drive FXIO_D[1:0]=01; transition to state0 if FXIO_D[2]=0, state1 if CMP = 0, state2 if CMP = 1 (FXIO_D[3]=1).

Table continues on the next page...

Table 319. State machine configuration (continued)

Register	Value	Configuration
SHIFTCTL2	0080_0206h	Configure for State mode using FXIO_D[4:2] as inputs to select next state and timer 0 output low to trigger next state.
SHIFTBUF2	0224_9249h	State2: Drive FXIO_D[1:0] = 10, transition to state1 with timer 0 output low.
TIMCMP0	0000_FFFFh	Configure baud rate of divide by 131072 of the FLEXIO clock. Set TIMCMP[7:0] as (baud rate divider ÷ 2) - 1.
TIMCFG0	0000_0000h	Configure timer always enabled.
TIMCTL0	0000_0003h	Configure single 16-bit counter.
TIMCFG1	0010_7600h	Configure timer enabled on trigger rising edge, disabled on trigger falling edge, decrement on trigger edge.
TIMCTL1	0F03_0303h	Configure timer output enabled on FXIO_D[3], external trigger 15 (comparator output) selected.

40.6.12 Keypad interface

The keypad interface can support a 3 × 4 keypad matrix using three timers and three shifters, although a larger matrix can be supported using additional shifters. The configuration is designed for four columns configured as active low open-drain pins and three rows configured as input pins with pull-up resistors enabled.

One shifter is configured in Logic mode to assert its output when any of the row inputs are low, indicating a key is pressed. Use a timer to filter the shifter output to ensure that the key is pressed for a minimum amount of time before performing the column scan.

A different shifter is configured for parallel transmit. Use this shifter to scan each column when a keypress is detected. When not scanning, the shifter output is configured to assert all active low open-drain column outputs to detect any keypress. Use a dedicated timer to control the transmit shifter.

The last shifter is configured for parallel receive. Use this shifter to capture the result of the column scanning so that you can decode which key (or keys) was pressed. This configuration captures the state of both row and column pins for each scan, although the row state can also be deduced by the shift order. Use a dedicated timer to control the receive shifter, which shifts at half the frequency of the transmit shifter.

When the result of the key scan is available in the receive shifter register, FLEXIO continues to monitor the row inputs and can trigger multiple scans from a single keypress. To support debouncing, you can decide how many consecutive scans must be considered as a single keypress.

NOTE

Because the pins used in Logic mode are fixed per shifter and the shifters that support parallel shifts are limited, this configuration is restricted to what pins and shifters it can use. Increasing the matrix beyond four-row inputs requires multiple shifters in Logic mode. Increasing beyond four-column outputs requires concatenating transmit shifters to create a larger shift register.

Table 320. Keypad interface configuration

Register	Value	Configuration
SHIFTCFG0	0003_0032h	Start bit enabled (logic 0) and stop bit enabled (logic 1), configured for 4-bit shift.
SHIFTCTL0	0101_0402h	Configure transmit using timer 1 on the rising edge of clock generating open-drain output on pins[7:4] (column outputs).
SHIFTBUF0	0804_0201h	Static data containing the column scan pattern; each column is scanned one-hot with dead time in between.
SHIFTCFG1	0000_0020h	In Logic mode, mask input 3.
SHIFTCTL1	0000_0007h	Configure Logic mode.
SHIFTBUF1	07FF_07FFh	Static data configuring Logic mode LUT. Output asserts if pins [3:1] are logic 0.
SHIFTCFG3	0007_0000h	Configured for 8-bit shift.
SHIFTCTL3	0280_0001h	Configure receive using timer 2 on falling edge of clock with input data on pins [7:0] (both rows and columns; pin 0 is don't care).
TIMCMP0	0000_00ffh	Configure prescanning glitch filter to 256 FLEXIO clock cycles. For different filter cycles, configure TIMCMP[15:0] as (filter cycles) - 1.
TIMCFG0	0103_6600h	Configure enable on trigger rising edge and disable on trigger falling edge. Initial clock state is logic 0 and is not affected by reset.
TIMCTL0	0540_0003h	Configure 16-bit counter using the shifter 1 flag (logic state) as trigger.
TIMCMP1	0000_0F3Fh	Configure eight shifts (twice for each column) at column scan rate of divide by 128. For a different scan frequency, define TIMCMP[7:0] as (scan divider ÷ 2) - 1.
TIMCFG1	0020_2622h	Enable on trigger rising edge, disable on compare. Start and stop bits are enabled.
TIMCTL1	0340_0001h	Configure dual 8-bit counter using timer 0 output as the trigger.
TIMCMP2	0000_0801h	Configure four shifts at half the frequency of the timer 1 trigger.

Table continues on the next page...

Table 320. Keypad interface configuration (continued)

Register	Value	Configuration
TIMCFG2	0110_2100h	Enable on timer 1 enable, disable on compare, decrement on trigger input with output initially negated, and not affected by reset.
TIMCTL2	0740_0001h	Configure dual 8-bit counter using timer 1 output as the trigger.
SHIFTBUF3	Keypad scan result	Keypad scan result can be read from SHIFTBUF3. Each byte is the result of one scan with both row and column pin state from the scan (pin 0 is not used). Use the shifter status flag to indicate when data can be written using interrupt or DMA request.

40.7 Memory map and registers

40.7.1 FLEXIO register descriptions

NOTE

Invalid register accesses, which include reading a write-only register, writing to a read-only register, or accessing an invalid address, result in a bus error.

40.7.1.1 FLEXIO memory map

FLEXIO1 base address: 425C_0000h

FLEXIO2 base address: 425D_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VERID)	32	R	0201_0003
4	Parameter (PARAM)	32	R	0420_0808
8	FLEXIO Control (CTRL)	32	RW	0000_0000
C	Pin State (PIN)	32	R	0000_0000
10	Shifter Status (SHIFTSTAT)	32	RW	0000_0000
14	Shifter Error (SHIFTERR)	32	RW	0000_0000
18	Timer Status Flag (TIMSTAT)	32	RW	0000_0000
20	Shifter Status Interrupt Enable (SHIFTSIEN)	32	RW	0000_0000
24	Shifter Error Interrupt Enable (SHIFTEIEN)	32	RW	0000_0000
28	Timer Interrupt Enable (TIMIEN)	32	RW	0000_0000
30	Shifter Status DMA Enable (SHIFTSDEN)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
38	Timer Status DMA Enable (TIMERSDEN)	32	RW	0000_0000
40	Shifter State (SHIFTSTATE)	32	RW	0000_0000
48	Trigger Status (TRGSTAT)	32	RW	0000_0000
4C	External Trigger Interrupt Enable (TRIGIEN)	32	RW	0000_0000
50	Pin Status (PINSTAT)	32	RW	0000_0000
54	Pin Interrupt Enable (PINIEN)	32	RW	0000_0000
58	Pin Rising Edge Enable (PINREN)	32	RW	0000_0000
5C	Pin Falling Edge Enable (PINFEN)	32	RW	0000_0000
60	Pin Output Data (PINOUTD)	32	RW	0000_0000
64	Pin Output Enable (PINOUTE)	32	RW	0000_0000
68	Pin Output Disable (PINOUTDIS)	32	RW	0000_0000
6C	Pin Output Clear (PINOUTCLR)	32	RW	0000_0000
70	Pin Output Set (PINOUTSET)	32	RW	0000_0000
74	Pin Output Toggle (PINOUTTOG)	32	RW	0000_0000
80 - 9C	Shifter Control (SHIFTCTL0 - SHIFTCTL7)	32	RW	0000_0000
100 - 11C	Shifter Configuration (SHIFTCFG0 - SHIFTCFG7)	32	RW	0000_0000
200 - 21C	Shifter Buffer (SHIFTBUF0 - SHIFTBUF7)	32	RW	0000_0000
280 - 29C	Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7)	32	RW	0000_0000
300 - 31C	Shifter Buffer Byte Swapped (SHIFTBUFBYS0 - SHIFTBUFBYS7)	32	RW	0000_0000
380 - 39C	Shifter Buffer Bit Byte Swapped (SHIFTBUFBBS0 - SHIFTBUFBBS7)	32	RW	0000_0000
400 - 41C	Timer Control (TIMCTL0 - TIMCTL7)	32	RW	0000_0000
480 - 49C	Timer Configuration (TIMCFG0 - TIMCFG7)	32	RW	0000_0000
500 - 51C	Timer Compare (TIMCMP0 - TIMCMP7)	32	RW	0000_0000
680 - 69C	Shifter Buffer Nibble Byte Swapped (SHIFTBUFNBS0 - SHIFTBUFNBS7)	32	RW	0000_0000
700 - 71C	Shifter Buffer Halfword Swapped (SHIFTBUFHWS0 - SHIFTBUFHWS7)	32	RW	0000_0000
780 - 79C	Shifter Buffer Nibble Swapped (SHIFTBUFNIS0 - SHIFTBUFNIS7)	32	RW	0000_0000
800 - 81C	Shifter Buffer Odd Even Swapped (SHIFTBUFOES0 - SHIFTBUFOES7)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
880 - 89C	Shifter Buffer Even Odd Swapped (SHIFTBUFEOS0 - SHIFTBUFEOS7)	32	RW	0000_0000
900 - 91C	Shifter Buffer Halfword Byte Swapped (SHIFTBUFHBS0 - SHIFTBUFHBS7)	32	RW	0000_0000

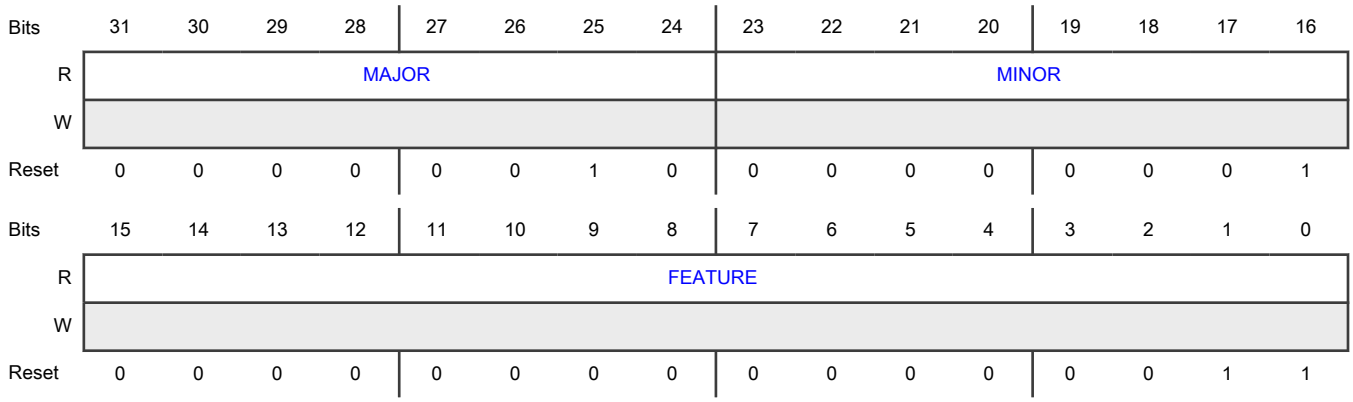
40.7.1.2 Version ID (VERID)

Indicates the version of FLEXIO.

Offset

Register	Offset
VERID	0h

Diagram



Fields

Field	Description
31-24 MAJOR	Major Version Number Indicates the major version number of the module specification.
23-16 MINOR	Minor Version Number Indicates the minor version number of the module specification.
15-0 FEATURE	Feature Specification Number Indicates the feature set number. 0000_0000_0000_0000 - Standard features implemented

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0000_0000_0000_0001 - State, logic, and parallel modes supported 0000_0000_0000_0010 - Pin control registers supported 0000_0000_0000_0011 - State, logic, and parallel modes, plus pin control registers supported

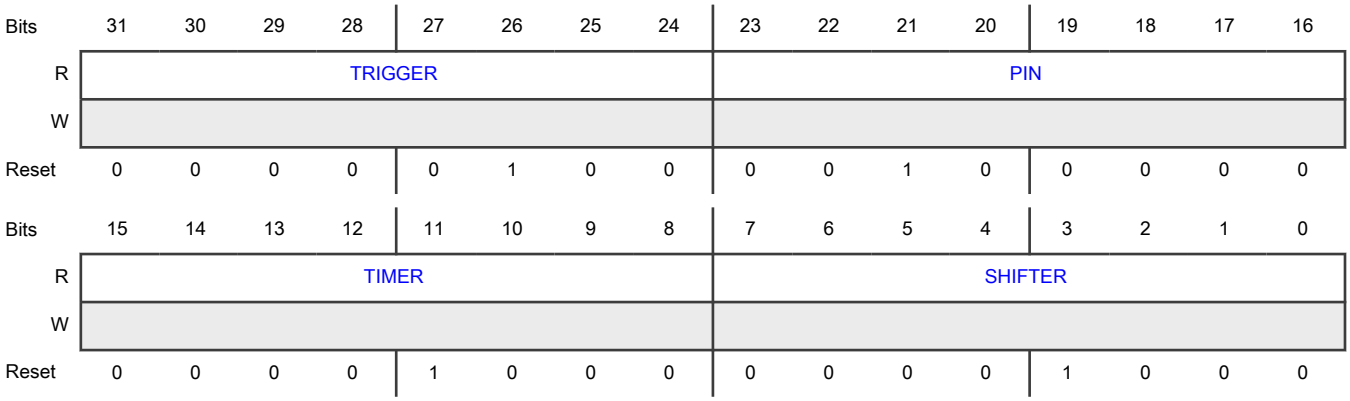
40.7.1.3 Parameter (PARAM)

Contains the number of shifters, timers, pins, and triggers.

Offset

Register	Offset
PARAM	4h

Diagram



Fields

Field	Description
31-24 TRIGGER	Trigger Number Indicates the number of external triggers implemented.
23-16 PIN	Pin Number Indicates the number of pins implemented.
15-8 TIMER	Timer Number Indicates the number of timers implemented.
7-0 SHIFTER	Shifter Number Indicates the number of shifters implemented.

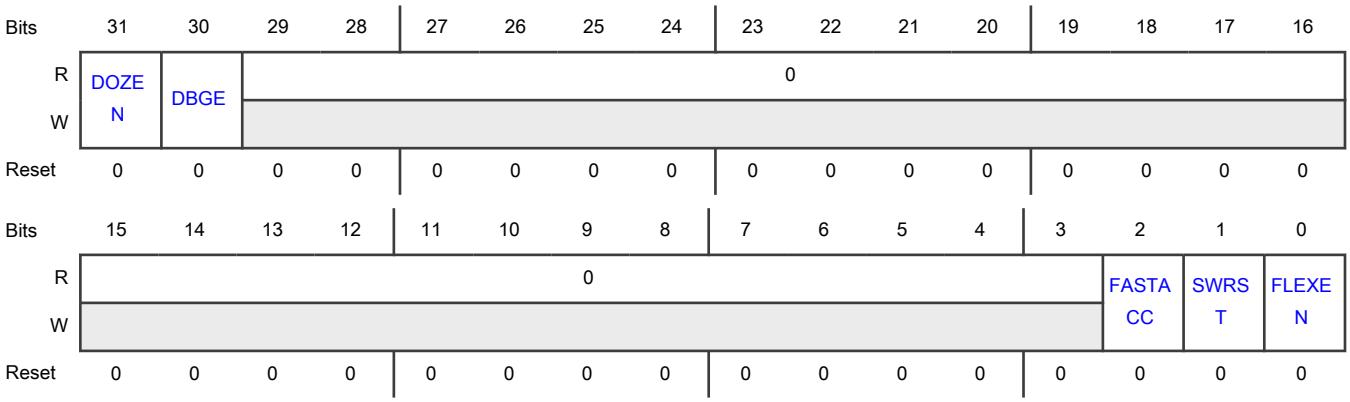
40.7.1.4 FLEXIO Control (CTRL)

Controls various aspects of the FLEXIO operation.

Offset

Register	Offset
CTRL	8h

Diagram



Fields

Field	Description
31 DOZEN	Doze Enable Disables FLEXIO operation in Doze modes. 0 - Enable 1 - Disable
30 DBGE	Debug Enable Enables the FLEXIO operation in Debug mode. 0 - Disable 1 - Enable
29-3 —	Reserved
2 FASTACC	Fast Access Configures fast or normal register accesses to FLEXIO registers, but requires the FLEXIO functional clock to be at least equal to the frequency of the bus clock. 0 - Normal 1 - Fast

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 SWRST	Software Reset Specifies whether software reset is enabled. The software reset does not affect this register but it affects all other logic in FLEXIO. All other register accesses are ignored until this field is cleared. The field remains 1 until software clears it and the reset has cleared in the FLEXIO clock domain. If you write 1 to this field, all FLEXIO registers except the Control register are reset. 0 - Disabled 1 - Enabled
0 FLEXEN	FLEXIO Enable Enables FLEXIO. 0 - Disable 1 - Enable

40.7.1.5 Pin State (PIN)

Indicates the status of the pin data input.

Offset

Register	Offset
PIN	Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PDI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PDI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PDI	Pin Data Input Indicates the input data on each of the FLEXIO pins.

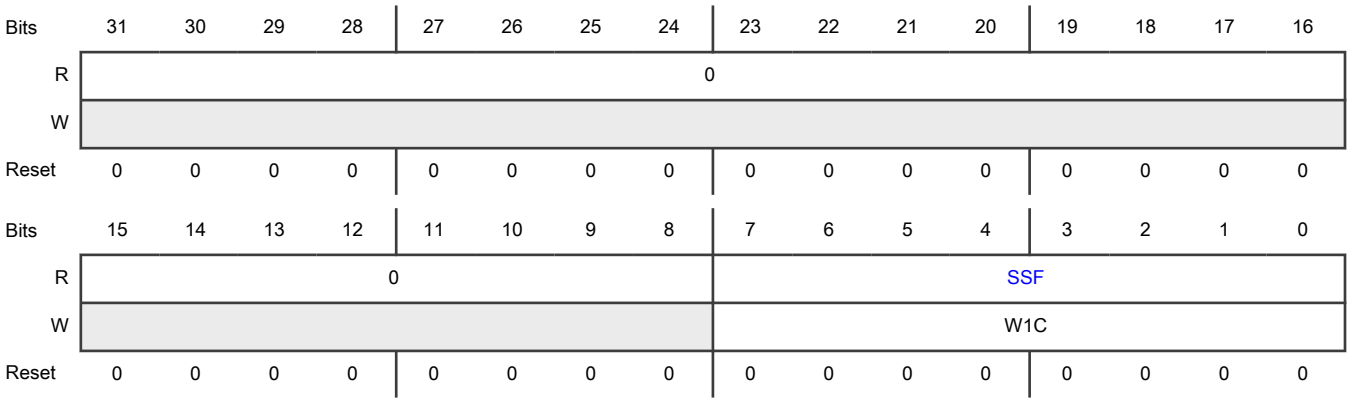
40.7.1.6 Shifter Status (SHIFTSTAT)

Contains shifter status flags.

Offset

Register	Offset
SHIFTSTAT	10h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 SSF	Shifter Status Flag Indicates the shifter status. This flag is updated in one of the following cases: • If SHIFTCTL*n*[SMOD] = 001b (Receive mode), the status flag is set when SHIFTBUF is loaded with data from the shifter (SHIFTBUF is full). The status flag is cleared when you read Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). • If SHIFTCTL*n*[SMOD] = 010b (Transmit mode), the status flag is set when SHIFTBUF data is transferred to the shifter (SHIFTBUF is empty) or when SHIFTCTL*n*[SMOD] is initially configured as 010b (Transmit mode). The status flag is cleared when you write to the SHIFTBUF register. • If SHIFTCTL*n*[SMOD] = 100b (Match Store mode), the status flag is set when a match occurs between SHIFTBUF and the shifter. The status flag is cleared when you read the SHIFTBUF register. • If SHIFTCTL*n*[SMOD] = 101b (Match Continuous mode), the status flag returns the current match result between SHIFTBUF and the shifter. You cannot clear the status flag by reading the SHIFTBUF register. • If SHIFTCTL*n*[SMOD] = 110b (State mode), the status flag for a shifter sets when it is selected by the current state pointer. • If SHIFTCTL*n*[SMOD] = 111b (Logic mode), the status flag returns the current value of the programmable logic block output.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can clear this status flag by writing a logic one to the flag for all modes except Match Continuous mode, State mode, and Logic mode. **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000 - Clear 0000_0001 - Set When writing 0000_0000 - No effect 0000_0001 - Clear the flag

40.7.1.7 Shifter Error (SHIFTERR)

Reports shifter errors.

Offset

Register	Offset
SHIFTERR	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								SEF							
W									W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
7-0 SEF	Shifter Error Flag Indicates shifter error flag status. This flag is set when one of the following events occurs: • If SHIFTCTL*n*[SMOD] = 001b (Receive mode), it indicates that either the shifter is ready to store new data into SHIFTBUF before the previous data is read from SHIFTBUF (SHIFTBUF overrun), or the received start or stop bit does not match the expected value. • If SHIFTCTL*n*[SMOD] = 010b (Transmit mode), it indicates that the shifter is ready to load new data from SHIFTBUF before new data is written into SHIFTBUF (SHIFTBUF underrun). • If SHIFTCTL*n*[SMOD] = 100b (Match Store mode), it indicates the occurrence of a match event before the previous match data is read from SHIFTBUF (SHIFTBUF overrun). • If SHIFTCTL*n*[SMOD] = 101b (Match Continuous mode), the error flag is set when a match occurs between SHIFTBUF and the shifter. • If SHIFTCTL*n*[SMOD] = 111b (Logic mode), the error flag is set when the output of the programmable logic block is asserted. For SHIFTCTL*n*[SMOD] = 101b (Match Continuous mode), the flag can also be cleared when you read Shifter Buffer (SHIFTBUF0 - SHIFTBUF7). **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000 - Clear 0000_0001 - Set When writing 0000_0000 - No effect 0000_0001 - Clear the flag

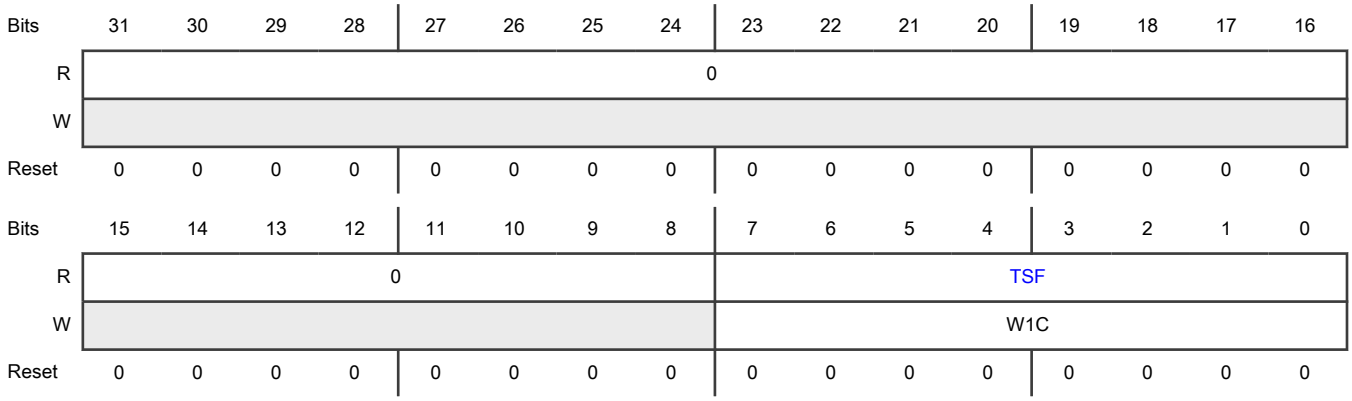
40.7.1.8 Timer Status Flag (TIMSTAT)

Reports timer status.

Offset

Register	Offset
TIMSTAT	18h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 TSF	Timer Status Flag Indicates timer status. This flag is set depending on Timer mode: • In 8-bit baud counter mode, this flag is set when the upper 8-bit counter equals zero and decrements. • In 8-bit high PWM mode, this flag is set when the upper 8-bit counter equals zero and decrements. • In 16-bit counter mode, this flag is set when the 16-bit counter equals zero and decrements. • In 16-bit counter disable mode, TSF is set when a timer disable event is detected. • In 8-bit word counter mode, TSF is set when the upper 8-bit counter equals zero and decrements. • In 8-bit low PWM mode, TSF is set when the upper 8-bit counter equals zero and decrements. • In 16-bit input capture mode, TSF is set when a timer disable event is detected and the flag is clear. In this mode, you must read Timer Control (TIMCTL0 - TIMCTL7) only when TSF is set. **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000 - Clear 0000_0001 - Set When writing 0000_0000 - No effect 0000_0001 - Clear the flag

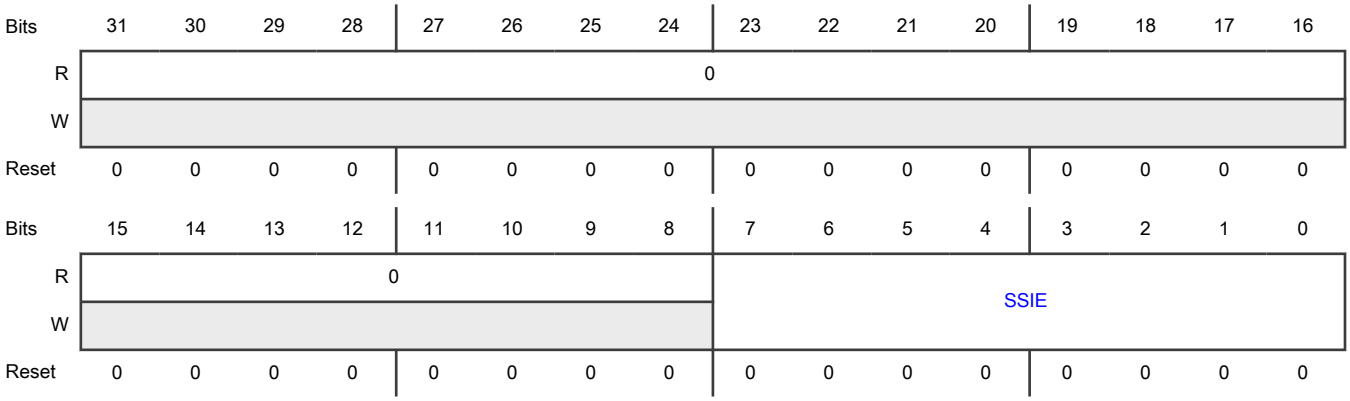
40.7.1.9 Shifter Status Interrupt Enable (SHIFTSIEN)

Enables shifter status interrupts.

Offset

Register	Offset
SHIFTSIEN	20h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 SSIE	Shifter Status Interrupt Enable Enables interrupt generation when the corresponding SHIFTSTAT[SSF] flag is set. If you write 0 to this field, SHIFTSTAT[SSF] is disabled; and if you write 1 to this field, SHIFTSTAT[SSF] is enabled. 0b - Disable 1b - Enable

40.7.1.10 Shifter Error Interrupt Enable (SHIFTEIEN)

Enables shifter error interrupts.

Offset

Register	Offset
SHIFTEIEN	24h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								SEIE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7-0 SEIE	Shifter Error Interrupt Enable Enables interrupt generation when the corresponding SHIFTERR[SEF] flag is set. If you write 0 to this field, SHIFTERR[SEF] is disabled; and if you write 1 to this field, SHIFTERR[SEF] is enabled. 0b - Disable 1b - Enable

40.7.1.11 Timer Interrupt Enable (TIMIEN)

Enables timer status interrupts.

Offset

Register	Offset
TIMIEN	28h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TEIE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7-0 TEIE	Timer Status Interrupt Enable Enables interrupt generation when the corresponding TIMSTAT[TSF] flag is set. If you write 0 to this field, TIMSTAT[TSF] is disabled; and if you write 1 to this field, TIMSTAT[TSF] is enabled. 0b - Disable 1b - Enable

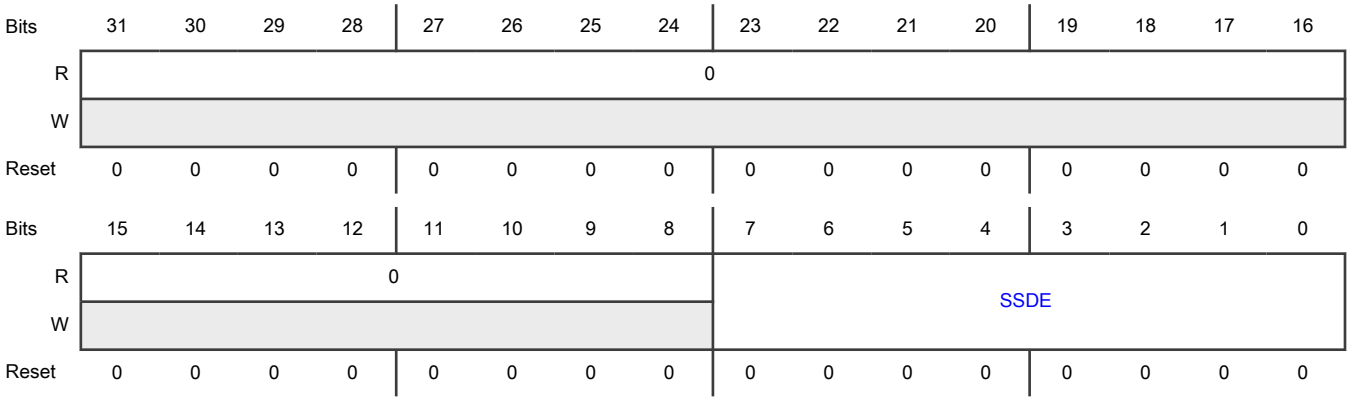
40.7.1.12 Shifter Status DMA Enable (SHIFTSDEN)

Enables shifter DMA requests.

Offset

Register	Offset
SHIFTSDEN	30h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 SSDE	Shifter Status DMA Enable Enables DMA request generation when the corresponding SHIFTSTAT[SSF] flag is set. If you write 0 to this field, SHIFTSTAT[SSF] is disabled; and if you write 1 to this field, SHIFTSTAT[SSF] is enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0b - Disable 1b - Enable

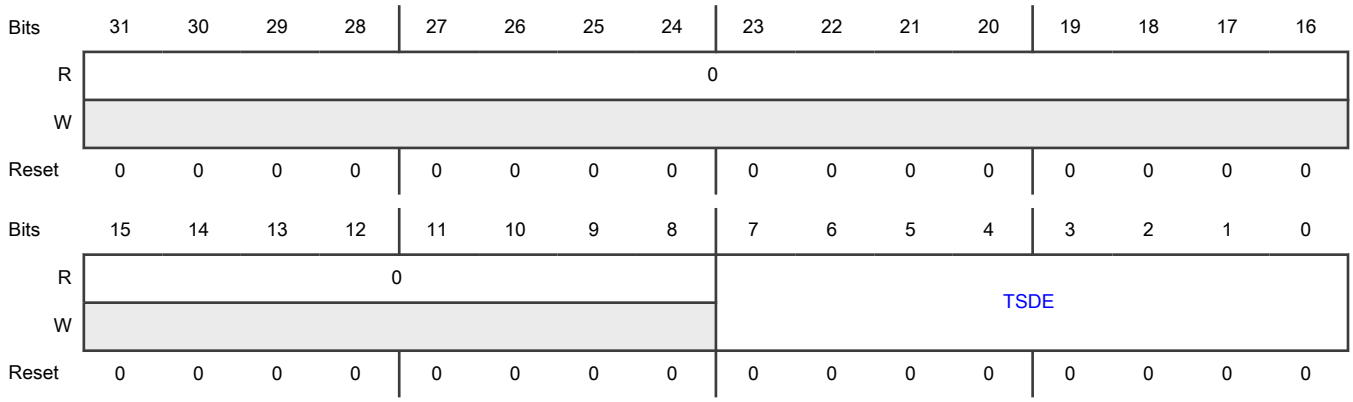
40.7.1.13 Timer Status DMA Enable (TIMERSDEN)

Enables DMA requests when the timer status flag is set.

Offset

Register	Offset
TIMERSDEN	38h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 TSDE	Timer Status DMA Enable Enables DMA request generation when the corresponding TIMSTAT[TSF] flag is set. When the timer status DMA request is enabled, reading or writing to a timer compare register clears the corresponding timer status register. The DMA must therefore read or write to the timer compare register as part of the DMA transfer; otherwise, the DMA request remains asserted. 0b - Disable 1b - Enable

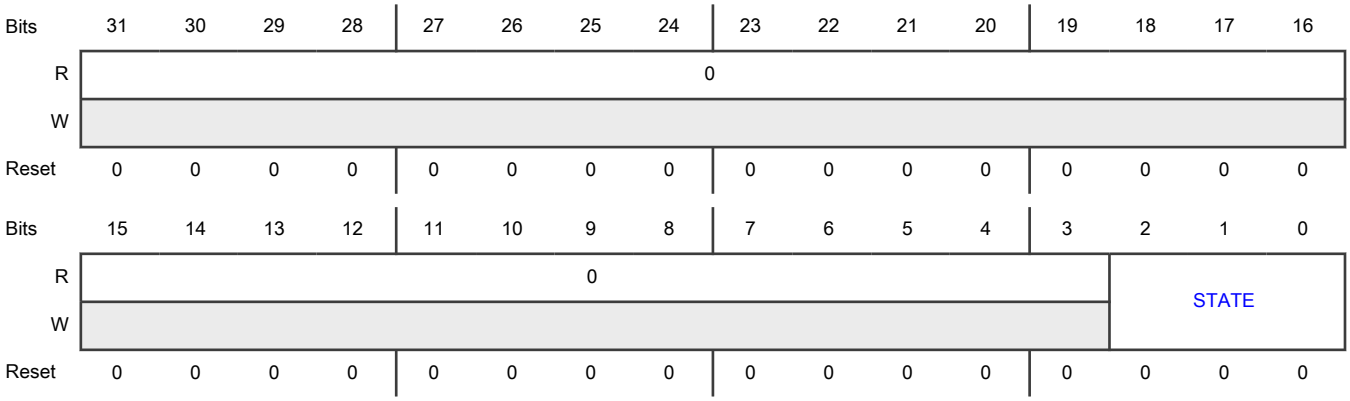
40.7.1.14 Shifter State (SHIFTSTATE)

Contains a pointer to track the current shifter.

Offset

Register	Offset
SHIFTSTATE	40h

Diagram



Fields

Field	Description
31-3 —	Reserved
2-0 STATE	Current State Pointer Maintains a pointer to track the current shifter (configured for State mode) enabled to drive outputs and compute the next state. Reading this register when the state pointer is updating can result in the return of an incorrect state. The value that you write to this field overrides the current state.

40.7.1.15 Trigger Status (TRGSTAT)

Contains external trigger status flags.

Offset

Register	Offset
TRGSTAT	48h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												ETSF			
W													W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved
3-0 ETSF	External Trigger Status Flag Specifies whether the external trigger status flag is set when a rising edge is detected on the corresponding external trigger input. **NOTE** This field behaves differently for register reads and writes. When reading 0000 - Clear 0001 - Set When writing 0000 - No effect 0001 - Clear the flag

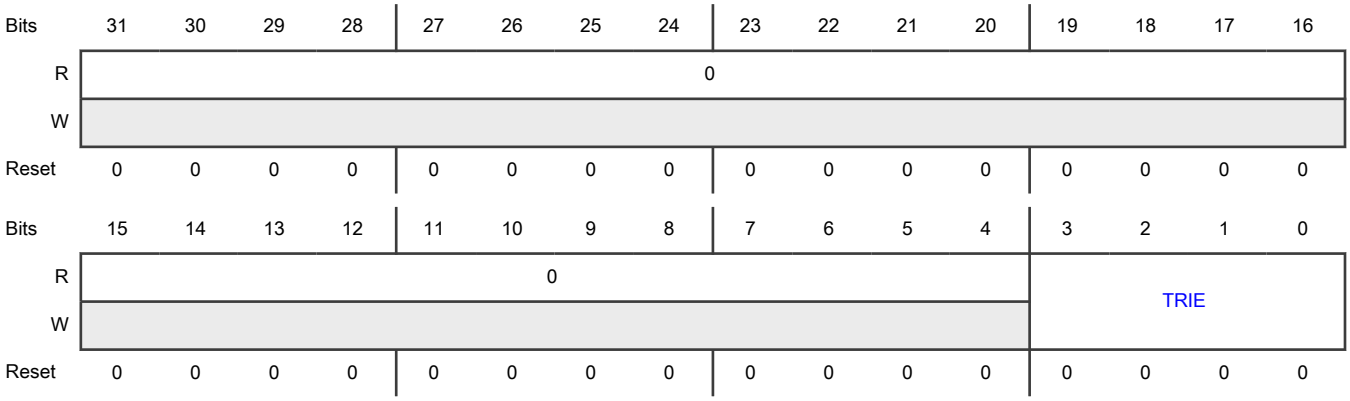
40.7.1.16 External Trigger Interrupt Enable (TRIGIEN)

Enables external trigger interrupts.

Offset

Register	Offset
TRIGIEN	4Ch

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 TRIE	External Trigger Interrupt Enable Enables interrupt generation when the corresponding TRGSTAT[ETSF] flag is set. If you write 0 to this field, TRGSTAT[ETSF] is disabled, and if you write 1 to this field, TRGSTAT[ETSF] is enabled. 0b - Disable 1b - Enable

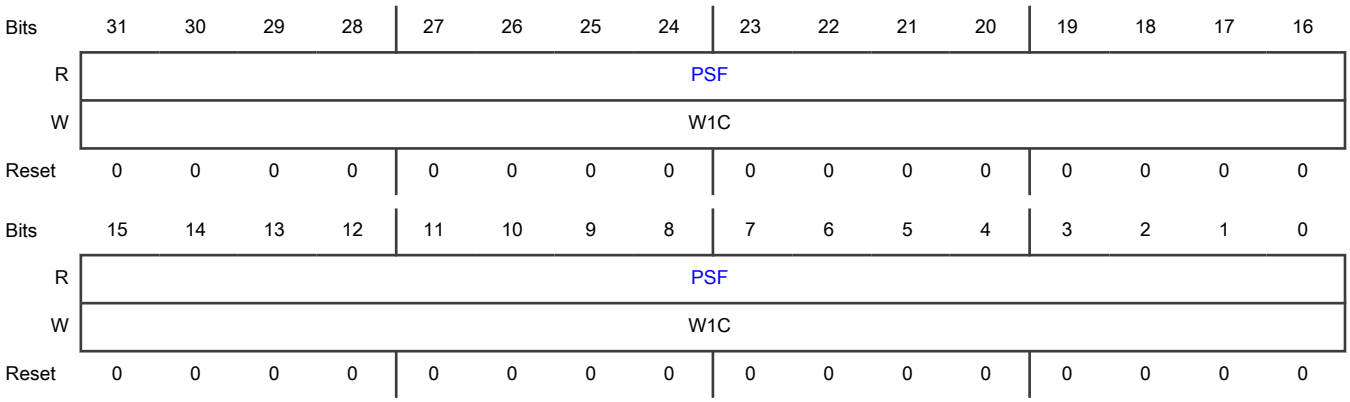
40.7.1.17 Pin Status (PINSTAT)

Contains pin status flags.

Offset

Register	Offset
PINSTAT	50h

Diagram



Fields

Field	Description
31-0 PSF	Pin Status Flag Indicates whether the pin status flag is set when a rising edge or falling edge (if configured) is detected on the corresponding pin, as configured by the pin. **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000_0000_0000_0000_0000_0000_0000 - Clear 0000_0000_0000_0000_0000_0000_0000_0001 - Set When writing 0000_0000_0000_0000_0000_0000_0000_0000 - No effect 0000_0000_0000_0000_0000_0000_0000_0001 - Clear the flag

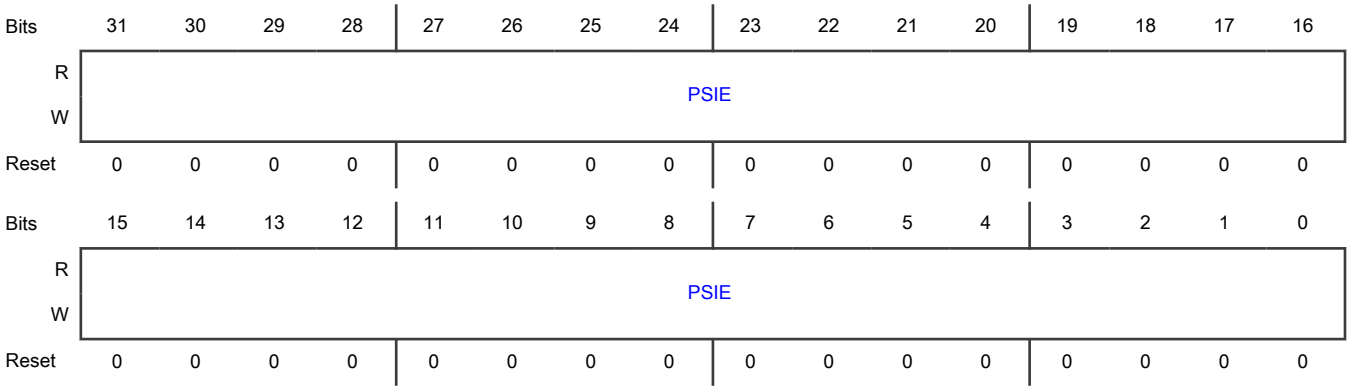
40.7.1.18 Pin Interrupt Enable (PINIEN)

Enables pin status interrupts.

Offset

Register	Offset
PINIEN	54h

Diagram



Fields

Field	Description
31-0	Pin Status Interrupt Enable

Table continues on the next page...

Field	Description
PSIE	Enables interrupt generation when the corresponding PINSTAT[PSF] flag is set. If you write 0 to this field, PINSTAT[PSF] is disabled, and if you write 1 to this field, PINSTAT[PSF] is enabled. 0b - Disable 1b - Enable

40.7.1.19 Pin Rising Edge Enable (PINREN)

Enables the pin status flag on a rising edge.

Offset

Register	Offset
PINREN	58h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PRE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PRE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PRE	Pin Rising Edge Specifies whether the pin status flag is set whenever a rising edge is detected on the pin. 0b - Not set 1b - Set

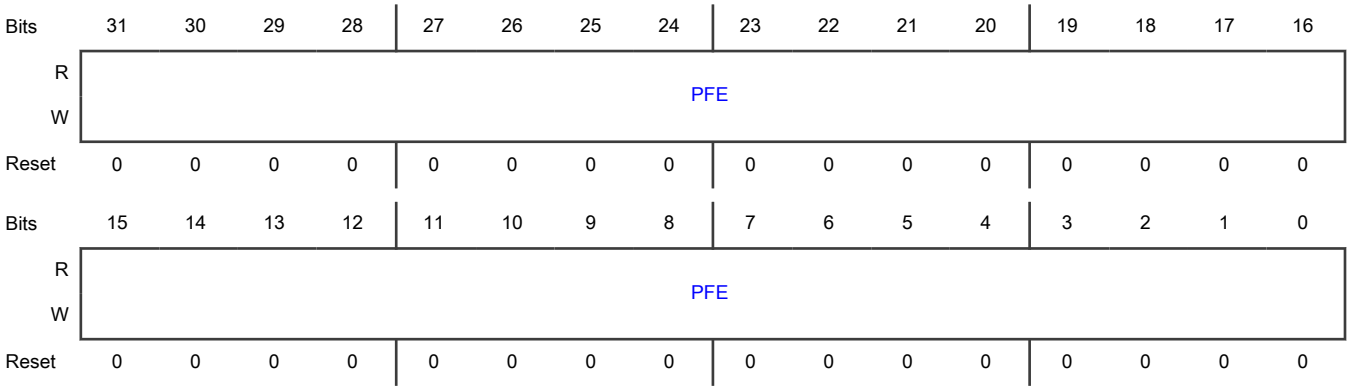
40.7.1.20 Pin Falling Edge Enable (PINFEN)

Enables the pin status flag on a falling edge.

Offset

Register	Offset
PINFEN	5Ch

Diagram



Fields

Field	Description
31-0 PFE	Pin Falling Edge Specifies whether the pin status flag is set whenever a falling edge is detected on the pin. 0b - Not set 1b - Set

40.7.1.21 Pin Output Data (PINOUTD)

Contains data output when direct pin output is enabled.

Offset

Register	Offset
PINOUTD	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OUTD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OUTD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OUTD	Output Data Configures the value driven on the corresponding pin when direct pin output is enabled. 0b - Logic zero 1b - Logic one

40.7.1.22 Pin Output Enable (PINOUTE)

Enables pin output.

Offset

Register	Offset
PINOUTE	64h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OUTE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OUTE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OUTE	Output Enable Enables direct output on the corresponding pin. If this field is 0, the pin is controlled by timer/shifter configuration, and if this field is 1, pin is an output and driven with the value of Pin Output Data (PINOUTD). 0b - Controlled by timer/shifter configuration 1b - Output; driven with value of PINOUTD

40.7.1.23 Pin Output Disable (PINOUTDIS)

Disables pin output.

Offset

Register	Offset
PINOUTDIS	68h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	OUTDIS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	OUTDIS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OUTDIS	Output Disable Configures the corresponding pins to disable direct output. If this field is 1, the corresponding fields in Pin Output Data (PINOUTD) and Pin Output Enable (PINOUTE) become 0. 0b - No effect 1b - Corresponding fields become 0

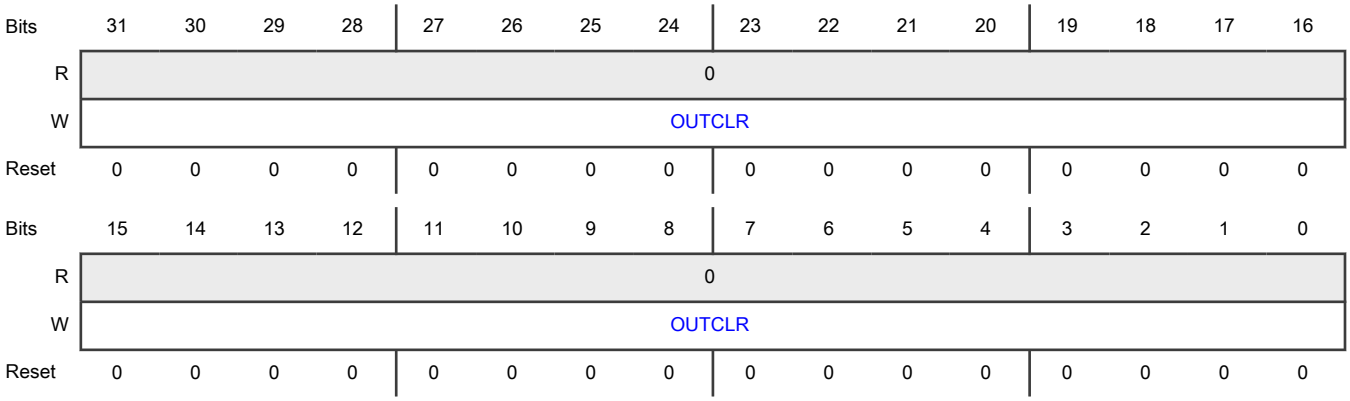
40.7.1.24 Pin Output Clear (PINOUTCLR)

Clears pin output.

Offset

Register	Offset
PINOUTCLR	6Ch

Diagram



Fields

Field	Description
31-0 OUTCLR	Output Clear Configures the corresponding pins to output zero. If this field is 1, the corresponding field in Pin Output Data (PINOUTD) becomes 0 and the one in Pin Output Enable (PINOUTE) becomes 1. 0b - No effect 1b - Corresponding field in PINOUTD becomes 0; corresponding field in PINOUTE becomes 1

40.7.1.25 Pin Output Set (PINOUTSET)

Sets pin output.

Offset

Register	Offset
PINOUTSET	70h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	OUTSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	OUTSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OUTSET	Output Set Configures the corresponding pins to output logic one. If this field is 1, the corresponding fields in Pin Output Data (PINOUTD) and Pin Output Enable (PINOUTE) become 1. 0b - No effect 1b - Corresponding fields become 1

40.7.1.26 Pin Output Toggle (PINOUTTOG)

Toggles pin output.

Offset

Register	Offset
PINOUTTOG	74h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	OUTTOG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	OUTTOG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OUTTOG	Output Toggle Configures the corresponding pins to toggle. If this field is 1, the corresponding field in Pin Output Data (PINOUTD) is inverted and the one in Pin Output Enable (PINOUTE) becomes 1. 0b - No effect 1b - Corresponding field in PINOUTD is inverted; corresponding field in PINOUTE becomes 1

40.7.1.27 Shifter Control (SHIFTCTL0 - SHIFTCTL7)

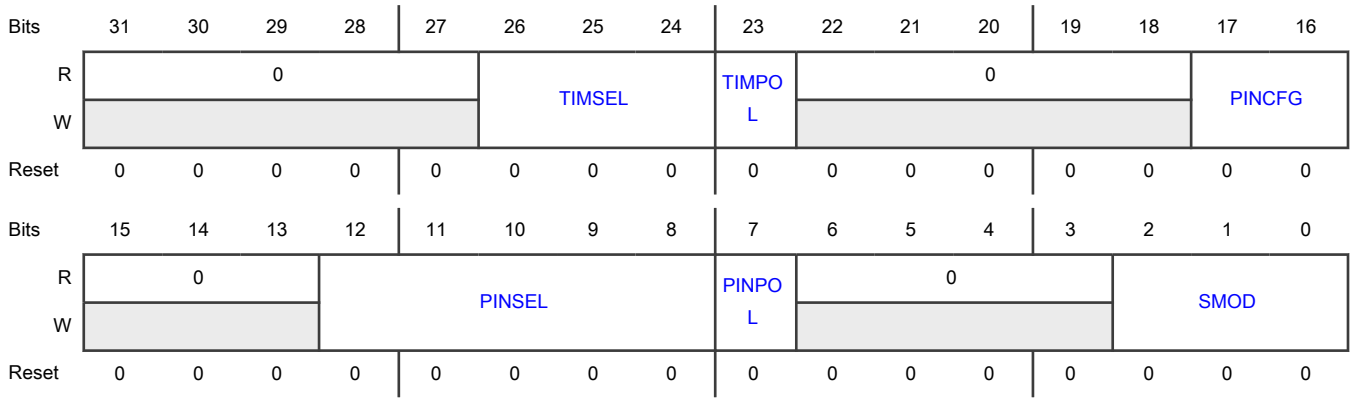
Provides shifter controls.

Offset

For n = 0 to 7:

Register	Offset
SHIFTCTLn	80h + (n × 4h)

Diagram



Fields

Field	Description
31-27 —	Reserved
26-24 TIMSEL	Timer Select Selects which timer is used for controlling the logic or shift register and generating the shift clock. TIMSEL = i selects TIMERi.
23	Timer Polarity

Table continues on the next page...

Table continued from the previous page...

Field	Description
TIMPOL	Determines whether the shift occurs on the positive edge or negative edge of the shift clock. 0 - Positive edge 1 - Negative edge
22-18 —	Reserved
17-16 PINCFG	Shifter Pin Configuration Specifies shifter pin configuration. For pins configured as an output (PINCFG = 11b), this field takes effect when you write to the register. **NOTE** When initially configuring PINCFG as 11b, FLEXIO may briefly drive the pin low. To avoid this, you can configure PINCFG as 10b along with the rest of the Control register and then perform a subsequent write to set PINCFG as 11b. Likewise, when changing the value of PINCFG from 11b to 00b, you must perform an initial write to set PINCFG as 10b and then perform a subsequent write to update the rest of the Control register with the value of PINCFG as 00b. 00 - Shifter pin output disabled 01 - Shifter pin open-drain or bidirectional output enable 10 - Shifter pin bidirectional output data 11 - Shifter pin output
15-13 —	Reserved
12-8 PINSEL	Shifter Pin Select Selects the pin that is used by the shifter input or output. PINSEL = i selects the FXIO_Di pin. For pins configured as an output (PINCFG = 11b), this field takes effect when you write to the register.
7 PINPOL	Shifter Pin Polarity Specifies the shifter pin polarity. For pins configured as an output (PINCFG = 11b), this field takes effect when you write to this register. 0 - Active high 1 - Active low
6-3 —	Reserved
2-0 SMOD	Shifter Mode Configures the mode of the shifter.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	000 - Disable 001 - Receive mode; capture the current shifter content into SHIFTBUF on expiration of the timer 010 - Transmit mode; load SHIFTBUF contents into the shifter on expiration of the timer 011 - Reserved 100 - Match Store mode; shifter data is compared to SHIFTBUF content on expiration of the timer 101 - Match Continuous mode; shifter data is continuously compared to SHIFTBUF contents 110 - State mode; SHIFTBUF contents store programmable state attributes 111 - Logic mode; SHIFTBUF contents implement programmable logic lookup table

40.7.1.28 Shifter Configuration (SHIFTCFG0 - SHIFTCFG7)

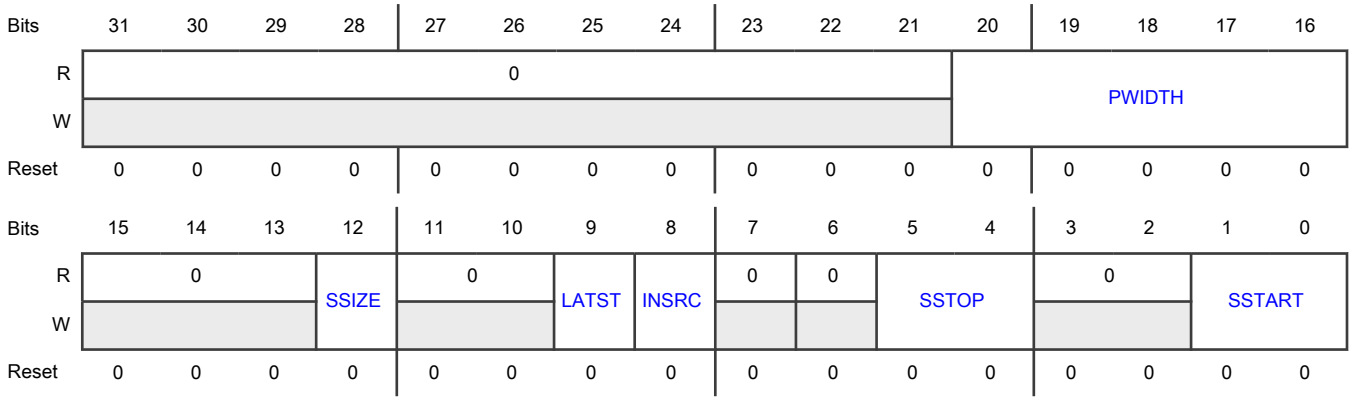
Provides fields for shifter configuration.

Offset

For n = 0 to 7:

Register	Offset
SHIFTCFGn	100h + (n × 4h)

Diagram



Fields

Field	Description
31-21 —	Reserved
20-16	Parallel Width

Table continues on the next page...

Table continued from the previous page...

Field	Description
PWIDTH	Configures the number of bits to be shifted on each shift clock for all shifters: • 1-bit shift for PWIDTH = 0 • 2-bit shift for PWIDTH = 1 • 4-bit shift for PWIDTH = 2...3 • 8-bit shift for PWIDTH = 4...7 • 16-bit shift for PWIDTH = 8...15 • 32-bit shift for PWIDTH = 16...31 For shifters that support parallel transmit (SHIFTER0, SHIFTER4, ...) or parallel receive (SHIFTER3, SHIFTER7, ...), this field, together with PINSEL, also selects the pins to be driven or sampled on each shift clock: FXIO_D[PINSEL+PWIDTH]:FXIO_D[PINSEL]. Shifters that do not support parallel transmit or parallel receive only support parallel shift when SHIFTCFG*n*[INSRC] = 1. If SHIFTCTL*n*[SMOD] = 110b (State mode), use this field to disable state outputs (see State mode).
15-13 —	Reserved
12 SSIZE	Shifter Size Configures the size of the Shift registers. A 24-bit Shift register shifts data only into bits [23:0] and does not update bits [31:24] during shift operations. When the Shift register is configured for a 24-bit shift, configuring PWIDTH as 8..15 performs a 12-bit shift and PWIDTH as 16..31 performs a 24-bit shift. 0 - 32-bit 1 - 24-bit
11-10 —	Reserved
9 LATST	Late Store Configures what happens when a receive or match Shift register is configured to both shift and store on the same cycle. 0 - Store the pre-shift register state 1 - Store the post-shift register state
8 INSRC	Input Source Selects the input source for the shifter. Configuring this field as 1 is not supported for the last shifter. 0 - Pin 1 - Shifter n+1 output

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 —	Reserved
6 —	Reserved
5-4 SSTOP	Shifter Stop Allows automatic stop bit insertion, if the selected timer has also enabled a stop bit, when SHIFTCTL*n*[SMOD] is 10b (Transmit mode). If SHIFTCTL*n*[SMOD] is 1b or 100b (Receive mode or Match Store mode), this field allows automatic stop bit checking if the selected timer has also enabled a stop bit. If SHIFTCTL*n*[SMOD] is 110b (State mode), this field disables state outputs (see State mode). If SHIFTCTL*n*[SMOD] is 111b (Logic mode), this field masks logic pin inputs (see Logic mode). 00 - Stop bit disabled for Transmitter, Receiver, and Match Store modes 01 - Stop bit disabled for Transmitter, Receiver, and Match Store modes; when timer is in stop condition, Receiver and Match Store modes store receive data on the configured shift edge 10 - Transmitter mode outputs stop bit value 0 in Match Store mode; if stop bit is not 0, Receiver and Match Store modes set error flag (when timer is in stop condition, these modes also store receive data on the configured shift edge) 11 - Transmitter mode outputs stop bit value 1 in Match Store mode; if stop bit is not 1, Receiver and Match Store modes set error flag (when timer is in stop condition, these modes also store receive data on the configured shift edge)
3-2 —	Reserved
1-0 SSTART	Shifter Start Allows automatic start bit insertion, if the selected timer has also enabled a start bit, when SHIFTCTL*n*[SMOD] is 10b (Transmit mode). If SHIFTCTL*n*[SMOD] = 1b (Receive mode) or 100b (Match Store mode), this field allows automatic start bit checking if the selected timer has also enabled a start bit. If SHIFTCTL*n*[SMOD] is 110b (State mode), this field disables state outputs (see State mode). If SHIFTCTL*n*[SMOD] = 111b (Logic mode), this field masks logic pin inputs (see Logic mode). 00 - Start bit disabled for Transmitter, Receiver, and Match Store modes; Transmitter mode loads data on enable 01 - Start bit disabled for Transmitter, Receiver, and Match Store modes; Transmitter mode loads data on first shift 10 - Transmitter mode outputs start bit value 0 before loading data on first shift; if start bit is not 0, Receiver and Match Store modes set error flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
	11 - Transmitter mode outputs start bit value 1 before loading data on first shift; if start bit is not 1, Receiver and Match Store modes set error flag

40.7.1.29 Shifter Buffer (SHIFTBUF0 - SHIFTBUF7)

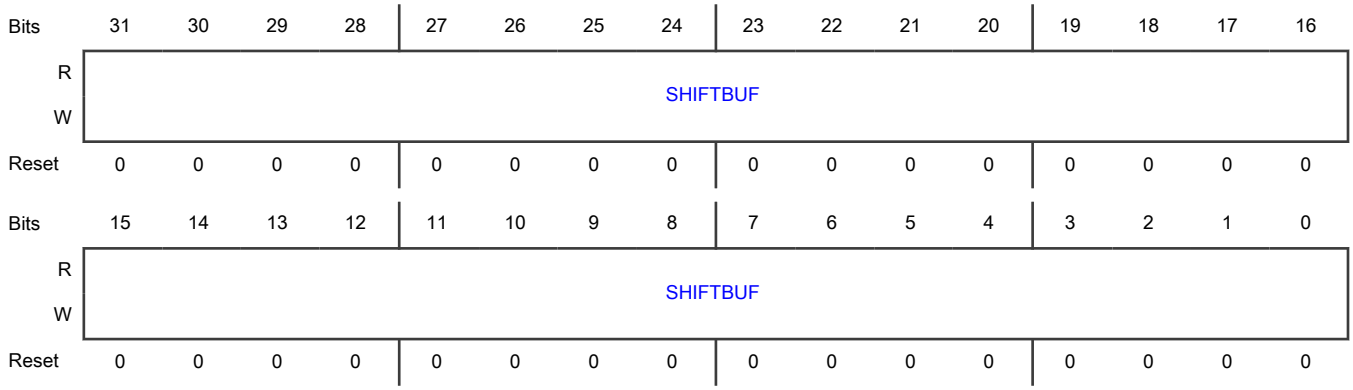
Contains shift buffer data.

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFn	200h + (n × 4h)

Diagram



Fields

Field	Description
31-0 SHIFTBUF	Shift Buffer Contains the data to be matched with the shifter contents and is used for various other functions, depending on the setting of SHIFTCTL*n*[SMOD]: • If SHIFTCTL0[SMOD] is 1b (Receive mode), shifter data is transferred into SHIFTBUF at the expiration of the timer. You must read this register only when the corresponding SHIFTSTAT[SSF] flag is set, indicating that new shifter data is available. • If SHIFTCTL0[SMOD] is 10b (Transmit mode), SHIFTBUF data is transferred into the shifter before the timer begins. • If SHIFTCTL0[SMOD] is 100b (Match Store mode), SHIFTBUF[31:16] contains the data to be matched with the shifter contents and SHIFTBUF[15:0] can be used to mask the match result (1 = mask, 0 = no mask). The match is checked when the timer expires. Shifter data [31:16] is written

Table continues on the next page...

Field	Description
	to SHIFTBUF[31:16] whenever a match event occurs. You must read this register only when the corresponding shifter status flag is set, indicating that new shifter data is available. • If SHIFTCTL0[SMOD] is 101b (Match Continuous mode), SHIFTBUF[31:16] contains the data to be matched with the shifter contents, and SHIFTBUF[15:0] can be used to mask the match result (1 = mask, 0 = no mask). • If SHIFTCTL0[SMOD] is 111b (Logic mode), SHIFTBUF[31:0] implements a 5-input, 32-bit programmable logic lookup table (see Logic mode). • If SHIFTCTL0[SMOD] is 110b (State mode), use SHIFTBUF[31:24] to drive the output value when this shifter is selected by the current state pointer and use SHIFTBUF[23:0] to configure the value of the next state transition (see State mode).

40.7.1.30 Shifter Buffer Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS7)

Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it is bit-swapped.

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFBISn	280h + (n × 4h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFBIS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFBIS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SHIFTBUFBIS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register are bit-swapped. Reads return SHIFTBUF[0:31].

40.7.1.31 Shifter Buffer Byte Swapped (SHIFTBUFBYS0 - SHIFTBUFBYS7)

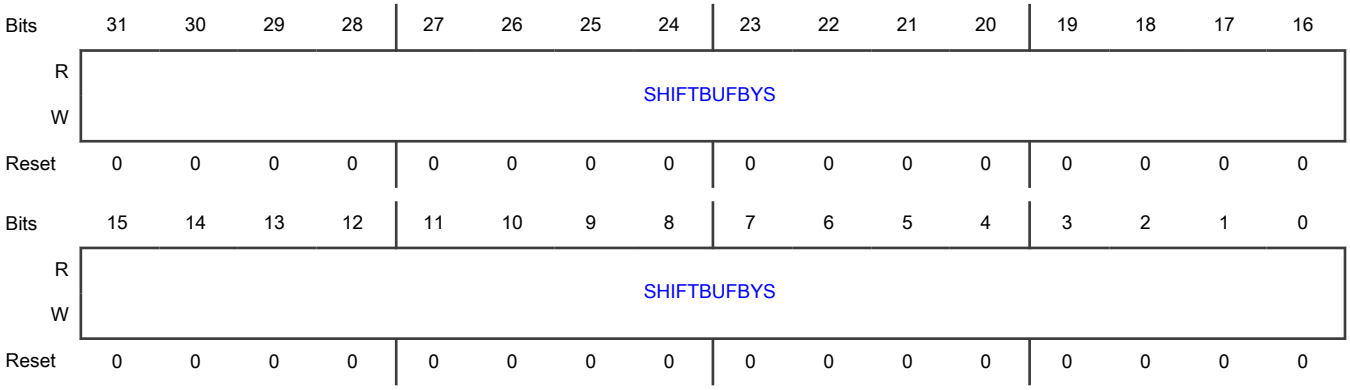
Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it is byte-swapped.

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFBYSn	300h + (n × 4h)

Diagram



Fields

Field	Description
31-0 SHIFTBUFBYS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register are byte-swapped. Reads return {SHIFTBUF[7:0], SHIFTBUF[15:8], SHIFTBUF[23:16], SHIFTBUF[31:24]}.

40.7.1.32 Shifter Buffer Bit Byte Swapped (SHIFTBUFBBS0 - SHIFTBUFBBS7)

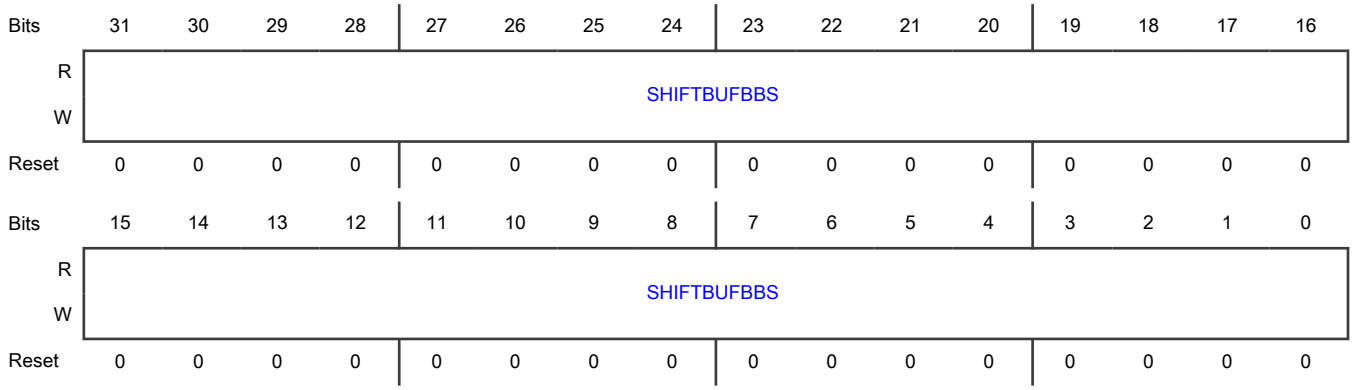
Contains the register data for Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but it is bit-swapped within each byte.

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFBBSn	380h + (n × 4h)

Diagram



Fields

Field	Description
31-0 SHIFTBUFBBS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), except that reads or writes to this register are bit-swapped within each byte. Reads return {SHIFTBUF[24:31], SHIFTBUF[16:23], SHIFTBUF[8:15], SHIFTBUF[0:7]}.

40.7.1.33 Timer Control (TIMCTL0 - TIMCTL7)

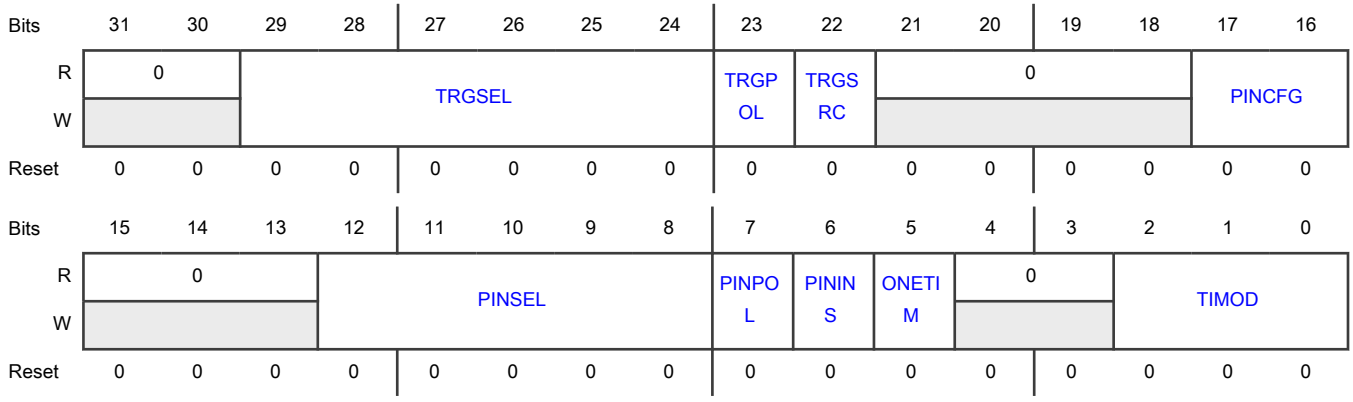
Controls various settings for timer *n*.

Offset

For n = 0 to 7:

Register	Offset
TIMCTLn	400h + (n × 4h)

Diagram



Fields

Field	Description
31-30 —	Reserved
29-24 TRGSEL	Trigger Select Selects the trigger. The valid values for TRGSEL depend on the configuration of Parameter (PARAM): • If TRGSRC = 1, the valid values for *n* depend on the settings of PARAM[PIN], PARAM[TIMER], and PARAM[SHIFTER]. • If TRGSRC = 0, the valid values for *n* depend on PARAM[TRIGGER]. See the chip-specific FLEXIO information for external trigger selection. **NOTE** For a pin, *n* = 0 to 31, for a shifter, *n* = 0 to 7, and for a timer, *n* = 0 to 7. If TRGSRC = 0, configure the trigger selection as *n* = external trigger *n* input. If TRGSRC = 1, you can configure the internal trigger to select an input pin as 2×*n* = pin *n* input. If TRGSRC = 1, you can configure the internal trigger to select a shifter or timer signal as: • 4×*n* + 1 = shifter *n* status flag • 4×*n* + 3 = timer *n* trigger output Following are the values for expanded internal trigger selection (TRGSRC = 1): • 0000 = Pin 0 • 0001 = Shifter 0 flag • 0010 = Pin 1 • 0011 = Timer 0 trigger • 0100 = Pin 2 • 0101 = Shifter 1 flag • 0110 = Pin 3 • 0111 = Timer 1 trigger • ... • This continues up to pin 31, shifter 7, and timer 7.
23 TRGPOL	Trigger Polarity Specifies whether the trigger is active high or active low. 0 - Active high 1 - Active low
22 TRGSRC	Trigger Source Specifies whether the selected trigger source is external or internal.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - External 1 - Internal
21-18 —	Reserved
17-16 PINCFG	Timer Pin Configuration Configures the direction of the timer pin. For pins configured as an output (PINCFG = 11b), this field takes effect when you write to the register. **NOTE** When you initially configure PINCFG as 11b, FLEXIO may briefly drive the pin low. To avoid this, configure PINCFG as 10b along with the rest of the Control register and then perform a subsequent write to set the value of PINCFG as 11b. Likewise, when changing the value of PINCFG from 11b to 00b, you must perform an initial write to set PINCFG as 10b, and then perform a subsequent write to update the rest of the Control register with PINCFG as 00b. 00 - Timer pin output disabled 01 - Timer pin open-drain or bidirectional output enable 10 - Timer pin bidirectional output data 11 - Timer pin output
15-13 —	Reserved
12-8 PINSEL	Timer Pin Select Selects the pin that is used by the timer input or output. PINSEL = i selects the FXIO_Di pin. For pins configured as an output (PINCFG = 11b), this field takes effect when you write to the register.
7 PINPOL	Timer Pin Polarity Specifies the timer pin polarity. For pins configured as an output (PINCFG = 11b), this field takes effect when you write to the register. 0 - Active high 1 - Active low
6 PININS	Timer Pin Input Select Specifies what selects the timer pin input. If this field is 1, the timer input pin is different from the timer output pin. PINSEL must select an even-numbered pin when this field is 1, which means that the output pin is even-numbered and input pin is odd-numbered. 0 - PINSEL selects timer pin input and output 1 - PINSEL + 1 selects the timer pin input; timer pin output remains selected by PINSEL

Table continues on the next page...

Table continued from the previous page...

Field	Description
5 ONETIM	Timer One Time Operation Configures the timer to perform a single enable or disable iteration. Clear the timer status flag for the timer to be enabled again. 0 - Generate the timer enable event as normal 1 - Block the timer enable event unless the timer status flag is clear
4-3 —	Reserved
2-0 TIMOD	Timer Mode Specifies the timer mode: • In 8-bit baud counter mode, the lower 8 bits of the counter and compare register are used to configure the baud rate of the timer shift clock. The upper 8 bits are used to configure the shifter bit count. • In 8-bit PWM high mode, the lower 8 bits of the counter and compare register are used to configure the high period of the timer shift clock. The upper 8 bits are used to configure the low period of the timer shift clock. The shifter bit count is configured using another timer or external signal. • In 16-bit counter mode, the full 16 bits of the counter and compare register are used to configure either the baud rate of the shift clock or the shifter bit count. • In 16-bit counter disable mode, the full 16 bits of the counter and compare register are used to configure either the baud rate of the shift clock or the shifter bit count. • In 8-bit word counter mode, the lower 8 bits of the counter and compare register are used to configure the shifter bit count. The upper 8 bits are used to configure the shifter word count. • In 8-bit PWM low mode, the lower 8 bits of the counter and compare register are used to configure the low period of the timer shift clock. The upper 8 bits are used to configure the high period of the timer shift clock. Use another timer or external signal to configure the shifter bit count. • In 16-bit input capture mode, the inverted value of the 16-bit counter is latched into the compare register when a timer counter disable condition is detected (as configured by TIMCFG*n*[TIMDIS]). This also sets the timer status flag. The timer counter is immediately restarted from FFFFh. 000 - Timer disabled 001 - Dual 8-bit counters baud mode 010 - Dual 8-bit counters PWM high mode 011 - Single 16-bit counter mode 100 - Single 16-bit counter disable mode 101 - Dual 8-bit counters word mode 110 - Dual 8-bit counters PWM low mode 111 - Single 16-bit input capture mode

40.7.1.34 Timer Configuration (TIMCFG0 - TIMCFG7)

Controls various aspects of timer configuration.

The options to enable or disable the timer using the timer *n* - 1 enable or disable are reserved when *n* is evenly divisible by 4 (timer 0, for example).

NOTE

The pin and trigger level and edges specified in this register refer to the signal state after being modified by the settings of TIMCTL*n*[PINPOL] and TIMCTL*n*[TRGPOL]. For example, "trigger low" means that a trigger is actually at logic level 1 if TIMCTL*n*[TRGPOL] is 1 (active low).

Offset

For n = 0 to 7:

Register	Offset
TIMCFGn	480h + (n × 4h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						TIMOUT		0		TIMDEC		0		TIMRST	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	TIMDIS			0	TIMENA			0		TSTOP		0		TSTART	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-26 —	Reserved
25-24 TIMOUT	Timer Output Configures the initial state of the timer output and whether it is affected by the timer reset. 00 - Logic one when enabled; not affected by timer reset 01 - Logic zero when enabled; not affected by timer reset 10 - Logic one when enabled and on timer reset 11 - Logic zero when enabled and on timer reset
23 —	Reserved
22-20	Timer Decrement

Table continues on the next page...

Table continued from the previous page...

Field	Description
TIMDEC	Configures the source of the timer decrement and that of the shift clock. 000 - Decrement counter on FLEXIO clock; shift clock equals timer output 001 - Decrement counter on trigger input (both edges); shift clock equals timer output 010 - Decrement counter on pin input (both edges); shift clock equals pin input 011 - Decrement counter on trigger input (both edges); shift clock equals trigger input 100 - Decrement counter on FLEXIO clock divided by 16; shift clock equals timer output 101 - Decrement counter on FLEXIO clock divided by 256; shift clock equals timer output 110 - Decrement counter on pin input (rising edge); shift clock equals pin input 111 - Decrement counter on trigger input (rising edge); shift clock equals trigger input
19 —	Reserved
18-16 TIMRST	Timer Reset Configures the condition that causes the timer counter (and optionally the timer output) to be reset. In 8-bit counter mode, the timer reset only resets the lower 8 bits that configure the baud rate. In all other modes, the timer reset resets full 16 bits of the counter. 000 - Never reset timer 001 - Timer reset on timer output high. 010 - Timer reset on timer pin equal to timer output 011 - Timer reset on timer trigger equal to timer output 100 - Timer reset on timer pin rising edge 101 - Reserved 110 - Timer reset on trigger rising edge 111 - Timer reset on trigger rising or falling edge
15 —	Reserved
14-12 TIMDIS	Timer Disable Configures the condition that causes the timer to be disabled and stop decrementing. 000 - Timer never disabled 001 - Timer disabled on timer n-1 disable 010 - Timer disabled on timer compare (upper 8 bits match and decrement) 011 - Timer disabled on timer compare (upper 8 bits match and decrement) and trigger low 100 - Timer disabled on pin rising or falling edge 101 - Timer disabled on pin rising or falling edge provided trigger is high

Table continues on the next page...

Table continued from the previous page...

Field	Description
	110 - Timer disabled on trigger falling edge 111 - Reserved
11 —	Reserved
10-8 TIMENA	Timer Enable Configures the condition that causes the timer to be enabled and start decrementing. 000 - Timer always enabled 001 - Timer enabled on timer n-1 enable 010 - Timer enabled on trigger high 011 - Timer enabled on trigger high and pin high 100 - Timer enabled on pin rising edge 101 - Timer enabled on pin rising edge and trigger high 110 - Timer enabled on trigger rising edge 111 - Timer enabled on trigger rising or falling edge
7-6 —	Reserved
5-4 TSTOP	Timer Stop Specifies whether the stop bit is enabled. The stop bit can be added on a timer compare (between each word) or on a timer disable. When stop bit is enabled, configured shifters output the contents of the stop bit when the timer is disabled. When stop bit is enabled on timer disable, the timer remains disabled until the next rising edge of the shift clock. If configured for both timer compare and timer disable, only one stop bit is inserted on timer disable. 00 - Disabled 01 - Enabled on timer compare 10 - Enabled on timer disable 11 - Enabled on timer compare and timer disable
3-2 —	Reserved
1 TSTART	Timer Start Specifies whether the start bit is enabled. If it is enabled, configured shifters output the contents of the start bit when the timer is enabled. The timer counter reloads from the compare register on the first rising edge of the shift clock. 0 - Disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enabled
0 —	Reserved

40.7.1.35 Timer Compare (TIMCMP0 - TIMCMP7)

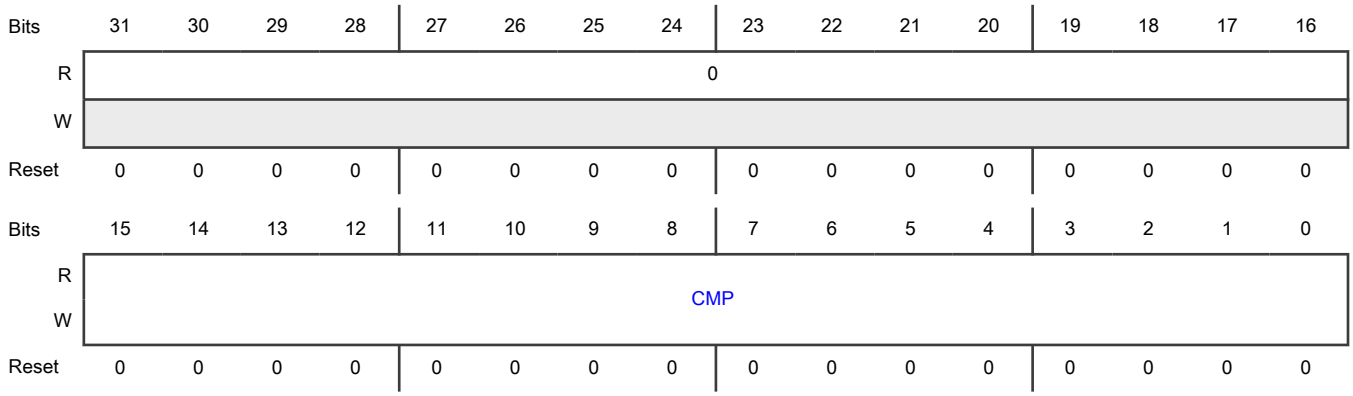
Contains the timer compare value.

Offset

For n = 0 to 7:

Register	Offset
TIMCMPn	500h + (n × 4h)

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 CMP	Timer Compare Value Loads into the timer counter when the timer is first enabled, when the timer is reset, and when the timer decrements down to zero. In 8-bit baud counter mode, the lower 8 bits configure the baud rate divider as (CMP[7:0] + 1) × 2. The upper 8 bits configure the number of bits in each word as (CMP[15:8] + 1) ÷ 2.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	In 8-bit PWM high mode, the lower 8 bits configure the high period of the output to (CMP[7:0] + 1) and the upper 8 bits configure the low period of the output to (CMP[15:8] + 1). In 16-bit counter mode, the compare value can be used to generate the baud rate divider (if shift clock source is timer output) as (CMP[15:0] + 1) × 2. When the shift clock source is a pin or trigger input, the compare register is used to set the number of bits in each word as (CMP[15:0] + 1) ÷ 2. In 16-bit counter disable mode, the compare value can be used to generate the baud rate divider (if shift clock source is timer output) as (CMP[15:0] + 1) × 2. When the shift clock source is a pin or trigger input, the compare register is used to set the number of bits in each word as (CMP[15:0] + 1) ÷ 2. In 8-bit word counter mode, the lower 8 bits configure the number of bits in each word as (CMP[7:0] + 1) ÷ 2. The upper 8 bits configure the number of words to transfer equal to (CMP[15:8] + 1) ÷ 2. In 8-bit PWM low mode, the lower 8 bits configure the low period of the output to (CMP[7:0] + 1) and the upper 8 bits configure the high period of the output to (CMP[15:8] + 1). In 16-bit input capture mode, the compare register is updated with the inverse of the timer counter value whenever the timer status flag is set. You must read this register only when the timer status flag is set.

40.7.1.36 Shifter Buffer Nibble Byte Swapped (SHIFTBUFNBS0 - SHIFTBUFNBS7)

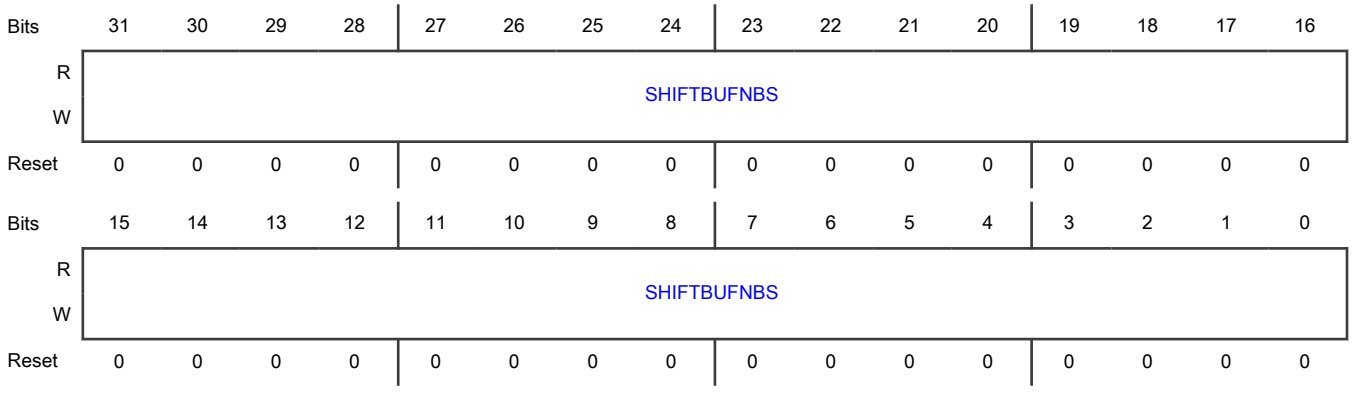
Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it is nibble-swapped within each byte.

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFNBSn	680h + (n × 4h)

Diagram



Fields

Field	Description
31-0 SHIFTBUFNBS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register are nibble-swapped within each byte. Reads return {SHIFTBUF[27:24], SHIFTBUF[31:28], SHIFTBUF[19:16], SHIFTBUF[23:20], SHIFTBUF[11:8], SHIFTBUF[15:12], SHIFTBUF[3:0], SHIFTBUF[7:4]}.

40.7.1.37 Shifter Buffer Halfword Swapped (SHIFTBUFHWS0 - SHIFTBUFHWS7)

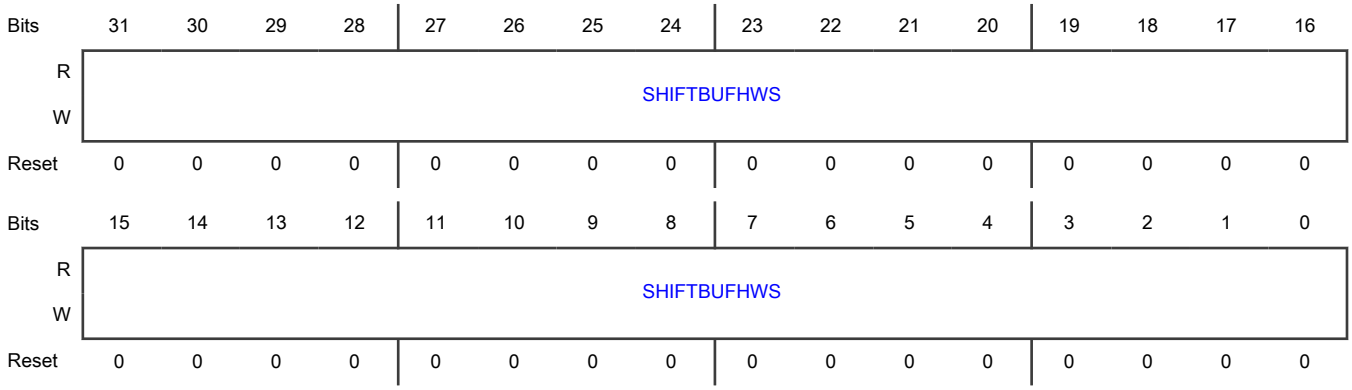
Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it is halfword-swapped.

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFHWSn	700h + (n × 4h)

Diagram



Fields

Field	Description
31-0 SHIFTBUFHWS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register are halfword-swapped. Reads return {SHIFTBUF[15:0], SHIFTBUF[31:16]}.

40.7.1.38 Shifter Buffer Nibble Swapped (SHIFTBUFNIS0 - SHIFTBUFNIS7)

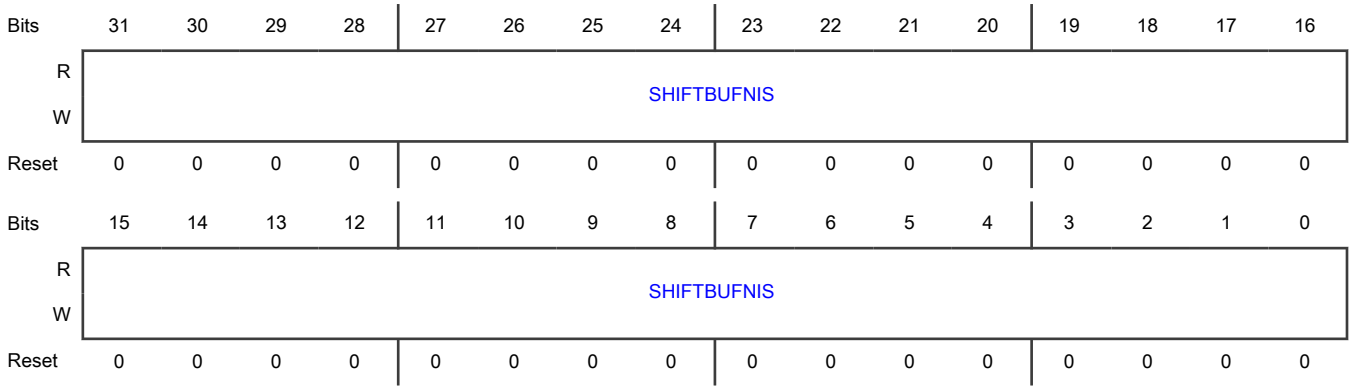
Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it is nibble-swapped.

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFNISn	780h + (n × 4h)

Diagram



Fields

Field	Description
31-0 SHIFTBUFNIS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register are nibble-swapped. Reads return {SHIFTBUF[3:0], SHIFTBUF[7:4], SHIFTBUF[11:8], SHIFTBUF[15:12], SHIFTBUF[19:16], SHIFTBUF[23:20], SHIFTBUF[27:24], SHIFTBUF[31:28]}.

40.7.1.39 Shifter Buffer Odd Even Swapped (SHIFTBUFOES0 - SHIFTBUFOES7)

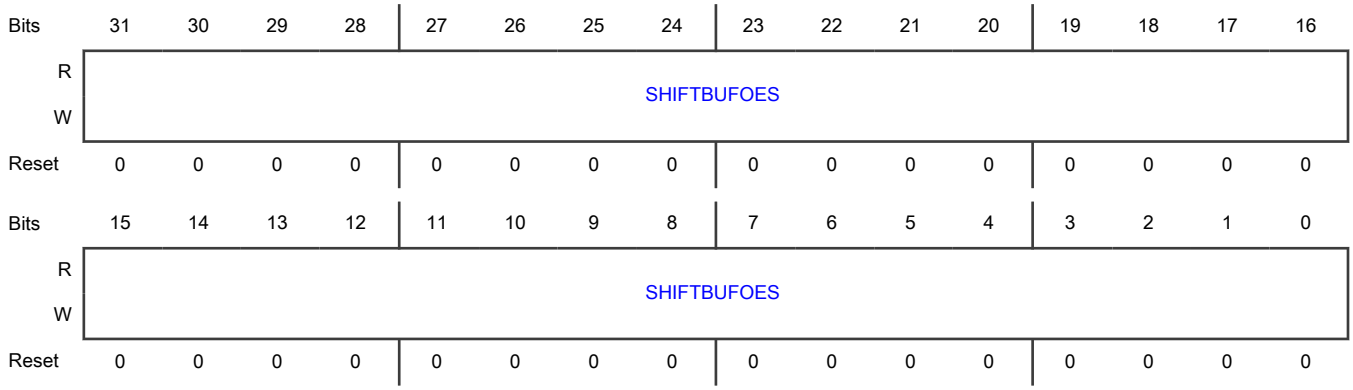
Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it has odd and even bits partitioned separately.

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFOESn	800h + (n × 4h)

Diagram



Fields

Field	Description
31-0 SHIFTBUFOES	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register have the odd and even bits partitioned separately. Only 32-bit accesses are supported for this register. Reads return {SHIFTBUF[31], SHIFTBUF[29], SHIFTBUF[27],SHIFTBUF[25],SHIFTBUF[23],SHIFTBUF[21],SHIFTBUF[19],SHIFTBUF[17],SHIFTBUF[15],SHIFTBUF[13],SHIFTBUF[11],SHIFTBUF[9], SHIFTBUF[7], SHIFTBUF[5], SHIFTBUF[3], SHIFTBUF[1], SHIFTBUF[30], SHIFTBUF[28], SHIFTBUF[26], SHIFTBUF[24], SHIFTBUF[22], SHIFTBUF[20], SHIFTBUF[18], SHIFTBUF[16], SHIFTBUF[14], SHIFTBUF[12], SHIFTBUF[10], SHIFTBUF[8], SHIFTBUF[6], SHIFTBUF[4], SHIFTBUF[2], SHIFTBUF[0]}.

40.7.1.40 Shifter Buffer Even Odd Swapped (SHIFTBUFEOS0 - SHIFTBUFEOS7)

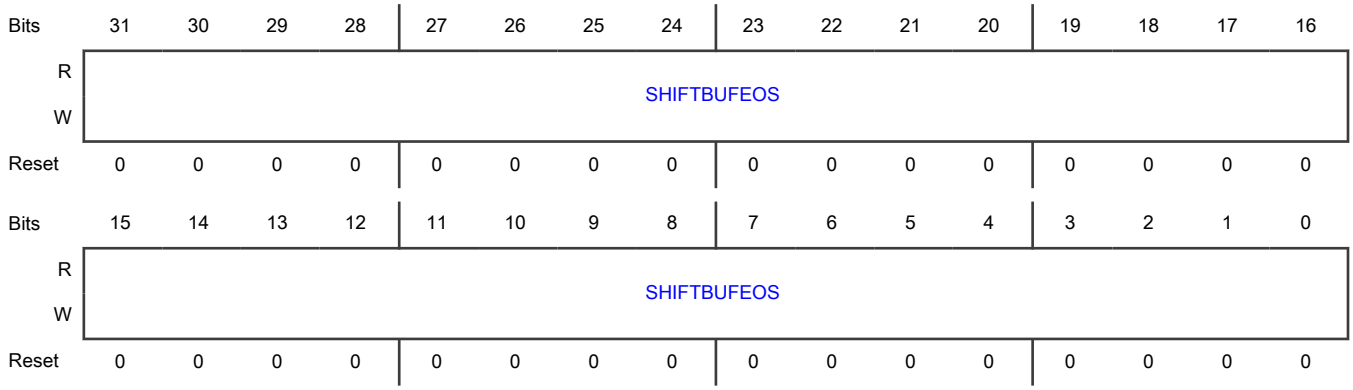
Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, with even and odd bits partitioned separately.

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFEOSn	880h + (n × 4h)

Diagram



Fields

Field	Description
31-0 SHIFTBUFEOS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register have the even and odd bits partitioned separately. Only 32-bit accesses are supported for this register. Reads return {SHIFTBUF[30], SHIFTBUF[28], SHIFTBUF[26], SHIFTBUF[24], SHIFTBUF[22], SHIFTBUF[20], SHIFTBUF[18], SHIFTBUF[16], SHIFTBUF[14], SHIFTBUF[12], SHIFTBUF[10], SHIFTBUF[8], SHIFTBUF[6], SHIFTBUF[4], SHIFTBUF[2], SHIFTBUF[0], SHIFTBUF[31], SHIFTBUF[29], SHIFTBUF[27],SHIFTBUF[25],SHIFTBUF[23],SHIFTBUF[21],SHIFTBUF[19],SHIFTBUF[17],SHIFTBUF[15],SHIFTBUF[13],SHIFTBUF[11],SHIFTBUF[9], SHIFTBUF[7], SHIFTBUF[5], SHIFTBUF[3], SHIFTBUF[1]}.

40.7.1.41 Shifter Buffer Halfword Byte Swapped (SHIFTBUFHBS0 - SHIFTBUFHBS7)

Contains Shifter Buffer (SHIFTBUF0 - SHIFTBUF7) content, but it is halfword byte-swapped.

Offset

For n = 0 to 7:

Register	Offset
SHIFTBUFHBSn	900h + (n × 4h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFHBS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFHBS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SHIFTBUFHBS	Shift Buffer Acts as an alias to Shifter Buffer (SHIFTBUF0 - SHIFTBUF7), but reads or writes to this register are halfword byte-swapped. Reads return {SHIFTBUF[23:16], SHIFTBUF[31:24], SHIFTBUF[7:0], SHIFTBUF[15:8]}.

40.8 Glossary

Baud rate Number of bits that LPUART transmits or receives per second

Break character Break character is generated when the transmitter is holding the data line at the space level for at least 1 character time

Idle character Idle character is generated when the transmitter is holding the data line at logic 1 for at least 1 character time

PWM Pulse width modulation

Chapter 41
Universal Serial Bus Controller (USB)

41.1 Chip-specific USB information

Table 321. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The chip includes two instances of USB 2.0 Controllers and PHY that can be used as USB OTG port, USB HOST port or USB Device port.

The chip supports the following USB Controller and PHY output signals.

USB Instance	Signal
USB1 Controller	OTG_PWR
USB2 Controller	OTG_OC
USB1 Controller	OTG_OC
USB2 Controller	OTG_PWR
USB1 PHY	OTG_ID
USB2 PHY	OTG_ID

41.2 Overview

USB provides a high-performance functionality that conforms to *Universal Serial Bus Specification Revision 2.0*.

USB comprises two independent USB controller cores (two on-the-go (OTG) controller cores). Each controller core supports UTMI interfaces according to its feature support. Both the controller cores are single-port cores. For the OTG cores, there is only one port, which can be used as either a downstream or an upstream port.

See Features for more information.

NOTE

The term "OTG" is used throughout this document. The USB core hardware does not support the automatic OTG function. It is a dual-role (Host and Device mode) core, which you can implement in software with the hardware assist.

41.2.1 Block diagram

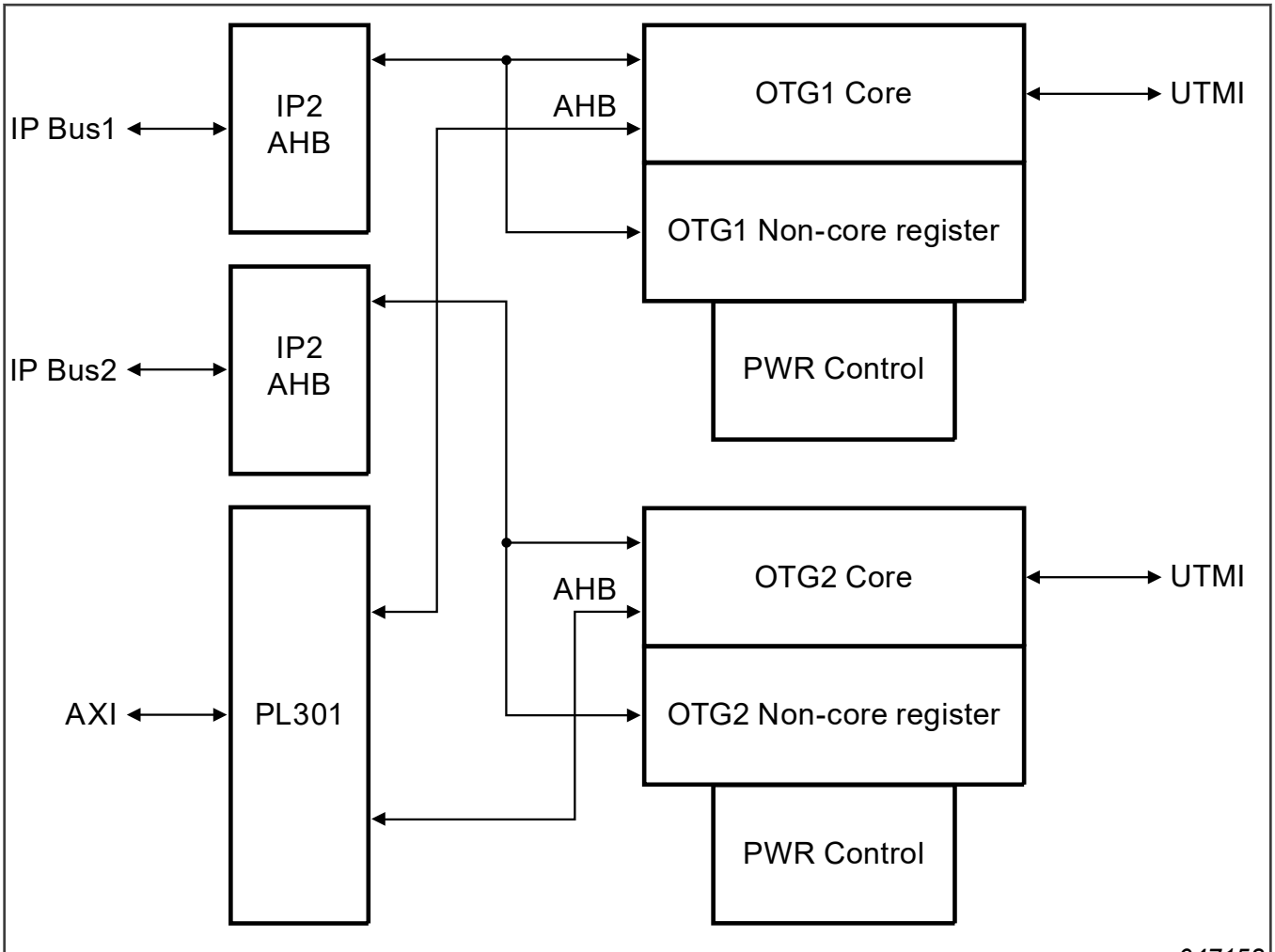


Figure 165. Block diagram

41.2.2 Features

- Includes USB 2.0 controller cores in this chip:
  - Controller core 0 is also named OTG1 core; its connected port is named OTG1 port.
  - Controller core 1 is also named OTG2 core; its connected port is named OTG2 port.
- Support for USB 2.0 controller core 0:
  - High-speed (HS), full-speed (FS), and low-speed (LS) OTG cores
  - HS, FS, and LS UTMI-compliant interface
  - HS, FS, and LS operations in Host mode (with UTMI transceiver)
  - HS and FS operations in Peripheral mode (with UTMI transceiver)
  - OTG signaling, session request protocol, and host negotiation protocol
  - Eight bidirectional endpoints
- USB 2.0 controller core 1:

- HS, FS, and LS OTG core
- HS, FS, and LS UTMI-compliant interface
- HS, FS, and LS operations in Host mode (with UTMI transceiver)
- HS and FS operations in Peripheral mode (with UTMI transceiver)
- Hardware support for OTG signaling, session request protocol, and host negotiation protocol
- Eight bidirectional endpoints

- Low-Power mode with local and remote wake-up capability
- Embedded DMA controller in each core

41.3 Functional description

These sections describe the functionality of the various building blocks of USB.

41.3.1 Submodule sections

41.3.1.1 USB 2.0 controller core 0

USB 2.0 controller 0 is an instantiation of an EHCI-compatible core that supports HS, FS, and LS operations.

In Host mode, this controller core supports HS, FS, and LS operations. In Device mode, it supports HS and FS operations.

41.3.1.1.1 Host mode

The controller supports direct connection of an HS, FS, or LS device with an on-chip UTMI transceiver. The system does not have a separate transaction translator block. The DMA and protocol engine blocks include the transaction translator function, which is normally associated with a USB 2.0 HS hub to support connection to FS and LS devices.

41.3.1.1.2 Peripheral (Device) mode

The following features are supported in this mode:

- Eight bidirectional endpoints
- HS or FS operation
- Support for HNP and SRP
- Remote wake-up capability

41.3.1.2 USB 2.0 controller core 1

The USB 2.0 controller core 1 is an instantiation of the EHCI-compatible core that supports the HS, FS, and LS operations USB 2.0 controller core 1 is identical as core 0.

In Host mode, this controller core supports the HS, FS, and LS operations. In Device mode, it supports HS and FS operations.

41.3.2 Host data structures

For detailed information, see the "Data Structures" chapter in *Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus, revision 1.0*.

41.3.3 Operation references

41.3.3.1 Host operational model

For detailed information, see the "Operational Model" chapter in *Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus, revision 1.0*.

41.3.3.2 Interrupts: host operational model

For detailed information about interrupt capability provided by the EHCI host controller hardware, see the "Interrupts" section in *Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus, revision 1.0*.

41.3.3.3 Host controller initialization

For detailed information, see the "Host Controller Initialization" section in *Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus, revision 1.0*.

41.3.4 EHCI deviations

For the purposes of a dual-role host or device controller with support for On-The-Go (OTG) applications, you must deviate from EHCI specification. Device and OTG operations are not specified in EHCI; therefore, the implementation supported in this core is proprietary.

The Host mode operation of the core is near EHCI compatible, with a few minor differences documented in this section. The following is a summary of these deviations.

Table 322. EHCI deviations

Deviations	Description
Embedded transaction translator	Allows direct attachment of the FS and LS devices in Host mode without the need for a companion controller (see Embedded transaction translator function).
Device operation	Generally disabled; therefore, Device mode is mostly transparent when in Host mode (exceptions are provided in the following sections). See Device operation for more information.
Embedded design interface	Does not have a PCI; therefore, the PCI configuration registers described in the EHCI specification are not applicable (see Embedded design interface for more information).
On-the-Go (OTG) operation	Includes an OTG for port 1.

41.3.4.1 Embedded transaction translator function

The OTG controller supports directly connected FS and LS devices, without requiring a companion controller, by including the capabilities of a USB 2.0 HS hub transaction translator.

Although there is no separate transaction translator block in the system, the transaction translator function that is normally associated with an HS hub is implemented within the DMA and protocol engine blocks. The embedded transaction translator function is an extension of the EHCI interface, but makes use of the standard data structures and operational models that already exist in the EHCI specification to support FS and LS devices.

41.3.4.1.1 Discovery: EHCI deviation

In a standard EHCI controller design, the EHCI host controller driver detects an FS or LS device by analyzing whether PORTSC1[PE] becomes 1 after the port reset operation.

PORTSC1[PE] becomes 1 in a standard EHCI controller implementation after the port reset operation and when the host and device negotiate an HS connection (that is, after chirp completes successfully).

Because this controller has an embedded transaction translator, PORTSC1[PE] always becomes 1 after the port reset operation, regardless of the host device chirp result. PORTSC1[PSPD] indicates the resulting port speed. Therefore, the standard EHCI host controller driver requires an alteration to handle directly connected FS and LS devices or hubs.

Table 323. Summary of EHCI

Standard EHCI	EHCI with embedded transaction translator
After PORTSC1[PE] becomes 1, following a connection and reset sequence, the device or hub is assumed to be in HS.	After PORTSC1[PE] becomes 1, following a connection and reset sequence, PORTSC*n* indicates the device or hub speed.
FS and LS devices are assumed to be downstream from an HS hub; therefore, all port-level controls are performed using the hub class through the nearest hub.	FS and LS devices can either be downstream from an HS hub or directly attached. When an FS or LS device is downstream from an HS hub, the port-level control is performed using the hub class through the nearest hub. When an FS or LS device is directly attached, the port-level control is accomplished using PORTSC*n*.
FS and LS devices are assumed to be downstream from an HS hub with HubAddr = X, where HubAddr > 0 and HubAddr is the address of the hub where the bus transitions from HS to either FS or LS (that is, a split target hub).	FS and LS devices can be either downstream from an HS hub with HubAddr = X [HubAddr > 0] or directly attached, where HubAddr = 0 and HubAddr is the address of the root hub where the bus transitions from HS to either FS or LS (that is, a split target hub is the root hub).

41.3.4.1.2 Operational model

The operational models support the behavior of the transaction translator (see the USB 2.0 specification) and support the EHCI controller moving packets between the system memory and a USB-HS hub. Because the embedded transaction translator exists within the host controller, there is no physical bus between the EHCI host controller driver and the USB FS and LS buses.

The following sections briefly discuss the operational model for how the EHCI and transaction translator operational models are combined without a physical bus existing between them. These sections assume that you are familiar with both the EHCI and USB 2.0 transaction translator operational models.

41.3.4.1.2.1 Microframe pipeline

The EHCI operational model uses the concept of H-frames and B-frames to describe the pipeline between the host (H) and the bus (B). The embedded transaction translator must use the same pipeline algorithms specified in the USB 2.0 specification for a hub-based transaction translator.

All periodic transfers begin at B-frame 0 (after SOF) and continue until the stored periodic transfers are complete. As an example of the microframe pipeline implemented in the embedded transaction translator, all periodic transfers that are tagged in EHCI to execute in H-frame 0 are ready to execute on the bus in B-frame 0.

When programming the S-mask and C-mask fields in the EHCI data structures to schedule periodic transfers for the embedded transaction translator, the EHCI host controller driver must follow the same rules specified in EHCI for programming those fields for downstream hub-based transaction translators.

After periodic transfers are exhausted, any stored asynchronous transfer is moved. Asynchronous transfers are opportunistic: they must execute whenever possible, and their operation is not tied to H-frame and B-frame boundaries. However, an asynchronous transfer cannot babble through the start of B-frame 0 (SOF).

41.3.4.1.2.2 Split-state machines

The start- and complete-split operational model differs from EHCI slightly because there is no bus medium between the EHCI controller and the embedded transaction translator.

A start- or complete-split operation occurs by requesting the split to the HS hub. The start-complete split operation is an internal operation to the embedded transaction translator. The following table summarizes the conditions where handshakes are emulated from an internal state instead of actual handshakes to the HS split bus traffic.

Table 324. Summary of handshake conditions

Condition	Emulate TT response
Start-split: All asynchronous buffers full	NAK
Start-split: All periodic buffers full	ERR
Start-split: Success for start of an asynchronous transaction	ACK
Start-split: Start periodic transaction	No handshake (OK)
Complete-split: Failed to find transaction in queue	Bus time out
Complete-split: Transaction in queue is busy	NYET
Complete-split: Transaction in queue is complete	Actual handshake from an LS or FS device

41.3.4.1.2.3 Asynchronous transaction scheduling and buffer management

Table 325. Asynchronous transaction scheduling and buffer management

USB 2.0 specification items	Implementation in the embedded transaction translator
11.17.3	Sequencing is provided and a packet-length estimator ensures that no FS or LS packet babbles exist in the SOF time.
11.17.4	Transaction tracking happens for two data pipes.

41.3.4.1.2.4 Periodic transaction scheduling and buffer management

Table 326. Periodic transaction scheduling and buffer management

USB 2.0 specification items	Implementation in the embedded transaction translator
11.18.6.[1-2]	• Terminate of pending start-split conditions: — End of frame (EOF), and not started in six microframes — Idle for more than four microframes • Terminate of pending complete-split conditions: — EOF — Idle for more than four microframes
11.18.[7-8]	• Transaction tracking for up to 16 data pipes • Complete-split transaction searching

Table continues on the next page...

Table 326. Periodic transaction scheduling and buffer management (continued)

USB 2.0 specification items	Implementation in the embedded transaction translator
	NOTE There is no data schedule mechanism for these transactions other than the microframe pipeline. The embedded TT assumes that the number of packets scheduled in a frame does not exceed the frame duration (1 ms); otherwise, it results in an undefined behavior.

41.3.4.1.2.5 Multiple transaction translators

HCSPARAMS[N_TT] indicates the maximum number of embedded transaction translators that USB currently supports.

41.3.4.2 Device operation

The coexistence of a device operational controller within the host controller has little effect on the EHCI compatibility for host operation, except as discussed in the following sections.

41.3.4.2.1 USBMODE

You cannot initialize the dual-role controller in Host or Device modes. Considering this, you must program USB Device Mode (USBMODE) for a host operation before the EHCI host controller driver can begin EHCI host operations.

41.3.4.2.2 Fields with nonzero values in USB core registers

Some of the reserved fields and reserved addresses in the capability and operational registers are used in Device mode. For Device mode, you must adhere to the following requirements:

- Always write 0 to all EHCI reserved fields (some of which are device fields) in the operational registers. This is an EHCI requirement of the device controller driver.
- Read operations by the host controller must properly mask EHCI reserved fields (some of which are device fields) because fields that are used exclusively for the device are undefined in Host mode.

41.3.4.2.3 SOF interrupt

The start of frame (SOF) interrupt for Device mode is shared as a free-running 125 µs interrupt for Host mode.

EHCI does not specify this interrupt, but it has been added for convenience and as a potential software time base. See USB Status (USBSTS) and Interrupt Enable (USBINTR) for more information.

41.3.4.3 Embedded design interface

This is an embedded USB host controller, defined by the EHCI specification, and thus does not implement the PCI configuration registers.

41.3.4.3.1 Frame adjust register

Because the optional PCI configuration registers are not included in this implementation, there is no corresponding bit-level timing adjustment (such as the one provided by the Frame Adjust register in the PCI configuration registers). The start of microframes is timed precisely to 125 µs using the transceiver clock as a reference clock. That is, a 60 MHz transceiver clock for 8-bit physical interfaces and full-speed serial interfaces, or a 30 MHz transceiver clock for 16-bit physical interfaces.

41.3.4.4 Miscellaneous variations from EHCI

41.3.4.4.1 Programmable physical interface behavior

This design supports multiple physical interfaces that can operate in different modes if you program the core with software-programmable physical interface modes. Software programmability allows you to select the physical interface part during the board design phase, instead of selection during the chip design phase. Port Status and Control (PORTSC1) provides the control fields for selecting the physical interface operating mode, a capability that EHCI does not define.

41.3.4.4.2 Discovery

41.3.4.4.2.1 Port reset

The port connect methods that EHCI specifies require you to set PORTSC1[PR] for a duration of 10 ms. Because of the complexity involved in supporting the attachment of devices that are not high speed, there are counters already present in the design that can count the 10 ms reset pulse to alleviate the software requirement needed to measure this duration. Perform the following procedure to establish the basic connection.

Table 327. Port reset

Step	Purpose	Programming	Notes
1	Change the port	Wait for the port change interrupt	A port connect change occurs to notify the host controller driver that a device is attached.
2	Reset the device	Write 1 to PORTSC1[PR]	—
3	Reset the device after 10 ms	Write 0 to PORTSC1[PR]	This step, which is necessary in a standard EHCI design, can be omitted with this implementation. If the EHCI host controller driver attempts to write 0 to PORTSC1[PR] when a reset is in progress, the write is ignored and the reset continues until completion.

A port enable change occurs to notify the host controller that the device in now operational, and at this point, the port speed is determined.

41.3.4.4.2.2 Port speed detection

After the port change interrupt indicates that a port is enabled, the EHCI stack must determine the port speed. Unlike the EHCI implementation, which reassigns the port owner for any device that does not connect at HS, this host controller supports direct attachment of non-HS devices.

Consider the following differences for port-speed detection:

- PORTSC1[PO] is read-only and always reads 0.
- PORTSC1[PSPD] provides the current operating speed of the port to the host controller driver.
- PORTSC1[HSP] signifies whether the port is in high speed or in full or low speed. This information is redundant with the aforementioned PORTSC1[PSPD].

41.3.4.4.3 Port Test Control mode

This mode behaves exactly as described in EHCI. An alternate host controller driver procedure is neither necessary nor supported.

41.3.5 Device data structures

This section defines the interface data structures used to communicate control, status, and data between the device controller driver (DCD) software and device controller.

The data structure definitions in this chapter support a 32-bit memory buffer address space. The interface consists of device queue heads and transfer descriptors.

NOTE

You must ensure that no interface data structure reachable from the device controller spans a 4 KB page boundary.

The data structures defined in this chapter are, from the device controller's perspective, a mix of read-only (RO) and read/write (R/W) fields. The device controller must preserve the RO fields on all data structure writes.

Figure 166 shows the organization of the device endpoint queue head (dQH).

Endpoint queue heads are arranged in an array, in a continuous area of memory indicated by Endpoint List Address (ENDPTLISTADDR). The even-numbered device queue heads in the list support receive endpoints (Out/Setup) and the odd-numbered queue heads in the list are used for transmit endpoints (In/Interrupt). That is, the device queue heads in the list with an even offset support receiver endpoints (Out) and the queue heads in the list with an odd offset are used for the transmit endpoint (In). The device controller indexes into this array based on the endpoint number received from the USB bus. All information necessary to respond to transactions (for all primed transfers) is contained in this list, enabling the device controller to readily respond to incoming requests without having to traverse a linked list.

NOTE

The endpoint queue head list must be aligned to a 2 KB boundary.

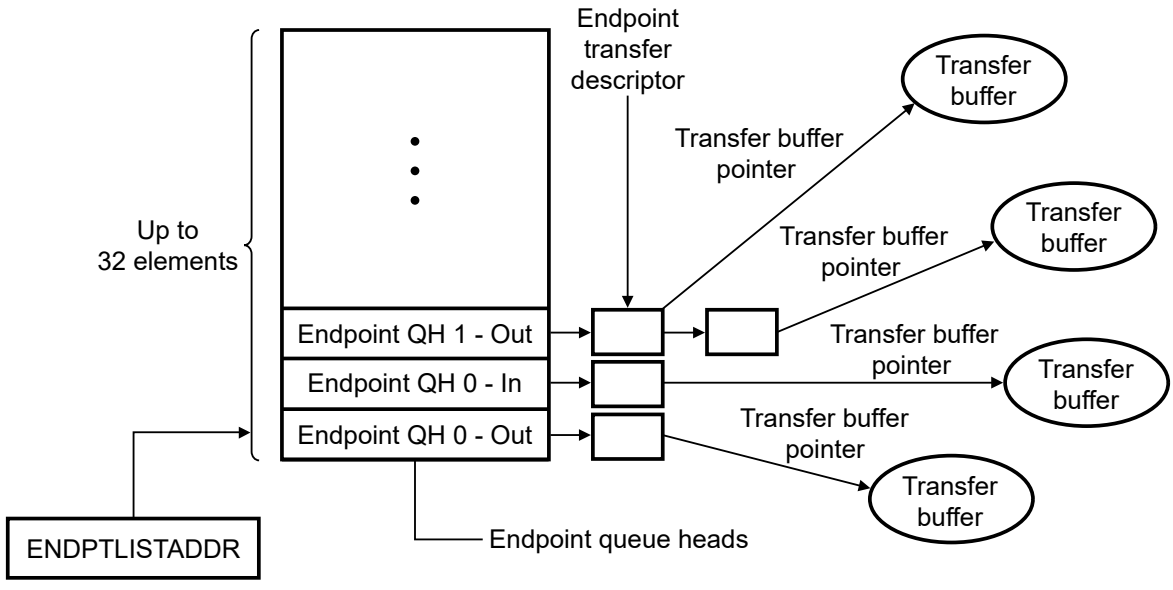


Figure 166. Endpoint queue head organization

41.3.5.1 Device endpoint queue head (dQH)

The dQH is where all transfers for a given endpoint are managed. It is a 48-byte data structure, but must be aligned on 64-byte boundaries.

During priming of an endpoint, the device transfer descriptor (dTD) is copied into the overlay area of the dQH, which starts at the next dTD pointer DWORD and continues through the end of the buffer pointer. After a transfer is complete, the dTD status DWORD is updated in the dTD indicated by the current dTD pointer. When a packet is in progress, the overlay area of the dQH is used as a staging area for the dTD so that the device controller can access the required information with minimal latency.

Table 328. Endpoint queue head (dQH) structure

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mult		ZLT	0		Maximum packet length											IOS	0														
Current dTD pointer																											0				
Next dTD pointer																											0				T[1]
0	Total bytes															IOC	0			MULTO		0		Status							
Buffer pointer (page 0)																				Current offset											
Buffer pointer (page 1)																				Reserved											
Buffer pointer (page 2)																				Reserved											
Buffer pointer (page 3)																				Reserved											
Buffer pointer (page 4)[1]																				Reserved											
Reserved																															
Setup buffer bytes 3…0																															
Setup buffer bytes 7…4																															

1. Transfer overlay starts at T and continues through the buffer pointer (page 4).

Host controller read/write Host controller read-only

41.3.5.1.1 Endpoint capabilities and characteristics

This DWORD specifies static information about the endpoint; in other words, this information does not change over the lifetime of the endpoint. The device controller software must not attempt to modify this information when the corresponding endpoint is enabled.

Table 329. Endpoint capabilities and characteristics

Field	Function
31-30 MULT	Multiplier Indicates the number of packets executed per transaction description. • 00 – Execute *n* transactions as demonstrated by the USB variable length packet protocol, where *n* is computed using dQH and dTD. • 01 – Execute one transaction • 10 – Execute two transactions • 11 – Execute three transactions **NOTE** You must write 00b to this field for non-ISO endpoints, and 01b, 10b, or 11b, as needed, for ISO endpoints.
29 ZLT	Zero-Length Termination Select Enables a zero-length packet to terminate transfers, where the total transfer length is a multiple of the maximum packet length. This field is not relevant for isochronous transfer types. • 0 – Enable a zero-length packet to terminate transfers equal to a multiple of the maximum packet length (default) • 1 – Disable the zero-length packet on transfers that are equal in length to a multiple maximum packet length
28-27 Reserved	Reserved for future use; must be 0.
26-16 Maximum Packet Length	Maximum Packet Length Corresponds to the maximum packet size of the associated endpoint (wMaxPacketSize). The maximum value this field can contain is 400h (1024).
15 IOS	Interrupt On Setup Used for control type endpoints to indicate whether USBSTS[UI] is configured in response to a setup being received.
14-0 Reserved	Reserved for future use; must be 0.

41.3.5.1.2 Transfer overlay-endpoint queue head

The seven DWORDS in the overlay area represent a transaction working space for the device controller.

The general operational model is that the device controller can detect whether the overlay area contains a description of an active transfer. If it does not contain an active transfer, it does not read the associated endpoint.

After an endpoint is ready, the device controller copies the dTD into the queue head overlay area. Until a transfer is expired, you must not write to the queue head overlay area or the associated transfer descriptor. After the transfer is complete, the device controller writes the results back to the original transfer descriptor and advances the queue.

See Device endpoint transfer descriptor (dTD) for a description of the overlay fields.

41.3.5.1.3 Current dTD pointer

The device controller uses the current dTD pointer to locate the transfer in progress. This word is for device controller (hardware) use only, and the DCD software must not modify it.

The following table describes the dTD pointer.

Table 330. Current dTD pointer

Bit	Description
31-5 Current dTD	Current dTD Points to the dTD that is represented in the transfer overlay area. The device controller modifies this field to the next dTD pointer during endpoint priming or queue advance.
4-0 Reserved	Reserved for future use; must be set to 0.

41.3.5.1.4 Setup buffer

The setup buffer is a dedicated storage for the 8-byte data that follows a setup PID.

NOTE

Each endpoint has a TX and an RX dQH associated with it. Only the RX queue head is used for receiving setup data packets.

Table 331. Multiple mode control (HCCPARAMS)

DWORD	Bits	Description
1	31-0	Setup Buffer 0 Contains bytes 3–0 of an incoming setup buffer packet. The device controller writes to it so you can read it.
2	31-0	Setup Buffer 1 Contains bytes 7–4 of an incoming setup buffer packet. The device controller writes to it so you can read it.

41.3.5.2 Device endpoint transfer descriptor (dTD)

The dTD informs the device controller of the location and quantity of data to be sent and received for a given transfer.

The DCD must not attempt to modify any field in an active dTD except the next like pointer, which you must modify only as described in Managing transfers with transfer descriptors.

Table 332 describes the dTD structure.

Table 332. dTD structure[1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Next link pointer																											0				T

Table continues on the next page...

Table 332. dTD structure[1] (continued)

0	Total bytes	IOC	0	MULTO	0	Status
Buffer pointer (page 0)				Current offset		
Buffer pointer (page 1)				0	Frame number	
Buffer pointer (page 2)				Reserved		
Buffer pointer (page 3)				Reserved		
Buffer pointer (page 4)				Reserved		

1. The shaded cells indicate host controller read/write fields and unshaded cells indicate host controller read-only fields.

Table 333 describes the dTD pointer.

Table 333. Next dTD pointer

Field	Function
31-5 Next Transfer Element Pointer	Next Transfer Element Pointer Contains the physical memory address of the next dTD to be processed. This field corresponds to memory address signals [31:5], respectively.
4-1 Reserved	Reserved for future use; must be 0.
0 T	Terminate Indicates to the device controller whether there are valid entries in the queue. • 0 – Pointer is valid (points to a valid transfer element descriptor) • 1 – Pointer is invalid

Table 334 describes the dTD token.

Table 334. dTD token

Field	Function
31 Reserved	Reserved for future use; must be 0.
30-16 Total Bytes	Total Bytes Specifies the total number of bytes to be moved with this transfer descriptor. This field is decremented by the number of bytes moved during the transaction, after the successful completion of the transaction.

Table continues on the next page...

Table 334. dTD token (continued)

Field	Function
	The maximum value you can store in the field is 5 × 4 KB (5000h). This is the maximum number of bytes that 5-page pointers can access. Although it is possible to create a transfer up to 20 KB, this assumes that the first offset into the first page is 0. When the offset cannot be predetermined, crossing past the fifth page can be guaranteed by limiting the total bytes to 16 KB. Therefore, the maximum recommended transfer is 16 KB (4000h). If the value of this field is 0 when the host controller fetches this transfer descriptor (and the active bit is 1), the device controller executes a zero-length transaction and retires the transfer descriptor. "In" transfers do not require that dTD be an even multiple of the Maximum Packet Length field in dQH. If you build such a transfer descriptor for an "In" transfer, the last transaction is always less than the value of the Maximum Packet Length field.
15 IOC	Interrupt On Complete Indicates whether USBSTS[UI] will be configured in response to the device controller being finished with this dTD.
14-12 Reserved	Reserved for future use; must be 0.
11-10 MULTO	Multiplier Override Can be used for transmit-ISO (that is, ISO-IN) to override the multiplier in the queue head. This field must be zero for all packet types that are not transmit-ISO. For example, if QH.multiplier = 3, the Maximum Packet Size field = 8, the Total Bytes field = 15, and the MULTO field = 0 (default), then three packets are sent: {Data2(8); Data1(7); Data0(0)}. If QH.multiplier = 3, the Maximum Packet Size field = 8, the Total Bytes field = 15, and the MULTO field = 2, then two packets are sent: {Data1(8); Data0(7)}. For maximum efficiency, you must calculate MULTO = greatest integer of (the Total Bytes field ÷ the Maximum Packet Size field). In cases where the Total Bytes field = 0, the MULTO field must be 1. For non-ISO and non-TX endpoints, you must write 0 to the MULTO field.
9-8 Reserved	Reserved for future use; must be 0.
7-0 Status	Status Used by the device controller to communicate individual command execution states back to the device controller software. This field contains the status of the last transaction performed on this dTD. The following are the bit encodings: • 0, 2, and 4 – Reserved • 3 – Transaction error • 5 – Data buffer error • 6 – Halted • 7 – Active

Table 335 describes the dTD buffer page pointer list.

Table 335. dTD buffer page pointer list

Field	Function
31-12 Buffer Pointer	Selects the page offset in memory for the packet buffer. Nonvirtual memory systems typically set the buffer pointers to a series of incrementing integers.
0, 11-0 Current Offset	Indicates the current offset that offsets into the 4 KB buffer where the packet will begin.
1, 10-0 Frame Number	Indicates the frame number in which a packet finishes. You typically use it to correlate relative completion times of packets on an ISO endpoint. The device controller writes to this field.

41.3.6 Device operational model

The function of the device operation is to transfer a request in the memory image to and from USB.

The device controller performs the data transfers (pointed to by the queue head) using a set of linked list transfer descriptors. The following sections explain the use of the device controller from the point-of-view of the DCD, and further describe how specific USB bus events relate to status changes in the device controller programmer's interface.

41.3.6.1 Device controller initialization

After hardware reset, the device is disabled until you write 1 to USBCMD[RS]. In the Disabled state, the pull-up on USB D+ is not active, which prevents an attach event from occurring. The minimum requirement is to have the queue heads setup for endpoint 0 before the device attach occurs.

After you enable the device, a USB reset occurs, followed by a setup packet arriving at endpoint 0. You must prepare a queue head so that the device controller can store the incoming setup packet.

Perform the following procedure to initialize a device:

1. Set Controller mode to Device mode in USB Device Mode (USBMODE).

   NOTE

   Transitioning from Host mode to Device mode requires a device controller reset before modifying USB Device Mode (USBMODE).

2. Allocate and initialize device queue heads in the system memory.

   At a minimum, initialize device queue heads 0 Tx and 0 Rx.

   NOTE

   You must initialize all device queue heads for control endpoints before the endpoint is enabled. Noncontrol device queue heads before the endpoint can be used.

   For information on device queue heads, see Device data structures.

3. Configure the ENDPOINTLISTADDR pointer.
4. Enable the microprocessor interrupt associated with the USB core.

   NXP recommends you to enable all device interrupts including USBSTS[UI], USBSTS[UEI], USBSTS[PCI], USBSTS[URI], and USBSTS[SLI].

   For a list of available interrupts, see Interrupt Enable (USBINTR) and USB Status (USBSTS) register tables.

5. Write 1 to USBCMD[RS]. After this field becomes 1 and the device is connected to a host, the host downstream port issues a bus reset. The DCD must monitor the reset event and adjust the software state as described in Bus reset.

NOTE

Endpoint 0 is designed as a control endpoint only; you do not need to configure it using Endpoint Control 0 (ENDPTCTRL0).

Because the first packet received is always a setup packet, it is not necessary to initially prime endpoint 0. The contents of the first setup packet require a response in accordance with the USB device framework command set.

41.3.6.2 Port state and control

From a chip or system reset, the device controller enters the powered state. A transition from the powered state to the attach state occurs when USBCMD[RS] = 1.

After receiving a reset on the bus, the port enters the defaultFS or defaultHS state in accordance with the reset protocol described in Appendix C.2 of *USB 2.0 Specification Rev. 2.0*.

The following state diagram depicts the state of a USB 2.0 device.

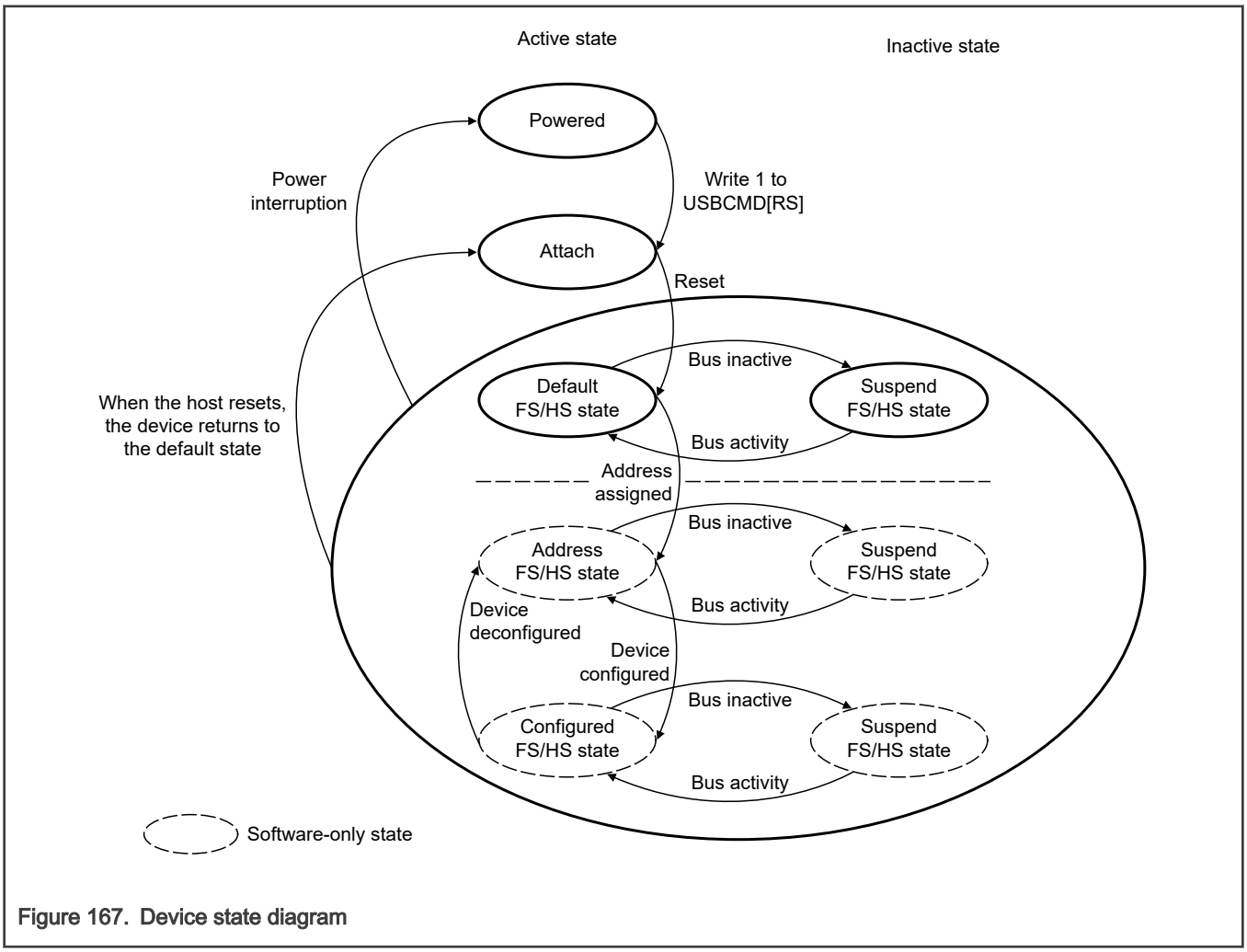


Figure 167. Device state diagram

The powered, attach, defaultFS, defaultHS, suspendFS, and suspendHS states are implemented in the device controller and communicated to the DCD using the following device controller state information fields:

- USBSTS[SLI]
- USBSTS[URI]

- USBSTS[PCI]
- PORTSC1[HSP]

It is the DCD's responsibility to maintain a state variable to differentiate between the defaultFS and defaultHS states, and the Address and Configured states. A change of state from Default to either Address or Configured is part of the enumeration process described in the device framework section of the USB 2.0 specification.

After entering the Address state, the DCD programs Device Address (DEVICEADDR).

Entry into the Configured state indicates that you have properly initialized all endpoints to be used in the operation of the device by programming the ENDPTCTRL*x* registers and initializing the associated queue heads.

41.3.6.2.1 Bus reset

The host uses a bus reset to initialize downstream devices.

When a bus reset is detected, the device controller renegotiates its attachment speed, resets the device address to 0, and notifies the DCD by interrupt (assuming USBINTR[URE] = 1). After a reset is received, all endpoints (except endpoint 0) are disabled, and the device controller cancels any primed transactions. The following table explains the concept of priming, and the tasks that DCD must perform when a reset is received.

Table 336. Bus reset

Step	Purpose	Programming
1	Clear all setup token semaphores	Read Endpoint Status (ENDPTSTAT) and write the same value back to ENDPTSTAT.
2	Clear all the endpoint-complete status fields	Read Endpoint Complete (ENDPTCOMPLETE) and write the same value back to ENDPTCOMPLETE.
3	Cancel primed status	Wait until all fields in Endpoint Prime (ENDPTPRIME) become 0, and then write FFFFFFFFh to Endpoint Flush (ENDPTFLUSH).

Read PORTSC1[PR] and ensure it is still active. A USB reset occurs for a minimum of 3 ms and the DCD must reach this point in the reset cleanup before the end of the reset occurs; otherwise, a hardware reset of the device controller is recommended. This is a rare scenario.

You can perform a hardware reset by writing 1 to USBCMD[RST]. By writing 0 to USBCMD[RS], a hardware reset causes the device to detach from the bus. Thus, the DCD must completely reinitialize the device controller after a hardware reset.

Additionally, you must free up all allocated dTDs because the device controller will no longer execute them. If this is the first time the DCD is processing a USB reset event, then it is likely that no dTDs are allocated.

At this time, the DCD can release the control back to the OS because no further changes to the device controller are permitted until a port change detect is indicated.

After a port change detect, the device reaches the default state and the DCD can read Port Status and Control (PORTSC1) to determine whether the device is operating in FS or HS mode. At this time, the device controller reaches its normal operating mode and DCD can begin enumeration according to *USB 2.0 specification chapter 9: Device Framework*.

NOTE

The device DCD may use the FS or HS mode information to determine the bandwidth mode of the device.

In some applications, it may not be possible to enable one or more pipes when in FS mode. There is no difference in DCD operations in FS and HS modes, except for the data rate issue.

41.3.6.2.2 Suspended state and Resume signal

Table 337. Operational models for Suspended state and Resume signal

Suspended	Resume
To conserve power, USB devices automatically enter Suspended state when the device observes no bus traffic for a specified period. When suspended, the USB device maintains any internal status, including its address and configuration. The attached devices must be prepared to suspend at any time when they are powered, regardless of whether they are assigned a nondefault address, are configured, or neither of the two conditions exist. Bus activity may cease because of the host entering a suspend mode of its own. In addition, a USB device must enter the Suspended state when the attached hub port is disabled. A USB device exits Suspend mode when there is bus activity. It may also request the host to exit Suspend mode or selectively suspend by using electrical signaling to indicate remote wake-up. The ability of a device to signal remote wake-up is optional. If the USB device is capable of remote wake-up signaling, the device must support the ability of the host to enable and disable this capability. When the device is reset, remote wake-up signaling must be disabled. The device controller moves into the Suspended state when suspend signaling is detected or activity is missing on the upstream port for more than a specific period. After the device controller enters the Suspended state, an interrupt notifies the DCD (assuming that PORTSC1[SUSP] is 1, meaning the device controller is suspended). The DCD response, when the device controller is suspended, is application-specific and may involve switching to a low-power operation. See the USB 2.0 specification for information about bus power limits in Suspended state.	If the device controller is suspended, its operation is resumed when an active signal is received on its upstream facing port. In addition, the device can signal the system to resume operation by forcing resume signaling to the upstream port by writing 1 to PORTSC1[FPR] when the device is in the Suspended state. Sending resume signal to an upstream port must cause the host to issue resume signaling and bring the suspended bus segment (one more device) back to the active condition. **NOTE** Before using resume signaling, the host must enable it via the set feature command defined in the device framework (chapter 9 of the USB 2.0 specification).

41.3.6.3 Managing endpoints

The USB 2.0 specification defines an endpoint (also called a device endpoint or an address endpoint) as a uniquely addressable portion of a USB device that can source or sink data in a communications channel between the host and the device.

The combination of the endpoint number and endpoint direction specifies the endpoint address.

The channel between the host and an endpoint at a specific device represents a data pipe. Endpoint 0 for a device is always a control type data channel used for device discovery and enumeration. Other types of endpoints that USB supports include bulk, interrupt, and isochronous. Each endpoint type has a specific behavior related to packet response and error handling.

The USB device controller hardware supports up to eight endpoint numbers.

Each endpoint direction is essentially independent, and you can configure it with differing behaviors in each direction. For example, the DCD can configure endpoint "1-In" as a bulk endpoint and endpoint "1-Out" as an isochronous endpoint. This helps to conserve the total number of endpoints required for device operation. The only exception is that control endpoints must use both directions on a single endpoint number to function as a control endpoint. Endpoint 0 is, for example, always a control endpoint and uses both directions.

Each endpoint direction requires a queue head allocated in memory. To support the eight endpoint numbers, 16 queue heads are required. The operation of an endpoint and use of queue heads are described later in this document.

See the USB 2.0 specification for more information about endpoint operations.

41.3.6.3.1 Endpoint initialization

After hardware reset, all endpoints except endpoint 0 are uninitialized and disabled. The DCD must configure and enable each endpoint by configuring ENDPTCTRL*x*.

Each 32-bit ENDPTCTRL*x* is split into an upper and a lower half. The lower half of ENDPTCTRL*x* is used to configure the receive or "Out" endpoint, and the upper half is used to configure the corresponding transmit or "In" endpoint. You must configure the control endpoints in the same way for both the upper and lower halves of ENDPTCTRL*x*; otherwise, the behavior is undefined. The following table shows the fields and corresponding values for device controller endpoint initialization.

Table 338. Device controller endpoint initialization: configuration settings

Step	Purpose	Programming
1	Reset the PID sequence.	Write 1 to ENDPTCTRL*x*[TXR] and ENDPTCTRL*x*[RXR].
2	Configure receive data toggle sequencing.	Write 0 to ENDPTCTRL*x*[TXI] and ENDPTCTRL*x*[RXI].
3	Select the endpoint type.	Configure ENDPTCTRL*x*[TXT] and ENDPTCTRL*x*[RXT].
4	Configure the endpoint stall condition.	Write 0 to ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS].

41.3.6.3.2 Stalling

The following are the occasions in which the device controller may need to return a STALL response to the host:

- In case of a functional stall: It is a condition that DCD sets, as described in the USB 2.0 device framework. This kind of a stall is only used on noncontrol endpoints, and can be enabled in the device controller by writing 1 to ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS], associated with the given endpoint and the given direction. In a functional stall condition, the device controller continues to return STALL responses to all transactions occurring on the respective endpoint and direction until DCD writes 0 to ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS].
- In case of a protocol stall: Unlike a function stall, it is used on control endpoints. The device controller automatically clears it at the start of a new control transaction (setup phase). When enabling a protocol stall, the DCD must enable ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS] (both directions) as a pair. A single write to ENDPTCTRL*x* can ensure that both ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS] become 1 at the same instant.

NOTE

Any write to ENDPTCTRL*x* in Operational mode must preserve ENDPTCTRL*x*[TXT] and ENDPTCTRL*x*[RXT], which means you must perform a read-modify-write operation.

Table 339. Device controller stall response matrix

USB packet	Value of ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS]	Effect on ENDPTCTRL*x*[TXS] and ENDPTCTRL*x*[RXS]	USB response
Setup packet that a noncontrol endpoint receives	N/A	None	STALL
In, Out, or Ping packet that a noncontrol endpoint receives	1	None	STALL
In, Out, or Ping packet that a noncontrol endpoint receives	0	None	ACK, NAK, or NYET
Setup packet that a control endpoint receives	N/A	Cleared	ACK
In, Out, or Ping packet that a control endpoint receives	1	None	STALL
In, Out, or Ping packet that a control endpoint receives	0	None	ACK, NAK, or NYET

41.3.6.3.3 Data toggle

Data toggle is a mechanism to maintain data coherency between the host and device for a given data pipe.

For more information about data toggle, see the USB 2.0 specification.

41.3.6.3.3.1 Data toggle reset

The DCD can reset ENDPTCTRL*x*[TXR] and ENDPTCTRL*x*[RXR] and cause the data toggle sequence to reset in the device controller by writing 1 to ENDPTCTRL*x*[TXR] and ENDPTCTRL*x*[RXR].

This is necessary only when configuring or initializing an endpoint, or returning from a STALL condition.

41.3.6.3.3.2 Data toggle inhibit

Writing 1 to ENDPTCTRL*x*[TXI] and ENDPTCTRL*x*[RXI] causes the device controller to ignore the data toggle pattern that is normally sent, and accepts all incoming data packets regardless of the data toggle state.

During normal operation, the device controller checks the DATA0 and DATA1 fields against the data toggle to determine whether the packet is valid. If the data PID does not match either ENDPTCTRL*x*[TXR] or ENDPTCTRL*x*[RXR] (that the device controller maintains for that endpoint), the data toggle is considered invalid. In this case, the device controller assumes that the packet is already received and discards the packet without reporting it to the DCD. To prevent the host controller from resending the same packet, the device controller responds to the error packet by acknowledging it with either an ACK or NYET response.

NOTE

This feature is for test purposes only; you must never use it during a normal device controller operation.

41.3.6.3.3.3 Priming transmit (TX) endpoints

Priming a transmit endpoint causes the device controller to fetch the device transfer descriptor (dTD) for the transaction that the dQH points to.

After the dTD is fetched, it is stored in the dQH until the device controller completes the transfer described by the dTD. Storing the dTD in the dQH allows the device controller to fetch the operating context needed to handle a request from the host, without the need to follow the linked list, starting at the dQH when the host request is received.

After the device has loaded the dTD, the leading data in the packet is stored in a FIFO in the device controller. This FIFO is split into virtual channels so that the leading data can be stored for any endpoint, with the maximum number of endpoints configured at the device synthesis time.

After a priming request is complete, Endpoint Status (ENDPTSTAT) indicates an endpoint state of primed transmit. For this, the device controller can respond to an "In" request from the host and meet the stringent bus turnaround time of HS USB.

Because only the leading data is stored in the device controller FIFO, the device controller must begin filling in leading data after the transaction starts. The FIFO must be sized to account for the maximum latency that the system memory bus can incur.

41.3.6.3.3.4 Priming receive (RX) endpoints

From the point of view of the DCD, priming receive endpoints is identical to priming transmit endpoints. Considering the device controller, the major difference in the operational model is that there is no data movement of the leading packet data because the data will be received from the host.

As part of the architecture, the FIFO for the receive endpoints is not partitioned into multiple channels like the transmit FIFO. Thus, the size of the receive FIFO does not scale with the number of endpoints.

41.3.6.4 Operational model for packet transfers

The host initiates all transactions on the USB bus, and in turn, the device must respond to any request from the host within the turnaround time stated in the USB 2.0 specification.

With USB 1.1 FS and LS rates, this turnaround time was significant and the USB 1.1 device controllers were architected in a way that enabled the device controller to access the main memory or interrupt a host protocol processor in order to respond to the USB 1.1 transaction. The architecture of the USB 2.0 device controller is different because the same methods do not meet the USB 2.0 HS turnaround time requirements from the increased clock rate.

A USB host sends requests to the device controller in an order that cannot be precisely predicted as a single pipeline, so it is not possible to prepare a single packet for the device controller to execute. However, the order of packet requests is predictable if the endpoint number and direction are considered. For example, if endpoint 3 (transmit direction) is configured as a bulk pipe, then the host is expected to send "In" requests to that endpoint. This device controller is architected in a way that enables it to prepare packets for each endpoint or direction in anticipation of the host request. The process of preparing the device controller to send or receive data in response to host-initiated transactions on the bus is referred to as priming the endpoint. This term is used throughout the chapter to describe the device controller operation so you can use priming to appropriately architect the DCD. Furthermore, the term flushing is used to describe the action of clearing a packet that is queued for execution.

41.3.6.4.1 Interrupt and bulk endpoint operational model

The behaviors of the device controller for interrupt and bulk endpoints are identical.

All valid "In" and "Out" transactions to bulk pipes handshake with a NAK unless the endpoint is primed. After the endpoint is primed, data delivery commences.

The device controller retires a dTD after the packets described in the transfer descriptor are completed. Each dTD describes "N" packets to be transferred according to the USB variable length transfer protocol. Equation 38, Table 340, and Table 341 describe how the device controller computes the number and length of packets that USB must send and receive. These vary according to the total number of bytes and maximum packet length. Also, Table 342 describes when TX-dTD and RX-dTD are considered complete.

With ZLT = 0, N = INT (number of bytes ÷ maximum packet length) + 1

With ZLT = 1, N = MAXINT (number of bytes ÷ maximum packet length)

N = MAXINT (number of bytes ÷ maximum packet length)

Equation 38. Formula for N

Table 340. Variable length transfer protocol example (ZLT = 0)

Bytes (dTD)	Maximum packet length (dQH)	N	P1	P2	P3
511	256	2	256	255	—
512	256	3	256	256	0
512	512	2	512	0	—

Table 341. Variable length transfer protocol example (ZLT = 1)

Bytes (dTD)	Maximum packet length (dQH)	N	P1	P2	P3
511	256	2	256	255	—
512	256	2	256	256	—
512	512	1	512	—	—

NOTE

You must write 00 to the MULT field in the dQH (see Device endpoint queue head (dQH)) for bulk, interrupt, and control endpoints.

Table 342. TX-dTD and RX-dTD completion status

TX-dTD is considered complete when:	RX-dTD is considered complete when:
All packets described in dTD are successfully transmitted. When this occurs, the total bytes in dTD become zero.	• All packets described in dTD are successfully received. When this occurs, the total bytes in dTD become zero. • A short packet (number of bytes < maximum packet length) is received. This is a successful transfer completion. DCD must check the total bytes in dTD to determine the number of bytes that are remaining. From the total bytes remaining in the dTD, the DCD can compute the actual bytes received. • A long packet is received: (number of bytes > maximum packet size) OR (total bytes received > total bytes specified). This is an error condition. The device controller discards the remaining packet and sets the Data Buffer Error bit in the Status field of the dTD (see Device endpoint transfer descriptor (dTD)). In addition, the endpoint is flushed and the USBERR interrupt becomes active.

After the successful completion of the packet(s) that dTD describes, the Active bit in the Status field of the dTD is cleared and the next pointer is followed when the Terminate field in dTD is clear. When the Terminate field is set, the device controller flushes the endpoint or direction and ceases operations for that endpoint or direction.

After an unsuccessful completion of a packet (in the case of a long packet), the dQH points to the dTD in error. To recover from this error condition, the DCD must properly reinitialize the dQH by clearing the Active bit and updating the next dTD pointer before attempting to reprime the endpoint.

NOTE

The device controller automatically retires all packet level errors, such as a missing handshake or CRC error. An interaction with the DCD is not required for handling such errors.

41.3.6.4.1.1 Interrupt and bulk endpoint bus response matrix

The following table shows the device controller's response to packets on interrupt and bulk endpoint, according to the device controller's state. In the table:

- Bus error refers to force bit stuff error.
- NYET or ACK: If the transfer descriptor has packets remaining (according to the USB variable length protocol), the response is ACK. Otherwise, the response is NYET.

A system error (SYSERR) should not occur when the latency FIFOs are correctly sized and the DCD is responsive.

Table 343. Interrupt and bulk endpoint bus response matrix

Token type	Stall	Not primed	Primed	Underflow	Overflow
Setup	Ignore	Ignore	Ignore	N/A	N/A
In	STALL	NAK	Transmit	Bus error	N/A
Out	STALL	NAK	Receive + NYET or ACK	N/A	NAK
Ping	STALL	NAK	ACK	N/A	N/A
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore

41.3.6.4.2 Control endpoint operation model

41.3.6.4.2.1 Setup phase

All requests to a control endpoint begin with a setup phase, followed by an optional data phase and a required status phase. The device controller always accepts the setup phase unless the setup lockout is engaged.

The setup lockout engages so that future setup packets are ignored. Lockout of setup packets ensures that when software is reading the setup packet stored in the queue head, that data is not written at the same time because this can potentially cause an invalid setup packet.

You can disable the setup lockout mechanism only one time at initialization (by writing 1 to USBMODE[SLOM]), and a new tripwire type semaphore ensures that the setup packet payload is extracted from the queue head without being corrupted by an incoming setup packet. Setup lockout is not necessary when using the tripwire. This is the preferred behavior because ignoring repeated setup packets because of long software interrupt latency leads to a compliance issue.

Leaving the Setup Lockout mode as 0 results in pre-2.3 hardware behavior.

After receiving an interrupt and inspecting Endpoint Setup Status (ENDPTSETUPSTAT) to determine that a setup packet is received on a particular pipe, perform the following procedure.

Table 344. Setup phase: associated procedure

Step	Purpose	Programming	Notes
1	Clear the corresponding bit.	Write 1 to ENDPTSETUPSTAT[ENDPTSETUPSTAT].	—
2	—	Write 1 to USBCMD[SUTW].	—
3	Duplicate contents of the Setup Buffer field (in dTD) into a local software byte array.	—	—
4	—	Read USBCMD[SUTW].	If the field is 1, continue to the next step; if the field is 0, go back to step 2.
5	—	Write 0 to USBCMD[SUTW].	—
6	Process the setup packet using a local software byte array copy, and execute the status and handshake phases.	—	After receiving a new setup packet, the status and handshake phases may still be pending from a previous control sequence. These must be flushed and deallocated before linking a new status or handshake dTD for the most recent setup packet.

41.3.6.4.2.2 Data phase

Following the setup phase, the DCD must create a device transfer descriptor for the data phase and prime the transfer.

After priming the packet, the DCD must immediately verify (by reading Endpoint Setup Status (ENDPTSETUPSTAT)) that a new setup packet has not been received. A prime completes when the associated field in Endpoint Prime (ENDPTPRIME) is 0 and the associated field in Endpoint Status (ENDPTSTAT) is 1. If the associated field in Endpoint Prime (ENDPTPRIME) becomes 0 and the associated field in Endpoint Status (ENDPTSTAT) has not become 1, prime fails. This is only due to improper setup of the dQH, the dTD, or a setup arriving during the prime operation. If a new setup packet is indicated after the associated field in Endpoint Prime (ENDPTPRIME) becomes 0, then the transfer descriptor is freed and the DCD must reinterpret the setup packet.

If a setup arrives after the data stage is primed, the device controller automatically clears the prime status (Endpoint Status (ENDPTSTAT)) to enforce data coherency with the setup packet.

NOTE

- You must write 00 to the MULT field in the dQH (see Device endpoint queue head (dQH)) for bulk, interrupt, and control endpoints.
- Error handling of data phase packets is performed in the same way as bulk packets.

41.3.6.4.2.3 Status phase

Similar to the data phase, the DCD must create a transfer descriptor (with byte length equal to 0) and prime the endpoint for the status phase.

The DCD must also perform the same checks associated with ENDPTSETUPSTAT[ENDPTSETUPSTAT], as described in the data phase.

NOTE

- You must write 00 to the MULT field in the dQH (see Device endpoint queue head (dQH)) for bulk, interrupt, and control endpoints.
- Error handling of data phase packets is performed in the same way as bulk packets.

41.3.6.4.2.4 Control endpoint bus response matrix

The following table shows the device controller's response to packets on a control endpoint, according to the device controller's state. In the table:

- Bus error refers to force bit stuff error.
- NYET or ACK: If the transfer descriptor has packets remaining (according to the USB variable length protocol), the response is ACK. Otherwise, the response is NYET.
- System error (SYSERR) must never occur when the latency FIFOs are correctly sized and the DCD is responsive.

Table 345. Control endpoint bus response matrix

Token type	Endpoint state					Setup lockout
	Stall	Not primed	Primed	Underflow	Overflow	
Setup	ACK	ACK	ACK	N/A	SYSERR	—
In	STALL	NAK	Transmit	Bus error	N/A	N/A
Out	STALL	NAK	Receive + NYET or ACK	N/A	NAK	N/A
Ping	STALL	NAK	ACK	N/A	N/A	N/A
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore	Ignore

41.3.6.4.3 Isochronous endpoint operational model

Isochronous endpoints are used for real-time scheduled delivery of data, and their operational model is significantly different from the host throttled bulk, interrupt, and control data pipes.

The device controller accomplishes real-time delivery by using the following techniques:

- Exact MULT packets per microframe are transmitted and received. MULT is a two-bit field in the dQH (see Device endpoint queue head (dQH)). The variable length packet protocol is not used for isochronous endpoints.
- NAK responses are not used. Instead, zero-length packets are sent in response to an "In" request for unprimed endpoints. For unprimed RX endpoints, the response to an "Out" transaction is to ignore the packet within the device controller.
- Prime requests always schedule the transfer described in the dTD for the next microframe. If the ISO-dTD is still active after that frame, then the ISO-dTD is considered ready until the DCD executes or cancels it.

An EHCI-compatible host controller uses the periodic frame list to schedule data exchanges to isochronous endpoints. The operational model for Device mode does not use such a data structure. Instead, the same dTD used for control, bulk, and interrupt endpoints is also used for isochronous endpoints. The difference between bulk and ISO endpoints is in the handling of the dTD:

- Priming an ISO endpoint is a delayed operation so that an endpoint is primed only after an SOF is received. After the DCD writes to the prime bit, the bit is cleared as usual to indicate to software that the device controller completed priming the dTD for transfer. Internal to the design, the device controller hardware masks that prime start until the next frame boundary. This behavior is hidden from the DCD but occurs so that the device controller can match the dTD to a specific microframe.
- With isochronous endpoints, the transaction must fully complete within one microframe. After an ISO transaction is started in a microframe, it retires the corresponding dTD when MULT transactions occur or the device controller finds a fulfillment condition.

  The Transaction Error bit in the Status field of the dTD indicates a fulfillment error condition. When a fulfillment error occurs on the frame after the transfer fails to complete, the device controller force retires the ISO-dTD and moves to the next ISO-dTD.

NOTE

Partially completed packets cause fulfillment errors. If no activity occurs on a primed ISO-dTD, the transaction stays primed indefinitely. This means it is up to you to discard transmit ISO-dTDs that pile up from a failure of the host or to move the data.

- The data level error handling for ISO packets is different. When a CRC error occurs on a received packet, the packet is not retried in the same way as bulk and control endpoints. Instead, the CRC error is noted by setting the Transaction Error bit in the Status field of the dTD, and the data is stored as usual for the application software to manage. The following table shows how a retired ISO packet is managed.

Table 346. Conditions that cause transmit and receive packets to retire

Conditions for transmit packet retirement (all conditions must be fulfilled)	Conditions for receive packet retirement (all conditions must be fulfilled)
• The MULT counter reaches zero. • A fulfillment error occurs. The Transaction Error bit is set in the Status field of the dTD: number of packets occurred > 0 AND number of packets occurred < MULT. **NOTE** For TX-ISO, the MULT counter can be loaded with a lesser value in the dTD MULTO field in hardware versions 2.3 and later. If MULTO is 0, the MULT counter is initialized to the multiplier in the queue head.	• The MULT counter reaches zero. • A non-MDATA PID is received. The exit criteria is only valid in hardware version 2.3 or later. Before this version, any PID sequence that did not exactly match the MULTO field was flagged as a transaction error because of a PID mismatch or fulfillment error. • An overflow error occurs. This happens when: — The packet received is greater than the maximum packet length (the Data Buffer Error bit is set in the Status field of the dTD). — The packet received exceeds the total bytes allocated in the dTD (the Data Buffer Error bit is set in the Status field of the dTD). • A fulfillment error occurs (the Transaction Error bit is set in the Status field of the dTD): number of packets occurred > 0 AND number of packets occurred < MULT. • A CRC error occurs (the Transaction Error bit is set in the Status field of the dTD).

NOTE

For ISO, when a dTD is retired, the next dTD is primed for the next frame. For a continuous microframe-to-microframe operation, the DCD must ensure that the dTD linked-list is ahead of the device controller by at least two microframes.

41.3.6.4.3.1 Isochronous pipe synchronization

When it is necessary to synchronize an isochronous data pipe to the host, the microframe number (specified by USB Frame Index (FRINDEX)) can be used as a marker.

To cause a packet transfer to occur at a specific microframe number ("N"), the DCD must interrupt on SOF during frame N-1. If USB Frame Index (FRINDEX) = N-1, the DCD must write to the prime bit. The device controller primes the isochronous endpoint in microframe N-1 so that the device controller is able to execute delivery during microframe N.

NOTE

Priming an endpoint towards the end of microframe N-1 does not guarantee delivery in microframe N. The delivery may actually occur in microframe N+1 if the device controller does not have enough time to complete the prime before the SOF for packet N is received.

41.3.6.4.3.2 Isochronous endpoint bus response matrix

The following table shows the device controller's response to packets on isochronous endpoints, according to the device controller's state. In the table:

- Bus error refers to force bit stuff error.
- Null packet refers to zero-length packet.

Table 347. Isochronous endpoint bus response matrix

Token type	Stall	Not primed	Primed	Underflow	Overflow
Setup	Stall	Stall	Stall	N/A	N/A
In	Null packet	Null packet	Transmit	Bus error	N/A
Out	Ignore	Ignore	Receive	N/A	Drop packet
Ping	Ignore	Ignore	Ignore	Ignore	Ignore
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore

41.3.6.5 Managing queue heads

The dQH points to the linked list of transfer tasks; the dTD depicts each of these tasks. An area of memory that Endpoint List Address (ENDPTLISTADDR) points to contains a group of all dQHs in a sequential list, as shown in Figure 168. The even elements in the list of dQHs are used for receive endpoints (Out and Setup) and the odd elements are used for transmit endpoints (In and Interrupt). Device transfer descriptors are linked, head to tail, starting at the queue head and ending at the Terminate bit of the dTD. After the dTD retires, it is no longer a part of the linked list from the queue head. Therefore, software is required to track all transfer descriptors because pointers do not exist within the queue head after the dTD is retired (see Software link pointers).

In addition to the current and next pointers, and the dTD overlay examined in Operational model for packet transfers, the dQH also contains the following parameters for the associated endpoint: Multiplier (MULT), Maximum Packet Length, and Interrupt On Setup (IOS). The next section discusses the complete initialization of the dQH, along with these fields.

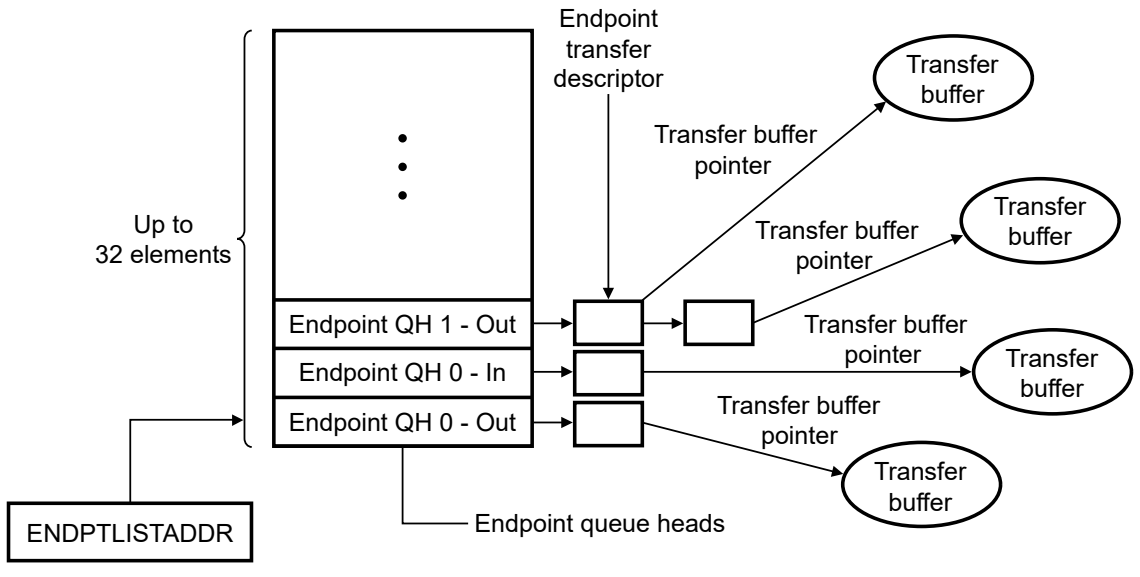


Figure 168. Endpoint queue head diagram

41.3.6.5.1 Queue head initialization

You must initialize a device queue head for each active endpoint. Perform the following procedure to initialize it.

Table 348. Queue head initialization

Step	Programming	Notes
1	Write to the wMaxPacketSize field as either the USB chapter 9 or application-specific protocol requires.	—
2	Write 0 to the MULT field in dQH for control, bulk, and interrupt endpoints.	Set the multiplier to 1, 2, or 3, for ISO endpoints, as the required bandwidth and in conjunction with the USB chapter 9 protocol. In FS mode, the MULT field in dQH can only be 1 for ISO endpoints.
3	Write 1 to the next Terminate field in dTD.	—
4	Write 0 to the Active bit in the Status field of dTD.	—
5	Write 0 to the Halt bit in the Status field of dTD.	—

NOTE

The DCD must only modify dQH if the associated endpoint is not primed and there are no outstanding dTDs.

41.3.6.5.2 Operational model for setup transfers

As discussed in Control endpoint operation model, the DCD gives special treatment to setup transfer. It does not use a dTD, but instead stores the incoming data from a setup packet in an 8-byte buffer within the dQH.

After receiving a notification of the setup packet, the DCD must manage the setup transfer as provided in the following procedure.

Table 349. Setup transfers: associated procedure

Step	Purpose	Programming	Notes
1	Copy setup buffer contents from dQH–RX to software buffer.	—	—
2	Acknowledge setup backup.	Write 1 to the corresponding bit in ENDPTSETUPSTAT[ENDPTSETUPSTAT].	• The acknowledgement must occur before continuing to process the setup packet. • After the acknowledgement has occurred, the DCD must not attempt to access the setup buffer in the dQH–RX. Only the local software copy must be examined.
3	Check for pending data or status dTDs from previous control transfers and flush, if any exist.	See Flushing and de-priming an endpoint.	—
4	—	Decode the setup packet, and prepare data phase (optional) and status phase transfers as required by either the USB chapter 9 or application-specific protocol.	—

NOTE

The device controller can receive setup packets before the completion of previous control transfers. Existing control packets in progress must be flushed and the new control packet must be completed.

41.3.6.6 Managing transfers with transfer descriptors

41.3.6.6.1 Software link pointers

The DCD software must maintain head and tail pointers to the linked list of dTDs for each respective queue head.

This is necessary because the dQH only maintains pointers to the currently working dTD and the next dTD to be executed. The operations described in the next section for managing the dTD assume that the DCD can use the head and tail of the dTD linked list. The following figure shows the software link pointers.

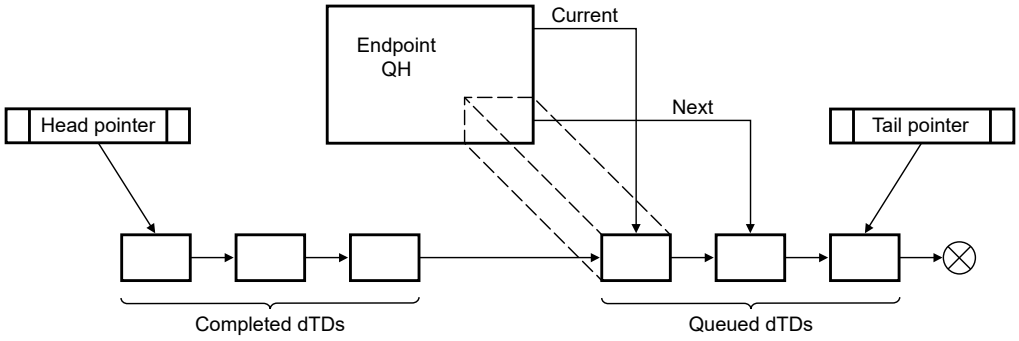


Figure 169. Software link pointers

NOTE

To conserve memory, the reserved fields at the end of the dQH can be used to store the head and tail pointers, but it still remains the responsibility of the DCD to maintain the pointers.

41.3.6.6.2 Building a transfer descriptor

Before executing a transfer from the linked list, you must build a dTD to describe the transfer.

As a prerequisite, allocate 8-DWORD dTD block of memory aligned to 8-DWORD boundaries. For example, bit address 4:0 is equal to 00000. Then perform the following procedure to build a dTD.

Table 350. Building a transfer descriptor: associated procedure

Step	Purpose	Programming
1	—	Write 0 to the first seven DWords.
2	—	Write 1 to the Terminate field in dTD.
3	—	Write the transfer size to the Total Bytes field in dQH.
4	—	Write 1 to the Interrupt On Complete (IOC) field in dTD, if desired.
5	Initialize the Status field.	Write 1 to the Active bit of the Status field and write 0 to all remaining Status bits.
6	Point to the start of the data buffer.	Fill in buffer pointer page 0 and the current offset.
7	—	Initialize buffer pointer page 1 through page 4 to be one greater than each of the previous buffer pointers.

41.3.6.6.3 Executing a transfer descriptor

To safely add a dTD, the DCD must perform the procedures listed in Table 351, based on the case. These procedures manage the event in which the device controller reaches the end of the dTD list at the same time a new dTD is added to the end of the list.

Determine whether the link list is empty by checking the DCD driver. This evaluates whether the pipe is empty (the internal representation of the linked list must indicate whether any packets are outstanding).

Table 351. Executing a transfer descriptor

Step	Case 1: If the link list is empty	Step	Case 2: If the link list is not empty
1	Write to the dQH next pointer and write 0 to the dQH Terminate bit as a single DWORD operation.	1	Add dTD to the end of linked list.
2	Clear the active and halt bits in dQH (in case they are set from a previous error).	2	Read the appropriate field in Endpoint Prime (ENDPTPRIME). If it is 1, it is done; otherwise, continue to read the field.
3	Prime the endpoint by writing 1 to the appropriate field in Endpoint Prime (ENDPTPRIME). You must check that the dTD and dQH content in external memory is updated before writing 1 to the field in Endpoint Prime (ENDPTPRIME).	3	Write 1 to USBCMD[ATDTW].
—	—	4	Read the appropriate status field in Endpoint Status (ENDPTSTAT) (store in a temporary variable for later use).
—	—	5	Read USBCMD[ATDTW]: • If the field is 0, go back to step 3. • If the field is 1, move forward to step 6.
—	—	6	Write 0 to USBCMD[ATDTW]: • If the status field read at step 4 is 1, the procedure is complete. • If the status field read at step 4 is 0, then go to step 1 of case 1.

41.3.6.6.4 Transfer completion

After you initialize a dTD and prime the associated endpoint, the device controller executes the transfer on the host-initiated request. The DCD is notified with a USB interrupt if the Interrupt On Complete (IOC) bit in the dTD is set. Alternately, the DCD can poll Endpoint Complete (ENDPTCOMPLETE) to find when the dTD was executed. After a dTD is executed, the DCD can also check the associated status bits to determine execution success or failure.

NOTE

Multiple dTDs can be completed in a single endpoint-complete notification. After clearing the notification, DCD must search the dTD linked list and retire all dTDs that are complete (the Active bit in the dTD Status field is cleared).

By reading the status fields of the completed dTDs, the DCD can determine whether the transfers completed successfully. The following combination of status bits determine the transfer completion success:

- Active = 0
- Halted = 0
- Transaction Error = 0

- Data Buffer Error = 0

If any combination other than the aforementioned one exists, the DCD must take proper action. Device error matrix indicates the transfer failure mechanisms.

In addition to checking the status bit, the DCD must read the Transfer Bytes field to determine the actual bytes transferred. When a transfer is complete, the total bytes transferred is decremented by the actual bytes transferred. For transmit packets, a packet is considered complete only after the actual byte count is zero. For receive packets, the host can send fewer bytes in the transfer according to the USB variable length packet protocol.

41.3.6.6.5 Flushing and de-priming an endpoint

The DCD must flush to de-prime another endpoint on a USB device reset or during a broken control transfer.

Some application-specific requirements may exist to stop transfers in progress. The DCD can perform the following procedure to stop a transfer in progress:

1. Write 1 to the corresponding field(s) in Endpoint Flush (ENDPTFLUSH).
2. Wait until all fields in Endpoint Flush (ENDPTFLUSH) become 0.

   NOTE

   This operation may take quite a long time, depending on the USB bus activity. You must not have this wait loop within an interrupt service routine.

3. Read Endpoint Status (ENDPTSTAT) to ensure that the corresponding fields are now 0 for all endpoints commanded to be flushed. The following is the reason for a flush failure, if the corresponding bits are still 1 after completion of step 2:

   In very rare cases, a packet is already in progress to a particular endpoint when the previous endpoint commands the flush operation by writing to Endpoint Flush (ENDPTFLUSH). A safeguard is in place to refuse the flush, to ensure that the packet in progress completes successfully. The DCD may need to repeatedly flush any endpoints that fail to flush by repeating steps 1–3 until each endpoint is successfully flushed.

41.3.6.6.6 Device error matrix

Table 352 summarizes packet errors that the device controller does not manage automatically.

Table 352. Device error matrix

Error	Direction	Packet type	Data buffer error bit	Transaction error bit
Overflow	RX	Any	1	0
ISO packet error	RX	ISO	0	1
ISO fulfillment error	Both	ISO	0	1

The device controller manages all errors on bulk, control, and interrupt endpoints, except for a data buffer overflow. However, for ISO endpoints, error packets are not retried and errors are tagged as indicated in Table 353.

Table 353. Error descriptions

Error	Description
Overflow	Indicates that the number of bytes received exceeded the maximum packet size or total buffer length.

Table continues on the next page...

Table 353. Error descriptions (continued)

Error	Description
	This error also sets the Halt bit in the dQH, and if there are dTDs remaining in the linked list for the endpoint, then those are not executed.
ISO packet error	Indicates a CRC error on the received ISO packet. Contents are not guaranteed to be correct.
ISO fulfillment error	Indicates that the host failed to complete the number of packets, defined in the dQH MULT field, within the given microframe. For scheduled data delivery, the DCD may need to readjust the data queue because a fulfillment error causes the device controller to cease data transfers on the pipe for one microframe. During the dead microframe, the device controller reports an error on the pipe and primes for the following frame.

41.3.6.7 Servicing interrupts

The interrupt service routine must consider that there are high-frequency, low-frequency, and error operations, and consider the order of interrupts accordingly.

41.3.6.7.1 High-frequency interrupts

You must manage high-frequency interrupts in the order shown in the following table. The most important of these is listed first because the DCD must acknowledge a setup buffer as promptly as possible.

Table 354. High-frequency interrupt events

Execution order	Interrupt	Action
1a	USB interrupt (see Endpoint Setup Status (ENDPTSETUPSTAT))	Copy the contents of the setup buffer and acknowledge the setup packet (as indicated in Figure 168). Process the setup packet according to USB 2.0 chapter 9 or the application-specific protocol.
1b	USB interrupt[1] (see Endpoint Complete (ENDPTCOMPLETE))	Handle the completion of dTD as indicated in Figure 168.
2	SOF interrupt	Perform the action that the application considers necessary. This interrupt may not be used in all applications.

1. It is common for multiple interrupts to stack up on any call to and during the interrupt service routine.

41.3.6.7.2 Low-frequency interrupts

Low-frequency interrupts do not occur often compared to high-frequency interrupts. You can handle low-frequency interrupts in any order.

Table 355. Low-frequency interrupt events

Interrupt	Action
Port change	Change software state information.
Sleep enable (suspend)	Change software state information. Perform low-power handling as necessary.
Reset received	Change software state information. Terminate pending transfers.

41.3.6.7.3 Error interrupts

Error interrupts are infrequent. You must place them last in the interrupt service routine.

Table 356. Error interrupt events

Interrupt	Action
USB error interrupt	This error is redundant because it combines a USB interrupt and an error status in the dTD. The DCD handles packet-level errors more appropriately by checking the dTD status field upon receipt of the USB interrupt (by using Endpoint Complete (ENDPTCOMPLETE)).
System error	Unrecoverable error that requires immediate reset of core; you must free up transfer buffers in progress and restart the DCD.

41.3.7 Modes of operation

USB includes two main modes of operation: Normal mode and Low-Power mode. Each USB OTG controller core can operate at high speed (480 Mbps), full speed (12 Mbps), and low speed (1.5 Mbps).

41.3.7.1 Normal mode

The OTG controller core can operate in Host mode and Device (Peripheral) mode.

Each USB controller core has its corresponding port, which can work in one or more interface modes.

Each controller supports only the interface type listed in this section. Selecting a different interface type in the PORTSC1[PTS] field results in an unpredictable behavior and may cause the system to hang.

- OTG1 port
  - This port supports on-chip UTMI transceiver only.
- OTG2 port
  - This port supports on-chip UTMI transceiver only.

41.3.7.2 Low-Power mode

Each USB controller core has a low-power mode (Suspend mode) to save power.

As the USB 2.0 specification describes, the device can go into the Suspend state after it sees a constant Idle state on the upstream-facing port (UFP). The OTG controller core enters Suspend mode after 3 ms of inactivity on the port when it is in Device Operation mode. The host controller does not suspend automatically; you can place it in Suspend mode.

Either the local Arm platform, the remote USB host, or a peripheral can initiate a wake-up sequence to resume USB communication. For more information on Suspend and Resume states, see Power control.

41.3.7.2.1 Power control

USB supports suspend and wake-up functionalities.

The power control block allows you to place the transceiver in Low-Power mode when the USB bus is in the Idle state, and supports local and remote wake-up to bring the transceiver out of Low-Power mode, when needed. Additionally, the power control block can wake up the Arm platform by generating an interrupt.

41.3.7.2.1.1 Entering low-power Suspend mode

In Host Operation mode, low-power Suspend mode is entered as follows:

1. If needed, set UTMIPHY_CFG1[COMMONONN] to power down most analog blocks and stop clocks during Suspend mode.
2. Write 0 to USBCMD[ASE] and USBCMD[PSE], and wait until USBCMD[AS] and USBCMD[PS] become 0.

3. Write 1 to PORTSC1[SUSP].
4. Write 1 to PORTSC1[PHCD].

NOTE

Step 3 shall be done in atomic operation. That is, interrupt should be disabled during this step.

For Device Operation mode, low-power Suspend mode is entered as follows:

1. If needed, set UTMIPHY_CFG1[COMMONONN] to power down most analog blocks and stop clocks during Suspend mode.
2. After the host drive is in the Idle state for 3 ms, an SLI interrupt is issued (USBSTS[SLI]).
3. Write 1 to PORTSC1[PHCD].

NOTE

Step 2 shall be done in atomic operation. That is, interrupt should be disabled during this step.

41.3.7.2.1.2 Wake-up events

The power control block monitors the USB bus when the USB core is in the USB Suspend state.

Depending on whether the controller is in Host or Device mode, the core monitors a number of wake-up conditions. If the related wake-up interrupt enable field is 1, an interrupt is generated to the Arm platform upon detection of a wake-up condition.

The USB wake-up interrupt also reactivates the Arm platform clocks if they are stopped during the Suspend state.

41.3.7.2.1.2.1 Host mode events

The host controller wakes up after any of the following events:

- Remote wake-up request: A peripheral can request the host to reactivate the bus by driving wake-up signaling on the DM or DP lines. The power control block sends a wake-up request to the USB core when a J-K transition on the DM or DP line is detected.
- Wake-up on overcurrent: If a wake-up on overcurrent is enabled (PORTSC1[WKOC] = 1), the power control block sends a wake-up request to the USB core when an overcurrent event is detected.
- Wake-up on disconnect: If wake-up on disconnect is enabled (PORTSC1[WKDC] = 1), the power control block sends a wake-up request to the USB core when a disconnection event is detected (a J-SE0 or K-SE0 transition on the DM or DP line).
- Wake-up on connect: If a wake-up on connect is enabled (PORTSC1[WKCN] = 1), the power control block sends a wake-up request to the USB core when a connection event is detected (an SE0-J or SE0-K transition on the DM or DP line).

41.3.7.3 Disable mode

To disable the USBPHY analog to achieve low power:

1. Write 0 to UTMIPHY_CFG2[DRVVBUS0] to disable the VBUS valid comparator.
2. Write 1 to UTMIPHY_CFG1[PHY_POR_SW] to reset all test registers and state machines in the PHY.

41.3.8 Clocking

The following are USB clock sources:

- AHB bus clock
- Peripheral access clock
- PL301 peripheral clock

See the "Clock Controller (CLKCTL)" chapter for clock setting, configuration, and gating information.

41.3.9 Reset

Table 357. USB high-speed PHY resets

Reset	Description
Chip reset	Resets the USB controller logic and registers to their default states.
USB controller reset (USBCMD[RST])	Resets the controller. The host or device controller writes 0 to this field after the reset process is complete. You cannot terminate the reset process early by writing 0 to this register.

41.3.10 Interrupts

41.3.10.1 USB core interrupts

The USB core uses a dedicated vector in the interrupt table. In the Interrupt section, you can find the vector numbers for each core.

With the exception of wake-up interrupts, the USB core controls all interrupt sources. See Interrupt Enable (USBINTR) for more information.

41.3.10.2 USB wake-up interrupts

The USB core has an associated wake-up interrupt. These interrupts are generated outside the USB core, but use the same vector as the corresponding USB core interrupt.

The power control blocks generate these interrupts, which run on a low-power, low-speed clock. The wake-up interrupt is designed to work even when the USB and Arm platform clocks are disabled so that a wake-up condition on the USB bus can reactivate the Arm platform clocks.

Because the wake-up interrupt is generated and cleared on a low-power, low-speed clock, this interrupt request responds very slowly to clear actions. For this reason, you must disable the wake-up interrupt to clear the request flag. Disabling the interrupt masks the request instantaneously because it is clocked by the Arm platform clock. You must wait for at least three low-power clock cycles before re-enabling this interrupt to allow sufficient time for the request flag to clear. Because this interrupt is used only during USB low-power modes, it is sufficient to enable the wake-up interrupt just prior to entering the USB Suspend mode.

41.4 External signals

Table 358. External signals

Signal	Description
OTGn_DN	DN OTG signal
OTGn_DP	DP OTG signal
OTGn_ID	ID signal
OTGn_OC	OTG external input for VBUS overcurrent detection
OTGn_PWR	To control PMIC to supply VBUS voltage
USB2_VBUS	USB 5 V signal
USB2_TXRTUNE	Transmitter resistor tune pin

See the chip-specific USB information for the external signals that this chip supports.

41.5 Initialization

See Host controller initialization and Device controller initialization for information on initializing the USB controller.

41.6 Register interface

USB configuration, control, and status registers are divided into the categories shown in the following table.

EHCI registers are listed along with device registers to show the complementary nature of host and device control.

Table 359. Core register sets

Offset	Register sets	Functional description
000h–07Ch	Identification registers	• Used to declare the target interface presence • Include the complete set of hardware configuration parameters
100h–124h	Static, read-only capability registers	• Define the software limits, restrictions, and capabilities of host and device controller implementation • Used as parameters for the host and device controller driver
080h–0FCh 140h–1FCh	Operational registers	• Dynamic control or status registers that can have read-only, read/write, or read/write-to-clear accesses • Used to control and monitor the operational state of the host and device controller

NOTE

- USB control registers support only DWORD (32-bit) accesses.
- You must not access reserved addresses in the memory map. If you do so, there is no bus error response.

41.6.1 Configuration, control, and status register sets

NOTE

Depending on implementation, "*x*" can have the following values: UOG1, UOG2.

Table 360. Device and host capability registers

Offset (hex)	Size (bytes)	Register name and mnemonic	Device mode	Host mode
000	4	Identification (ID)	O[1]	O
004	4	Hardware General (HWGENERAL)	O	O
008	4	Host Hardware Parameters (HWHOST)	X[2]	O
00C	4	Device Hardware Parameters (HWDEVICE)	O	X
010	4	TX Buffer Hardware Parameters (HWTXBUF)	O	O
014	4	RX Buffer Hardware Parameters (HWRXBUF)	O	O

Table continues on the next page...

Table 360. Device and host capability registers (continued)

Offset (hex)	Size (bytes)	Register name and mnemonic	Device mode	Host mode
018–07F	—	Reserved	—	—
080	4	General Purpose Timer 0 Load (GPTIMER0LD)	O	O
084	4	General Purpose Timer 0 Controller (GPTIMER0CTRL)	O	O
088	4	General Purpose Timer 1 Load (GPTIMER1LD)	O	O
08C	4	General Purpose Timer 1 Controller (GPTIMER1CTRL)	O	O
090	4	System Bus Configuration (SBUSCFG)	O	O
094–09F	—	Reserved	—	—
100	1	Capability Registers Length (CAPLENGTH)	O	O
101	—	Reserved	—	—
102	2	Host Controller Interface Version (HCIVERSION)	X	O
104	4	Host Controller Structural Parameters (HCSPARAMS)	X	O
108	4	Host Controller Capability Parameters (HCCPARAMS)	X	O
10C–11F	—	Reserved	—	—
120	2	Device Controller Interface Version (DCIVERSION)	O	X
122	2	Reserved	—	—
124	4	Device Controller Capability Parameters (DCCPARAMS)	O	X
128–13F	—	Reserved	—	—
140	4	USB Command (USBCMD)	O	O
144	4	USB Status (USBSTS)	O	O
148	4	Interrupt Enable (USBINTR)	O	O
14C	4	USB Frame Index (FRINDEX)	O	O
150	4	Reserved	—	—
154	4	Frame List Base Address (PERIODICLISTBASE)	X	O
	—	Device Address (DEVICEADDR)	O	X
158	4	Next Asynchronous Address (ASYNCLISTADDR)	X	O

Table continues on the next page...

Table 360. Device and host capability registers (continued)

Offset (hex)	Size (bytes)	Register name and mnemonic	Device mode	Host mode
	4	Endpoint List Address (ENDPTLISTADDR)	O	X
15C	4	Reserved	—	—
160	4	Programmable Burst Size (BURSTSIZE)	O	O
164	4	TX FIFO Fill Tuning (TXFILLTUNING)	X	O
168	4	Reserved	—	—
170	4	Reserved	—	—
178	4	Endpoint NAK (ENDPTNAK)	O	X
17C	4	Endpoint NAK Indication Enable (ENDPTNAKEN)	O	X
184	4	Port Status and Control (PORTSC1)	O	O
188–1A3	—	Reserved	—	—
1A4	4	On-The-Go Status and Control (OTGSC)	O	O
1A8	4	USB Device Mode (USBMODE)	O	O
1AC	4	Endpoint Setup Status (ENDPTSETUPSTAT)	O	X
1B0	4	Endpoint Prime (ENDPTPRIME)	O	X
1B4	4	Endpoint Flush (ENDPTFLUSH)	O	X
1B8	4	Endpoint Status (ENDPTSTAT)	O	X
1BC	4	Endpoint Complete (ENDPTCOMPLETE)	O	X
1C0 1C4 1C8 1CC 1D0 1D4 1D8 1DC	64	Endpoint Control 0 (ENDPTCTRL0) Endpoint Control 1 (ENDPTCTRL1) Endpoint Control 2 (ENDPTCTRL2) Endpoint Control 3 (ENDPTCTRL3) Endpoint Control 4 (ENDPTCTRL4) Endpoint Control 5 (ENDPTCTRL5) Endpoint Control 6 (ENDPTCTRL6) Endpoint Control 7 (ENDPTCTRL7)	O	X

1. O means that the register is available in Host and Device operation modes.
2. X means that the register is reserved in Host and Device operation modes.

41.6.2 Identification registers

USB uses identification registers to declare the target interface presence. They include hardware configuration parameters.

41.6.3 OTG operations

41.6.3.1 Register fields used for OTG operations

The behavior descriptions of Device mode and Host mode are provided in Register interface. However, for OTG operations, it is necessary to perform tasks independent of Controller mode.

NOTE

The only way to transit Controller mode out of Host or Device mode is by using USBCMD[RST]. Therefore, it is also necessary for the OTG tasks to be performed independent of a controller reset and Controller mode.

The following figure shows Controller mode.

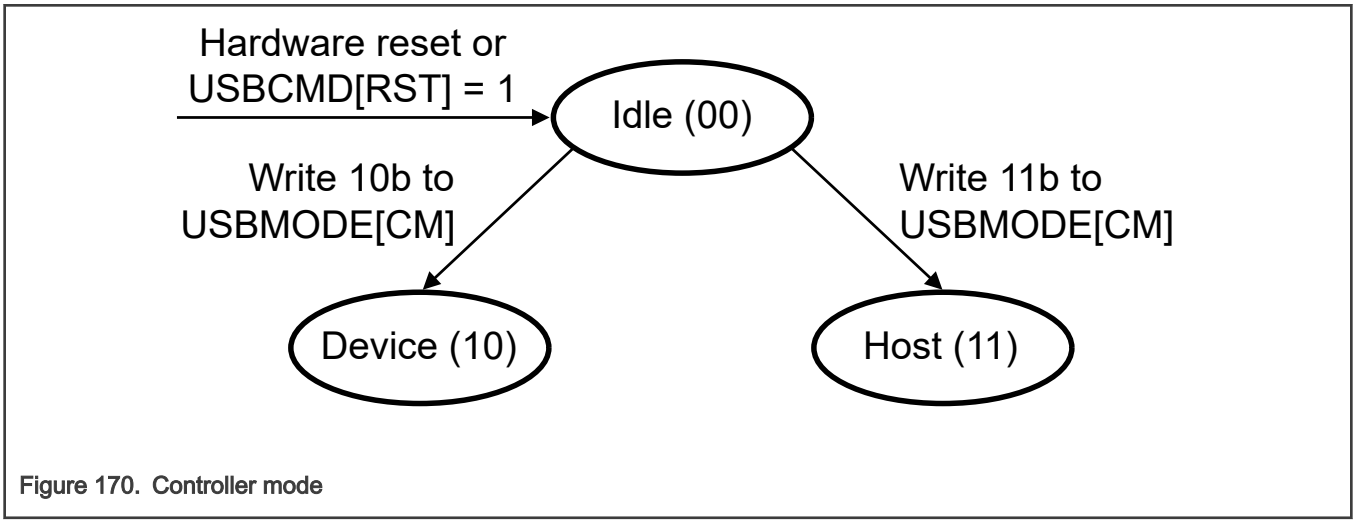


Figure 170. Controller mode

Following are the registers and register fields that are used for OTG operations, which are independent of Controller mode and are also not affected by a write to USBCMD[RST]:

- All identification registers
- All device and host capability registers
- All fields in On-The-Go Status and Control (OTGSC)

Additionally, the following fields of Port Status and Control (PORTSC1) are used:

- Physical interface serial select
- Physical interface data width
- Physical interface low power
- Physical interface wake signals
- Port indicators
- Port power

41.7 Register descriptions

There are two kinds of registers in the USB module: USB core (USBC) registers and USB noncore (USBNC) registers.

USBC registers are used to control USB core functions and are more independent of USB features. Each USB controller core has its own core registers.

USBNC registers are additional to USBC registers and are more dependent on USB features. For detailed descriptions of USBC registers, see Register interface. Section USBNC memory map describes only the USBNC registers.

NOTE

For reserved bits, preserve the value when writing (read the reset value, then write this value back).

41.7.1 USBC memory map

41.7.1.1 USBC register descriptions

41.7.1.1.1 USBC memory map

USB.USB_OTG1 base address: 4C10_0000h

USB.USB_OTG2 base address: 4C20_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Identification (ID)	32	R	E401_FA05
4	Hardware General (HWGENERAL)	32	R	0000_0015
8	Host Hardware Parameters (HWHOST)	32	R	1002_0001
C	Device Hardware Parameters (HWDEVICE)	32	R	0000_0011
10	TX Buffer Hardware Parameters (HWTXBUF)	32	R	8008_0B08
14	RX Buffer Hardware Parameters (HWRXBUF)	32	R	0000_0808
80	General Purpose Timer 0 Load (GPTIMER0LD)	32	RW	0000_0000
84	General Purpose Timer 0 Controller (GPTIMER0CTRL)	32	RW	0000_0000
88	General Purpose Timer 1 Load (GPTIMER1LD)	32	RW	0000_0000
8C	General Purpose Timer 1 Controller (GPTIMER1CTRL)	32	RW	0000_0000
90	System Bus Configuration (SBUSCFG)	32	RW	0000_0002
100	Capability Registers Length (CAPLENGTH)	8	R	40
102	Host Controller Interface Version (HCIVERSION)	16	R	0100
104	Host Controller Structural Parameters (HCSPARAMS)	32	R	0001_0011
108	Host Controller Capability Parameters (HCCPARAMS)	32	R	0000_0006
120	Device Controller Interface Version (DCIVERSION)	16	R	0001
124	Device Controller Capability Parameters (DCCPARAMS)	32	R	0000_0188
140	USB Command (USBCMD)	32	RW	0008_0000
144	USB Status (USBSTS)	32	RW	See section
148	Interrupt Enable (USBINTR)	32	RW	0000_0000
14C	USB Frame Index (FRINDEX)	32	RW	0000_0000
154	Device Address (DEVICEADDR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
154	Frame List Base Address (PERIODICLISTBASE)	32	RW	0000_0000
158	Next Asynchronous Address (ASYNCLISTADDR)	32	RW	0000_0000
158	Endpoint List Address (ENDPTLISTADDR)	32	RW	0000_0000
160	Programmable Burst Size (BURSTSIZE)	32	RW	0000_0000
164	TX FIFO Fill Tuning (TXFILLTUNING)	32	RW	0000_0808
178	Endpoint NAK (ENDPTNAK)	32	RW	0000_0000
17C	Endpoint NAK Indication Enable (ENDPTNAKEN)	32	RW	0000_0000
180	Configure Flag (CONFIGFLAG)	32	R	0000_0001
184	Port Status and Control (PORTSC1)	32	RW	1000_0000
1A4	On-The-Go Status and Control (OTGSC)	32	RW	See section
1A8	USB Device Mode (USBMODE)	32	RW	0000_0000
1AC	Endpoint Setup Status (ENDPTSETUPSTAT)	32	RW	0000_0000
1B0	Endpoint Prime (ENDPTPRIME)	32	RW	0000_0000
1B4	Endpoint Flush (ENDPTFLUSH)	32	RW	0000_0000
1B8	Endpoint Status (ENDPTSTAT)	32	R	0000_0000
1BC	Endpoint Complete (ENDPTCOMPLETE)	32	RW	0000_0000
1C0	Endpoint Control 0 (ENDPTCTRL0)	32	RW	0080_0080
1C4	Endpoint Control 1 (ENDPTCTRL1)	32	RW	0000_0000
1C8	Endpoint Control 2 (ENDPTCTRL2)	32	RW	0000_0000
1CC	Endpoint Control 3 (ENDPTCTRL3)	32	RW	0000_0000
1D0	Endpoint Control 4 (ENDPTCTRL4)	32	RW	0000_0000
1D4	Endpoint Control 5 (ENDPTCTRL5)	32	RW	0000_0000
1D8	Endpoint Control 6 (ENDPTCTRL6)	32	RW	0000_0000
1DC	Endpoint Control 7 (ENDPTCTRL7)	32	RW	0000_0000

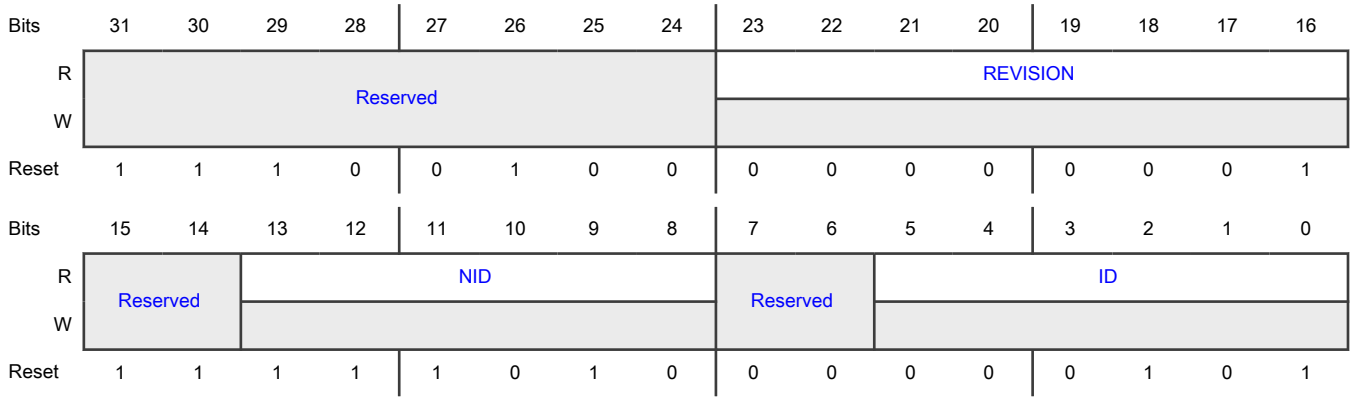
41.7.1.1.2 Identification (ID)

Identifies the USB 2.0 high-speed core and its revision.

Offset

Register	Offset
ID	0h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 REVISION	Revision Number Indicates the revision number of the controller core.
15-14 —	Reserved
13-8 NID	Complement Version Indicates the complement version of the ID.
7-6 —	Reserved
5-0 ID	Configuration Number Indicates the configuration number (05h) and signifies that the peripheral is USB 2.0 HS core.

41.7.1.1.3 Hardware General (HWGENERAL)

Indicates general hardware parameters as defined in system-level issues and core configuration.

NOTE

The reset value can vary from instance to instance. Therefore, for more accurate information, see field descriptions and ignore their reset values.

Offset

Register	Offset
HWGENERAL	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				SM		PHYM				PHYW		Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1

Fields

Field	Description
31-12 —	Reserved
11-10 SM	Serial Interface Mode Capability Indicates serial interface mode capability. 00 - No serial engine; always use parallel signaling 01 - Serial engine present; always use serial signaling for FS/LS 10 - Software programmable; reset to use parallel signaling for FS/LS 11 - Software programmable; reset to use serial signaling for FS/LS
9-6 PHYM	Transceiver Type Indicates the transceiver type. 0000 - UTMI/UMTI+ 0001 - ULPI DDR 0010 - ULPI 0011 - Serial only 0100 - Software programmable: reset to UTMI/UTMI+ 0101 - Software programmable: reset to ULPI DDR 0110 - Software programmable: reset to ULPI 0111 - Software programmable: reset to serial 1000 - IC-USB 1001 - Software programmable: reset to IC-USB 1010 - HSIC 1011 - Software programmable: reset to HSIC 1100 - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1111 - Reserved
5-4 PHYW	PHY Width Indicates the data width of the transceiver connected to the controller core. 00 - 8-bit wide data bus (software nonprogrammable) 01 - 16-bit wide data bus (software nonprogrammable) 10 - Reset to 8-bit wide data bus (software programmable) 11 - Reset to 16-bit wide data bus (software programmable)
3-0 —	Reserved

41.7.1.1.4 Host Hardware Parameters (HWHOST)

Indicates the host hardware parameters, the number of ports, and the mode supported.

Offset

Register	Offset
HWHOST	8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												NPORT			HC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31-4 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
3-1 NPORT	Number of Ports Indicates the number of downstream ports supported by the host controller. It is defined as NPORT + 1. **NOTE** A value of 000 indicates a single-port host controller.
0 HC	Host Capable Indicates whether Host Operation mode is supported. 0 - Not supported 1 - Supported

41.7.1.1.5 Device Hardware Parameters (HWDEVICE)

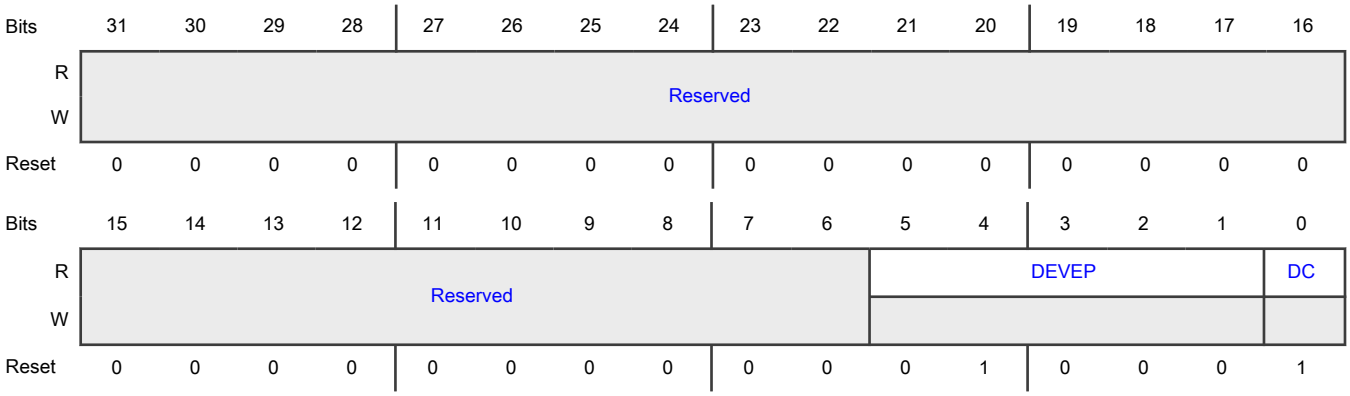
Indicates the device hardware parameters such as the device endpoint number and the mode supported.

NOTE
This register is only available in the OTG core.

Offset

Register	Offset
HWDEVICE	Ch

Diagram



Fields

Field	Description
31-6	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
5-1 DEVEP	Device Endpoint Number Indicates the device endpoint number.
0 DC	Device Capable Indicates whether Device Operation mode is supported. 0 - Not supported 1 - Supported

41.7.1.1.6 TX Buffer Hardware Parameters (HWTXBUF)

Indicates TX buffer hardware parameters, such as the default burst size and total buffer size.

Offset

Register	Offset
HWTXBUF	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TXCHANADD							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TXBURST							
W																
Reset	0	0	0	0	1	0	1	1	0	0	0	0	1	0	0	0

Fields

Field	Description
31-24 —	Reserved
23-16 TXCHANADD	TX Channel Add Indicates the TX FIFO buffer size.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	The TX FIFO buffer size is (2^TXCHANADD) × 4 bytes. This field becomes 08h, so the buffer size is 256 × 4 bytes. For the OTG controller operating in Device mode, this is the FIFO buffer size per endpoint. As the OTG controller has 8 TX endpoints, there are eight of these buffers. For the OTG controller operating in Host mode, or for host-only controller, the entire buffer memory is used as a single TX buffer. Therefore, there is only one of this buffer.
15-8 —	Reserved
7-0 TXBURST	TX Burst Indicates the default burst size for memory to TX buffer transfer. This is the reset value of fields in BURSTSIZE.

41.7.1.1.7 RX Buffer Hardware Parameters (HWRXBUF)

Indicates RX buffer hardware parameters, such as the default burst size and total buffer size.

Offset

Register	Offset
HWRXBUF	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXADD								RXBURST							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

Fields

Field	Description
31-16	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
15-8 RXADD	RX Add Indicates the RX FIFO buffer size. The total buffer size for all receive endpoints is (2^RXADD). RX buffer size is (2^RXADD) × 4 bytes. This field becomes 08h, so buffer size is 256 × 4 bytes. There is a single receive FIFO buffer in the USB controller. The buffer is shared for all endpoints for the OTG controller in Device mode.
7-0 RXBURST	RX Burst Indicates the default burst size for memory to RX buffer transfer. This is the reset value of fields in Programmable Burst Size (BURSTSIZE).

41.7.1.1.8 General Purpose Timer 0 Load (GPTIMER0LD)

Controls the load value of the count timer in General Purpose Timer 0 Controller (GPTIMER0CTRL).

Offset

Register	Offset
GPTIMER0LD	80h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								GPTLD							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPTLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
23-0 GPTLD	General Purpose Timer Load Value Indicates the general purpose timer load value. This value represents the time in microseconds minus 1 for the timer duration. The value of this field is loaded to GPTIMER0CTRL[GPTCNT], if GPTIMER0CTRL[GPTRST] is 1. For example, for a one millisecond timer, load 1000 - 1 = 999 (or 0003E7h). **NOTE** The maximum value is FFFFFFh or 16.777215 seconds.

41.7.1.1.9 General Purpose Timer 0 Controller (GPTIMER0CTRL)

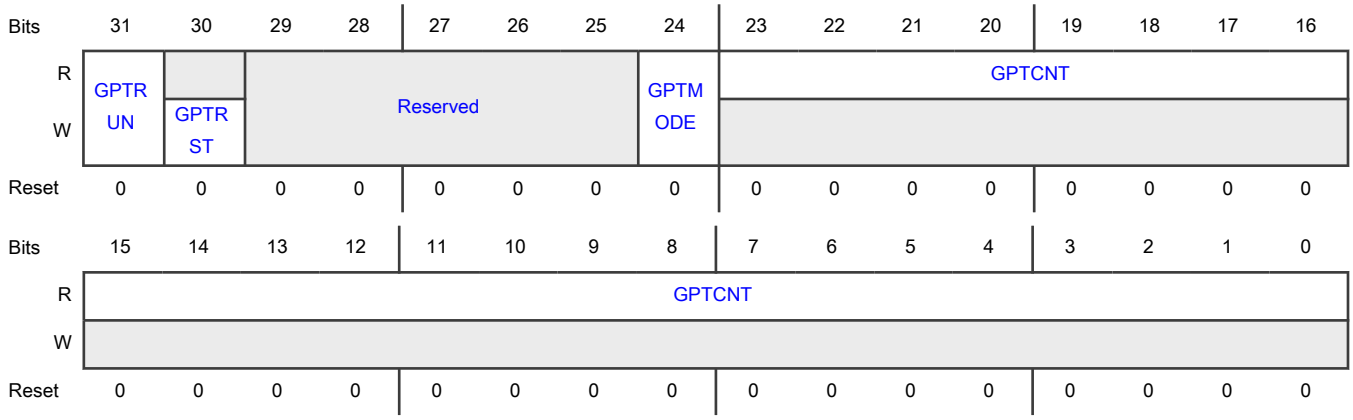
Contains the control for the general purpose countdown timer 0. A data field can be queried to determine the running count value. This timer has granularity on 1 µs and you can program it to a little over 16 seconds. There are two counter modes that are described in the register. When the timer counter value transitions to zero, an interrupt could be generated if enabled.

USBSTS[TI0] is the interrupt status field, and USBINTR[TIE0] is the interrupt enable field.

Offset

Register	Offset
GPTIMER0CTRL	84h

Diagram



Fields

Field	Description
31	General Purpose Timer Run

Table continues on the next page...

Table continued from the previous page...

Field	Description
GPTRUN	Specifies whether the general purpose timer is running or has stopped counting. GPTIMER0CTRL[GPTCNT] is not impacted when writing 1 or 0 to this field. 0 - Stop counting 1 - Run
30 GPTRST	General Purpose Timer Reset Loads the counter value from GPTIMER0LD[GPTLD]. 0 - No action 1 - Load counter value
29-25 —	Reserved
24 GPTMODE	General Purpose Timer Mode Specifies the general purpose timer mode. In One Shot mode, the timer counts down to zero, generates an interrupt, and halts until you reset the counter. In Repeat mode, the timer counts down to zero, generates an interrupt, and automatically reloads the counter value from GPTIMER1LD[GPTLD] to start again. 0 - One Shot mode 1 - Repeat mode
23-0 GPTCNT	General Purpose Timer Counter Indicates the count value of the countdown timer.

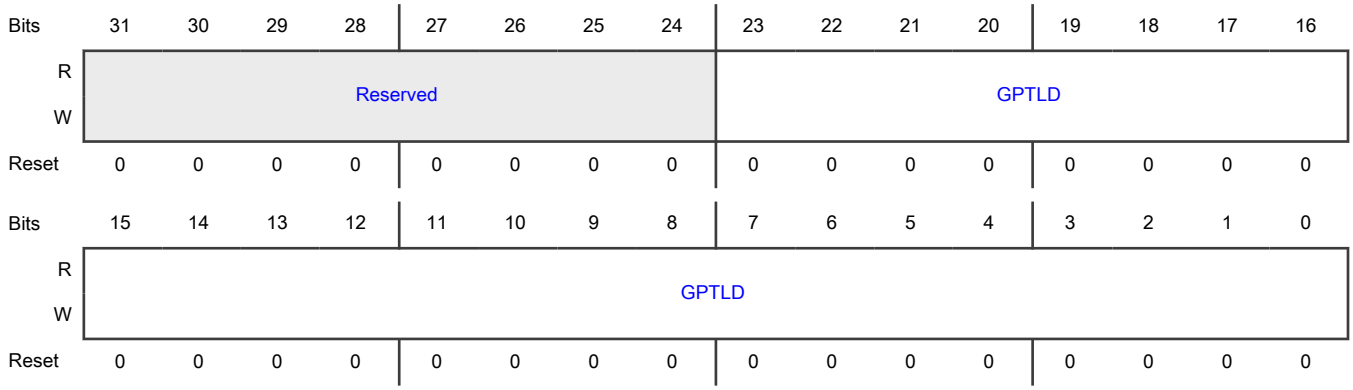
41.7.1.1.10 General Purpose Timer 1 Load (GPTIMER1LD)

Controls the load value of the count timer in General Purpose Timer 1 Controller (GPTIMER1CTRL).

Offset

Register	Offset
GPTIMER1LD	88h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 GPTLD	General Purpose Timer Load Value Indicates the general purpose timer load value. This value represents the time in microseconds minus 1 for the timer duration. The value of this field is loaded to GPTIMER0CTRL[GPTCNT] if GPTIMER0CTRL[GPTRST] is 1. For example, for a one millisecond timer, load 1000 - 1 = 999 (or 0003E7h). **NOTE** The maximum value is FFFFFFh or 16.777215 seconds.

41.7.1.1.11 General Purpose Timer 1 Controller (GPTIMER1CTRL)

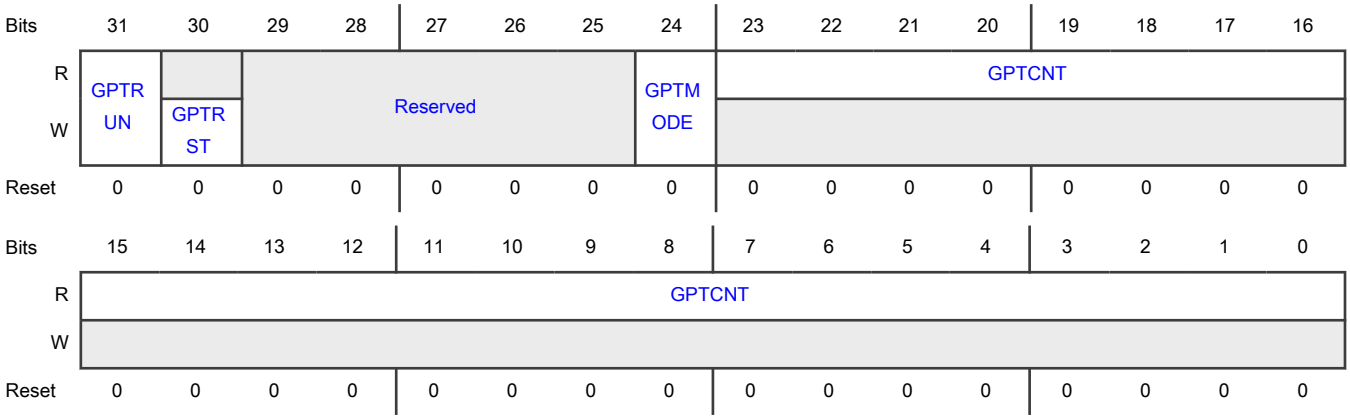
Contains the control for the general purpose countdown timer 1. A data field can be queried to determine the running count value. This timer has granularity on 1 µs and you can program it to a little over 16 seconds. There are two counter modes that are described in this register. When the timer counter value transitions to zero, an interrupt can be generated if it is enabled.

USBSTS[TI1] is the interrupt status field, and USBINTR[TIE1] is the interrupt enable field.

Offset

Register	Offset
GPTIMER1CTRL	8Ch

Diagram



Fields

Field	Description
31 GPTRUN	General Purpose Timer Run Specifies whether the general purpose timer is running or has stopped counting. GPTIMER0CTRL[GPTCNT] is not impacted when writing 1 or 0 to this field. 0 - Stop counting 1 - Run
30 GPTRST	General Purpose Timer Reset Loads counter value from GPTIMER0LD[GPTLD]. 0 - No action 1 - Load counter value
29-25 —	Reserved
24 GPTMODE	General Purpose Timer Mode Indicates the general purpose timer mode. In One Shot mode, the timer counts down to zero, generates an interrupt, and stops until you reset the counter. In Repeat mode, the timer counts down to zero, generates an interrupt, and automatically reloads the counter value from GPTLD bits to start again. 0 - One Shot mode 1 - Repeat mode
23-0 GPTCNT	General Purpose Timer Counter Indicates the count value of the countdown timer.

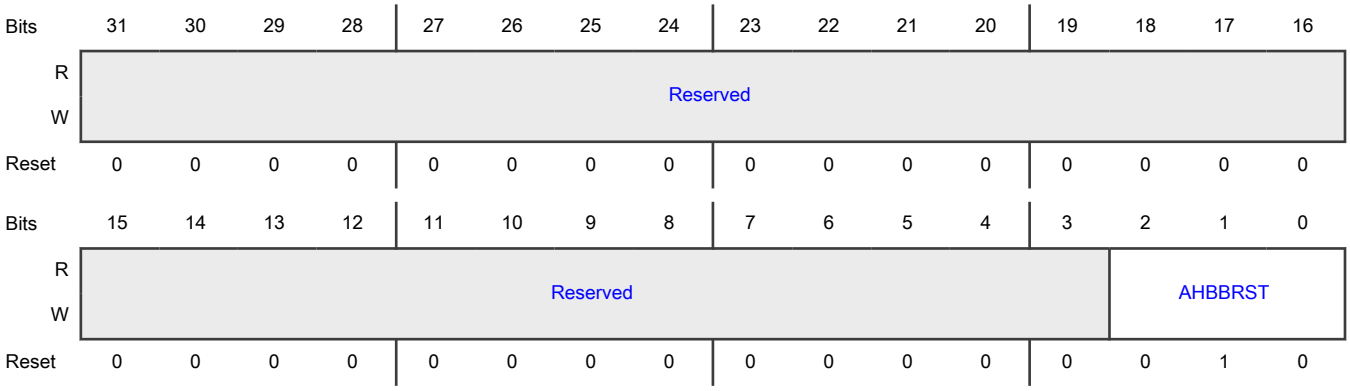
41.7.1.1.12 System Bus Configuration (SBUSCFG)

Defines the system bus configuration.

Offset

Register	Offset
SBUSCFG	90h

Diagram



Fields

Field	Description
31-3 —	Reserved
2-0 AHBBRST	AHB Controller Interface Burst Configuration Controls the AHB controller transfer type sequence (or priority). **NOTE** This register overrides Programmable Burst Size (BURSTSIZE) when the value of AHBBRST is not zero. 000 - Incremental burst of unspecified length only 001 - INCR4 burst, then single transfer 010 - INCR8 burst, INCR4 burst, then single transfer 011 - INCR16 burst, INCR8 burst, INCR4 burst, then single transfer 100 - Reserved; do not use 101 - INCR4 burst, then incremental burst of unspecified length 110 - INCR8 burst, INCR4 burst, then incremental burst of unspecified length 111 - INCR16 burst, INCR8 burst, INCR4 burst, then incremental burst of unspecified length

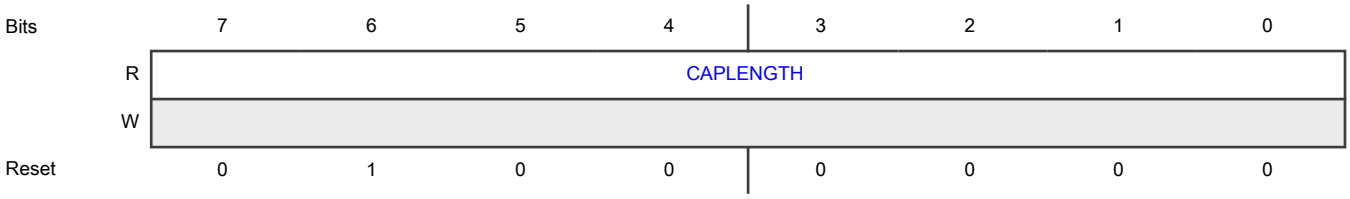
41.7.1.1.13 Capability Registers Length (CAPLENGTH)

Contains the address offset to the operational registers relative to the CAPLENGTH register.

Offset

Register	Offset
CAPLENGTH	100h

Diagram



Fields

Field	Description
7-0 CAPLENGTH	Capability Length Indicates the capability length. This field is used as an offset to add to the register base to find the beginning of the operational register. The default value is 40h.

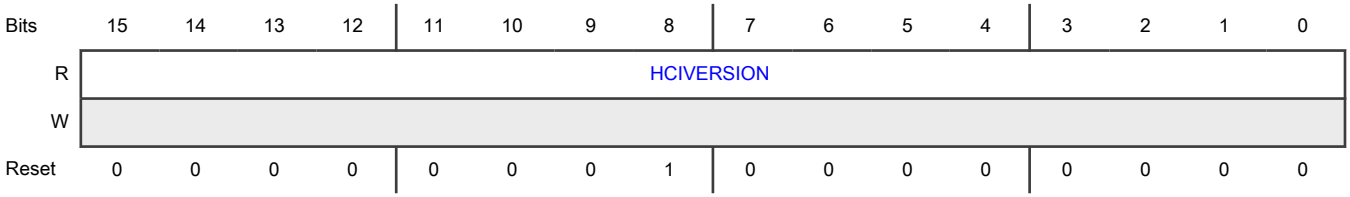
41.7.1.1.14 Host Controller Interface Version (HCIVERSION)

Contains a BCD encoding of the EHCI revision number that this host controller supports. The most significant byte of this register represents a major revision and the least significant byte indicates the minor revision.

Offset

Register	Offset
HCIVERSION	102h

Diagram



Fields

Field	Description
15-0 HCIVERSION	Host Controller Interface Version Number Indicates the host controller interface version number. The default value is 10h, which specifies EHCI rev1.0.

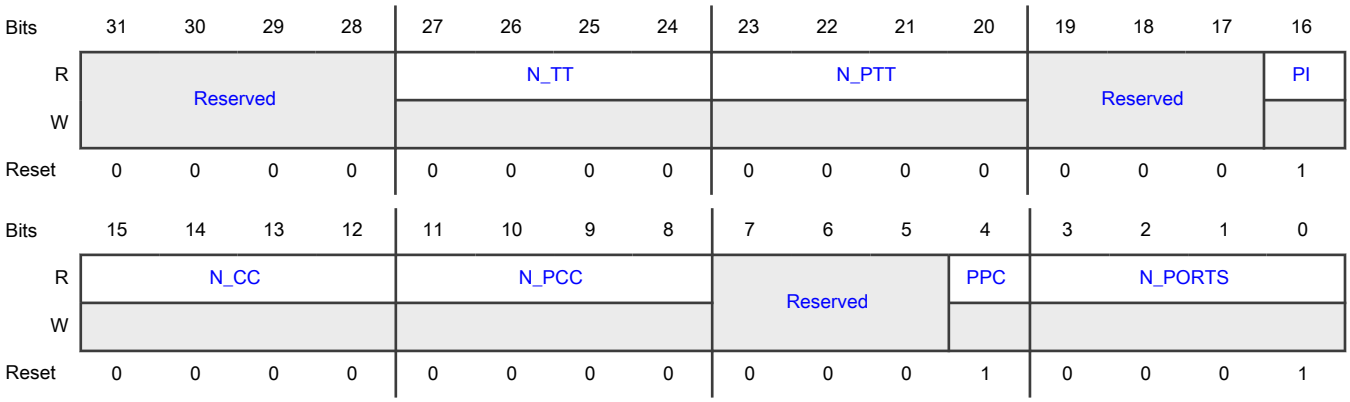
41.7.1.1.15 Host Controller Structural Parameters (HCSPARAMS)

Indicates the port steering logic capabilities of the host control structural parameters (HCSPARAMS).

Offset

Register	Offset
HCSPARAMS	104h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-24 N_TT	Number of Transaction Translators Indicates the number of embedded transaction translators associated with the USB 2.0 host controller. The value of this field is 0001b for multi-port host, and 0000b for single-port host. The default value of this field is 0000b.
23-20 N_PTT	Number of Ports Per Transaction Translator Indicates the number of ports assigned to each transaction translator within the USB 2.0 host controller. The value of this field is equal to that of N_PORTS for a multi-port host, and 0000b for a single-port host. The default value of this field is 0000b.
19-17 —	Reserved
16 PI	Port Indicators (P_INDICATOR) Indicates whether the ports support port indicator control.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If this field is 1, the port status and control registers include an R/W field for controlling the state of the port indicator. The value of this field is 1b in all controller cores.
15-12 N_CC	Number of Companion Controller Indicates the number of companion controllers associated with this USB 2.0 host controller. The value of this field is 0000b in all controller cores. If this field is 0, there is no internal companion controller and port-ownership hand-off is not supported. If this field is 1, internal companion controllers exist and port-ownership hand-offs are supported. 0000 - No internal companion controller exists 0001 - Internal companion controllers exist
11-8 N_PCC	Number of Ports Per Companion Controller Indicates the number of ports supported per internal companion controller. This field is used to indicate the port routing configuration to the system software. For example, if N_PORTS has a value of 6 and N_CC has a value of 2, then N_PCC could have a value of 3. The convention is that the first N_PCC ports are assumed to be routed to companion controller 1, the next N_PCC ports to companion controller 2, and so on. In the previous example, N_PCC could have been 4, where the first 4 are routed to companion controller 1 and the last 2 are routed to companion controller 2. The value of this field must be consistent with that of N_PORTS and N_CC. The value of this field is 0000b in all controller cores.
7-5 —	Reserved
4 PPC	Port Power Control Indicates whether the host controller implementation includes port power control. The value of this field affects the functionality of PORTSC1[PP] in each port status and control register. 0 - No port power switches 1 - Port power switches exist
3-0 N_PORTS	Number of Ports Indicates the number of downstream ports. This field indicates the number of physical downstream ports implemented on this host controller. The value of this field determines the number of port registers that are addressable in the operational register. The range of valid values is 1h–Fh. A zero in this field is undefined. The value of this field is always 0001b because all controller cores are single-port hosts.

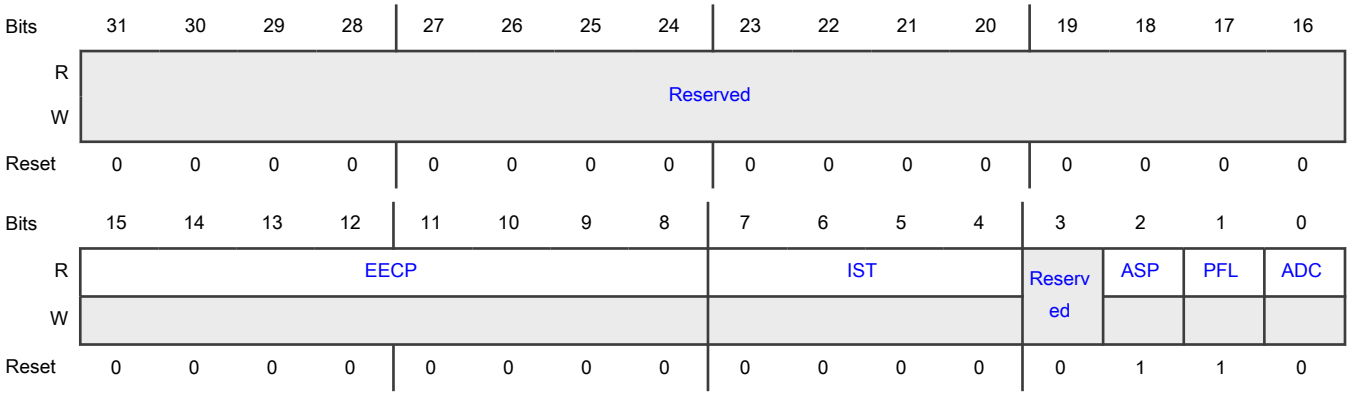
41.7.1.1.16 Host Controller Capability Parameters (HCCPARAMS)

Identifies multiple mode control (time-base bit functionality) and addressing capability.

Offset

Register	Offset
HCCPARAMS	108h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 EECP	EHCI Extended Capabilities Pointer Indicates the existence of a capabilities list. A value of 00h indicates that no extended capabilities are implemented. A nonzero value in this field indicates the offset of the first EHCI extended capability in the PCI configuration space. The pointer value must be 40h or greater (if implemented) to maintain the consistency of the PCI header defined for this class of device. **NOTE** The value of this field is 00h in all controller cores.
7-4 IST	Isochronous Scheduling Threshold Indicates, relative to the current position of the executing host controller, where you can reliably update the isochronous schedule. When bit[7] of this field is 0, the value of the least significant three bits indicates the number of microframes during which a host controller can hold a set of isochronous data structures (one or more) before flushing the state. When bit[7] is 1, then the host software assumes the host controller may cache an isochronous data structure for an entire frame. The value of this field is 00h in all controller cores.
3 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 ASP	Asynchronous Schedule Park Capability Indicates whether the park feature for HS queue heads is supported. If this field is 1, then the host controller supports the park feature for high-speed queue heads in the asynchronous schedule. You can enable or disable this feature and set it to a specific level by using USBCMD[ASPE] and USBCMD[ASP]. **NOTE** ASP bit reset value: 00b for OTG controller core; 11b for host-only controller core.
1 PFL	Programmable Frame List Flag Indicates the frame list length. If this flag is 0, you must use a frame list length of 1024 elements with this host controller. You must write 0 to USBCMD[FS_2] and USBCMD[FS_1]. If this flag is 1, you can specify and use a smaller frame list and configure the host controller via USBCMD[FS_2] and USBCMD[FS_1]. The frame list must always be aligned on a 4 KB page boundary. This requirement ensures that the frame list is always physically contiguous. The value of this flag is 1 in all controller cores.
0 ADC	Addressing Capability Indicates 64-bit addressing capability. This field is 0b in all controller cores, and 64-bit addressing capability is not supported.

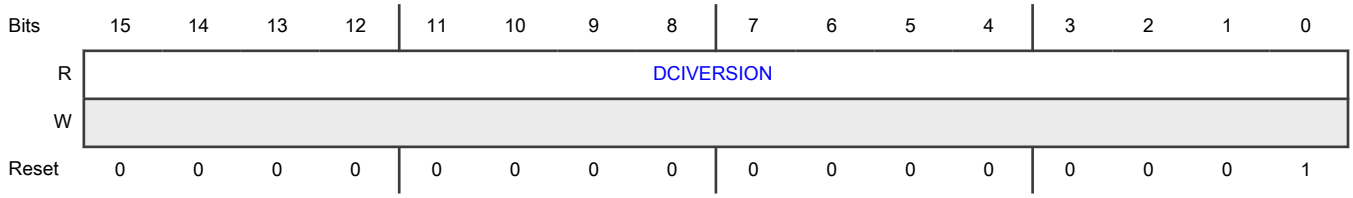
41.7.1.1.17 Device Controller Interface Version (DCIVERSION)

Indicates the 2-byte BCD encoding of the device controller interface version number.

Offset

Register	Offset
DCIVERSION	120h

Diagram



Fields

Field	Description
15-0 DCIVERSION	Device Controller Interface Version Number Indicates the device controller interface version number. The default value of this field is 01h, which specifies rev 0.1.

41.7.1.1.18 Device Controller Capability Parameters (DCCPARAMS)

Indicates the overall device capability of the controller.

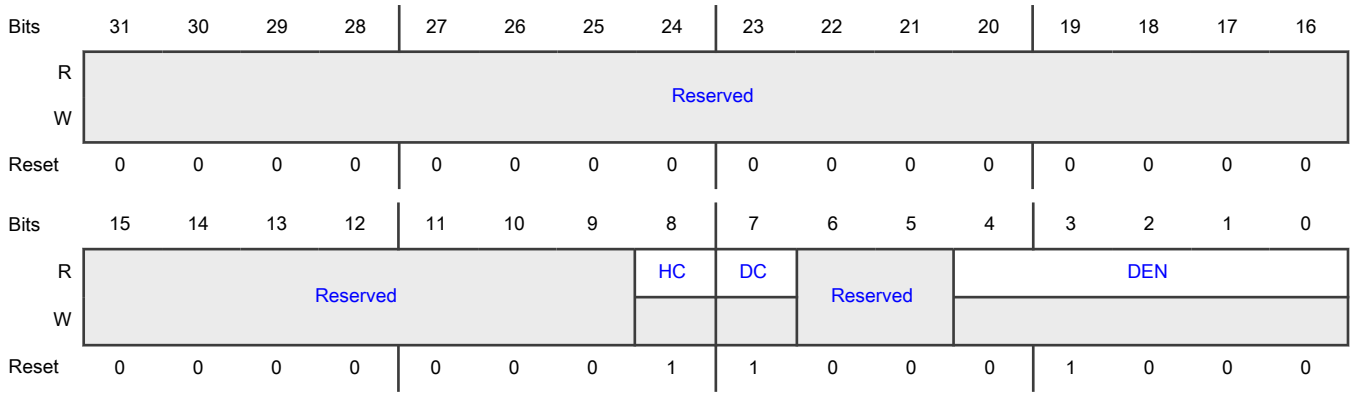
NOTE

This register is only available in the OTG controller core.

Offset

Register	Offset
DCCPARAMS	124h

Diagram



Fields

Field	Description
31-9	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
8 HC	Host Capable Indicates whether the controller is capable of operating as an EHCI-compatible USB 2.0 host controller. 0 - Not host capable 1 - Host capable
7 DC	Device Capable Indicates whether the controller is capable of operating as a USB 2.0 device. 0 - Not device capable 1 - Device capable
6-5 —	Reserved
4-0 DEN	Device Endpoint Number Indicates the number of endpoints built into the device controller. Valid values are 0–15. 0_0000 - Not device capable 0_0001 - Device capable

41.7.1.1.19 USB Command (USBCMD)

Indicates the command that the serial bus host or device controllers must execute. Writing to the register causes a command to be executed.

Offset

Register	Offset
USBCMD	140h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								ITC							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FS_2	ATDTW	SUTW	Reserved	ASPE	Reserved	ASP		Reserved	IAA	ASE	PSE	FS_1		RST	RS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23-16 ITC	Interrupt Threshold Control Sets the maximum rate at which the host or device controller issues interrupts. ITC contains the maximum interrupt interval, measured in microframes. The following values are the valid maximum interrupt intervals: 0000_0000 - Immediate (no threshold) 0000_0001 - 1 microframe 0000_0010 - 2 microframes 0000_0100 - 4 microframes 0000_1000 - 8 microframes 0001_0000 - 16 microframes 0010_0000 - 32 microframes 0100_0000 - 64 microframes
15 FS_2	Frame List Size 2 (Host mode only) Specifies the size of the frame list that controls which fields in USB Frame Index (FRINDEX) you must use for the frame list current index. This field is both readable and writable only if HCCPARAMS[PFL] is 1; otherwise, it is only readable. **NOTE** This field is made up of USBCMD bits 15, 3, and 2. 0b000 - 1024 elements (4096 bytes) (default value) 0b001 - 512 elements (2048 bytes) 0b010 - 256 elements (1024 bytes) 0b011 - 128 elements (512 bytes) 0b100 - 64 elements (256 bytes) 0b101 - 32 elements (128 bytes) 0b110 - 16 elements (64 bytes) 0b111 - 8 elements (32 bytes)
14 ATDTW	Add dTD TripWire (Device mode only) Works as a semaphore to ensure proper addition of a new dTD to an active (primed) endpoint's linked list. You write 1 and 0 to this field; however, the hardware can also write 0 to the field when the state machine is in the hazard region for which adding a dTD to a primed endpoint may go unrecognized.
13	Setup TripWire (Device mode only)

Table continues on the next page...

Table continued from the previous page...

Field	Description
SUTW	Works as a semaphore to ensure that the setup data payload of 8 bytes is extracted from a queue head by the DCD without being corrupted. If the Setup Lockout mode is off (USBMODE[SLOM] is 1), then there is a hazard when new setup data arrives while the DCD is copying the setup data payload from the queue head of a previous setup packet. You write 1 and 0 to this field; however, the hardware can also write 0 to the field if a hazard is detected.
12 —	Reserved
11 ASPE	Asynchronous Schedule Park Mode Enable Enables Park mode. If HCCPARAMS[ASP] is 1, then this field defaults to 1h, with R/W as its access type. Otherwise, the field is 0, with access type RO. **NOTE** ASPE bit reset value: '0b' for OTG controller . 0 - Disable 1 - Enable
10 —	Reserved
9-8 ASP	Asynchronous Schedule Park Mode Count Contains a count of the number of successive transactions that the host controller is allowed to execute from an HS queue head on the asynchronous schedule, before continuing traversal of the asynchronous schedule. If HCCPARAMS[ASP] is 1, then this field defaults to 3h, with R/W as the access type. Otherwise, the field defaults to 0, with access type RO. Valid values are 1h to 3h. You must not write 0 to this field when USBCMD[ASPE] is 1. Doing so leads to undefined behavior. This field is 3h for all controller cores.
7 —	Reserved
6 IAA	Interrupt on Async Advance Doorbell Acts as a doorbell for the host controller to issue an interrupt the next time it advances asynchronous schedule. You must write 1 to this field to ring the doorbell. After the host controller evicts all appropriate cached schedule states, it writes 1 to USBSTS[AAI]. If USBINTR[AAE] is 1, then the host controller asserts an interrupt at the next interrupt threshold.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	The host controller writes 0 to this field after it writes 1 to USBSTS[AAI]. You must not write 1 to this field when the asynchronous schedule is inactive. Doing so yields undefined results. This field is only used in Host mode. Writing 1 to this field in Device mode leads to undefined results.
5 ASE	Asynchronous Schedule Enable Determines whether the host controller skips processing the asynchronous schedule. If this field is 1, Next Asynchronous Address (ASYNCLISTADDR) is used to access the periodic schedule. If this field is 0 (the default value), the asynchronous schedule is not processed. Only the host controller uses this field. 0 - Do not process the asynchronous schedule 1 - Access the asynchronous schedule
4 PSE	Periodic Schedule Enable Determines whether the host controller skips processing the periodic schedule. If this field is 1, Frame List Base Address (PERIODICLISTBASE) is used to access the periodic schedule. If this field is 0 (the default value), the periodic schedule is not processed. Only the host controller uses this field. 0 - Do not process the periodic schedule 1 - Process the periodic schedule
3-2 FS_1	Frame List Size 1 See the description of USBCMD[FS_2].
1 RST	Controller Reset Resets the controller. Use this field to reset the controller. The host or device controller writes 0 to this field after the reset process is complete. You cannot terminate the reset process early by writing 0 to this field. In Host Operation mode, the controller resets its internal pipelines, timers, counters, state machines, and so on, to their initial values when you write 1 to this field. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. You must not write 1 to this field when USBSTS[HCH] is 0. Attempting to reset an actively running host controller results in undefined behavior. In Device Operation mode, the controller resets its internal pipelines, timers, counters, state machines, and so on, to their initial values when you write 1 to this field. Writing 1 to this field when the chip is in the attached state is not recommended because the effect on an attached host is undefined. To ensure that the chip is not in an attached state before initiating a device controller reset, all primed endpoints must be flushed and USBCMD[RS] must be 0.
0 RS	Run/Stop Specifies whether the controller is executing the operation.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	In Host Operation mode, the controller proceeds with the execution of the schedule if the value of this field is 1b. The controller continues the execution as long as this field is 1. If this field becomes 0 (the default value), the host controller completes the current transaction on the USB and then halts. The Halted bit in the Status field of the dTD token indicates when the controller has finished the transaction and has entered the Stopped state. You must not write 1 to this field unless the controller is in the Halted state (that is, USBSTS[HCH] = 1). In Device Operation mode, writing 1 to this field causes the controller to enable a pull-up on D+ and initiate an attach event. This control field is not directly connected to the pull-up enable because the pull-up becomes disabled upon transitioning into high-speed mode. You must use this field to prevent an attach event before the controller is properly initialized. Writing 0 to this field leads to a detach event. 0 - Stop executing 1 - Running

41.7.1.1.20 USB Status (USBSTS)

Indicates various states of the host and device controller and any pending interrupts. This register does not indicate status resulting from a transaction on the serial bus.

Offset

Register	Offset
USBSTS	144h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved						TI1	TI0	Reserved				UPI	UAI	Reserved	NAKI
W							W1C	W1C					W1C	W1C		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AS	PS	RCL	HCH	Reserved	ULPII	Reserved	SLI	SRI	URI	AAI	SEI	FRI	PCI	UEI	UI
W						W1C		W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	u	0	0	0	0	0	0	0

Fields

Field	Description
31-26 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
25 TI1	General Purpose Timer Interrupt 1 (GPTINT1) Flag Indicates when the counter in General Purpose Timer 1 Controller (GPTIMER1CTRL) transitions to 0. This flag is set when this happens. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt did not occur 1 - Interrupt occurred When writing 0 - No effect 1 - Clear the flag
24 TI0	General Purpose Timer Interrupt 0 (GPTINT0) Flag Indicates when the counter in General Purpose Timer 0 Controller (GPTIMER0CTRL) transitions to 0. This flag is set when this happens. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt did not occur 1 - Interrupt occurred When writing 0 - No effect 1 - Clear the flag
23-20 —	Reserved
19 UPI	USB Host Periodic Interrupt Flag Indicates the cause of a periodic interrupt and whether a short packet is detected. The host controller sets this field when the cause of an interrupt is the completion of a USB transaction, where the transfer descriptor (TD) has the IOC field set and the TD is from a periodic schedule. The host controller also sets this field when a short packet (the actual number of bytes received is less than expected) is detected and the packet is on a periodic schedule. The device controller does not use this field; the field is always 0. **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When reading 0 - Interrupt did not occur 1 - Interrupt occurred When writing 0 - No effect 1 - Clear the flag
18 UAI	USB Host Asynchronous Interrupt Flag Indicates the cause of an asynchronous interrupt and whether a short packet is detected. The host controller sets this flag when the cause of an interrupt is the completion of a USB transaction, where the transfer descriptor (TD) has the IOC field set and the TD is from an asynchronous schedule. The host controller also sets this flag when a short packet (the actual number of bytes received is less than expected) is detected and the packet is on an asynchronous schedule. The device controller does not use this field; the field is always 0. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt did not occur 1 - Interrupt occurred When writing 0 - No effect 1 - Clear the flag
17 —	Reserved
16 NAKI	NAK Interrupt Indicates whether a NAK interrupt is detected. Hardware sets this field, when both ENDPTNAK[EPTN] and ENDPTNAK[EPRN] are set for a particular endpoint, and their corresponding fields (ENDPTNAKEN[EPTNE] and ENDPTNAKEN[EPRNE]) are 1. When all enabled TX and RX endpoint NAK fields are cleared, this field is cleared automatically.
15 AS	Asynchronous Schedule Status Reports the current real status of the asynchronous schedule. If this field is 0, the asynchronous schedule status is disabled, and if this field is 1, the status is enabled. The host controller is not required to immediately disable or enable the asynchronous schedule when you transition USBCMD[ASE]. When this field and USBCMD[ASE] have the same value, the asynchronous schedule is either enabled (1) or disabled (0).

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field is only used in Host Operation mode. 0 - Disabled 1 - Enabled
14 PS	Periodic Schedule Status Reports the current real status of the periodic schedule. If this field is 0, the periodic schedule is disabled, and if this field is 1, the status is enabled. The host controller is not required to immediately disable or enable the periodic schedule when software transitions USBCMD[PSE]. When this field and USBCMD[PSE] have the same value, the periodic schedule is either enabled (1) or disabled (0). This field is only used in Host Operation mode. 0 - Disabled 1 - Enabled
13 RCL	Reclamation Detects an empty asynchronous schedule. This field is only used in Host Operation mode. 0 - Not detected 1 - Detected
12 HCH	HC Halted Indicates whether the host controller is halted. This field is 0 whenever USBCMD[RS] is 1. The controller writes 1 to this field after it has stopped executing (because of USBCMD[RS] = 0), either by software or by the controller hardware (for example, in case of an internal error). This field is only used in Host Operation mode. The default value is 0b for the OTG core This is because the OTG core is not operating as host by default (see USBMODE[CM]). **NOTE** The reset value of HCH is 0b for the OTG controller core
11 —	Reserved
10 ULPII	ULPI Interrupt Flag Indicates an event completion in the ULPI viewport. Hardware sets this field to 1b when there is an event completion in the ULPI viewport. You can only use this flag if the controller supports UPLI Interface mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - Event completion did not occur 1 - Event completion occurred When writing 0 - No effect 1 - Clear the flag
9 —	Reserved
8 SLI	Device Controller Suspend Flag Indicates whether the device controller entered a suspended state. When a device controller enters a suspended state from an active state, this flag is set to 1. The device controller clears this flag upon exiting from a suspended state. This flag is used only in Device Operation mode. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Did not enter Suspended state 1 - Entered Suspended state When writing 0 - No effect 1 - Clear the flag
7 SRI	SOF Received Flag Indicates whether a start of (micro) frame (SOF) is received. When the device controller detects an SOF, this flag is set to 1. When the SOF is extremely late, the device controller automatically sets this field to indicate that an SOF is expected. Therefore, this flag is set roughly every 1 ms in device FS mode and every 125 ms in HS mode. It is synchronized to the actual SOF that is received. Because the device controller is initialized to FS before connect, this flag sets at an interval of 1 ms during the prelude to connect and chirp. In Host mode, this field sets every 125 µs; the host controller driver can use it as a time base. **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When reading 0 - SOF not received 1 - SOF received When writing 0 - No effect 1 - Clear the flag
6 URI	USB Reset Received Flag Indicates whether a USB reset is received. When the device controller detects a USB reset and enters the default state, this flag is set to 1. This flag is used only in Device Operation mode. **NOTE** This field behaves differently for register reads and writes. When reading 0 - USB reset not received 1 - USB reset received When writing 0 - No effect 1 - Clear the flag
5 AAI	Interrupt on Async Advance Flag Indicates the assertion of the interrupt source. You can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing 1 to USBCMD[IAA]. This flag is used only in Host Operation mode. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt did not occur 1 - Interrupt occurred When writing 0 - No effect 1 - Clear the flag
4	System Error Flag Indicates an error response to a read on the system interface.

Table continues on the next page...

Table continued from the previous page...

Field	Description
SEI	This flag is set to 1 when an error response is seen to a read on the system interface. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Error response did not occur 1 - Error response occurred When writing 0 - No effect 1 - Clear the flag
3 FRI	Frame List Rollover Flag Indicates whether the frame list index rolled over. The host controller sets this flag to 1 when the frame list index rolls over from its maximum value to 0. The exact value at which the rollover occurs depends on the frame list size. For example, if the frame list size (as programmed in USBCMD[FS_1] and USBCMD[FS_2]) is 1024, USB Frame Index (FRINDEX) rolls over every time FRINDEX[13] toggles. Similarly, if the size is 512, the host controller sets this flag to 1 every time FHINDEX[12] toggles. This field is used only in Host Operation mode. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Frame list index did not roll over 1 - Frame list index rolled over When writing 0 - No effect 1 - Clear the flag
2 PCI	Port Change Detect Flag Indicates port change detection. The host controller sets this flag to 1 when a connect status occurs on any port, a port enable or disable change occurs, or PORTSC1[FPR] becomes 1 as the result of a J-K transition on the suspended port. The device controller sets this flag to 1 when the port controller enters the FS or HS operational state. When the port controller exits the FS or HS operation state because of reset or suspend events, the notification mechanisms are controlled by USBSTS[URI] and PORTSC1[SUSP], respectively. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Port change not detected 1 - Port change detected When writing 0 - No effect 1 - Clear the flag
1 UEI	USB Error Interrupt (USBERRINT) Flag Indicates an error condition after the completion of a USB transaction. The host or device controller sets this flag after the completion of a USB transaction results in an error condition. This flag is set along with USBSTS[UI], if the TD on which the error interrupt occurred also has its IOC field set in the dTD token. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt did not occur 1 - Interrupt occurred When writing 0 - No effect 1 - Clear the flag
0 UI	USB Interrupt (USBINT) Flag Indicates the cause of an interrupt. The host or device controller sets this flag when the cause of an interrupt is the completion of a USB transaction where the transfer descriptor (TD) has its interrupt on complete (IOC) field set in the dTD token. The host or device controller also sets this field when a short packet is detected (the actual number of bytes received is less than the expected number of bytes). **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt did not occur 1 - Interrupt occurred When writing 0 - No effect 1 - Clear the flag

41.7.1.1.21 Interrupt Enable (USBINTR)

Enables interrupts to software. An interrupt is generated when a field in this register is 1 and the corresponding interrupt source is active. USB Status (USBSTS) still shows interrupt sources even if they are disabled by USBINTR, allowing you to poll the interrupt events.

Offset

Register	Offset
USBINTR	148h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						TIE1	TIE0	Reserved				UPIE	UAIE	Reserved	NAKE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					Reserved	Reserved	SLE	SRE	URE	AAE	SEE	FRE	PCE	UEE	UE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-26 —	Reserved
25 TIE1	General Purpose Timer 1 Interrupt Enable Enables general purpose timer 1. When this field is 1 and USBSTS[TI1] is 1 too, the controller issues an interrupt. 0 - Disable 1 - Enable
24 TIE0	General Purpose Timer 0 Interrupt Enable Enables general purpose timer 0. When this field is 1 and USBSTS[TI0] is 1 too, the controller issues an interrupt. 0 - Disable 1 - Enable
23-20 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
19 UPIE	USB Host Periodic Interrupt Enable Enables USB host periodic interrupt. When this field is 1 and USBSTS[UPI] is 1 too, the host controller issues an interrupt at the next interrupt threshold. 0 - Disable 1 - Enable
18 UAIE	USB Host Asynchronous Interrupt Enable Enables USB host asynchronous interrupt. When this field is 1 and USBSTS[UAI] is 1 too, the host controller issues an interrupt at the next interrupt threshold. 0 - Disable 1 - Enable
17 —	Reserved
16 NAKE	NAK Interrupt Enable Enables NAK interrupt. When this field is 1 and USBSTS[NAKI] is 1 too, the controller issues an interrupt. This field is only used in Device Operation mode. 0 - Disable 1 - Enable
15-11 —	- These bits are reserved and should be set to zero.
10 —	Reserved
9 —	Reserved
8 SLE	Sleep Interrupt Enable Enables sleep interrupt. When this field is 1 and USBSTS[SLI] is 1 too, the controller issues an interrupt. This field is only used in Device Operation mode. 0 - Disable 1 - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 SRE	SOF Received Interrupt Enable Enables the SOF received interrupt. When this field is 1 and USBSTS[SRI] is 1 too, the controller issues an interrupt. 0 - Disable 1 - Enable
6 URE	USB Reset Interrupt Enable Enables the USB reset interrupt. When this field is 1 and USBSTS[URI] is 1 too, the controller issues an interrupt. This field is used only in Device Operation mode. 0 - Disable 1 - Enable
5 AAE	Async Advance Interrupt Enable Enables asynchronous advance interrupt. When this field is 1 and USBSTS[AAI] is 1 too, the controller issues an interrupt. This field is used only in Host Operation mode. 0 - Disable 1 - Enable
4 SEE	System Error Interrupt Enable Enables system error interrupt. When this field is 1 and USBSTS[SEI] is 1 too, the controller issues an interrupt. This field is used only in Host Operation mode. 0 - Disable 1 - Enable
3 FRE	Frame List Rollover Interrupt Enable Enables the frame list rollover interrupt. When this field is 1 and USBSTS[FRI] is 1 too, the controller issues an interrupt. This field is used only in Host Operation mode. 0 - Disable 1 - Enable
2 PCE	Port Change Detect Interrupt Enable Enables port change detect interrupt. When this field is 1 and USBSTS[PCI] is 1 too, the controller issues an interrupt.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Disable 1 - Enable
1 UEE	USB Error Interrupt Enable Enables USB error interrupts. When this field is 1 and USBSTS[UEI] is 1 too, the controller issues an interrupt. 0 - Disable 1 - Enable
0 UE	USB Interrupt Enable Enables USB interrupt. When this field is 1 and USBSTS[UI] is 1 too, the controller issues an interrupt. 0 - Disable 1 - Enable

41.7.1.1.22 USB Frame Index (FRINDEX)

Indexes the periodic frame list.

The host controller uses this register to index the periodic frame list. The register updates every 125 microseconds (once per microframe). Bits[*n*:3] are used to select a particular entry in the periodic frame list during periodic schedule execution. The number of bits used for the index depends on the size of the frame list that you define by using USBCMD[FS_2] and USBCMD[FS_1].

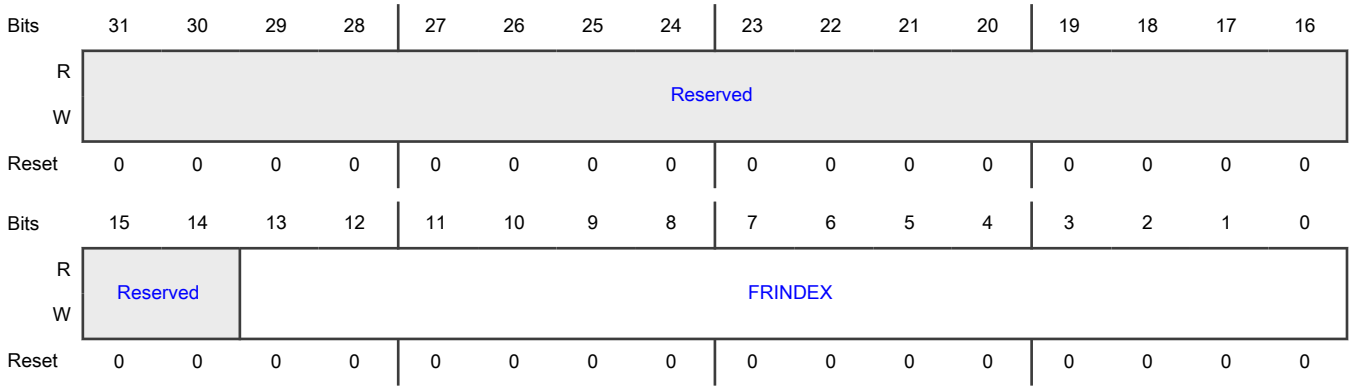
You must write to this register as a DWord. Byte writes produce undefined results. You cannot write to this register unless the host controller is in the Halted state, as indicated by USBSTS[HCH]. A write to this register, when USBCMD[RS] is 1, produces undefined results. Writes to this register also affect the SOF value.

In Device mode, this register is read-only and the device controller updates FRINDEX[13:3] from the frame number indicated by the SOF marker. Whenever the USB bus receives an SOF, FRINDEX[13:3] is checked against the SOF marker. If FRINDEX[13:3] is different from the SOF marker, FRINDEX[13:3] is set to the SOF value and FRINDEX[2:0] is 0 (that is, SOF for 1 ms frame). If FRINDEX[13:3] is equal to the SOF value, FRINDEX[2:0] increments (that is, SOF for 125 µs microframe).

Offset

Register	Offset
FRINDEX	14Ch

Diagram



Fields

Field	Description
31-14 —	Reserved
13-0 FRINDEX	Frame Index Increments at the end of each time frame (microframe). Bits[*n*:3] are used for the frame list current index. This means that each location of the frame list is accessed eight times (frames or microframes) before moving to the next index. In Host mode, the following values of *n* are based on the values of USBCMD[FS_2] and USBCMD[FS_1]. In Device mode, the value is the current frame number of the last frame transmitted. It is not used as an index. In either mode, bits 2:0 indicate the current microframe. The following values are represented as USBCMD (frame list size) number elements *n*. 00_0000_0000_0000 - (1024) 12 00_0000_0000_0001 - (512) 11 00_0000_0000_0010 - (256) 10 00_0000_0000_0011 - (128) 9 00_0000_0000_0100 - (64) 8 00_0000_0000_0101 - (32) 7 00_0000_0000_0110 - (16) 6 00_0000_0000_0111 - (8) 5

41.7.1.1.23 Device Address (DEVICEADDR)

Contains the device address. The upper seven bits of this register represent the device address.

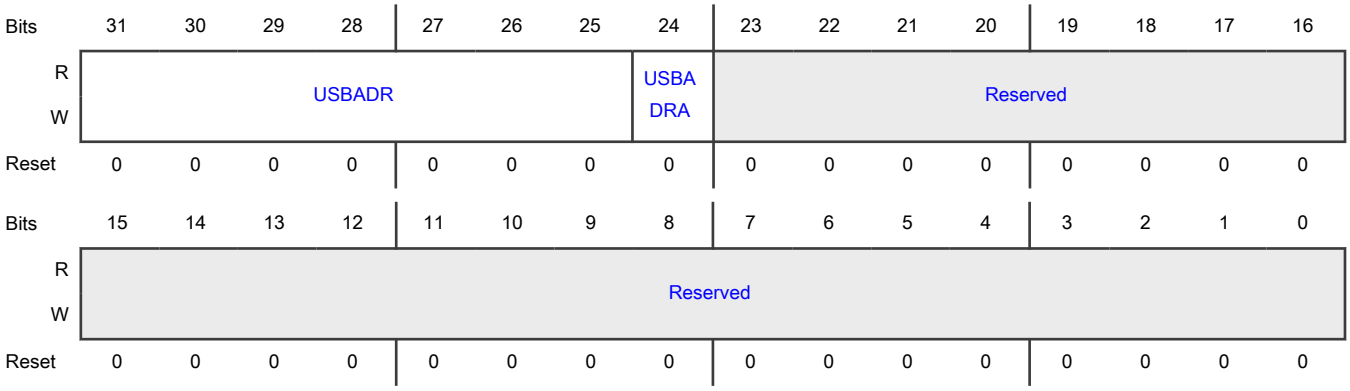
After a controller or USB reset, the device address is set to the default address (0) that matches all incoming addresses. You must reprogram the address after receiving a SET_ADDRESS descriptor.

Only the device controller uses this register.

Offset

Register	Offset
DEVICEADDR	154h

Diagram



Fields

Field	Description
31-25 USBADR	Device Address Represents the USB device address.
24 USBADRA	Device Address Advance Accelerates the SET_ADDRESS sequence by allowing the DCD to preset DEVICEADDR[USBADR] before the status phase of the SET_ADDRESS descriptor. When this field is 0 (the default value), any writes to USBADR are instantaneous. When you write 1 to this field at the same time or before writing to the USBADR field, a write to USBADR is staged and held in a hidden register. After an "In" transaction occurs on endpoint 0 and is ACKed, USBADR loads from the holding register. Hardware automatically writes 0 to this field on the following conditions: • The "In" transaction is ACKed to endpoint 0 (USBADR is updated from the staging register). • "Out" or "Setup" transactions occur to endpoint 0 (USBADR is not updated). • Device reset occurs (USBADR is reset to 0). **NOTE** After the status phase of the SET_ADDRESS descriptor, the DCD has 2 ms to program the USBADR field. This mechanism ensures that this specification is met when the DCD is busy and cannot write to this field within 2 ms from the SET_ADDRESS status phase. If the DCD writes to the USBADR field when USBADRA = 1 after the SET_ADDRESS data phase (before the prime of the status phase), the USBADR field is programmed instantly at the correct time and meets the 2 ms USB requirement.
23-0 —	Reserved

41.7.1.1.24 Frame List Base Address (PERIODICLISTBASE)

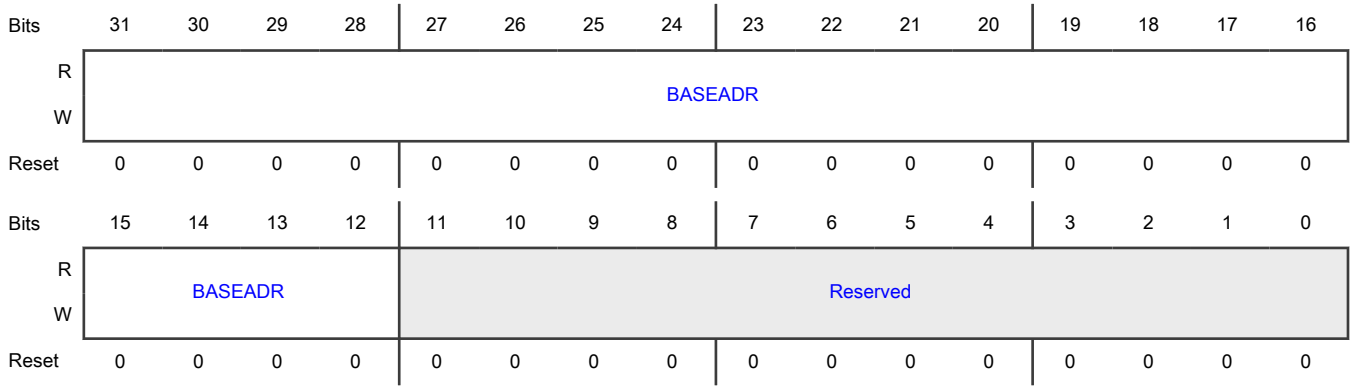
Contains the beginning address of the periodic frame list in the system memory.

The host controller driver (HCD) loads this register before the host controller starts the schedule execution. The memory structure referenced by this physical memory pointer is assumed to be 4-KB aligned. The contents of this register are combined with USB Frame Index (FRINDEX) to enable the host controller to step through the periodic frame list in sequence.

Offset

Register	Offset
PERIODICLISTBASE	154h

Diagram



Fields

Field	Description
31-12 BASEADR	Base Address (Low) Specifies the base address. These bits of this field correspond to memory address signals [31:12], respectively. Only the host controller uses this field.
11-0 —	Reserved

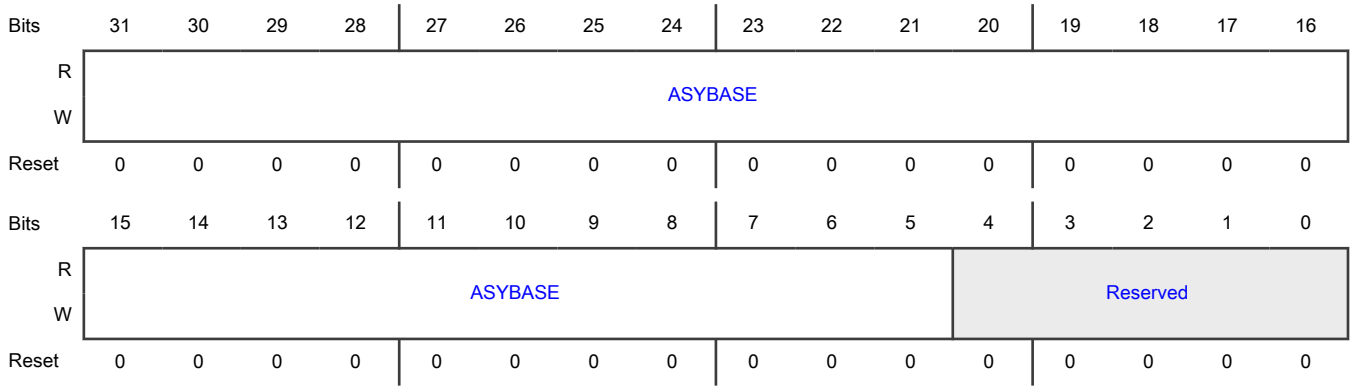
41.7.1.1.25 Next Asynchronous Address (ASYNCLISTADDR)

Contains the address of the next asynchronous queue head to be executed by the host. You cannot modify bits [4:0] of this register; they always return a 0 when read.

Offset

Register	Offset
ASYNCLISTADDR	158h

Diagram



Fields

Field	Description
31-5 ASYBASE	Link Pointer Low (LPL) Specifies the link pointer. The bits of this field correspond to memory address signals [31:5], respectively. This field can only reference a queue head. Only the host controller uses this field.
4-0 —	Reserved

41.7.1.1.26 Endpoint List Address (ENDPTLISTADDR)

Contains the address of the top of the endpoint list in system memory, when in Device mode. You cannot modify bits[10:0] of this register; they always return a 0 when read.

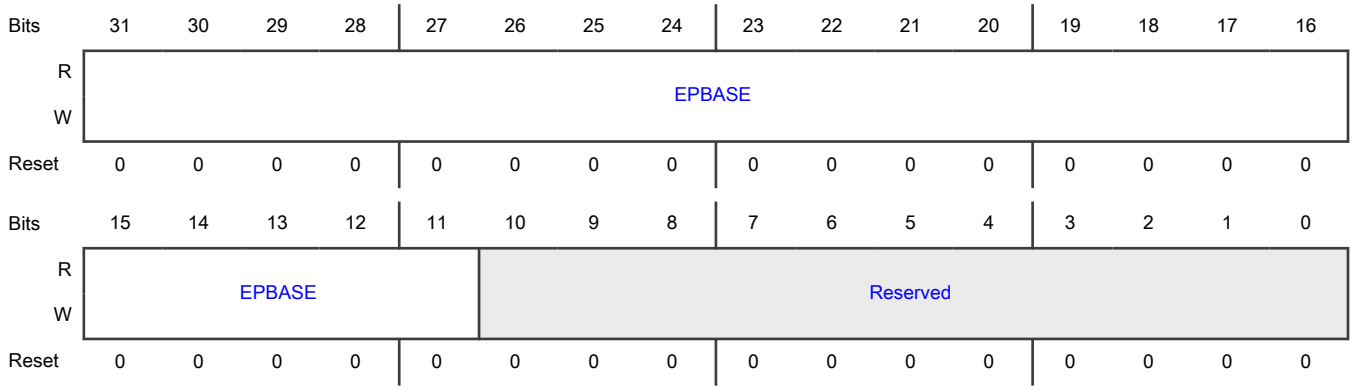
The memory structure referenced by this physical memory pointer is assumed to be 64 bytes.

This is a device-controller-only register.

Offset

Register	Offset
ENDPTLISTADDR	158h

Diagram



Fields

Field	Description
31-11 EPBASE	Endpoint List Pointer (Low) Specifies the endpoint list pointer. These bits of this field correspond to memory address signals [31:11], respectively. This field references a list of up to 32 queue heads (that is, one queue head per endpoint and direction).
10-0 —	Reserved

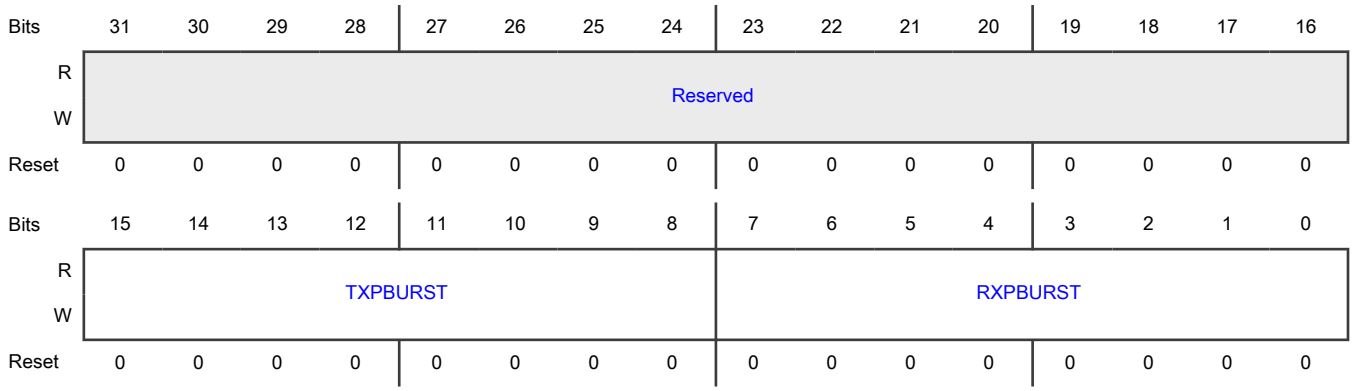
41.7.1.1.27 Programmable Burst Size (BURSTSIZE)

Controls the burst size used during data movement on the AHB controller interface. This register is ignored if SBUSCFG[AHBBRST] has a nonzero value.

Offset

Register	Offset
BURSTSIZE	160h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 TXPBURST	Programmable TX Burst Size Represents the maximum length of a burst, in 32-bit words, when moving data from the system memory to the USB bus. HWTXBUF[TXBURST] determines the default value of this field.
7-0 RXPBURST	Programmable RX Burst Size Represents the maximum length of a burst, in 32-bit words, when moving data from the USB bus to the system memory. HWRXBUF[RXBURST] determines the default value of this field.

41.7.1.1.28 TX FIFO Fill Tuning (TXFILLTUNING)

Controls performance tuning associated with how the host controller posts data to the TX latency FIFO before moving the data onto the USB bus.

The specific areas of performance include how much data to post into the FIFO and an estimate for how long that operation should take in the target system.

Following are some related definitions:

T_0 = Standard packet overhead

T_1 = Time to send data payload

T_{ff} = Time to fetch packet into the TX FIFO up to the specified level

T_s = Total packet flight time (send-only) packet

$T_s = T_0 + T_1$

T_p = Total packet time (fetch and send) packet

$T_p = T_{ff} + T_0 + T_1$

After the discovery of a transmit (out or setup) packet in the data structures, the host controller checks to ensure T_p remains before the end of the [micro]frame. If so, it proceeds to prefill the TX FIFO. If the remaining time in the frame of [micro]frame is < T_s at any time during this operation, then the packet attempt ceases and the packet is tried at a later time.

Although this is not an error condition and the host controller eventually recovers, a mark is created in the scheduler health counter to note the occurrence of a back-off event. When a back-off event is detected, the partial packet fetched may need to be discarded from the latency buffer to make room for periodic traffic that will begin after the next SOF. Too many back-off events can waste bandwidth and power on the system bus and should therefore be minimized (not necessarily eliminated). You can minimize back-off events by using TXFILLTUNING[TXSCHHEALTH](T_{ff}).

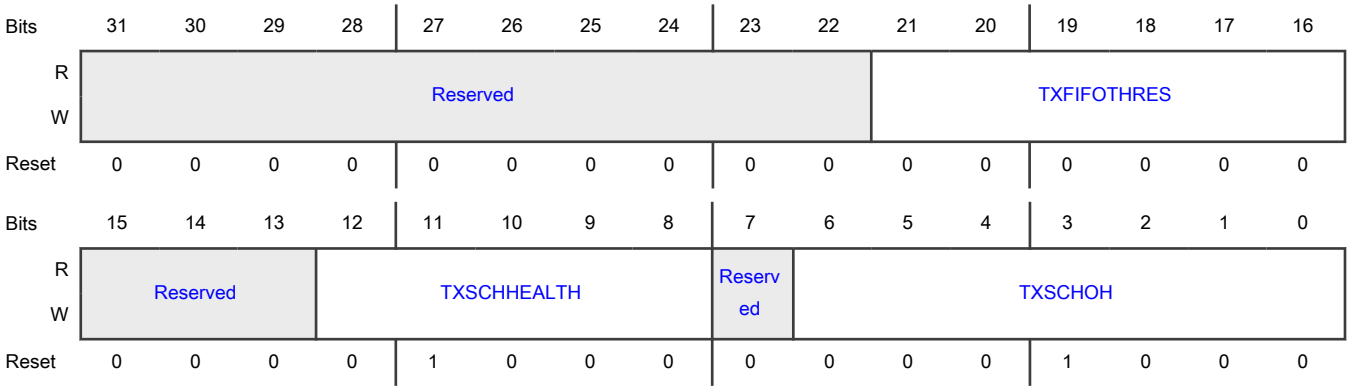
NOTE

The reset value could vary from instance to instance. In that case, see the field descriptions of this register and ignore the reset value in the summary table.

Offset

Register	Offset
TXFILLTUNING	164h

Diagram



Fields

Field	Description
31-22 —	Reserved
21-16 TXFIFOTHRES	FIFO Burst Threshold Controls the number of data bursts that are posted to the TX latency FIFO in Host mode before the packet begins on to the bus. The minimum value is 2 and this value should be as low as possible to maximize USB performance. A higher value can be used in systems with unpredictable latency and/or insufficient bandwidth, where the FIFO may underrun because the data transferred from the latency FIFO to USB occurs before it can be replenished from system memory. This value is ignored if USBMODE[SDIS] is 1.
15-13 —	Reserved
12-8 TXSCHHEALTH	Scheduler Health Counter Increments when the host controller fails to fill the TX latency FIFO to the level that you program in TXFILLTUNING[TXFIFOTHRES] before running out of time to send the packet before the next SOF. This health counter measures the number of times this occurs to provide feedback for selecting a proper scheduler overhead (TXFILLTUNING[TXSCHOH]). Writing to this register clears the counter that has the maximum value of 31.
7 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
6-0 TXSCHOH	Scheduler Overhead Adds an additional fixed offset to the schedule time estimator described as T_{ff}. As an approximation, the value selected for this register must limit the number of back-off events captured in TXFILLTUNING[TXSCHHEALTH] to less than 10 per second in a highly utilized bus. Selecting a value that is too high for this register is not recommended because it can needlessly reduce USB utilization. The time unit represented in this register is 1.267 µs when a device is connected in High-Speed mode. The time unit represented in this register is 6.333 µs when a device is connected in Low-Speed or Full-Speed mode. The default value is 08h for the OTG controller core .

41.7.1.1.29 Endpoint NAK (ENDPTNAK)

Configures RX and TX endpoint NAKs. Endpoint sends a NAK handshake for the "In", "Out", and "Ping" tokens.

Offset

Register	Offset
ENDPTNAK	178h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								EPTN							
W									W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								EPRN							
W									W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23-16 EPTN	TX Endpoint NAK Flag Indicates a transmit endpoint NAK.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Each TX endpoint has one bit in this field, and that bit sets when the device sends a NAK handshake on a received "In" token for the corresponding endpoint. Bit [*n*] - Endpoint #[*n*], where *n* is 0–7. **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000 - No NACK 0000_0001 - NACK When writing 0000_0000 - No effect 0000_0001 - Clear the flag
15-8 —	Reserved
7-0 EPRN	RX Endpoint NAK Flag Indicates a receive endpoint NAK. Each RX endpoint has one bit in this field, and that bit sets when the device sends a NAK handshake on a received "Out" or "Ping" token for the corresponding endpoint. Bit[*n*] - Endpoint #[*n*], where *n* is 0–7. **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000 - No NACK 0000_0001 - NACK When writing 0000_0000 - No effect 0000_0001 - Clear the flag

41.7.1.1.30 Endpoint NAK Indication Enable (ENDPTNAKEN)

Enables RX and TX endpoint NAKs.

Offset

Register	Offset
ENDPTNAKEN	17Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								EPTNE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								EPRNE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23-16 EPTNE	TX Endpoint NAK Enable Enables transmit endpoint NAK. Each bit in this field enables the corresponding TX endpoint NAK bit. If this bit is set and the corresponding TX endpoint NAK bit is set, the NAK interrupt bit is set. Bit[*n*] - Endpoint #[*n*], where *n* is 0–7.
15-8 —	Reserved
7-0 EPRNE	RX Endpoint NAK Enable Enables receive endpoint NAK. Each bit in this field enables the corresponding RX endpoint NAK bit. If this bit is set and the corresponding RX endpoint NAK bit is set, the NAK interrupt bit is set. Bit[*n*] - Endpoint #[*n*], where *n* is 0–7.

41.7.1.1.31 Configure Flag (CONFIGFLAG)

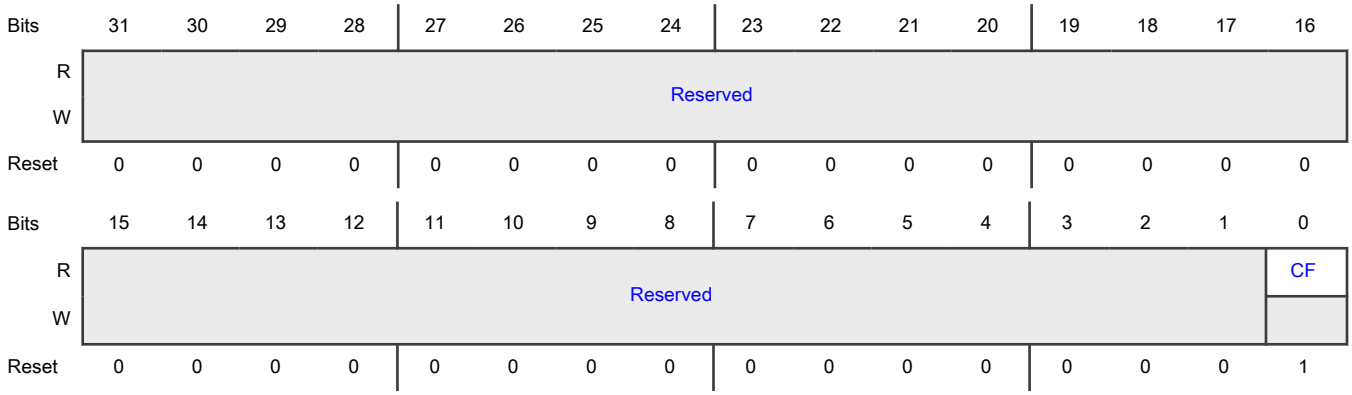
Controls port-routing control logic.

This register is not used in this implementation. A read from this register returns a constant of a 0000_0001h to indicate that all port routings default to this host controller.

Offset

Register	Offset
CONFIGFLAG	180h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 CF	Configure Flag Controls the default port-routing control logic. You must set this flag as the last action in its process of configuring the host controller. If this flag is 0, the port routing control logic routes each port to an implementation dependent on classic host controller by default. If this flag is 1, the port routing control logic routes all ports to this host controller by default. 0 - Port routing to classic host controller 1 - Port routing to this host controller

41.7.1.1.32 Port Status and Control (PORTSC1)

Determines and implements ports that need service.

A host controller can implement one to eight port status and control registers. The number is determined by HCSPARAMS[N_PORTS]. You can read this parameter register to determine how many ports need service.

All controller cores on this product are single-port, so there is only one port status and control register for each controller core.

This register is only reset by POR or controller core reset. The following are the initial conditions of a port:

- No device connected
- Port disabled

If the port supports power control, this state remains until software supplies port power.

A device controller operating in Device mode includes only PORTSC1 and does not support power control in that mode. Port control in Device mode is only used for status port reset, suspend, and current connect status. It is also used to initiate Test mode or force signaling, and allows you to place the PHY into Low-Power Suspend mode and disable the PHY clock.

Offset

Register	Offset
PORTSC1	184h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PTS_1		STS	PTW	PSPD		PTS_2	PFSC	PHCD	WKOC	WKDC	WKCN	PTC			
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIC		PO	PP	LS		HSP	PR	SUSP	FPR	OCC	OCA	PEC	PE	CSC	CCS
W											W1C		W1C		W1C	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-30 PTS_1	Parallel Transceiver Select 1 Indicates parallel transceiver select 1. Bit field {bit25, bit31, bit30}: 000b - UTMI/UTMI+ 001b - Reserved 010b - ULPI 011b - Serial/USB 1.1 PHY/IC-USB (FS only) 100b - HSIC Parallel Transceiver Select (bit25, bit31, bit30). For the OTG core, it is read-only. For host1, host2, and host3 cores, it is read/write. **NOTE** All USB port interface modes are listed in this field description, but not all are supported. For details related to each controller core, see Features. The behavior is unknown when an unsupported interface mode is selected.
29 STS	Serial Transceiver Select Selects the serial transceiver. The serial interface engine can be used in combination with UTMI+/ULPI physical interface to provide FS and LS signaling instead of the parallel interface signals.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this field is 1, the serial interface engine is used instead of parallel interface signals. This field has no effect unless PORTSC1[PTS_1] is 1 (to select the UTMI+/ULPI interface). The serial/USB1.1 PHY/IC-USB uses the serial interface engine for FS and LS signaling regardless of the value of this field. 0 - Parallel interface signals 1 - Serial interface engine
28 PTW	Parallel Transceiver Width Selects the parallel transceiver width. This field has no effect if the serial interface engine is used. 0 - 8-bit UTMI interface [60 MHz] 1 - 16-bit UTMI interface [30 MHz]
27-26 PSPD	Port Speed Indicates the speed at which the port is operating. 00 - FS 01 - LS 10 - HS 11 - Undefined
25 PTS_2	Parallel Transceiver Select 2 See the description of PORTSC1[PTS_1].
24 PFSC	Port Force Full Speed Connect Specifies the speed at which the port is connected. If this field is 1, the port is forced to connect at FS only. It disables the chirp sequence that allows the port to identify itself as HS. 0 - Normal operation 1 - Forced to full speed
23 PHCD	PHY Low-Power Suspend - Clock Disable (PLPSCD) Disables the PHY clock. Reading this field indicates the status of the PHY clock. **NOTE** You cannot disable the PHY clock if it is being used as the system clock. In Device mode, you can place the PHY into Low-Power Suspend mode when the device is not running (USBCMD[RS] is 0) or the host has signaled suspend (PORTSC1[SUSP] is 1). This field becomes 0

Table continues on the next page...

Table continued from the previous page...

Field	Description
	automatically when the host initials resume. Before forcing a resume from the device, the device controller driver must write 0 to this field. In Host mode, you can place the PHY into Low-Power Suspend mode when the downstream device is placed into Suspend mode or when no downstream device is connected. Low-Power Suspend mode is completely under your control. 0 - Enable 1 - Disable
22 WKOC	Wake on Overcurrent Enable (WKOC_E) Enables wake-up on overcurrent. Writing 1 to this field enables the port to be sensitive to overcurrent conditions as wake-up events. This field is 0 if PORTSC1[PO] is 0.
21 WKDC	Wake on Disconnect Enable (WKDSCNNT_E) Enables wake-up on disconnect. Writing 1 to this field enables the port to be sensitive to device disconnects as wake-up events. This field is 0 if PORTSC1[PO] is 0 or is in Device mode. 0 - Disable 1 - Enable
20 WKCN	Wake on Connect Enable (WKCNNT_E) Enables port sensitivity. Writing 1 to this field enables the port to be sensitive to device connects as wake-up events. This field is 0 if PORTSC1[PO] is 0 or is in Device mode. 0 - Disable 1 - Enable
19-16 PTC	Port Test Control Defines port test control. See Port Test Control mode for the operational model used for these test modes, and chapter 7 of the USB Specification Revision 2.0 for details on each test mode. FORCE_ENABLE_FS and FORCE ENABLE_LS are extensions to the test mode support specified in the EHCI specification. Writing the FORCE_ENABLE_{HS/FS/LS} values to this field forces the port into the connected and enabled states at the selected speed. Writing the TEST_MODE_DISABLE value to this field allows the port state machines to progress normally from that point. **NOTE** *Low-speed operations are not supported as a peripheral device.* Any value other than 0 indicates that the port is operating in Test mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Value specific test 0000 - TEST_MODE_DISABLE 0001 - J_STATE 0010 - K_STATE 0011 - SE0 (host) or NAK (device) 0100 - Packet 0101 - FORCE_ENABLE_HS 0110 - FORCE_ENABLE_FS 0111 - FORCE_ENABLE_LS 1000-1111 - Reserved
15-14 PIC	Port Indicator Control Specifies the port indicator. Writing to this field has no effect if HCSPARAMS[PI] is 0. See the USB Specification Revision 2.0 for a description on how these fields must be used. This field is 0 if PORTSC1[PO] is zero. 00 - Port indicators are off 01 - Amber 10 - Green 11 - Undefined
13 PO	Port Owner Indicates port ownership. This field to used to release ownership of the port to a selected host controller (in the event that the attached device is not an HS device). This field unconditionally becomes 0 when the configured field in Configure Flag (CONFIGFLAG) makes a 0 to 1 transition. The field unconditionally becomes 1 whenever the configured field is 0. Write 1 to this field when the attached device is not an HS device. If this field is 1, it means that an internal companion controller owns and controls the port. Port owner handoff is not supported in all controller cores; therefore, this field is always 0.
12 PP	Port Power Works in accordance with HCSPARAMS[PPC]. In other words, the functioning of this field depends on the value of HCSPARAMS[PPC]: • PPC = 0b, PP = 0b (read-only): A controller in Device mode does not have port power control switches. • PPC = 1b, PP = 1b or 0b (read/write): A controller in Host mode requires port power control switches.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	OTG controller requires port power control switches. This field represents the current setting of the switch (0 means off; 1 means on). When power is not available on a port (that is, PP = 0), the port is nonfunctional and does not report attachments, detachments, and so on. When an overcurrent condition is detected on a powered port and PPC is 1, this field, in each affected port, may be transitioned by the controller driver from 1 to 0 (removing power from the port). This feature is implemented in all controller cores (PPC = 1). 0 - Off 1 - On
11-10 LS	Line Status Indicates the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. In Host mode, the host controller driver may not necessarily use LineState (unlike EHCI) because the port controller state machine and port routing manage the connection of LS and FS. In Device mode, the device controller driver may not necessarily use LineState. The following is the encoding of bits [11:10]. 00 - SE0 01 - K-state 10 - J-state 11 - Undefined
9 HSP	High-Speed Port Indicates whether the host or device connected to the port is in HS mode. The field is redundant for PORTSC1[PSPD] but is included for compatibility. 0 - Not in HS mode 1 - In HS mode
8 PR	Port Reset Specifies whether the port is in reset. In Host mode (read/write), the bus-reset sequence (as defined in the USB Specification Revision 2.0) starts when you write 1 to this field. This field automatically changes to zero after the reset sequence is complete. This behavior is different from that of EHCI where the host controller driver is required to write 0 to this field after the reset duration is timed in the driver. In Device mode, this is a read-only status field. Device reset from the USB bus is also indicated in USB Status (USBSTS). This field is 0 if PORTSC1[PP] is 0. 0 - Port not in reset 1 - Port in reset

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 SUSP	Suspend Specifies whether the port is in the Suspended state. In Host mode (read/write), this field works with PORTSC1[PE] to define the port states as follows: In case of {PE, SUSP} port state: 0x - Disable 10 - Enable 11 - Suspend When in the Suspended state, downstream propagation of data is blocked on this port (except for port reset). Blocking occurs at the end of the current transaction if a transaction was in progress when this field was 1. The port is sensitive to resume detection in the Suspended state. The status of the field does not change until the port is suspended; there may be a delay in suspending a port if a transaction is in progress on USB. The host controller unconditionally writes 0 to this field when you write 0 to PORTSC1[FPR]. The host controller ignores a write of zero to this field. If host software writes 1 to this field when the port is not enabled (that is, PORTSC1[PE] is 0), the results are undefined. This field is 0 if PORTSC1[PP] is 0 in Host mode. In Device mode, this is a read-only status field. 0 - Port not in Suspended state 1 - Port in Suspended state
6 FPR	Force Port Resume Specifies whether force resume is detected or driven on the port. In Host mode, write 1 to this field to drive resume signaling. The host controller writes 1 to this field if a J-to-K transition is detected when the port is in the Suspended state. When this field transitions to 1 because a J-to-K transition is detected, USBSTS[PCI] also becomes 1. This field automatically changes to 0 after the resume sequence is complete. This behavior is different from that of EHCI where the host controller driver is required to write 0 to this field after the resume duration is timed in the driver. When the host controller owns the port, the resume sequence follows the defined sequence documented in USB specification revision 2.0. The resume signaling (full-speed "K") is driven on the port until this field remains 1 (that is, until the port has switched to the high-speed idle state). Writing 0 to this field has no effect because the port controller times the resume operation, clears the field, and the port control state switches to HS or FS idle. This field is not-EHCI compatible and remains 0 if PORTSC1[PP] is 0 in Host mode. In Device mode, you must write 1 to this field to drive resume signaling after the device is in the Suspended state for 5 ms or more. The device controller writes 1 to this field if a J-to-K transition is detected when the port is in the Suspended state. The field becomes 0 when the device returns to normal operation. Additionally, when this field becomes 0 because of a K-to-J transition detection, USBSTS[PCI] also becomes 1. 0 - No resume (K-state) detected or driven on port

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Resume detected or driven on port
5 OCC	Overcurrent Change Flag Indicates whether an overcurrent change occurred. Hardware sets this field to 1 when there is a change in an overcurrent active condition. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No change occurred 1 - Change occurred When writing 0 - No effect 1 - Clear the flag
4 OCA	Overcurrent Active Indicates whether the port currently has an overcurrent condition. This field automatically transitions from 1 to 0 when the overcurrent condition is removed. 0 - Overcurrent condition exists 1 - No overcurrent condition exists
3 PEC	Port Enable and Disable Change Flag Indicates whether the port enabled and disabled status has changed. In Host mode (for the root hub), this flag is set to 1 only when a port is disabled because of disconnect on the port or because of appropriate conditions existing at the EOF2 point (see chapter 11 of the USB specification). This field is 0 if PORTSC1[PP] is 0. In Device mode, the device port is always enabled; therefore, this field is always 0. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No change occurred 1 - Change occurred When writing 0 - No effect 1 - Clear the flag
2	Port Enable and Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
PE	Enables a port. In Host mode, this field is reset to 0b. Only the host controller can enable ports as part of reset and enable. You cannot enable a port by writing 1 to this field, and either a fault condition (such as a disconnect event) or host software can disable a port. The status of this field does not change until the port state actually changes. There may be a delay in disabling or enabling a port because of other host controller and bus events. When this field is 0 (the port is disabled), downstream propagation of data is blocked except for reset. In Host mode, this field is 0 if PORTSC1[PP] is 0. In Device mode, this field is always 1 (the device port is always enabled). 0 - Disable 1 - Enable
1 CSC	Connect Status Change Flag Indicates whether a change has occurred in the port's current connect status. In Host mode, the host or device controller sets this field for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the connect status (PORTSC1[CCS]) changes twice before you clear PORTSC1[CSC]. The hub hardware sets an already-set field (that is, CSC remains set). You can clear this field by writing 1 to it. In Host mode, this field is 0 if PORTSC1[PP] is 0. In Device Controller mode, this field is undefined. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No change occurred 1 - Change occurred When writing 0 - No effect 1 - Clear the flag
0 CCS	Current Connect Status Indicates the current port connection status. In Host mode, if this field is 1, the device is present on port, and if this field is 0 (default), no device is present. This value reflects the current state of the port, and may not correspond directly to the event that causes this field to be set. In Host mode, this field is 0 if PORTSC1[PP] is 0. In Device mode, if this field is 1, it indicates that the device is successfully attached and is operating in either high speed or full speed, as indicated by PORTSC1[HSP]. If this field is 0, it indicates that the device is not

Table continues on the next page...

Table continued from the previous page...

Field	Description
	attached successfully or is forcibly disconnected by writing 0 to USBCMD[RS]. It does not state the device being disconnected or suspended.

41.7.1.1.33 On-The-Go Status and Control (OTGSC)

Provides OTG status and control.

This register is available only in OTG controller core and includes fields for:

- OTG interrupt enables (read/write)
- OTG interrupt status (read/write to clear)
- OTG status inputs (read-only)
- OTG controls (read/write)

The status inputs are debounced using a 1 ms time constant. Values on the status inputs that do not persist for more than 1 ms do not cause an update of the status input register, or cause an OTG interrupt.

See USB Device Mode (USBMODE) for more information.

Offset

Register	Offset
OTGSC	1A4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	DPIE	EN_1MS	BSEIE	BSVIE	ASVIE	AVVIE	IDIE	Reserved	DPIS	STATUS_...	BSEIS	BSVIS	ASVIS	AVVIS	IDIS
W										W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	u	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	DPS	TOG_1MS	BSE	BSV	ASV	AVV	ID	Reserved		IDPU	DP	OT	Reserved	VC	VD
W																
Reset	0	0	u	u	u	u	u	u	0	0	1	0	0	0	0	0

Fields

Field	Description
31 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
30 DPIE	Data Pulse Interrupt Enable Enables the data pulse interrupt. Writing 1 to the field enables the interrupt.
29 EN_1MS	1 Millisecond Timer Interrupt Enable Enables the 1 ms timer interrupt. Writing 1 to the field enables the interrupt.
28 BSEIE	B Session End Interrupt Enable Enables the B session end interrupt. Writing 1 to the field enables the interrupt.
27 BSVIE	B Session Valid Interrupt Enable Enables the B session valid interrupt. Writing 1 to the field enables the interrupt.
26 ASVIE	A Session Valid Interrupt Enable Enables the A session valid interrupt. Writing 1 to the field enables the interrupt.
25 AVVIE	A VBUS Valid Interrupt Enable Enables the A VBUS valid interrupt. Writing 1 to the field enables the interrupt.
24 IDIE	USB ID Interrupt Enable Enables the USB ID interrupt. Writing 1 to the field enables the interrupt.
23 —	Reserved
22 DPIS	Data Pulse Interrupt Status Flag Indicates the status of the data pulse interrupt. This field becomes 1 when data bus pulsing occurs on a DP or DM line. Data bus pulsing is detected only when USBMODE[CM] = 11b and PORTSC1[PP] = 0b. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Data pulse did not occur 1 - Data pulse occurred When writing 0 - No effect 1 - Clear the flag
21 STATUS_1MS	1 Millisecond Timer Interrupt Status Flag Indicates the status of the timer interrupt. This field becomes 1 once every millisecond.

Table continues on the next page...

Table continued from the previous page...

Field	Description
20 BSEIS	B Session End Interrupt Status Flag Indicates the status of the B session end interrupt. This field becomes 1 when VBUS has fallen below the B session end threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No change 1 - Fallen below the threshold When writing 0 - No effect 1 - Clear the flag
19 BSVIS	B Session Valid Interrupt Status Flag Indicates the status of the B session valid interrupt. This field becomes 1 when VBUS has either risen above or fallen below the B session valid threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No change 1 - Risen above or fallen below the threshold When writing 0 - No effect 1 - Clear the flag
18 ASVIS	A Session Valid Interrupt Status Flag Indicates the status of the A session valid interrupt. This field becomes 1 when VBUS has either risen above or fallen below the A session valid threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No change 1 - Risen above or fallen below the threshold When writing 0 - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Clear the flag
17 AVVIS	A VBUS Valid Interrupt Status Flag Indicates the status of the A VBUS valid interrupt. This field becomes 1 when VBUS has either risen above or fallen below the VBUS valid threshold on A device. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No change 1 - Risen above or fallen below the threshold When writing 0 - No effect 1 - Clear the flag
16 IDIS	USB ID Interrupt Status Flag Indicates the status of the USB ID interrupt. This field becomes 1 when a change on the ID input is detected. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Change not detected 1 - Change detected When writing 0 - No effect 1 - Clear the flag
15 —	Reserved
14 DPS	Data Bus Pulsing Status Indicates the status of data bus pulsing. If this field is 1, it indicates that data bus pulsing is detected on the port.
13 TOG_1MS	1 Millisecond Timer Toggle Toggles once per millisecond.
12	B Session End

Table continues on the next page...

Table continued from the previous page...

Field	Description
BSE	Indicates that VBUS is below the B session end threshold.
11 BSV	B Session Valid Indicates that VBUS is above the B session valid threshold.
10 ASV	A Session Valid Indicates that VBUS is above the A session valid threshold.
9 AVV	A VBUS Valid Indicates that VBUS is above the A VBUS valid threshold.
8 ID	USB ID Indicates the USB ID. 0 - A device 1 - B device
7-6 —	Reserved
5 IDPU	ID Pullup Provides control over the ID pull-up resistor. If this field is 0, the functionality is off and the ID input is not sampled. If this field is 1 (default), the functionality is on and the ID input is sampled.
4 DP	Data Pulsing Controls pullup on the DP line. Writing 1 to this field causes the pullup on the DP line to be asserted for data pulsing during SRP.
3 OT	OTG Termination Controls pulldown on the DM line. This field must be 1 when the OTG device is in Device mode to control the pulldown on the DM line.
2 —	Reserved
1 VC	VBUS Charge Controls VBUS pulsing during SRP. Writing 1 to this field causes the VBUS line to be charged.
0 VD	VBUS Discharge Discharges VBUS through a resistor.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Writing 1 to this field causes VBUS to discharge through a resistor.

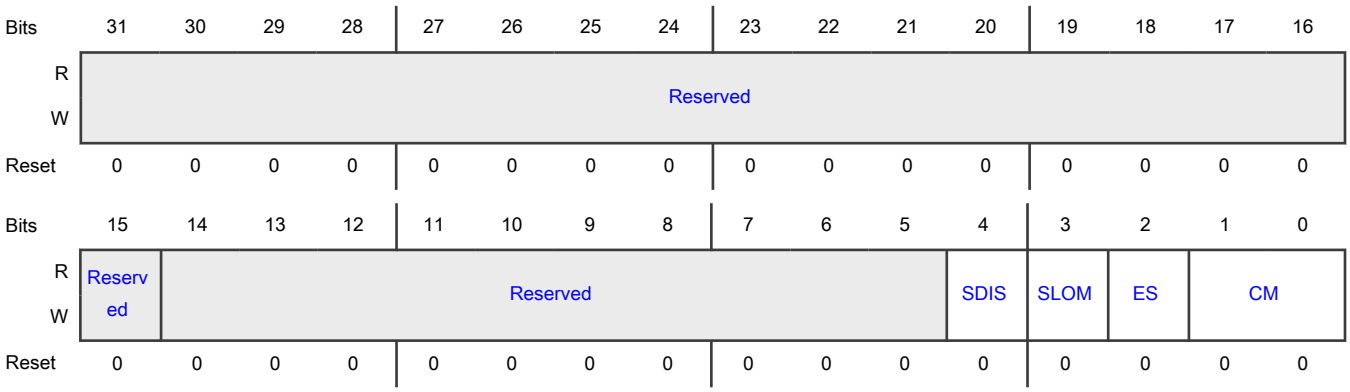
41.7.1.1.34 USB Device Mode (USBMODE)

Selects the USB device mode.

Offset

Register	Offset
USBMODE	1A8h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 —	Reserved
14-5 —	Reserved
4 SDIS	Stream Disable Mode Disables double priming for low-bandwidth systems. In Device mode, writing 1 to this field disables double priming on both RX and TX for low-bandwidth systems. This mode ensures that when the RX and TX buffers are sufficient to contain an entire packet, the standard double buffering scheme is disabled to prevent overruns and underruns in bandwidth limited systems.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE In High-Speed mode, all packets received are responded to with an NYET handshake when stream disable is active. In Host mode, writing 1 to this field ensures that overruns and underruns of the latency FIFO are eliminated for low bandwidth systems where the RX and TX buffers are sufficient to contain the entire packet. Enabling stream disable also ensures that the TX latency is filled to capacity before the packet is launched onto the USB. **NOTE** The time duration to prefill the FIFO becomes significant when stream disable is active. See TX FIFO Fill Tuning (TXFILLTUNING) and TXTTFILLTUNING (MPH only) to characterize the adjustments needed for the scheduler when using this feature. The use of this feature substantially limits the overall USB performance that can be achieved. 0 - Disable 1 - Enable
3 SLOM	Setup Lockout Mode Specifies whether the setup lockouts are on. In Device mode, this field controls the behavior of the setup lock mechanism. See Control endpoint operation model for more information. DCD requires you to use the setup data buffer tripwire in USB Command (USBCMD). 0 - On (default) 1 - Off
2 ES	Endian Select Selects the endianness. You can use this field to change the byte alignment of the transfer buffers to match the host microprocessor. The fields in the microprocessor interface and the data structures are unaffected by the value of this field because they are based on the 32-bit word. 0 - Little endian (default) 1 - Big endian
1-0 CM	Controller Mode Specifies the controller mode. Controller mode is defaulted to the proper mode for host-only and device-only implementations. For those designs that contain both host & device capability, the controller defaults to an idle state and needs to be initialized to the desired operating mode after reset. For combination host and device controllers, you can write to this register only once after reset. If it is necessary to switch modes, you must reset the controller by writing to USBCMD[RST] before reprogramming this register. For the OTG controller core, the reset value is 00b.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	00 - Idle (default for host and device combination) 01 - Reserved 10 - Device controller (default for device-only controller) 11 - Host controller (default for host-only controller)

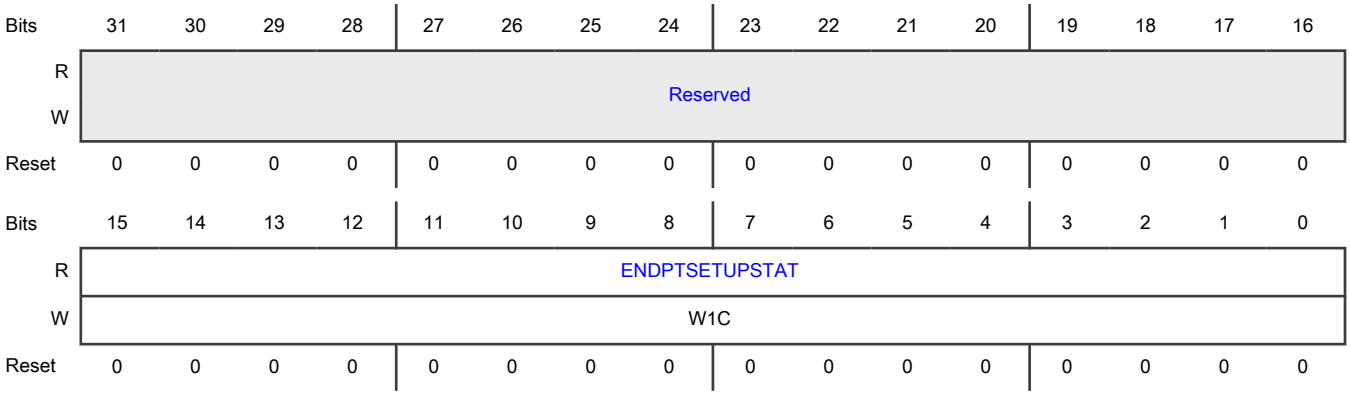
41.7.1.1.35 Endpoint Setup Status (ENDPTSETUPSTAT)

Indicates the endpoint setup status.

Offset

Register	Offset
ENDPTSETUPSTAT	1ACh

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 ENDPTSETUPSTAT	Endpoint Setup Status Flag Indicates the endpoint setup status and whether the setup transaction is received. For every setup transaction that is received, a corresponding bit in this field becomes 1. You must clear or acknowledge the setup transfer by writing 1 to the respective bit after it has read the setup data from the queue head.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	The response to a setup packet includes many operations in an order. The total response time is crucial to limit bus timeouts if the setup lockout mechanism is engaged. See Managing endpoints in the device operational model. This register is used only in Device mode. **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000_0000_0000 - Not received 0000_0000_0000_0001 - Received When writing 0000_0000_0000_0000 - No effect 0000_0000_0000_0001 - Clear the flag

41.7.1.1.36 Endpoint Prime (ENDPTPRIME)

Primes the endpoint.

When you set the prime bit for a given endpoint, the device controller loads the transfer descriptor, pointed to by the queue head, so that the endpoint is ready to transmit or receive when the host sends a request ("In" or "Out" token). The endpoint NAKs all requests from the host until the endpoint is primed. The controller automatically reprimes the endpoint with a new transfer descriptor when one is found via the next_dtd pointer of the current transfer descriptor. Therefore, you must set the prime bit only when a descriptor is added to the queue head.

Offset

Register	Offset
ENDPTPRIME	1B0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								PETB							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PERB							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23-16 PETB	Prime Endpoint Transmit Buffer Parses a new transfer descriptor from the queue head and prepares for a transmit buffer. For each endpoint, a corresponding bit is used to request that a buffer is prepared for a transmit operation to respond to a USB "In" or "Interrupt" transaction. You must write 1 to the corresponding bit when posting a new transfer descriptor to an endpoint queue head. Hardware automatically uses this bit to begin parsing for a new transfer descriptor from the queue head and preparing a transmit buffer. Hardware writes 0 to this field when the associated endpoint is successfully primed. PETB[*n*] - Endpoint [*n*], *n* is in 0–7 **NOTE** Hardware temporarily writes 1 to this field during hardware repriming operations when a dTD is retired and the dQH is updated.
15-8 —	Reserved
7-0 PERB	Prime Endpoint Receive Buffer Parses a new transfer descriptor from the queue head and prepares for a receive buffer. For each endpoint, a corresponding bit is used to request that a buffer is prepared for a receive operation when a USB host initiates a USB "Out" transaction. You must write 1 to the corresponding bit when posting a new transfer descriptor to an endpoint queue head. Hardware automatically uses this field to begin parsing for a new transfer descriptor from the queue head and preparing a receive buffer. Hardware writes 0 to this field when an associated endpoint is successfully primed. PERB[*n*] - Endpoint [*n*], *n* is in 0–7 **NOTE** Hardware temporarily writes 1 to this field during hardware repriming operations when a dTD is retired and the dQH is updated.

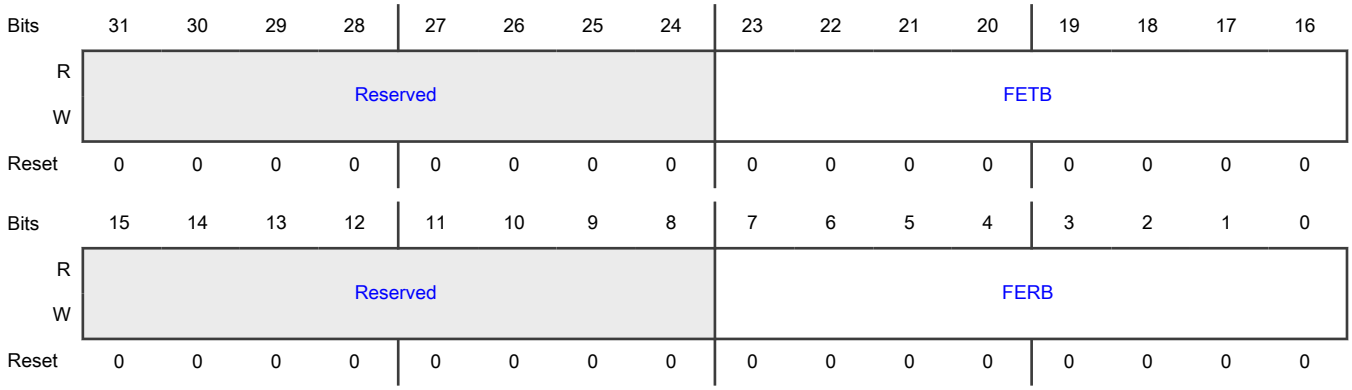
41.7.1.1.37 Endpoint Flush (ENDPTFLUSH)

Flushes the endpoint.

Offset

Register	Offset
ENDPTFLUSH	1B4h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 FETB	Flush Endpoint Transmit Buffer Clears any primed transmit buffers. When you write 1 to this field, it causes the associated endpoints to clear any primed transfer buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FETB[*n*]: Endpoint [*n*], *n* is in 0–7
15-8 —	Reserved
7-0 FERB	Flush Endpoint Receive Buffer Clears any primed receive buffers. When you write 1 to this field, it causes the associated endpoints to clear any primed receive buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FERB[*n*]: Endpoint [*n*], *n* is in 0–7

41.7.1.1.38 Endpoint Status (ENDPTSTAT)

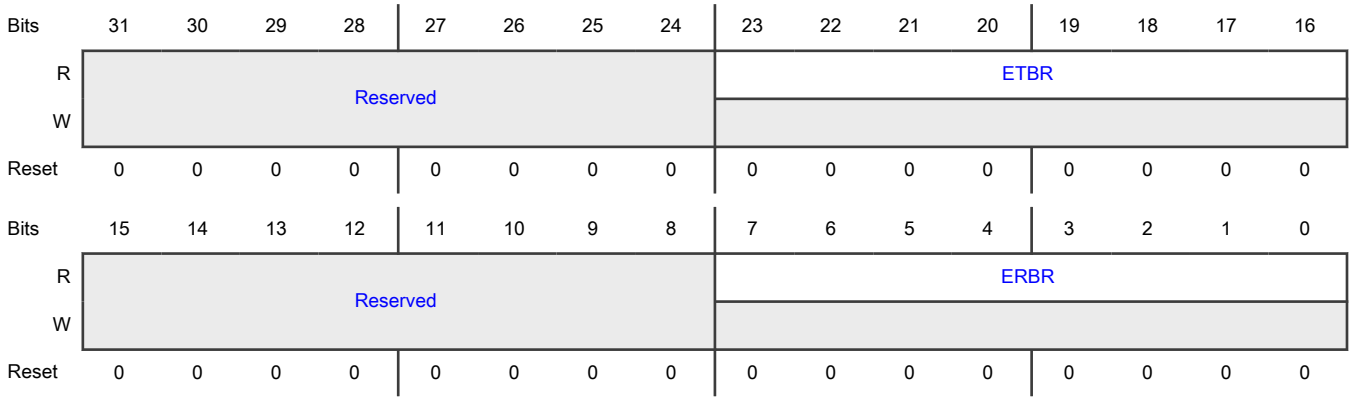
Indicates endpoint transmit and receive buffer ready status.

This register is only used in Device mode.

Offset

Register	Offset
ENDPTSTAT	1B8h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 ETBR	Endpoint Transmit Buffer Ready Indicates the endpoint transmit buffer ready status. One bit for each endpoint indicates the status of the respective endpoint buffer. Hardware writes 1 to this field as a response to receiving a command from a corresponding field in Endpoint Prime (ENDPTPRIME). There is always a delay between writing 1 to the corresponding field in Endpoint Prime (ENDPTPRIME) and the endpoint indicating the ready status. This delay time varies based on the current USB traffic and the number of fields with value 1 in Endpoint Prime (ENDPTPRIME). This field becomes 0 via the USB reset, USB DMA system, or through Endpoint Flush (ENDPTFLUSH). ETBR[*n*]: Endpoint [*n*], *n* is in 0–7 **NOTE** Hardware temporarily clears this field during hardware endpoint repriming operations when a dTD is retired and the dQH is updated.
15-8 —	Reserved
7-0 ERBR	Endpoint Receive Buffer Ready Indicates the endpoint receive buffer ready status. One bit for each endpoint indicates status of the respective endpoint buffer. Hardware writes 1 to this field as a response to receiving a command from a corresponding field in Endpoint Prime (ENDPTPRIME). There is always a delay between writing 1 to the corresponding field in Endpoint Prime (ENDPTPRIME) and the endpoint indicating the ready status. This delay time varies based on the current USB traffic and the number of fields with value 1 in Endpoint Prime (ENDPTPRIME). This field becomes 0 via the USB reset, USB DMA system, or through Endpoint Flush (ENDPTFLUSH). ERBR[*n*]: Endpoint [*n*], *n* is in 0–7

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE Hardware temporarily clears this field during hardware endpoint repriming operations when a dTD is retired and the dQH is updated.

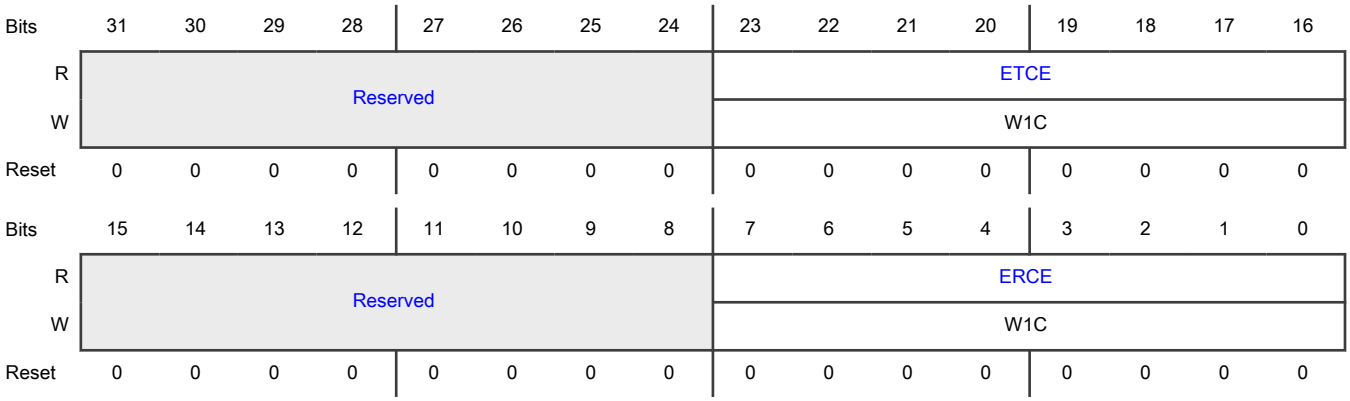
41.7.1.1.39 Endpoint Complete (ENDPTCOMPLETE)

This register is only used in device mode.

Offset

Register	Offset
ENDPTCOMPLETE	1BCh

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 ETCE	Endpoint Transmit Complete Event Flag Indicates whether a transmit event (Out or Setup) occurred. You must read the corresponding endpoint queue to determine the endpoint status. If the corresponding IOC bit is set in the transfer descriptor, then this field is set along with USBSTS[UI]. ETCE[*n*]: Endpoint [*n*], *n* is in 0–7 **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When reading 0000_0000 - Transmit did not complete 0000_0001 - Transmit completed When writing 0000_0000 - No effect 0000_0001 - Clear the flag
15-8 —	Reserved
7-0 ERCE	Endpoint Receive Complete Event Flag Indicates whether a received event (Out or Setup) occurred. You must read the corresponding endpoint queue to determine the transfer status. If the corresponding IOC bit is set in the transfer descriptor, then this field is set along with USBSTS[UI]. ERCE[*n*]: Endpoint [*n*], *n* is in 0–7 **NOTE** This field behaves differently for register reads and writes. When reading 0000_0000 - Receive did not complete 0000_0001 - Receive completed When writing 0000_0000 - No effect 0000_0001 - Clear the flag

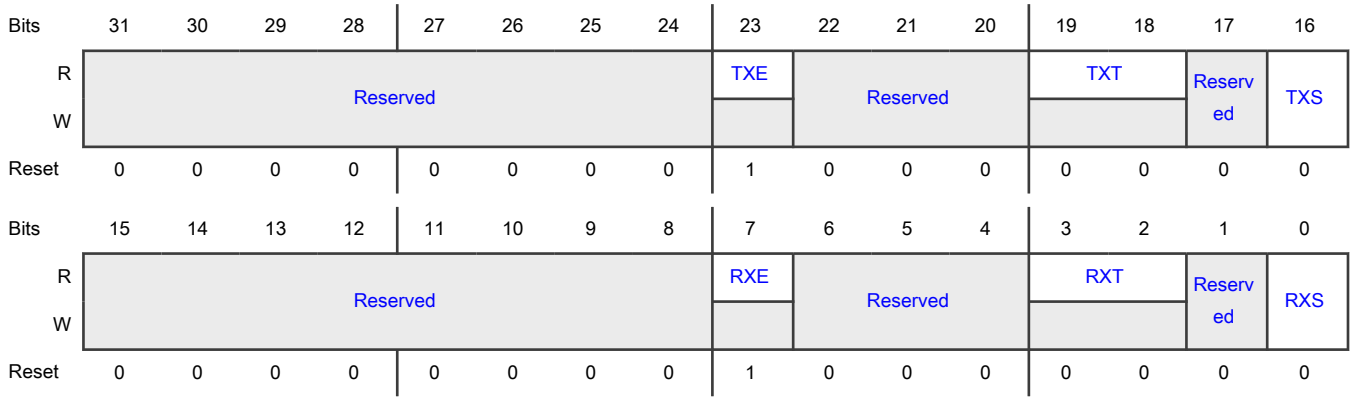
41.7.1.1.40 Endpoint Control 0 (ENDPTCTRL0)

Forces the endpoint to return a STALL handshake to the host. Every device uses this register as a control endpoint.

Offset

Register	Offset
ENDPTCTRL0	1C0h

Diagram



Fields

Field	Description
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables transmit endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
22-20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. If this field is 0, endpoint 0 is fixed as a control endpoint.
17 —	Reserved
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. If you write 1 to this field, it forces the endpoint to return a STALL handshake to the host. The field continues returning a STALL response until you write 0 to it or if the field automatically becomes 0 after it receives a new Setup request. After receiving a Setup request, hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) is cleared.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and the hardware continuing to write 0 to this field. In most systems, it is unlikely for the DCD to observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1 or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled
15-8 —	Reserved
7 RXE	RX Endpoint Enable Enables the receive endpoint. Endpoint 0 is always enabled. 0 - Disable 1 - Enable
6-4 —	Reserved
3-2 RXT	RX Endpoint Type Indicates the receive endpoint type. If this field is 0, endpoint 0 is fixed as a control endpoint.
1 —	Reserved
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled. If you write 1 to this field, it forces the endpoint to return a STALL handshake to the host. The field continues returning a STALL response until you write 0 to it or if the field automatically becomes 0 after it receives a new Setup request. After receiving a Setup request, hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) is cleared.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and the hardware continuing to write 0 to this field. In most systems, it is unlikely for the DCD to observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled

41.7.1.1.41 Endpoint Control 1 (ENDPTCTRL1)

Enables endpoint control 1.

This register is used for endpoint 1 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Offset

Register	Offset
ENDPTCTRL1	1C4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TXE		TXI	Reserved	TXT		TXD	TXS
W										TXR						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RXE		RXI	Reserved	RXT		RXD	RXS
W										RXR						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables transmit endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets
21 TXI	TX Data Toggle Inhibit Inhibits transmit data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet. 0 - Allow 1 - Inhibit
20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. This field automatically becomes 1 after receiving a "Setup" request if this endpoint is configured as a control endpoint. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or if it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (of a maximum of 50 clocks) between the clearing of ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled
15-8 —	Reserved
7 RXE	RX Endpoint Enable Enables the receive endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
6 RXR	RX Data Toggle Reset Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0 - Allow

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Inhibit
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled

41.7.1.1.42 Endpoint Control 2 (ENDPTCTRL2)

Enables endpoint control 2.

This register is used for endpoint 2 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Offset

Register	Offset
ENDPTCTRL2	1C8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables transmit endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets

Table continues on the next page...

Table continued from the previous page...

Field	Description
21 TXI	TX Data Toggle Inhibit Inhibits transmit data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to it causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet. 0 - Allow 1 - Inhibit
20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. This field automatically becomes 1 after receiving a Setup request if this endpoint is configured as a control endpoint. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or if it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (of a maximum 50 clocks) between the clearing of ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled
15-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7 RXE	RX Endpoint Enable Enables receive endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
6 RXR	RX Data Toggle Reset Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0 - Allow 1 - Inhibit
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0	RX Endpoint Stall

Table continues on the next page...

Table continued from the previous page...

Field	Description
RXS	Specifies whether the receive endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled

41.7.1.1.43 Endpoint Control 3 (ENDPTCTRL3)

Enables endpoint control 3.

This register is used for endpoint 3 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Offset

Register	Offset
ENDPTCTRL3	1CCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables the transmit endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets
21 TXI	TX Data Toggle Inhibit Inhibits transmit data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet. 0 - Allow 1 - Inhibit
20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled
15-8 —	Reserved
7 RXE	RX Endpoint Enable Enables the receive endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
6 RXR	RX Data Toggle Reset Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0 - Allow 1 - Inhibit

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled

41.7.1.1.44 Endpoint Control 4 (ENDPTCTRL4)

Enables endpoint control 4.

This register is used for endpoint 4 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Offset

Register	Offset
ENDPTCTRL4	1D0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables the transmit endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets
21 TXI	TX Data Toggle Inhibit Inhibits transmit data toggle sequencing. This field is used only for testing; you must always write 0 to this field. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Allow 1 - Inhibit
20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. This field automatically becomes 1 after receiving a Setup request if this endpoint is configured as a control endpoint. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or if it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (of maximum 50 clocks) between the clearing of ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely for the DCD to observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled
15-8 —	Reserved
7 RXE	RX Endpoint Enable Enables the receive endpoint.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
6 RXR	RX Data Toggle Reset Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to it causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0 - Disable 1 - Enable
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled. This field automatically becomes 1 after receiving a Setup request if this endpoint is configured as a control endpoint. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (of maximum 50 clocks) between the clearing of ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely for the DCD to observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled

41.7.1.1.45 Endpoint Control 5 (ENDPTCTRL5)

Enables endpoint control 5.

This register is used for endpoint 5 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Offset

Register	Offset
ENDPTCTRL5	1D4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables the transmit endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets
21 TXI	TX Data Toggle Inhibit Inhibits transmit data toggle sequencing. This field is used only for testing; you must always write 0 to this field. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet. 0 - Allow 1 - Inhibit
20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. This field automatically becomes 1 after receiving a Setup request if this endpoint is configured as a control endpoint. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (of a maximum 50 clocks) between the clearing of ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled
15-8 —	Reserved
7 RXE	RX Endpoint Enable Enables the receive endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
6 RXR	RX Data Toggle Reset Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0 - Allow

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Inhibit
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between the clearing of ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled

41.7.1.1.46 Endpoint Control 6 (ENDPTCTRL6)

Enables endpoint control 6.

This register is used for endpoint 6 in Device Operation mode.

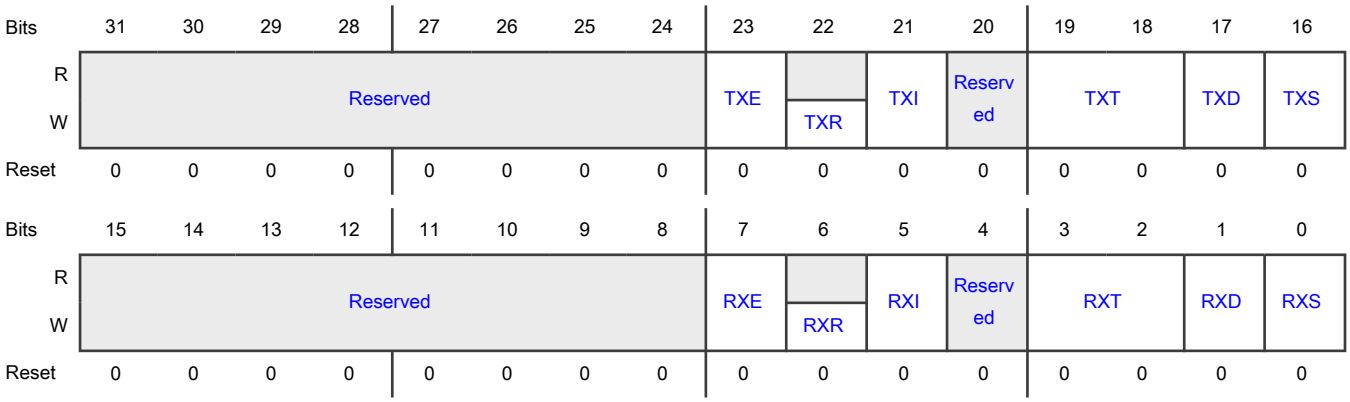
NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Offset

Register	Offset
ENDPTCTRL6	1D8h

Diagram



Fields

Field	Description
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables the transmit endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets

Table continues on the next page...

Table continued from the previous page...

Field	Description
21 TXI	TX Data Toggle Inhibit Inhibits transmit data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet. 0 - Allow 1 - Inhibit
20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled
15-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7 RXE	RX Endpoint Enable Enables the receive endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
6 RXR	RX Data Toggle Reset Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0 - Allow 1 - Inhibit
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0	RX Endpoint Stall

Table continues on the next page...

Table continued from the previous page...

Field	Description
RXS	Specifies whether the receive endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled

41.7.1.1.47 Endpoint Control 7 (ENDPTCTRL7)

Enables endpoint control 7.

This register is used for endpoint 7 in Device Operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of the opposite direction is disabled, then the unused direction type must be changed from the default control type to any other type (bulk type). Leaving an unconfigured endpoint control causes undefined behavior for the data PID tracking on the active endpoint or direction.

Offset

Register	Offset
ENDPTCTRL7	1DCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TXE		TXI	Reserved	TXT		TXD	TXS
W										TXR						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RXE		RXI	Reserved	RXT		RXD	RXS
W										RXR						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23 TXE	TX Endpoint Enable Enables the transmit endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
22 TXR	TX Data Toggle Reset Resets transmit data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets
21 TXI	TX Data Toggle Inhibit Inhibits transmit data toggle sequencing. This field is only used for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet. 0 - Allow 1 - Inhibit
20 —	Reserved
19-18 TXT	TX Endpoint Type Indicates the transmit endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
17 TXD	TX Endpoint Data Source Specifies the transmit endpoint data source. This field is 0 by default, which refers to dual port memory buffer and DMA engine. You must always write 0 to this field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
16 TXS	TX Endpoint Stall Specifies whether the transmit endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled
15-8 —	Reserved
7 RXE	RX Endpoint Enable Enables the receive endpoint. You must enable an endpoint only after it is configured. 0 - Disable 1 - Enable
6 RXR	RX Data Toggle Reset Resets receive data toggle. If you write 1 to this field, it resets the PID sequence. When a configuration event is received for this endpoint, you must write 1 to this field to synchronize the data PIDs between the host and device. 0 - Does not reset 1 - Resets
5 RXI	RX Data Toggle Inhibit Inhibits receive data toggle sequencing. This field is used only for testing; you must always write 0 to it. Writing 1 to this field causes this endpoint to ignore the data toggle sequence and always accept data packets regardless of their data PID. 0 - Allow

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Inhibit
4 —	Reserved
3-2 RXT	RX Endpoint Type Specifies the receive endpoint type. 00 - Control 01 - Isochronous 10 - Bulk 11 - Interrupt
1 RXD	RX Endpoint Data Sink Specifies the data sink of the RX endpoint. By default, this field is 0, which refers to dual port memory buffer or DMA engine. You must always write 0 to this field.
0 RXS	RX Endpoint Stall Specifies whether the receive endpoint is stalled. If this endpoint is configured as a control endpoint, this field automatically becomes 1 after receiving a Setup request. Hardware continues to write 0 to this field until the associated endpoint setup status field (ENDPTSETUPSTAT[ENDPTSETUPSTAT]) becomes 0. You can write 1 to this field to force the endpoint to return a STALL handshake to the host. This control continues to stall until you write 0 to this field or when it becomes 0 automatically for control endpoints. **NOTE** Control endpoint only: There is a slight delay (a maximum of 50 clocks) between clearing ENDPTSETUPSTAT[ENDPTSETUPSTAT] and hardware continuing to write 0 to this field. In most systems, it is unlikely that the DCD will observe this delay. However, if DCD observes that this field is not becoming 1, you must continually write to this field until it becomes 1, or until a new setup is received (check ENDPTSETUPSTAT[ENDPTSETUPSTAT] for that). 0 - Endpoint OK 1 - Endpoint stalled

41.7.2 USBNC memory map

41.7.2.1 USBNC register descriptions

41.7.2.1.1 USBNC memory map

USB.USBNC_OTG1 base address: 4C10_0200h

USB.USBNC_OTG2 base address: 4C20_0200h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	USB OTG Control 1 (CTRL1)	32	RW	3000_1000
4	USB OTG Control 2 (CTRL2)	32	RW	DF00_0000
30	PHY Configure 1 (UTMIPHY_CFG1)	32	RW	1533_6DC5
34	PHY Configure Register 2 (UTMIPHY_CFG2)	32	RW	9019_0020
3C	PHY Status Register (UTMIPHY_STATUS)	32	R	See section

41.7.2.1.2 USB OTG Control 1 (CTRL1)

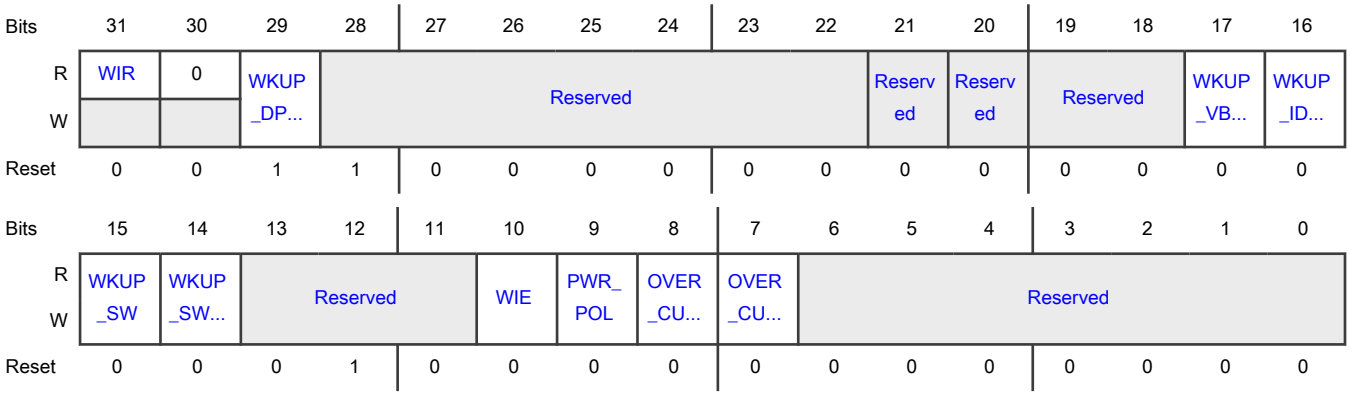
Controls the integration-specific features of the USB OTG1 and OTG2 modules.

These features are not directly related to the USB functionality, but they control special features, interfacing on the USB ports, and power control and wake-up functionality.

Offset

Register	Offset
CTRL1	0h

Diagram



Fields

Field	Description
31 WIR	Wake-Up Interrupt Request Indicates whether a wake-up interrupt request is received. This field becomes 0 by disabling the wake-up interrupt (writing 0 to CTRL1[WIE]). This field indicates that a wake-up interrupt request is received on the OTG1 port and becomes 0 by disabling the wake-up interrupt (writing 0 to the OWIE field). 0 - No wake-up interrupt request received

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Wake-up interrupt request received
30 —	Reserved
29 WKUP_DPDM_EN	Wake-Up on DP/DM Change Enable Enables wake-up on DP/DM change. If this field is 0, wake-up on DP/DM changes are to be disabled only when VBUS is 0. If this field is 1, wake-up on DP/DM changes are to be enabled (default) (it is for device only). 0 - Disable 1 - Enable
28-22 —	Reserved
21 —	Reserved
20 —	Reserved
19-18 —	Reserved
17 WKUP_VBUS_EN	Wake-Up After VBUS Change Enable Enables wake-up after VBUS change. 0 - Disable 1 - Enable
16 WKUP_ID_EN	Wake-Up ID Enable Enables wake-up on ID change. 0 - Disable 1 - Enable
15 WKUP_SW	Software Wake-Up Forces software wake-up. 0 - Inactive 1 - Force wake-up
14 WKUP_SW_EN	Software Wake-Up Enable Enables software wake-up.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Disable 1 - Enable
13-11 —	Reserved
10 WIE	Wake-Up Interrupt Enable Enables the OTG1 wake-up interrupt. CTRL1[WIR] also becomes 0 when disabling the interrupt. Prior to entering Suspend mode, you must turn off the wake-up interrupt enable after receiving a wake-up interrupt and turn it on again. 0 - Disable 1 - Enable
9 PWR_POL	Power Polarity Specifies the PMIC power pin polarity. 0 - Active low 1 - Active high
8 OVER_CUR_POL	Overcurrent Polarity Specifies the polarity of the OTG1 and OTG2 port overcurrent event, representing an overcurrent condition. 0 - High active (high on this signal represents an overcurrent condition) 1 - Low active (low on this signal represents an overcurrent condition)
7 OVER_CUR_DIS	Overcurrent Disable Disables overcurrent detection. 0 - Enable 1 - Disable
6-0 —	Reserved

41.7.2.1.3 USB OTG Control 2 (CTRL2)

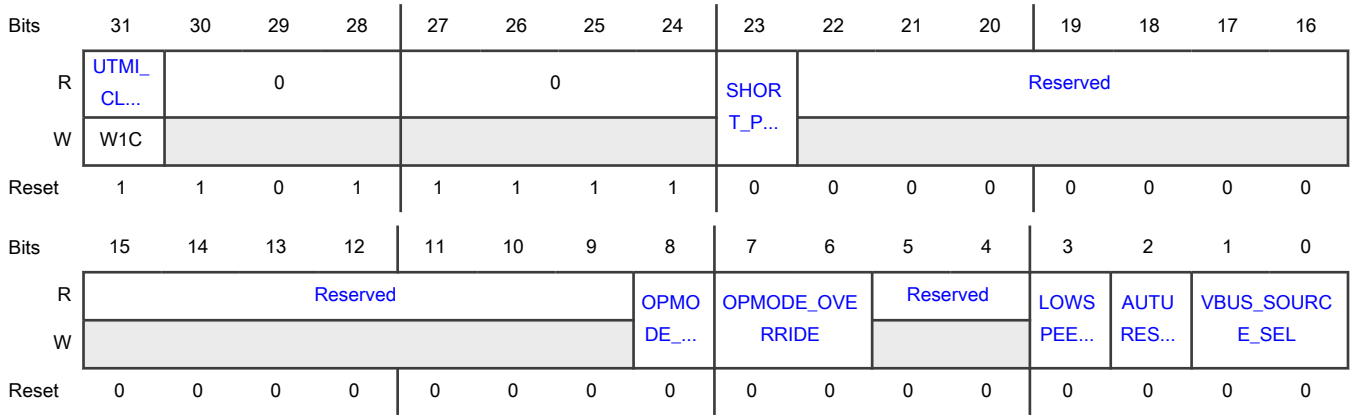
Controls the integration-specific features of the USB module.

These features are not directly related to USB functionality, but they control special features, interfacing on the USB ports, and power control and wake-up functionality.

Offset

Register	Offset
CTRL2	4h

Diagram



Fields

Field	Description
31 UTMI_CLK_VLD	UTMI Clock Valid Flag Indicates whether the UTMI clock from the USB PHY is valid. 0 - Default
30-28 —	Reserved
27-24 —	Reserved
23 SHORT_PKT_EN	Short Packet Interrupt Enables short packet interrupt. When enabled, the USB controller issues an interrupt when a short packet is received in Device mode, even if IoC of QTD is not set. 0 - Default
22-9 —	Reserved
8 OPMODE_OVERRIDEEN	UTMI_OpMode[1:0] Override Enable Overrides the controller's utmi_OpMode[1:0] signal to the PHY. 0 - Override disable 1 - Override enable
7-6 OPMODE_OVERRIDE	UTMI_OpMode[1:0] Override Value Overrides the controller's utmi_OpMode[1:0] signal to the PHY. 00 - Override to 2'b00

Table continues on the next page...

Table continued from the previous page...

Field	Description
	01 - Override to 2'b01 10 - Override to 2'b10 11 - Override to 2'b11
5-4 —	Reserved
3 LOWSPEED_EN	Low-Speed Enable Enables low speed. This field is 1 when AUTURESUME_EN is 1 and works on low speed. 0 - Default
2 AUTURESUME_EN	Auto Resume Enable Enables auto resume. This field is for the OTG1/OTG2 core UTMI PHY. The UH core has no such feature because the HSIC PHY does not support auto resume. To set USB, it sends resume with a 32 KHz clock when it detects remote wake-up from device. This feature is useful if the device drives a very short remote wakeup (minimum value is 1 ms for the USB2 spec) and system 24 MHz is off during Suspend mode. 0 - Default
1-0 VBUS_SOURCE_SEL	VBUS Source Select Specifies VBUS source select when detecting a VBUS wake-up event; it is for the UTMI PHY only. 00 - vbus_valid 01 - sess_valid 10 - sess_valid 11 - sess_valid

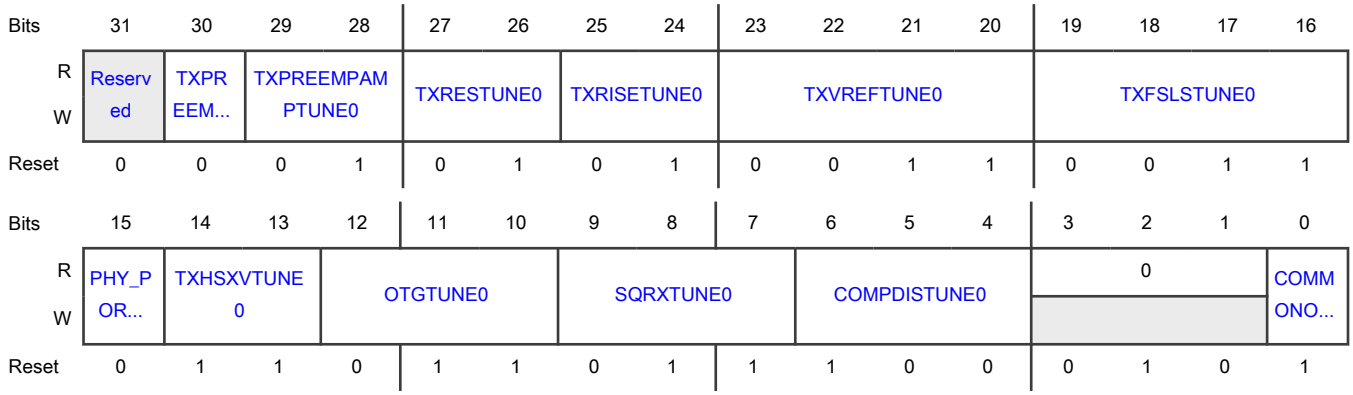
41.7.2.1.4 PHY Configure 1 (UTMIPHY_CFG1)

Holds the internal control register bits of the PHY clock.

Offset

Register	Offset
UTMIPHY_CFG1	30h

Diagram



Fields

Field	Description
31 —	Reserved
30 TXPREEMPPULSETUNE0	HS Transmitter Pre-Emphasis Duration Control Controls the duration for which the HS pre-emphasis current is sourced onto DP or DM. The HS transmitter pre-emphasis duration is defined in terms of unit amounts. One unit of pre-emphasis duration is approximately 580 ps and is defined as 1X pre-emphasis duration. This signal is valid only if either TXPREEMPAMPTUNE or TXPREEMPAMPTUNE is 1'b1. If TXPREEMPPULSETUNE is not used, set it to the design default. 0 - Design default; long pre-emphasis current duration 1 - Short pre-emphasis current duration
29-28 TXPREEMPAMPTUNE0	HS Transmitter Pre-Emphasis Current Control Controls the amount of current sourced to DP and DM after a J-to-K or K-to-J transition. The HS transmitter pre-emphasis current is defined in terms of unit amounts. One unit amount is approximately 600 µA and is defined as 1X pre-emphasis current. If these signals are not used, set them to the design default (2'b01). 00 - HS transmitter pre-emphasis is disabled 01 - HS transmitter pre-emphasis circuit sources 1x pre-emphasis current 10 - HS transmitter pre-emphasis circuit sources 2x pre-emphasis current 11 - HS transmitter pre-emphasis circuit sources 3x pre-emphasis current
27-26 TXRESTUNE0	USB Source Impedance Adjustment Adjusts the driver source impedance to compensate for added series resistance on the USB (in some applications, there can be significant series resistance on the D+ and D– paths between the transceiver and cable).

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE Any setting other than the default can result in source impedance variation across process, voltage, and temperature conditions that do not meet the USB 2.0 specification limits. If this bus is not used, leave it at the default setting. 00 - Source impedance increased by approximately 1.3 Ω 01 - 0, design default 10 - Source impedance decreased by approximately 3.6 Ω 11 - Source impedance decreased by approximately 5.6 Ω
25-24 TXRISETUNE0	HS Transmitter Rise and Fall Time Adjustment Adjusts the rise and fall times of the high-speed waveform. 00 - +5.4% 01 - 0, design default 10 - -7.2% 11 - -8.1%
23-20 TXVREFTUNE0	HS DC Voltage Level Adjustment Adjusts the high-speed DC level voltage. When the PHY is in a host role where the HOSTDISCONNECT signal is used, any adjustment of the HS transmit voltage level tune bits (TXVREFTUNE) might also necessitate a change to the HS disconnect detection threshold tune bits (COMPDISTUNE). This change may be required because the HS disconnect detection threshold might need to be increased (or decreased) if the HS transmit voltage level is increased (or decreased). 0000 - -6% 0001 - -4% 0010 - -2% 0011 - 0, design default 0100 - +2% 0101 - +4% 0110 - +6% 0111 - +8% 1000 - +10% 1001 - +12% 1010 - +14% 1011 - +16% 1100 - +18% 1101 - +20%

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1110 - +22% 1111 - +24%
19-16 TXFSLSTUNE0	FS/LS Source Impedance Adjustment Adjusts the low- and full-speed single-ended source impedance when driving high. The following adjustment values are based on nominal process, voltage, and temperature. 0000 - +6.27% 0001 - +3.02% 0011 - 0, design default 0111 - -3.23% 1111 - -6.25%
15 PHY_POR_SW	PHY Software POR Specifies whether you must perform the PHY power-on reset. This signal resets all test registers and state machines in the USB 2.0 femtoPHY. The POR signal must be asserted for a minimum of 10 µs. 0 - Do not perform 1 - Perform
14-13 TXHSXVTUNE0	Transmitter High-Speed Crossover Adjustment Adjusts the voltage at which the DP and DM signals cross when transmitting in HS mode. 00 - Reserved 01 - -17.31 mV 10 - -16.69 mV 11 - 0, design default
12-10 OTGTUNE0	VBUS Valid Threshold Adjustment Adjusts the voltage level for the VBUS valid threshold. If this bus is not used, leave it at the default setting (011). 000 - -9% 001 - -6% 010 - -3% 011 - 0, design default 100 - +3% 101 - +6% 110 - +9% 111 - +12%
9-7	Squelch Threshold Adjustment

Table continues on the next page...

Table continued from the previous page...

Field	Description
SQRXTUNE0	Adjusts the voltage level for the threshold used to detect valid high-speed data. 000 - +15.19% 001 - +10.04% 010 - +5.14% 011 - 0, design default 100 - -5.04% 101 - -10.19% 110 - -15.09% 111 - -20.24%
6-4 COMPDISTUNE0	Disconnect Threshold Adjustment Adjusts the voltage level for the threshold used to detect a disconnect event at the host. If this bus is not used, leave it at the default setting. When the PHY is in a host role where the HOSTDISCONNECT signal is used, any adjustment of the HS transmit voltage level tune bits (TXVREFTUNE) can also necessitate a change to the HS disconnect detection threshold tune bits (COMPDISTUNE). This change might be required because the HS disconnect detection threshold may need to be increased (or decreased) if the HS transmit voltage level is increased (or decreased). The default value is 3'b100. 000 - -9.71% 001 - -6.85% 010 - -3.36% 011 - 0, design default 100 - +4.04% 101 - +8.22% 110 - +13.18% 111 - +18.39%
3-1 —	Reserved
0 COMMONONN	Common Block Power-Down Control This bit is muxed with ipt_mode and then connected to COMMONONN of the USBPHY. This signal controls the power-down signals in the REFCLK_LOGIC, Bias, and PLL blocks when the USB 2.0 femtoPHY is in Suspend, or Sleep mode. 0 - In Suspend or Sleep modes, the REFCLK_LOGIC, Bias, and PLL blocks remain powered. With this setting, the input reference clock must remain on and valid during suspend or sleep. 1 - In Suspend mode, the REFCLK_LOGIC, Bias, and PLL blocks are powered down. In Sleep mode, the Bias and PLL blocks are powered down.

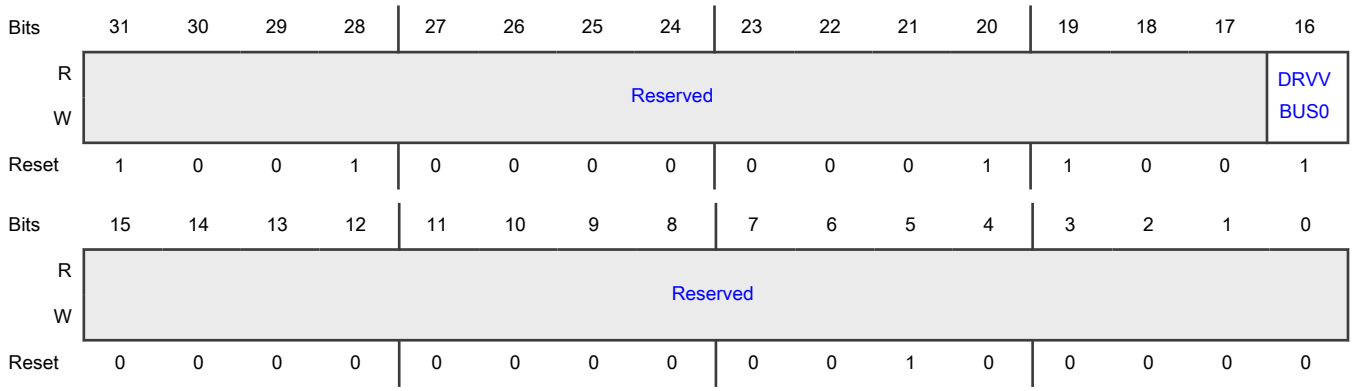
41.7.2.1.5 PHY Configure Register 2 (UTMIPHY_CFG2)

This register is used to hold the internal control register bits of the PHY.

Offset

Register	Offset
UTMIPHY_CFG2	34h

Diagram



Fields

Field	Description
31-17 —	Reserved
16 DRVVBUS0	Drive VBUS This controller signal controls the VBUS Valid comparator. This signal drives 5 V on VBUS through an external charge pump. When OTGDISABLE0 is set to 1'b0 and DRVVBUS0 is asserted, the Bandgap circuitry and VBUS Valid comparator are powered, even in Suspend or Sleep mode. 0 - The VBUS Valid comparator is disabled. 1 - The VBUS Valid comparator is enabled.
15-0 —	Reserved

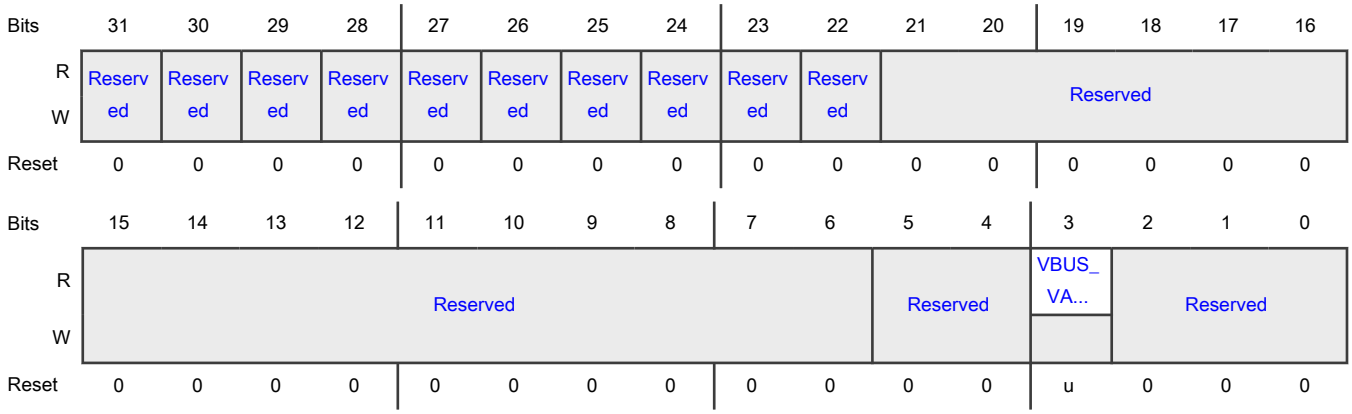
41.7.2.1.6 PHY Status Register (UTMIPHY_STATUS)

This register is used to hold the internal status register bits of the PHY.

Offset

Register	Offset
UTMIPHY_STATUS	3Ch

Diagram



Fields

Field	Description
31 —	Reserved
30 —	Reserved
29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25 —	Reserved
24 —	Reserved
23 —	Reserved
22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
21-6 —	Reserved
5-4 —	Reserved
3 VBUS_VALID	VBUS Valid Signal VBUS valid signal from PHY directly.
2-0 —	Reserved

Chapter 42
Universal Serial Bus 2.0 PHY (USB2_PHY)

42.1 Overview

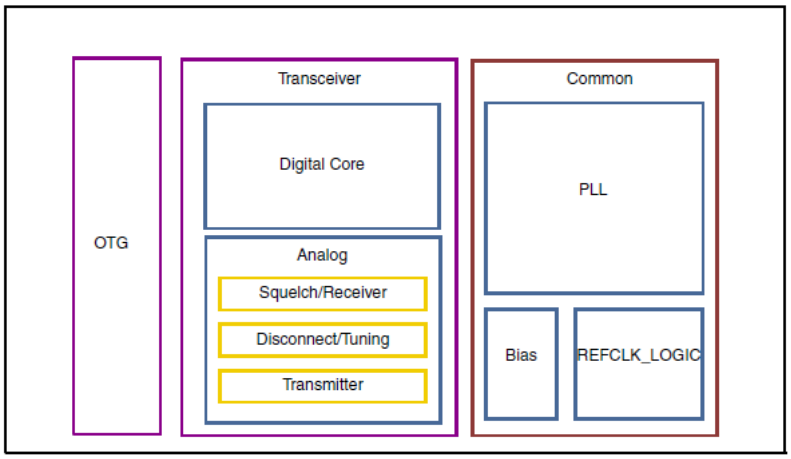
The USB 2.0 PHY, connects a USB controller to a USB system. This chapter provides an introduction to the USB 2.0 PHY and its features.

NOTE

The sections in this chapter are Synopsys Proprietary. Used with permission.

42.1.1 Block Diagram

The following figure shows the USB 2.0 PHY functional block diagram for a one-port macro.



42.1.1.1 System Level Block Diagram

The following figure shows the USB 2.0 PHY and its external interface connections.

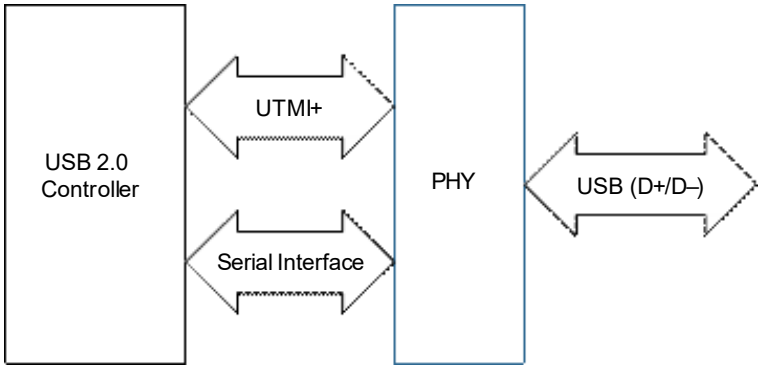


Figure 171. System Level Block Diagram

42.1.2 Features

The USB 2.0 PHY provides the following features.

General Features:

- Implements low power dissipation while active, idle, or on standby
- Fully integrates high-, full-, and low-speed (Host mode only) termination and signal switching
- Implements one parallel data interface and clock for high-, full-, and low-speed (Host mode only) USB data transfers
- Provides on-chip PLL to reduce clock noise and eliminate the need for an external clock generator
- Provides Universal Asynchronous Receiver/Transmitter (UART) support to enable the USB 2.0 PHY to receive and transmit asynchronous, serial data

USB 2.0 Features:

- Supports 480-Mbps high-speed, 12-Mbps full-speed, and 1.5-Mbps low-speed (Host mode only) data transmission rates
- Supports 8/16-bit unidirectional parallel interfaces for HS, FS, and LS (Host mode only) modes of operation, in accordance with the UTMI+ specification
- Provides dual (HS/FS) mode host support
- Implements data recovery from serial data on the USB connector
- Implements SYNC/End-of-Packet (EOP) generation and checking
- Implements Non Return to Zero Invert (NRZI) encoding and decoding Implements logic to support suspend, sleep, resume, and remote wakeup operations
- Supports battery charging to enable devices to draw current in excess of the USB 2.0 specification for charging and/or powering up from dedicated charging ports or charging downstream ports

42.1.3 Interfaces

Each USB 2.0 PHY port has three distinct external interfaces:

- USB Data Plus (D+) and Data Minus (D–) lines: These lines are USB 1.1 and 2.0 specificationcompliant. The USB 2.0 PHY supports high-speed, 480-Mbps transfers, as well as USB 1.1 fullspeed and low-speed transfers through the USB.
- USB 2.0 Transceiver Macrocell Interface (UTMI+): The USB 2.0 PHY supports the following modes through the UTMI+:
  - High-Speed (HS)
  - Full-Speed (FS)
  - Low-Speed Preamble (LS Preamble)

  The UTMI+ contains a receive port, a transmit port, and associated control lines to interface with a USB host controller or device controller. The receive and transmit ports can be configured as 8/16-bit parallel ports for all applicable modes of operation.
- Serial interface: This is the USB 1.1 controller interface that supports full-speed (FS-Serial mode) and low-speed (LS-Serial mode) data transmission rates to and from a USB 1.1 controller.

The USB 2.0 PHY handles low-level USB protocol and signaling. The USB 2.0 PHY supports SYNC detection, data serialization and deserialization, and data recovery.

42.1.4 Standards Compliance

The USB 2.0 PHY complies with the following standards:

- Universal Serial Bus Specification, Revision 2.0, with a High-Speed (HS) Transmit to Receive Inter- Packet Delay requirement of 38 bit-times provided by the UTMI specification version 1.05.
- Embedded Host Supplement to the USB Revision 2.0 Specification, Revision 2.0, USB Implementers Forum, Inc. ADP (Attach Detach Protocol) is not supported.
- UTMI+ Specification, Revision 1.0 (Level 3)

42.2 Clock Generation

The following figure shows the USB 2.0 PHY clock generation and distribution architecture, and the functional behavior of blocks that generate various clocks in the USB 2.0 PHY.

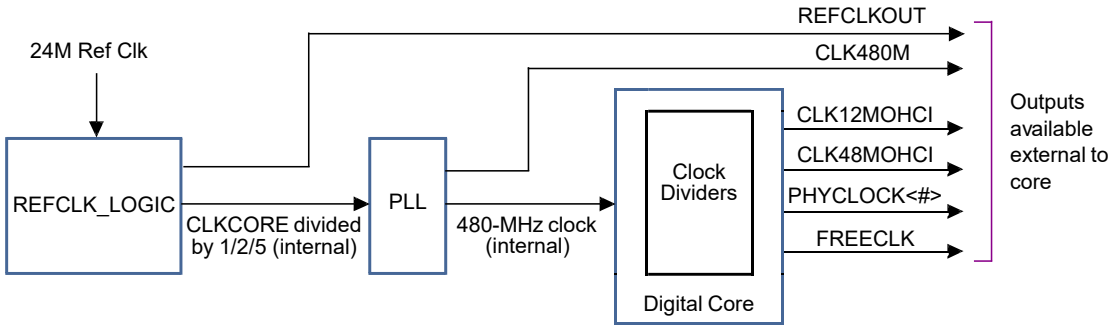


Figure 172. Clock Generation

42.3 Device Initialization

According to the USB 2.0 specification, a host must provide 100 ms between first detecting the D+ line at a high state and initiating a USB reset (SE0). The device pulling up the D+ line is expected to complete initialization within the 100-ms interval and be ready to respond to the host.

If the device requires more than 100 ms to initialize the device might connect the D+ resistance prior to initialization. After the 100-ms interval, the host initiates a USB reset. If the D+ pull-up resistance is enabled and remains enabled after the 100-ms interval, and if the device has not completed initialization, the device does not respond to the host correctly, and the transaction times out.

42.4 Tuning

The USB 2.0 PHY has an internal tuning circuit that ensures that the driver's impedance is within the USB 2.0 specification.

The USB 2.0 PHY performs two types of tuning:

- Incremental tuning: Incremental tuning occurs during an EOP, SOF, or SYNC pattern.
- Full tuning: Full tuning occurs during a power-on reset or after resume operation.

Chapter 43
Ethernet Controller with TSN (ENET_QoS)

43.1 Chip-specific ENET information

Table 361. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing

The chip includes has one 1G-bit Ethernet with AVB support (ENET) plus a separate 1Gbit Ethernet QoS with TSN support.

Table 362. Ethernet QoS with TSN Configuration

Parameter	Configuration	Description
RXFIFO_SIZE	8192	Receive FIFO Size (in Bytes)
TXFIFO_SIZE	8192	Transmit FIFO Size (in Bytes)
L3_L4_FILTER_NUM	8	Number of Layer 3 and Layer 4 Packet Filters
NUM_TXQ	5	Number of Transmit Queues
NUM_RXQ	5	Number of Receive Queues
EST_DEP	512	Depth of one instance of Gate Control List

43.3 Overview

NOTE

This section and all its sub-sections are Synopsys Proprietary. Used with permission.

The EQoS-TSN module supports the 10/100/1000Mbps applications in compliance with IEEE802.3-2015 specifications. It can support advanced applications such as Time-Sensitive Networking (TSN) and Audio Video Bridge (AVB). Other prominent features of the MAC core include a flexible receive parser, media clock recovery and generation, and safety.

43.3.1 Block diagram

The following figure shows the block diagram of this module. The module has four main blocks to perform all functions. The AHB interface is connected to all DMA channels. The DMA arbiter helps in arbitration of all paths (transmit and receive) in channels. Each channel has a separate set of control and status registers (CSR) for managing the transmit and receive functions, descriptor handling, and interrupt handling.

NOTE

In multi queue/channel configurations, enable only Q0/CH0 on Tx and Rx for half-duplex operation. If you want to enable single queue/channel in full-duplex operation, any queue/channel can be enabled.

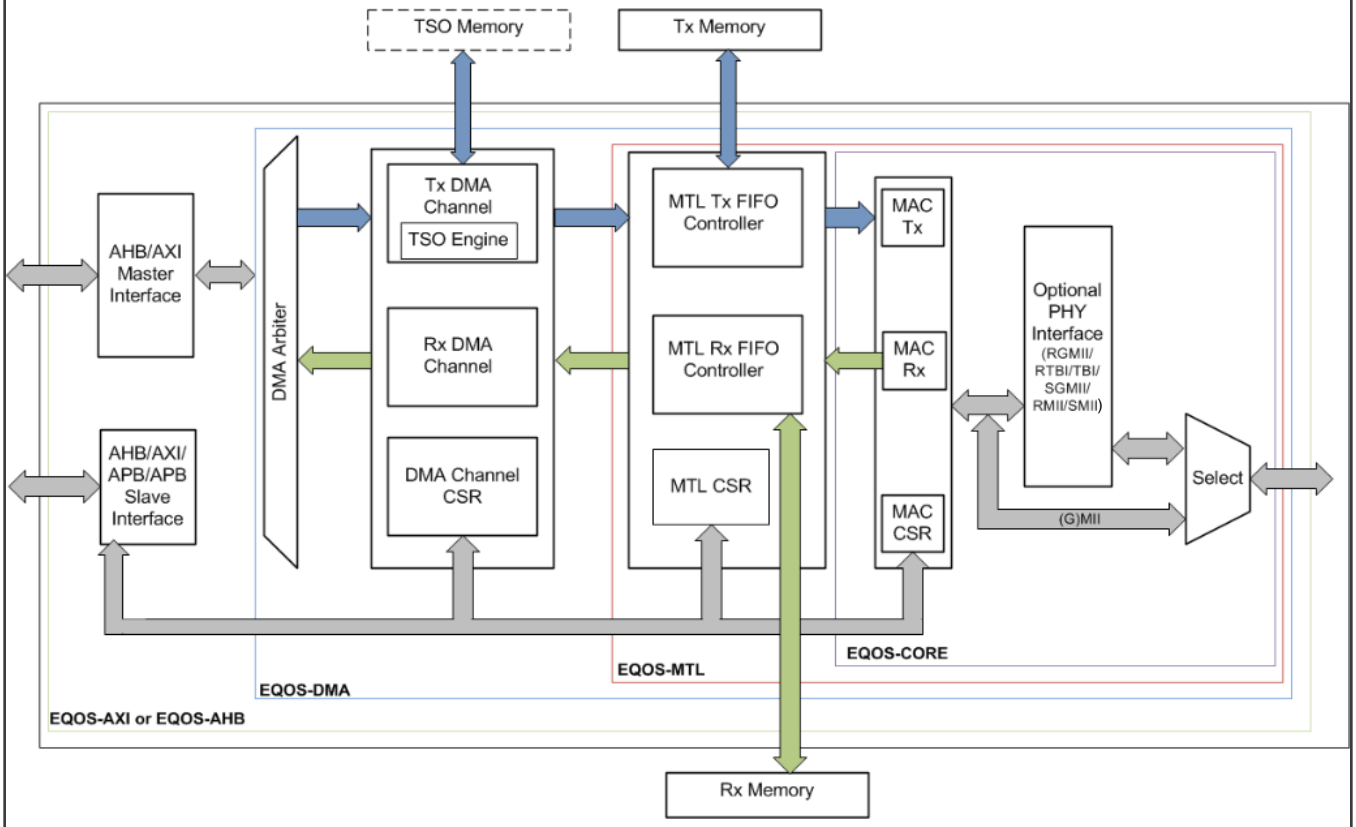


Figure 173. System-Level Block Diagram

43.3.2 Features

This module supports the following features in addition to the default features defined in IEEE802.3 specification

- RMII (10/100Mbps), RGMII (10/100/1000Mbps)
- Time aware shaper (IEEE802.1Qbv), Time synchronization (IEEE1588-2008), and Frame pre-emption (IEEE802.1Qbu) for Time-Sensitive Networking (TSN)
- Media clock recovery and generation for AVB
- RMII specification version 1.2 from RMII consortium
- RGMII specification version 2.6 from HP/Marvell
- Separate interface for Transmit, Receive and Control path
- Broadcast/Multicast Packet duplication
- Full duplex flow control operations (IEEE 802.3x pause packet and priority flow control)
- Network statistics with MAC management (RMON) counters
- Flexibility to control the Pulse per second (PPS) output signal
- MDIO (clause 22 and Clause 45) interface for configuration and management of PHY device
- Preamble and start of packet data (SFD) insertion/deletion

- Automatic CRC and pad generation/stripping option, option to disable CRC checking
- Programmable inter-packet gap
- Source address insertion/replacement
- VLAN insertion, replacement and deletion in transmitted packets with per-packet or static-global control
- Insertion, replacement and deletion of up to two VLAN tags
- Insertion, replacement and deletion of queue/channel based VLAN tags
- IEEE802.1Q VLAN tag detection and option to delete from received packets
- Programmable safety watchdog timeout limits
- Flexible address filtering modes:
  - Up to 2 additional 48-bit perfect (DA) address filters with masks for each byte
  - Up to 2 48-bit SA address comparison check with masks for each byte
  - 64-bit Hash filter for multicast and unicast (DA) addresses
  - Option to pass all multi-cast addressed packets
  - Promiscuous mode to pass all packets without any filtering for network monitoring
  - Pass all incoming packets (as per filter) with a status report
- Additional packet filtering:
  - VLAN tag-based: Perfect match and Hash-based filtering. Filtering based on either outer or inner VLAN tag is possible
  - Layer 3 and Layer 4-based: TCP or UDP over IPv4 or IPv6
  - Extended VLAN tag based filtering 4 filter selections

43.4 Functional description

NOTE

This section and all its sub-sections are Synopsys Proprietary. Used with permission.

The following sections describe functional details of the ENET module.

43.4.1 Architecture

This section describes the interfaces, protocols, and functionality. Following are the main blocks of the ENET QOS block.

- DMA Controller
- MAC Transaction Layer (MTL)
- MAC
- Interrupts

43.4.1.1 DMA controller

The DMA has independent Transmit (Tx) and Receive (Rx) engines, and a control and status register (CSR) space. The Tx engine transfers data from the system memory to the MTL interface, whereas the Rx engine transfers data from the MTL interface to the system memory. The DMA engine uses descriptors to efficiently move data from source to destination with minimal Host intervention. The DMA is designed for packet-oriented data transfers such as packets in Ethernet. The DMA controller can be programmed to interrupt the Host for situations such as Packet Transmit and Receive Transfer completion, and other normal or error conditions.

The DMA and Host communicate through the following two data structures:

- Control and Status registers (CSR)
- Descriptor lists and data buffers

The DMA supports up to 5 Tx and 5 Rx Descriptor lists (or DMA channels). The base address of each list is written to the respective Tx Descriptor List Address register and Rx Descriptor List Address register. The descriptor list is forward linked and the next descriptor is always considered at a fixed offset to the current one. The offset is controlled by the DSL field of DMA_Ch[n]_Control register. The number of descriptors in the list is programmed in the respective Tx (or Rx) Descriptor Ring Length register. Once the DMA processes the last descriptor in the list, it automatically jumps back to the descriptor in the List Address register to create a descriptor ring.

The descriptor lists reside in the physical memory address space of the application. Each descriptor can point to a maximum of two buffers in the system memory. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

A data buffer resides in the application physical memory space and consists of an entire packet or part of a packet but cannot exceed a single packet. Buffers contain only data. Buffer status is maintained in the descriptor. Data chaining refers to packets that span multiple data buffers. However, a single descriptor cannot span multiple packets. The DMA skips to the data buffer of next packet when end of packet (EOP) is detected.

This module supports the ring structure for the DMA descriptor. For more information on Descriptor, see the Descriptors .

Following sections describes the operation of DMA Controller.

43.4.1.1.1 DMA Controller Bus Burst Access

The DMA engines attempt to transfer data in a burst of maximum size as programmed in the PBL bit fields of Transmit Control and Receive Control registers of the respective DMA. The Rx and Tx descriptors are always accessed in the maximum possible (limited by PBL or 16 * 8/bus width) burst length for 16 bytes to be read. The burst transfers initiated by the DMA can be split into multiple burst transfers as per the AHB requirements and the settings of DMA_SysBus_Mode register.

The Tx DMA initiates a data transfer only when sufficient space is available in the MTL Tx Queue to accommodate either of the following:

- Bytes corresponding to the configured burst (PBL * bus_width/8)
- Remaining bytes in Tx Buffer without EOP
- Number of bytes till EOP

The Rx DMA initiates a data transfer in the following conditions:

- Sufficient data is available in MTL Rx Queue to accommodate the configured burst
- EOP (when it is less than the configured burst length) is detected in the Rx Queue

43.4.1.1.2 DMA application data buffer alignment

The Tx and Rx data buffers do not have any restrictions on start address alignment. For example, in systems with 32-bit memory, the start address for buffers can be aligned to any of the four bytes. However, the DMA always initiates write transfers with address aligned to the bus width and dummy data (old data) in the invalid byte lanes. This typically happens during the transfer of the beginning or end of an Ethernet packet. For the receiver, the software driver should discard the dummy bytes based on the start address of the buffer and size of the packet.

Table 363. Data buffer alignment examples

Examples	
Buffer Read	If the Tx buffer address is 32'h00000FF2 (for 32-bit data bus), and 15 bytes need to be transferred, the DMA reads five full words from address 32'h00000FF0, but when transferring data to the MTL Tx queue, the extra

Table continues on the next page...

Table 363. Data buffer alignment examples (continued)

	bytes (the first two bytes) are dropped or ignored. Similarly, the last 3 bytes of the last transfer are also ignored. The DMA always ensures that it transfers a full 32-bit data to the MTL Tx queue, unless it is the end of packet.
Buffer Write	If the Rx buffer address is 32'h0000FF2 (for 64-bit data bus) and 16 bytes of a received packet need to be transferred, the DMA writes 3 full words from address 32'h00000FF0. However, the first 2 bytes of the first transfer and the last 6 bytes of the third transfer have dummy data. The DMA considers the offset address only if it is the first Rx buffer of the packet. The DMA ignores the offset address and performs full word writes for the middle and the last Rx buffer of the packet.

43.4.1.1.3 DMA buffer size calculations

The DMA does not update the size fields in the Tx and Rx descriptors. The DMA updates only the status fields (RDES and TDES) of the descriptors. The driver has to perform the size calculations.

The Tx DMA transfers the exact number of bytes (indicated by buffer size field of TDES2) towards the MAC. If a descriptor is marked as first (FD bit of TDES3 is set), the DMA marks the first transfer from the buffer as SOP. If a descriptor is marked as last (LD bit of TDES3), the DMA marks the last transfer from that data buffer as EOP to the MTL.

The Rx DMA transfers data to a buffer until the buffer is full or the end of packet is received from the MTL. When the FD bit of a descriptor is set, the amount of valid data in a buffer is accurately indicated by the buffer size field (programmed in DMA Channel Receive Control register) minus the data buffer pointer offset. The offset is zero when the data buffer pointer is aligned to the data bus width. If a descriptor is marked as last, the buffer may not be full (as indicated by the buffer size in Bits [14:1] of Receive Control register). To compute the amount of valid data in this final buffer, the driver must read the packet length (PL bits of RDES3[14:0]) and subtract the sum of the buffer sizes of the preceding buffers in this packet. The Rx DMA always transfers the start of next packet with a new descriptor.

43.4.1.2 MAC transaction layer

The MAC Transaction Layer (MTL) provides the FIFO memory Interface to buffer and regulate the packets between system memory and the MAC. It also enables the data to be transferred between the system clock and MAC clock domains. The MTL layer has two data paths: Transmit path and Receive Path. The size of the FIFO is 8 Kb for Tx and Rx. The MTL communicates with the host through Application Transmit Interface (ATI) on the Transmit path and Application Receive Interface (ARI) on the receive path.

43.4.1.2.1 Transmit path

Internal DMA handles all transactions for the transmit path through ATI. Internal DMA pushes the Ethernet packets read from system memory into the corresponding queue. The packet is then popped out and transferred to the MAC when the queue threshold is reached (threshold mode) or complete packet is in the queue (store-and-forward mode). When EOP is transferred, the status of the transmission is taken from the MAC and transferred back to the internal DMA.

43.4.1.2.1.1 Transmit control word

The following control information related to packet transmission is provided as a part of the Control Word through ATI interface:

- Packet Length (if DCB is enabled with WFQ scheduling algorithm)
- CRC Pad Control
- Source Address Insertion Control
- VLAN Insertion and Replacement Control and VLAN Tag for Outer and Inner VLAN Tags
- TCP/IP Checksum Insertion Control
- One-step Timestamping Control Correction
- Transmit Timestamp Enable

43.4.1.2.1.2 Transmit operation

The following two modes of operation triggers reading of the data towards the MAC:

- Threshold mode: In Threshold (or cut-through) mode, as soon as the number of bytes in the Queue cross the configured threshold level (or when the end of packet is written before the threshold is crossed), the data is ready to be popped out and forwarded to the MAC. The threshold level is configured by using the TTC bits of MTL_TxQ0_Operation_Mode register corresponding to an MTL queue.
- Store-and-forward mode:

In store-and-forward mode, the MTL pops out the packet towards the MAC only when one or more of the following conditions are true:

- A complete packet is stored in the Queue
- The Tx FIFO becomes almost full
- The ATI watermark becomes low

The watermark becomes low when the requested queue does not have space to accommodate the requested burst length on the ATI. Therefore, the MTL when operating in the store-and-forward mode allows packet transmission even if the packet length is bigger than the Tx queue size.

The application can flush complete content of the Tx queue by setting Bit 0 (FTQ) of Transmit Operation Mode register corresponding to an MTL queue. This bit is self-clearing and initializes the Queue pointers to the default state. If the FTQ bit is set during a packet transfer from the MTL to the MAC, the MTL stops further transfer because the queue is considered to be empty. Therefore, an underflow event occurs at the MAC transmitter.

43.4.1.2.1.3 Initialization flow

Upon reset, the MTL is ready to manage the flow of data to and from the DMA, and MAC. For configurations with a single Transmit Queue, there are no initialization requirements for enabling the MTL. For configurations with multiple Transmit Queues, initialize the queue size for each of the queues by programming the TQS bits of MTL_TxQ0_Operation_Mode register corresponding to a Transmit queue. Also, initialize the MAC block. Internal DMA controllers must be individually enabled through their respective CSRs.

43.4.1.2.2 Receive path

The MTL Rx module receives the packets from the MAC and pushes them into the Rx Queue. The status (fill level) of the queue is indicated to DMA when it crosses the configured Receive threshold (RTC bits[1:0] of MTL_RxQ0_Operation_Mode register of corresponding MTL queue), or the complete packet is received. The MTL also indicates the fill level of the queue so that the DMA can initiate preconfigured burst transfers towards the AHB interface.

For more information on queues, see Multiple Channels and Queues Support.

43.4.1.2.2.1 Receive Operation

During a Receive operation, the MTL is a slave for the MAC. The general sequence of events is as follows:

1. When the MAC receives a packet, it indicates the availability of receive data.
2. The MAC indicates the SOP and EOP delimiters.
3. The MTL accepts the data and pushes it into corresponding Rx queue.
4. After the EOP is transferred, the MAC drives the status word which is also pushed into the corresponding Rx queue by the MTL.
5. If IEEE 1588 timestamp feature is enabled and the 64-bit timestamp is available along with the packet status, it is pushed into the Rx queue as a part of the status word. Therefore, two additional locations are taken per packet to store the timestamp in the Rx queue.
6. The MTL takes the data out of the queue and sends it to the DMA depending on the following modes:

43.4.1.2.2.1.1 Threshold mode

In the (default) Threshold mode, the MTL reads the data and indicates its availability to the application or DMA when one of the following occurs:

- Data bytes equal to the threshold amount are written to the Rx queue (RTC bits[1:0] of MTL_RxQ0_Operation_Moderegister of corresponding MTL queue)
- A full packet of data is received into the queue

43.4.1.2.2.1.2 Store and forward mode

In the store-and-forward mode (when Bit 5 of MTL_RxQ0_Operation_Mode register of a queue is set to 1), the initial Rx queue locations are reserved for the status words before writing the SOP. A packet is read out only after it is completely written into the Rx queue. In this mode, all error packets are dropped (if configured through Bit 4 of MTL_RxQ0_Operation_Mode register of a queue) such that only valid packets are read and forwarded to the application.

43.4.1.2.2.2 Multi packet receive operation

In Threshold mode, the packet status is available immediately after the packet data. In store-and-forward mode, the packet data is available after the packet status. The MTL is capable of storing any number of packets into the queue as long as it is not full.

If the MAC receives a packet when the corresponding Rx queue is full, the MTL ignores that packet and overflow pulse is generated on the corresponding Rx queue. In addition, the MTL increments the overflow counter in the MTL_RxQ0_Missed_Packet_Overflow_Cnt of corresponding queue.

43.4.1.2.2.3 Error handling in receive operation

Following things would happen, If the MTL Rx queue is full before it receives the EOP data from the MAC.

1. An overflow is declared
2. The whole packet (including the status word) is dropped
3. The overflow counter in the DMA (Overflow Counter register of corresponding MTL queue) is incremented.

This is true even if Bit 4 (FEP) of MTL_RxQ0_Operation_Mode register of corresponding MTL queue is set.

If the start address of such a packet has already been transferred to the Read Controller, the rest of the packet is dropped and a dummy EOP is written to the queue along with the status word with overflow status. The status indicates a partial packet because of overflow. In such packets, the Packet Length field is invalid. If the MTL Receive Queue is configured to operate in the store-and-forward mode and the length of the received packet is more than the queue size, overflow occurs and all such packets are dropped.

The MTL Rx Control logic can filter error and undersized packets, if enabled by using the FEP and FUP bits of MTL_RxQ0_Operation_Mode register of corresponding MTL queue. If the start address of such a packet has already been transferred to the Rx Queue Read Controller, that packet is not filtered. The start address of the packet is transferred to the Read Controller after the packet crosses the receive threshold set by Bits[1:0] of MTL_RxQ0_Operation_Mode register of corresponding MTL queue.

If the MTL Receive Queue is configured to operate in the store-and-forward mode, all error packets can be filtered and dropped. For the application or DMA to flush the error packet being read from the queue, it must assert the flush signal. The MTL then stops transferring data to the application (DMA). It internally reads the rest of the packet and drops it. The MTL then starts the transfer of next packet (if available).

43.4.1.3 MAC

MAC can support RMII or RGMII Phy interface. MAC consists of Mac Transmit interface (MTI), MAC Receive Interface (MRI) and MAC Control Interface (MCI).

43.4.1.3.1 MAC transmission

The MAC transmission process is as follows:

1. Transmission is initiated when the MAC Transaction Layer (MTL) pushes in data with the SOP signal asserted.
2. When the SOP signal is detected, the MAC accepts the data and begins transmitting to the RMII or RGMII.
3. After the EOP is transferred to the MAC, the MAC does one of the following:
   - The MAC completes the normal transmission and gives the transmission status to the MTL.
   - If a normal collision (in half-duplex mode) occurs during transmission, the MAC gives the Transmit Status with retry bit set to the MTL. The MAC gives the Retry request till one of the following is true:
     — Packet is successfully transmitted
     — Maximum retry requests expire
     — When maximum retry requests expire, the MAC aborts the packet transmission with Excessive Collision Transmit Status. The MAC accepts and drops all further data until the next SOP is received. The MTL block should retransmit the same packet from SOP on observing a Retry request (in the Status) from the MAC.
   - If any one of the following happens, the MAC aborts the packet transmission:
     — No carrier (half-duplex mode)
     — Loss of carrier (half-duplex mode)
     — Excessive deferral (half-duplex mode)
     — Late collisions (half-duplex mode)
     — Jabber
     — The MAC accepts and drops all further data until the next SOP is received
4. The MAC issues an underflow status if the MTL is not able to provide the data continuously during the transmission. The MAC accepts and drops all further data until the next SOP is received.
5. During the normal transfer of a packet from MTL, if the MAC receives a SOP without getting an EOP for the previous packet, it ignores the SOP and considers the new packet as continuation of the previous packet.

43.4.1.3.2 MAC reception

A receive operation is initiated when the MAC detects an SFD on RMII, RGMII, or MII. The MAC strips the preamble and SFD before proceeding to process the packet. The header fields are checked for filtering and the FCS field used to verify the CRC for the packet. The received packet is stored in a shallow buffer until the address filtering is performed. The packet is dropped in the MAC if it fails the address filter.

43.4.2 Using PHY interfaces

This module support the following modes:

- RMII 10/100 Mbps interface
- RGMII 10/100/1000 Mbps interface

The internal Phy_intf_sel input signal decides which mode has been selected. Phy_intf_sel signal gets sampled at reset. PS and FES bit of MAC configuration register selects the speed of mode.

The module supports the IEEE802.3-2015 specification for RMII (RMII specification version 1.2 from the RMII consortium) and RGMII specification.

NOTE

RMII reference clock (50 MHz) can be fed to IP from an external source or internally from PLL on SoC.

The module supports can access the phy registers through Station Management Agent (SMA) which is a two wire Station management interface (MIM). One wire is management data clock (MDC) and other wire is management data input/output

(MDIO). As per IEEE 802.3 specification, maximum operating frequency of MDC is 2.5 Mhz which can be derived from system clock using a divider. The CR bit of MAC_MDIO_Address register can be programmed to generate different MDC clock frequency.

SMA supports both MDIO clause 45 and clause 22 frame structure as per IEEE802.3 specification. The C45E bit of MAC_MDIO_Address register can be set to enable the clause 45 frame structure.

43.4.3 VLAN and double VLAN insertion, deletion, replacement and tagging

These sections describes the VLAN and Double VLAN features.

43.4.3.1 Double VLAN processing

The module can support processing of two VLAN tags in double VLAN tagging processing. These two VLAN tags can also be named as inner (C-VLAN) and outer (S-VLAN) VLAN tag. If packet is having only one tag then it would be considered as outer tag.

It supports the following function in Double VLAN processing feature using the MAC_VLAN_Incl, MAC_Inner_VLAN_Incl and MAC_VLAN_Tag registers.

- Insertion, replacement, or deletion of up to two VLAN tags in the Transmit path.
- Packet filtering and stripping based on any one of the two VLAN Tags in the Receive path. Stripping and providing up to two VLAN Tags in the Receive path as a part of the Receive status.

43.4.3.1.1 Transmit path

Following table describes the features supported by the MAC on the Transmit side.

Table 364. Double VLAN Processing Features in Transmit Path

Feature	Description
Support for C-VLAN and S-VLAN Tag types	The inner or outer VLAN tag can be of C-VLAN and S-VLAN type. The VLAN type is specified through the CSVL bit of MAC_VLAN_INCL and MAC_INNER_VLAN_INCL registers. The module supports processing of any sequence of outer and inner VLAN tags. **NOTE** The module does not support the C-VLAN S-VLAN sequence. The MAC does not check whether the packet provided by the application has a valid sequence of the VLAN Tag types or the insertion or replacement operation results in invalid sequence of VLAN Tag type. Therefore, the application must provide correct sequence of VLAN Tag types and program the MAC in such a way that it results in correct sequence of VLAN Tag types in the transmitted packet. The application must ensure the following: • The inner tag should not be S-VLAN when outer C-VLAN Tag insertion is enabled. • The outer tag should not be C-VLAN when inner S-VLAN Tag insertion is enabled. • The inner tag should not be S-VLAN when outer tag should be replaced with C-VLAN. • The outer tag should not be C-VLAN when inner tag should be replaced with S-VLAN.
VLAN Tag deletion	VLAN tag deletion for outer or inner tag can be enabled through VLC field in the MAC_VLAN_Incl or MAC_Inner_VLAN_Incl register, respectively. When VLAN deletion is enabled, the MAC deletes the tag present at the corresponding position. When a packet has only one tag, it is considered as the outer tag. If inner tag deletion is enabled and the packet has only one tag, the MAC does not delete the inner tag.

Table continues on the next page...

Table 364. Double VLAN Processing Features in Transmit Path (continued)

VLAN Tag Insertion or Replacement	VLAN tag insertion or replacement for outer or inner tag can be enabled through VLC field in the MAC_VLAN_Incl or MAC_Inner_VLAN_Incl register, respectively. When VLAN tag insertion or replacement is enabled, the VLTI bit in the MAC_VLAN_Incl or MAC_Inner_VLAN_Incl register is used to determine whether the VLAN tag should be taken from the register or the Control Word.

43.4.3.1.2 Receive path

Following table describes the features supported by the MAC on the receive side and the corresponding bits in the MAC_VLAN_Tag register.

Table 365. Double VLAN Processing in Receive Path

Feature	Description
Outer or inner VLAN tag-based filtering	The MAC can filter packets based on the outer or inner VLAN tag through the ERIVLT bit.
C-VLAN or S-VLAN tag-based filtering	The MAC can filter packets based on the C-VLAN or S-VLAN type based on the ERSVLM bit.
Outer and Inner VLAN Tag stripping	The MAC can strip the outer and inner VLAN Tags from received frame based on the EVLS and EIVLS bits.
16-bit outer and inner VLAN Tag and Type in Rx status	The MAC can provide the 16-bit outer and inner VLAN Tag and Type in the Rx status based on the EVLRXS and EIVLRXS bits, respectively.
Disabling or skipping checking of outer VLAN Tag type	The MAC can disable or skip checking of outer VLAN Tag type to match C-VLAN or SVLAN based on the DOVLTC bit.

43.4.3.1.3 Source address and VLAN insertion, replacement, or deletion

The module supports the SA and VLAN insertion, replacement and deletion. The source address (SA) and VLAN fields are Tx packet-related control information that are provided as part of the control word through interfaces. IP supports the feature to insert or replace the source address based on the information in the MAC Address Registers, and also the feature to insert, replace or delete the VLAN fields (VLAN Type and VLAN Tag) based on the setting of the VLTI bit in the MAC_VLAN_Incl register. SA insertion or replacement feature can be enabled for all Transmit packets or selective packets. Similarly, VLAN insertion, replacement, or deletion feature can be enabled for all Tx packets or selective packets.

43.4.3.1.4 Programming source address insertion or replacement

The software can use the SA insertion or replacement feature to instruct the MAC to do the following for Tx packets:

- Insert the content of the MAC Address Registers in the SA field
- Replace the content of the SA field with the content of the MAC Address Registers

When SA insertion is enabled, the application must ensure that the packets sent to the MAC do not have the SA field. The MAC does not check whether the SA field is present in the Transmit packet and it inserts the content of MAC Address Registers in the SA field. Similarly, when SA replacement is enabled, the application must ensure that the SA field is present in the packets sent to the MAC.

The MAC replaces the six bytes following the Destination Address field in the Transmit packet with the content of the MAC Address Registers.

- SARC bit of MAC_Configuration register can be program to enable the SA insertion or replacement for all Tx packets.

- SA insertion for selective packet can be enabled by programming the SA Insertion Control field (bits [25:23] of TDES3) in the first Transmit descriptor of the packet. When Bit 25 of TDES3 is set, the SA Insertion Control field indicates insertion or replacement by MAC Address1 registers. When Bit 25 of TDES3 is reset, it indicates insertion or replacement by MAC Address 0 registers.

If MAC Address1 Registers are not enabled, the MAC Address0 registers are used for insertion or replacement irrespective of the value of the most-significant bit of the SA Insertion Control field.

43.4.3.1.5 Programming VLAN insertion, replacement, or deletion

The software can use the VLAN insertion, replacement, or deletion feature to instruct the MAC to do the following for Tx packets:

- Delete the VLAN Type and VLAN Tag fields
- Insert or replace the VLAN Type and VLAN Tag fields

Insertion or replacement is done based on the setting of VLTI bit in the MAC_VLAN_Incl register as described in the following table:

Table 366. VLAN Insertion or Replacement Based on VLTI Bit

Condition	Description
VLTI bit is set	The MAC inserts or replaces the following: • VLAN Type field (C-VLAN or S-VLAN as indicated by the CSVL bit of MAC_VLAN_Incl register) • VLAN Tag field depending on the Content of the VT field of Transmit context descriptor of the packet
VLTI bit is reset	The MAC inserts or replaces the following: • VLAN Type field (C-VLAN or S-VLAN as indicated by the CSVL bit of MAC_VLAN_Incl register) • VLAN Tag field with the VLT field of MAC_VLAN_Incl register

When VLAN replacement or deletion is enabled, the MAC checks if the VLAN Type field (0x8100 or 0x88a8) is present after the DA and SA fields in the Transmit packet. The replace or delete operation does not occur if the VLAN Type field is not detected in two bytes following the DA and SA fields. However, when VLAN insertion is enabled, the MAC does not check the presence of VLAN Type field in the Transmit packet and just inserts the VLAN Type and VLAN Tag fields.

VLAN insertion, replacement, or deletion feature can be enabled for all Tx packets by programming the VLC and VLP fields of MAC_VLAN_Incl register.

VLAN insertion, replacement, or deletion for selective packets can be enabled by programming the VTIR field of TDES2 Normal Descriptor.

In addition, the VLP (VLAN Priority control) bit must be reset in MAC_VLAN_Incl register (for outer VLAN) and MAC_Inner_VLAN_Incl register (in inner VLAN) for the MAC to take the control inputs from the host, depending on the configuration.

43.4.3.1.6 Queue/Channel Based VLAN Tag Insertion on Tx

This module supports channel/queue based VLAN tag insertion on all transmitted packets.

43.4.3.1.6.1 Accessing Queue/Channel Specific VLAN Tag Registers

Queue/Channel specific VLAN tag registers are accessed using indirect addressing through the MAC_VLAN_Incl register. VLAN type and tag value can be independently programmed for each queue/channel.

43.4.4 Packet Filtering

There are various packets filtering scheme available in MAC receiver. Filtering sequence in their precedence order of received packet is shown in the following figure.

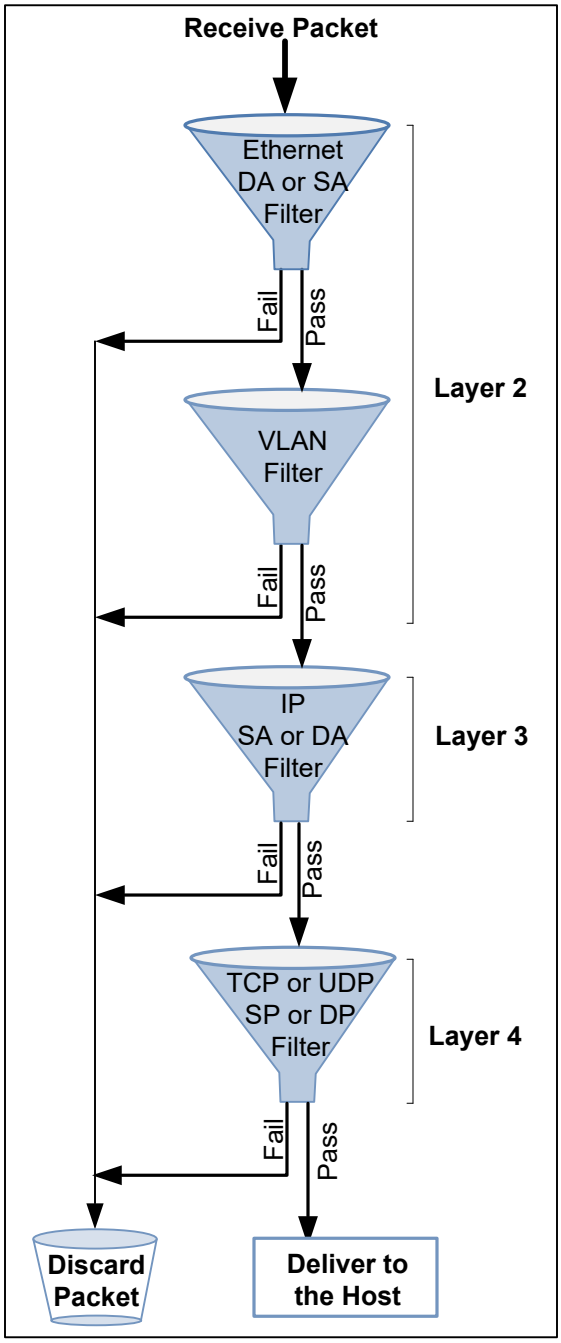


Figure 174. Packet Filtering Sequence

If any of the Layer filters are not enabled, that filter is bypassed and the subsequent filter is applied. A packet that fails any of the filters, is discarded. However, depending on the entry address and the number of parsed entries of the frame offset, discarded packets can be routed to RX_DMA_CH0 and a parser interrupt can be generated.

Even if no parser matched, the frame can be routed into the DMA channel indicated in DMA_CH_NO[2:0].

43.4.4.1 Source address or destination address filtering

The Address Filtering Module of the MAC checks the source address (SA) and destination address (DA) fields of each incoming packet. Programming of different type of address filtering is described in the following sections.

43.4.4.1.1 Unicast destination address filtering

MAC supports up to 64 MAC addresses for unicast perfect filtering. If perfect filtering is selected (HUC bit of MAC_Packet_Filter register is reset), the MAC compares all 48 bits of received unicast address with the programmed MAC address for any match. Default MacAddr0 is always enabled. The MacAddr1 to MacAddr63 addresses are selected with an individual enable bit. Each byte of MacAddr1 to MacAddr31 can be masked during comparison with corresponding received DA byte by setting the corresponding Mask Byte Control bit in the register. This enables group address filtering for the DA.

In hash filtering mode (when HUC bit is set), the MAC performs imperfect filtering for unicast addresses using a 64-bit Hash table. For hash filtering, the MAC uses the upper 6 bits CRC(by CRC32) of the received destination address to index the content of the Hash table. A value of 000000 selects Bit 0 of selected register, and a value of 111111 selects Bit 63 of Hash Table register. If the corresponding bit (indicated by the 6-bit CRC) is set to 1, the unicast packet is considered to have passed the Hash filter; otherwise, the packet is considered to have failed the Hash filter.

43.4.4.1.2 Multicast destination address filtering

PM bit of MAC_Packet_Filter register can be enabled to pass all multicast packets or disabled to do multicast addresses filtering based on HMC bit of MAC_Packet_Filter register. The multicast address is compared with the programmed MAC Destination Address registers (1–31). Also, Group address filtering is supported.

In Hash filtering mode, the MAC performs imperfect filtering using a 64-bit Hash table. The MAC uses the upper 6-bits CRC(by CRC32) of received multicast address to index the content of the Hash table. A value of 000000 selects Bit 0 of selected register and a value of 111111 selects Bit 63 of the Hash Table register. If the corresponding bit is set to 1, the multicast packet is considered to have passed the Hash filter. Otherwise, the packet is considered to have failed the Hash filter.

43.4.4.1.3 Hash or perfect address filtering

HPF bit of MAC_Packet_Filter register must be configured for enabling or disabling the destination address filtering by either Hash filter or Perfect filter. HUC and HMC bit of MAC_Packet_Filter register must be configured to select the Hash filtering or Perfect filtering for Unicast/Multicast packet.

43.4.4.1.4 Broadcast address filtering

The MAC does not filter any broadcast packets by default. DBF bit of MAC_Packet_filter register must be set to reject the broadcast packets.

43.4.4.1.5 Unicast source address filtering

By default, MAC compares the source address field with the value programmed in the source address registers. Bit 30 of MAC address register [1-2] can be configure to use it for source address instead of destination address comparison. The MAC also supports group filtering with source address(SA) . You can filter a group of addresses by masking one or more bytes of the address. The MAC drops the packets that fail the SA filter if the SAF bit is set in MAC_Packet_Filter register. Otherwise, the result of the SA filter is given as a status bit in the Receive Status word. When the SAF bit is set, the SA filter and DA filter result is AND'ed to decide whether the packet needs to be forwarded. This means that the packet is dropped if either filter fails. The packet is forwarded to the application only if the packet passes both filters in-order.

43.4.4.1.6 Inverse filtering

For DA and SA filtering, you can invert the filter-match result at the final output by setting the DAIF and SAIF bits of MAC_Packet_Filter register. The DAIF bit is applicable for both Unicast and Multicast DA packets. The result of the unicast or multicast destination address filter is inverted in this mode. Similarly, when the SAIF bit is set, the result of unicast SA filter is reversed.

Following two tables summarize the DA and SA filtering based on the type of packets received.

Table 367. Destination Address Filtering

Packet Type	PR	HPF	HUC	DAIF	HMC	PM	DBF	DA Filter Operation
Broadcast	1	X	X	X	X	X	X	Pass
	0	X	X	X	X	X	0	Pass
	0	X	X	X	X	X	1	Fail
Unicast	1	X	X	X	X	X	X	Pass all packets
	0	X	0	0	X	X	X	Pass on Perfect/Group filter match
	0	X	0	1	X	X	X	Fail on Perfect/Group filter match
	0	0	1	0	X	X	X	Pass on Hash filter match
	0	0	1	1	X	X	X	Fail on Hash filter match
	0	1	1	0	X	X	X	Pass on Hash or Perfect/Group filter match
	0	1	1	1	X	X	X	Fail on Hash or Perfect/Group filter match
Multicast	1	X	X	X	X	X	X	Pass all packets
	X	X	X	X	X	1	X	Pass all packets
	0	X	X	0	0	0	X	Pass on Perfect/Group filter match and drop Pause packets if PCF = 0x
	0	0	X	0	1	0	X	Pass on Hash filter match and drop Pause packets if PCF = 0x
	0	1	X	0	1	0	X	Pass on Hash or Perfect/Group filter match and drop Pause packets if PCF = 0x
	0	X	X	1	0	0	X	Fail on Perfect/Group filter match and drop Pause packets if PCF = 0x
	0	0	X	1	1	0	X	Fail on Hash filter match and drop Pause packets if PCF = 0x
	0	1	X	1	1	0	X	Fail on Hash or Perfect/Group filter match and drop Pause packets if PCF = 0x

Table 368. Source Address Filtering

Packet Type	PR	SAIF	SAF	SA Filter Operation
Unicast	1	X	X	Pass all packets.

Table continues on the next page...

Table 368. Source Address Filtering (continued)

Packet Type	PR	SAIF	SAF	SA Filter Operation
	0	0	0	Pass status on Perfect or Group filter match but do not drop packets that fail
	0	1	0	Fail status on Perfect or Group filter match but do not drop packet
	0	0	1	Pass on Perfect or Group filter match and drop packets that fail
	0	1	1	Fail on Perfect or Group filter match and drop packets that fail

NOTE

When the RA bit of MAC_Packet_Filter register is set, all packets are forwarded to the system along with the correct result of the address filtering in the Rx Status.

43.4.4.2 VLAN Filtering

VLAN tag filter can be done by either perfect match or hash table. Depending upon the configuration, MAC can compare either lower 12 bits or all 16 bits of received VLAN tag for perfect match. MAC forwarded the VLAN tagged packet with match status while it drops all the packets which don't matches.

MAC uses 16-bit VLAN hash table for group address filtering based on the VLAN tag. If hash filtering is enabled then most significant 4-bit of CRC-32 of VLAN tag are used to index the content of the VLAN Hash Table Register. A value of 1 in the VLAN Hash Table register, corresponding to the index, indicates that the VLAN tag of the packet matched and the packet should be forwarded. A value of 0 indicates that VLAN-tagged packet should be dropped.

The MAC also supports the inverse matching for VLAN packets. In the inverse matching mode, when the VLAN tag of a packet matches the perfect or hash filter, the packet should be dropped. If the VLAN perfect and VLAN hash match are enabled, a packet is considered as matched if either the VLAN hash or the VLAN perfect filter matches. When inverse match is set, a packet is forwarded only when both perfect and hash filters indicate mismatch.

Table below shows the different possibilities for VLAN matching and the final VLAN match status. When the RA bit of MAC_Packet_Filter register is set, all packets are received and the VLAN match status is indicated in the VF bit of RDES2 Normal Descriptor (Write-Back Format). When the RA bit is not set and the VTFE bit is set in MAC_Packet_Filter register, the packet is dropped if the final VLAN match status is Fail.

When VLAN VID is programmed to 0 in the VL field of MAC_VLAN_Tag register, all VLAN-tagged packets are considered as perfect matched but the status of the VLAN hash match depends on the VTHM and VTIM bits in MAC_VLAN_Tag register.

Table 369. VLAN Match Status

VID	VLAN Perfect Filter Match Result	VTHM Bit	VLAN Hash Filter Match Result	VTIM Bit	Final VLAN Match Status
VID = 0	Pass	0	X	X	Pass
	Pass	1	X	0	Pass
	Pass	1	Fail	1	Pass
	Pass	1	Pass	1	Fail
VID!= 0	Pass	X	X	0	Pass

Table continues on the next page...

Table 369. VLAN Match Status (continued)

VID	VLAN Perfect Filter Match Result	VTHM Bit	VLAN Hash Filter Match Result	VTIM Bit	Final VLAN Match Status
	Fail	0	X	0	Fail
	Fail	1	Fail	0	Fail
	Fail	1	Pass	0	Pass
	Fail	0	X	1	Pass
	Pass	X	X	1	Fail
	Fail	1	Pass	1	Fail
	Fail	1	Fail	1	Pass

In this table, X represents any value.

43.4.4.3 VLAN Filter Fail Packets Queue

When VLAN filtering is enabled, the VLAN Filter Fail Packets can be routed to a programmable Queue (VFFQ) when Bit RA = 1 or VTFE = 0 of MAC_Packet_Filter register and the enable bit (VFFQE of MAC_RXQ_CTRL4) for the queue is set.

The packets that passes the VLAN filtering, are routed to Rx queues based on the VLAN TAG priority field by programming the PSRQ in the corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers. The packets that fail the VLAN filter are discarded if RA=0 or VTFE=1. However, when RA=1 or VTFE=0, the VLAN filter fail packets are still forwarded to the application. In such a scenario, when the VLAN Filter Fail Queue Enable (VFFQE) bit is set, the VLAN filter fail packets are forwarded to the Rx queue number programmed in the VFFQ bit. If VFFQE=0, the Rx queue number is determined by the VLAN priority mapping as per the PSRQ fields.

Table below shows the Rx Queue routing table for Unicast tagged packets, with DA/SA Filter enabled.

Table 370. Rx Queue Routing table for Unicast Tagged Packets

RA	VFTE	SA/DA Filter Result	VLAN Filter Result	VFFQE	Queue Routing
X	X	PASS	PASS	X	PSRQ
0	0	PASS	FAIL	0	PSRQ
0	0	PASS	FAIL	1	VFFQ
0	X	FAIL	X	X	DROPPED
0	1	PASS	FAIL	X	DROPPED
1	X	FAIL	X	0	UFFQ*/PSRQ
1	X	FAIL	X	1	UFFQ*/VFFQ
1	X	PASS	FAIL	0	PSRQ
1	X	PASS	FAIL	1	VFFQ

X : Don't care condition * : When UFFQE is enabled else PSRQ

43.4.4.4 Extended Receive VLAN Filtering and Routing

When Extended Rx VLAN filtering and routing is enabled then both perfect filtering and Hash filtering can be enabled. The overall VLAN filter result is based on the Perfect filter result and Hash filter result (if enabled). Filter result is passed to application/software as part of the status bit. Extended Routing will happen only if VLAN filter has passed. Routing is only based on Perfect filter result. Each Perfect filter has a DMA channel enable and DMA channel number filed which should be programmed for routing.

For more details about routing, see details on 'Extended VLAN Based DMA Selection' in Dynamic (Per Packet) Mapping.

43.4.4.4.1 Comparison mode

Software/Application has following comparison option for each VLAN tag filter

- It can program the MAC to compare the inner or outer VLAN tag with either 12 bits or 16 bits programmed VID.
- It can choose that VID comparison is for SVLAN or CVLAN type frames, if type check is enabled for a filter.

43.4.4.4.2 Filtering

Perfect Filtering is done based on the MAC_VLAN_Tag_Filter registers. For each VLAN Tag Filter, the MAC will compare the relevant VLAN Tag ID and gives a result. If any one of the VLAN Tag Filters gives a match then the frame is considered to have passed the VLAN Tag Filters. If the frame mismatches all the filters, then the frame is considered to have failed the VLAN filter. This behavior is applicable only when the Inverse Filtering is not enabled in MAC_VLAN_Tag_Ctrl Register.

If inverse Filtering is enabled and the frame has mis-matched all the relevant filters then it is considered to have passed the VLAN filter. If the frame matches any one of the relevant filters then it is considered as a fail. If none of the enabled filters can perform a comparison or if none of the filters are enabled, then the frame is bypassed to the application.

The overall filter result and the programming on the VTFE and RA bits of the MAC Filter Register determine if the frame will be dropped or forwarded to the application. If RA = 1 or VTFE = 0, it does not matter if the filter result is a pass or fail. The frame is always forwarded. If RA = 0 and VTFE = 1, only then, if the VLAN Tag Filter result is a pass does the MAC forward the frame. If the frame is forwarded to the application, then the relevant filter result is indicated through the Status bits.

43.4.4.4.3 Filter Status

The Extended Receive VLAN Filtering and Routing feature provides two status bits to indicate the comparison result of the VLAN tags.

By default, the MAC indicates the VLAN Filter Status through one bit in the status – VF field in RDES2. When Extended RX VLAN Filtering and Routing is enabled, two status bits are used to indicate the comparison result of VLAN tags. The Outer VLAN Tag Filter Pass and Inner VLAN Tag Filter Pass bits are defined in the following positions. The status indicated through these bits is highly dependent on the programming as explained below.

In RDES2:

- Bit 15 – Outer VLAN Tag Filter Status
- Bit 14 – Inner VLAN Tag Filter Status

In ARI status: MAC Filter Status:

- Bit 15 – Outer VLAN Tag Filter Status
- Bit 14 – Inner VLAN Tag Filter Status

In MRI Status:

- Bit 47 – Outer VLAN Tag Filter Status
- Bit 46 – Inner VLAN Tag Filter Status

Outer VLAN Tag Filter Status (OTS)

- In perfect Filtering, without inverse filtering enabled, if this bit is set, it indicates that the frame's Outer VLAN Tag has matched one of the VLAN Tag Filters.
- If this bit is reset, it indicates that the frame's Outer VLAN Tag has either failed the relevant Outer VLAN Tag Filters or bypassed them.
- If none of the filters are enabled for Outer VLAN Tag Comparison, then this bit is reset.
- If Inverse Filtering is enabled and this bit is set, then the frame's VLAN Tag has passed all the relevant VLAN Tag Filters. If it is reset, then it has failed at least one of or bypassed all the Filters programmed for Outer VLAN Tag Comparison.
- This bit is valid for both Single and Double VLAN Tagged frames.

Inner VLAN Tag Filter Status (ITS)

- In perfect Filtering, without inverse filtering enabled, if this bit is set, it indicates that the frame's Inner VLAN Tag has matched one of the VLAN Tag Filters.
- If this bit is reset, it indicates that the frame's Inner VLAN Tag has either failed the relevant Inner VLAN Tag Filters or bypassed them.
- If none of the filters are enabled for Inner VLAN Tag Comparison, then this bit is reset.
- If Inverse Filtering is enabled and this bit is set, then the frame's VLAN Tag has passed all the relevant VLAN Tag Filters. If it is reset, then it has failed at least one of or bypassed all the Filters programmed for Inner VLAN Tag Comparison.
- This bit is valid for only Double VLAN Tagged frames, when Double VLAN Processing is enabled.

The application must look at the status bits and the programming to determine if the Frame has passed or failed the VLAN Filter.

Following two tables 'OTS and ITS Bit Values with At Least 1 Perfect Filter Enabled' and 'OTS and ITS Bit Values with Only VLAN Hash Filter Enabled' show the possible Filter combinations and the corresponding filter results. These tables explain the scenarios when Double VLAN Processing and Hash VLAN filter are enabled in the design.

Legend for the table 'OTS and ITS Bit Values with At Least 1 Perfect Filter Enabled'

- VTIM: VLAN Tag Inverse Match Enable – bit 17 in VLAN_Tag_Ctrl Register.
- HFO: Hash Filter enabled for Outer VLAN Tag Comparison - bit 25 and bit 27 in VLAN_Tag_Ctrl Register.
- HFI: Hash Filter enabled for Inner VLAN Tag Comparison – bit 25 and bit 27 in VLAN_Tag_Ctrl Register.
- PFO – Perfect Filter comparison enabled for Outer VLAN Tag - Any of the MAC_VLAN_Tag_Filter Registers is enabled (bit 16 is set) and programmed for Outer VLAN Tag comparison (bit 20 is set to 0).
- PFI – Perfect Filter comparison enabled for Inner VLAN Tag - Any of the MAC_VLAN_Tag_Filter Registers is enabled (bit 16 is set) and programmed for Inner VLAN Tag comparison (bit 20 is set to 1).
- OTS – Outer VLAN Tag Filter Status
- ITS – Inner VLAN Tag Filter Status

Following table shows the possible values of status bits (OTS and ITS) when at least one Perfect filter is enabled.

Table 371. OTS and ITS Bit Values with At Least 1 Perfect Filter Enabled

VTIM	HFO	HFI	PFO	PFI	OTS	ITS
0	0	0	0	1	0	1/0
0	0	0	1	0	1/0	0
0	0	0	1	1	1/0	1/0
0	1	0	1	1	1/0	1/0

Table continues on the next page...

Table 371. OTS and ITS Bit Values with At Least 1 Perfect Filter Enabled (continued)

VTIM	HFO	HFI	PFO	PFI	OTS	ITS
0	1	0	1	0	1/0	0
0	1	0	0	1	1/0	1/0
0	0	1	1	1	1/0	1/0
0	0	1	1	0	1/0	1/0
0	0	1	0	1	0	1/0
1	0	0	0	1	0	1/0
1	0	0	1	0	1/0	0
1	0	0	1	1	1/0	1/0
1	1	0	1	1	1/0	1/0
1	1	0	1	0	1/0	0
1	1	0	0	1	1/0	1/0
1	0	1	1	1	1/0	1/0
1	0	1	1	0	1/0	1/0
1	0	1	0	1	0	1/0

Following table shows the possible values of status bits (OTS and ITS) when none of the perfect filters are enabled and only the VLAN Hash Filter is enabled.

Table 372. OTS and ITS Bit Values with Only VLAN Hash Filter Enabled

VTIM	HFO	HFI	OTS	ITS
0	0	0	0	0
0	1	0	1/0	0
0	0	1	1/0	1/0
1	0	0	1/0	0
1	1	0	1/0	0
1	0	1	1/0	1/0

With no perfect filters enabled, any VLAN packet is considered to have bypassed the perfect filter. If Hash Filter is enabled for one of the Tags, then the respective Status bit depends on the Filter's result. The Status bits are set to 0 if VLAN Hash Filter is also not enabled.

If the value of ITS/OTS is shown as 1/0; then it indicates that the final result is dependent on the enabled relevant filter's result.

Example 1: The second row of table 'OTS and ITS Bit Values with Only VLAN Hash Filter Enabled' indicates that at least one Perfect Filter is enabled for Outer VLAN tag comparison and none of the filters are enabled for Inner VLAN tag comparison. Inverse VLAN Filtering is not enabled. The bit OTS is given as 1/0. If the received frame passes at least one of the enabled Outer VLAN Tag filters then the bit is set to 1. If the frame doesn't pass any of the enabled Outer VLAN Tag filters, then the bit is set to 0.

Example 2: Last Row of table 'OTS and ITS Bit Values with Only VLAN Hash Filter Enabled' indicates that Inverse Filtering is enabled, Hash Filter and at least one perfect filter is enabled for Inner VLAN Tag comparison, then if the received frame's Inner VLAN tag mismatches with both the Hash Filter and all the enabled Perfect filters, then the frame will have the ITS bit set to 1 else it is set to 0. OTS will be set to 0 as no comparison is performed.

43.4.4.4.4 Stripping

Each of the VLAN Tags has individual control over stripping. The programming options of always strip, never strip, strip on pass and strip on fail are available. Inner or Outer VLAN Tag Stripping is based on the pass or fail results of the individual tag. If a tag is bypassed by all the relevant filters, stripping is not applicable for the tag.

- If strip on Pass is enabled for the outer VLAN Tag, then the stripping occurs only if the Outer VLAN tag has passed the relevant filters. The Outer VLAN Tag Filter Result bit will be set.
- If strip on Fail is enabled for the outer VLAN Tag, then stripping occurs only if the Outer VLAN Tag has failed relevant filters. The Outer VLAN Tag Filter Result Bit will be reset.
- If the Outer VLAN tag of the received frame is bypassed by the entire filter (no comparison has been made), then the tag is not stripped, though the Status Bit is still 0.
- As multiple filters are enabled, it is possible that the received VLAN frame could have matched any one or more of the filters. The VLAN Tag's value is not always deterministic from the filter status bits.

If the software/application strips the VLAN Tag based on the filter result, it might lose the VID. So the suggested use is, if stripping is enabled for any of the tags, the tag can be put in the status. For this the software/application will have to enable the respective "VLAN Tag in Status" bit - 24 or 31 in the MAC VLAN Tag Control Register.

See section Programming Guidelines for Extended VLAN Filtering and Routing on Receive

43.4.4.5 Layer 3 and Layer 4 Filtering

IP supports the Layer 3 based packet filtering which is IP Source or Destination Address filtering in the IPv4 or IPv6 packets, and Layer 4 based packet filtering which is Source or Destination Port number filtering in TCP or UDP. When layer 3 and layer 4 packet filtering is enabled then packets are filtered in the following manner.

- Matched Packets: The MAC forwards the packets that match all enabled fields to the application along with the status. The MAC gives the matched field status only if the IPC bit of MAC_Configuration register is set and one of the following conditions is true:
  - All enabled Layer 3 and Layer 4 fields match
  - At least one of the enabled field matches and other fields are bypassed or disabled

When multiple Layer 3 and Layer 4 filters are enabled, any filter match is considered as a match. If more than one filter matches, the MAC provides the status of the lowest filter where Filter 0 is the lowest filter and Filter 3 is the highest filter.

- Unmatched Packets: The MAC drops the packets that do not match any of the enabled fields. You can use the inverse match feature to block or drop a packet with specific TCP or UDP over IP fields and forward all other packets.
- Non-TCP or UDP IP Packets: By default, all non-TCP or UDP IP packets are bypassed from the Layer 3 and Layer 4 filters. MAC can optionally be programmed to drop all non-TCP or UDP over IP packets.
- IP has a control register “MAC_L3_L4_Control” to control the Layer L3 and L4 packet filtering along with address registers to program the Layer L3 and L4 fields to be matched.

43.4.4.6 Flexible Receive Parser

When this function is enabled, all incoming packets are parsed as per the programmable instruction stored in memory. It does the following function

- Packet Filter (Accept or Reject)
- DMA Channel Selection

Flexible receive parser block operates over the first 128 bytes data of receive packet based on 64 bytes of instruction on each packet. Following figure shows the functional block diagram of Flexible Receive Parser.

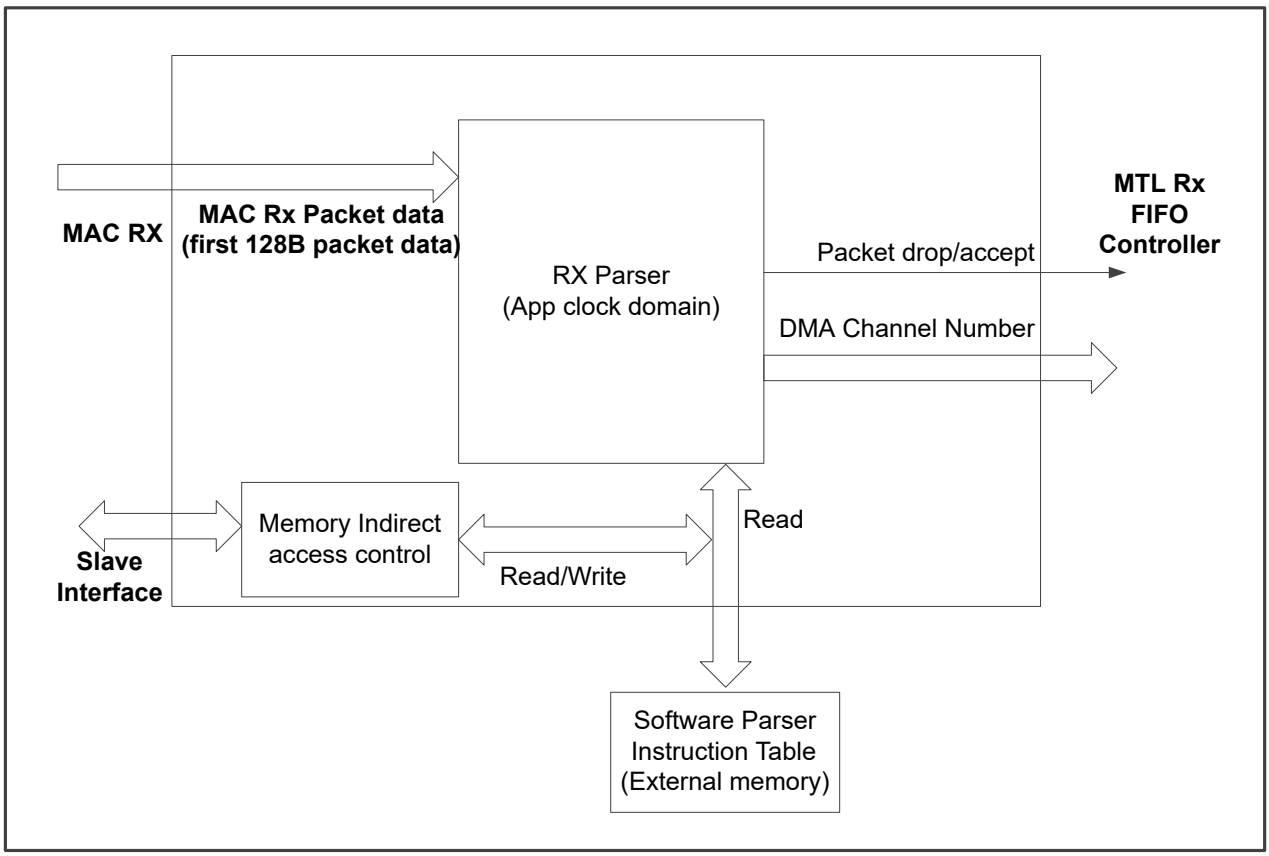


Figure 175. Functional block diagram of flexible recieve parser

43.4.4.6.1 Instruction Table Format of the Flexible Receive Parser

Software must build the 96-bit wide instruction before enabling the receive parser feature. Each instruction is programmed as an entry in the instruction memory.

The format of each entry in the Instruction table is shown in the following table.

Table 373. Rx Parser Instruction Entry Format

Field	Field Name	Description
31:0	MATCH_DATA	The 4-byte data. This data is used for comparing with incoming packet data starts at the frame offset as defined in [79:72] of the entry. The comparison is done only on those bits whose corresponding mask bits are set to 1 (MATCH_EN bits of [63:32]). See the Notes (below this table) on the byte order relative to Ethernet arrival data and endianness supported by QoS IP on the slave interface. The programming of this instruction table should adhere to the same.
62:32	MATCH_EN	When the MATCH_EN is set to 1, the corresponding Packet data bit is used for comparing. Otherwise, corresponding data bit is don't care.
64	AF	Accept frame
65	RF	Reject frame
66	IM	Inverse match
67	NIC	Next instruction control
71:68	Reserved	-
79:72	Frame Offset	[77:72] indicates the frame offset in terms of 4 bytes. (Here [79:78] always 0)Frame offset in the packet data to be compared for Match.This is in terms of 4 bytes. Value Actual Frame offset in bytes 0 01 42 8… ..63 252 The max value programmed in this Instruction Table is 128 B. The 'Actual Frame Offset' should not cross this limit.
87:80	OK Index	If (NIC==0) Memory Index to be used next when ENTRY_MATCH==1 and neither AF or RF is set to 1. If (NIC==1) Memory Index to be used next when ENTRY_MATCH==0 Note: ENTRY_MATCH is a calculated value based on MATCH_DATA, MATCH EN, and IM. It is not a field.
92:88	DMA CH NO	Indicates the DMA Channel Number (1-bit for each). This will be used when ENTRY_MATCH==1 and AF==1 (Accept frame). The encoding is as follows: • bit[0]- DMA Channel Number 0 • bit[1]- DMA Channel Number 1 • bit[2]- DMA Channel Number 2 • bit[3]- DMA Channel Number 3 • bit[4]- DMA Channel Number 4. Note: The DMA Channel number is encoded using bit wise as QoS IP support multicasting/ broadcast across DMA Channels. See section 3.3.12 for more details and proper usage.
128:93	Reserved	0 (This field is only for software view and reserved for future enhancements. The memory width remains as 93-bits for now)

The parser begins parsing from the 0th entry, for each received packet. The subsequent parsing entry location (next entry or OK_INDEX[]) depends on the parser result of the current entry. Following figure shows the detailed flow.

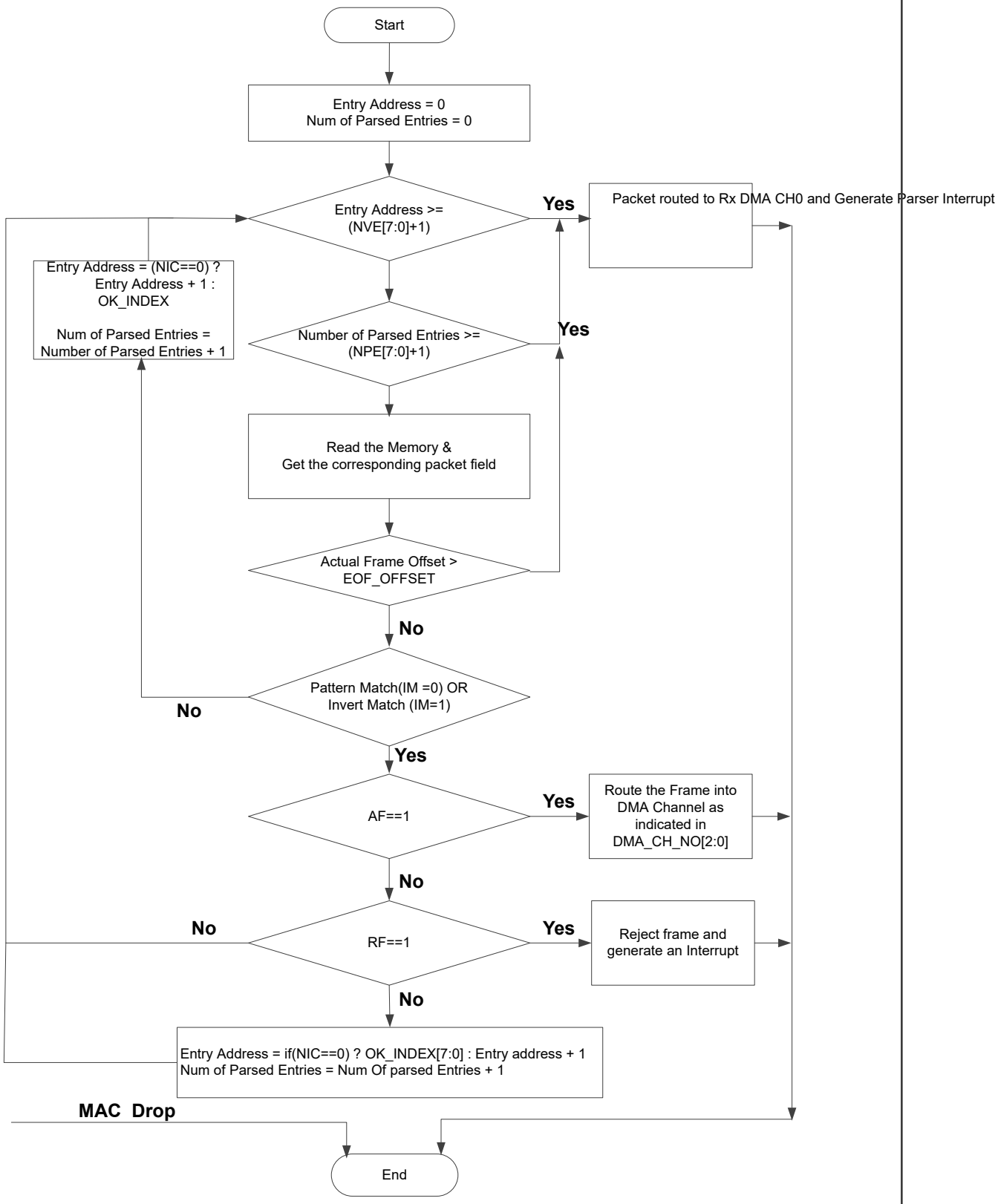


Figure 176. Flexible recieve parser flow chart

In the above figure, MAC Drop means the packet is dropped by the MAC due to another filter condition (for example, DA filter) and will not reach the Flexible Receive Parser.

The EOF_OFFSET shown in the figure is equal to the least among the following:

- Actual Packet size
- 128 Byte when frame pre-emption feature is not enabled/configured
- 128 Byte when frame pre-emption feature is enabled and packet is pre-emptible.
- 64 Byte when frame pre-emption feature is enabled and packet is Express Packet

43.4.4.6.2 Number of Valid Entries (NVE)

NVE indicates the number of valid entries in the instruction table. The default value is 64. Software can change this value by writing into MTL_RXP_Control_Status registers field NVE[7:0].

While parsing, if the Entry address is found to be is greater than NVE[7:0]+1, an error is flagged in the MTL_RXP_Interrupt_Control_Status register bit[0], NVEOVIS (Number of Valid Entries overflow status). IP generates an interrupt status (MTL_Interrupt_Status, bit[23], MTLPIS (MTL Parser Interrupt Status)).

43.4.4.6.3 Number of Parsable Entries (NPE)

NPE indicates the number of entries that can be parsed on an incoming packet. The default value is 64. Software can change this value by writing into MTL_RXP_Control_Status registers field NPE[7:0] (Number of Parsable Entries) bits [21:16].

While parsing, if the number of parsed entries is more than NPE[7:0]+1, an error is flagged in the MTL_RXP_Interrupt_Control_Status register bit[1], NPEOVIS (Number of Parsable Entries overflow status). IP generates an interrupt status (MTL_Interrupt_Status, bit[23], MTLPIS (MTL Parser Interrupt Status).

While NVE[] indicates is the number of valid entries in the Rx parser memory built by software, NPE[] indicates the worstcase parsable entries considering the various possible paths that the parser can take. (NVE[] >= NPE[]).

43.4.4.6.4 Frame Offset

It indicates the frame offset, in terms of 4 bytes which is used while comparing against the current table entry.

In an entry, an error is flagged in the TL_RXP_Interrupt_Control_Status register bit[2], FOOVIS (Frame Offset Overflow Interrupt Status) in any of the following cases:

When Frame Offset is more than

- Packet size
- 128B if pre-emption is disabled
- 128B if pre-emption is enabled and packet is pre-empitble
- 64B if pre-emption is enabled and packet is an Express packet

IP generates an interrupt status (MTL_Interrupt_Status bit[23]) and MTLPIS (MTL Parser Interrupt Status).

NOTE

If the frame ends after the (frame_offset *4) bytes but before ((frame_offset *4)+4) bytes, the Receive parser declares Frame Offset Error if comparison is enabled (via MASK bits) for the non-received bytes; from (frame_offset *4) to ((frame_offset *4)+4).

43.4.4.6.5 Frame Reject

In cases where an Entry matches (considering the Inverse match, IM bit in the instruction table) and Frame Accept bit is not set, and Frame Reject bit (RF) is set, the frame is dropped.

When a frame is dropped due to RF = 1, IP generates an interrupt status (MTL_Interrupt_Status, bit[23], MTLPIS (MTL Parser Interrupt Status).

43.4.4.6.6 Out of Order Processing

The Rx parser flow can be out of order. The decision tree need not be based on the order in which packet arrives. That is, the decision tree can be out of order; the decision tree need not be based on the same order as the packet data arrival (data within one packet). This implies that the next entry and next frame offset (OK_INDEX[]/FRAME_OFFSET[]) can be less than the current entry address and current Frame Offset.

43.4.4.6.7 DMA Channel Selection

When the Rx parser is enabled (FRPE bit[16] is set to 1 in MTL_Operation_Mode register) the Rx parser overrides the DMA selection criteria of the MAC.

When the Rx parser is disabled (FRPE bit[16] is set to 0 in MTL_Operation_Mode register) the MAC/MTL determines the DMA channel number.

43.4.4.6.8 Receive Queue Selection

The Rx Queue is determined by the existing MAC functionality, irrespective of the enabling/disabling of the Rx Parser. This is because the Rx parser is mainly designed for selecting the DMA channel and Rx queuing is decided by the MAC to enable the proper functioning of the Pause/PFC feature.

NOTE

When there are multiple queues and multiple DMA Channels are enabled, a particular DMA channel might receive out of order packets. This is because queue selection is based on MAC criteria (VLAN Priority and other criteria) and Rx DMA channel selection is based on Rx parser.

43.4.4.6.9 MAC Packet Filtering/Drop/Error Handling

The MAC does the packet filtering based on the received packet fields. Disable the Packet Filtering features inside the MAC when the Rx parser is enabled. To disable the Packet Filtering features inside the MAC,

1. Set Promiscuous Mode (PR, bit[0] of MAC_Packet_Filter register)
2. Set all other bits in the MAC_Packet_Filter register to its default values

When MAC decides to drop the packet (due to Rx MAC dependent Error such as GMII Error, Receive Watch Dog Error, CRC Error and Giant Frame Error) the packet is dropped irrespective of the Rx parser decision.

43.4.4.6.10 PAD Strip/CRC Strip Handling

The pad strip and CRC strip are controlled by the software and applicable to all the received packets. When these features are enabled, software must build the Rx parser instructions accordingly.

NOTE

Enable PAD and CRC stripping in real use case.

43.4.4.6.11 VLAN Strip Handling

The MAC supports stripping the VLANs, outer as well as inner VLAN. When MAC is programmed to strip the VLAN, the Rx parser considers those VLAN tags to be part of the incoming packet.

43.4.4.6.12 Multicast and Broadcast Support

IP supports multicast and broadcast packet and the packet is routed to the highest queue number.

When Rx Parser is enabled, the packet routing is decided by the DMA Channel Numbers (DMA CH NO, Bits [95:88]) in the Instruction Table.

When Rx Parser is disabled, the packet routing is determined by DCS (DMA Channel Select) field of the MAC_AddressX_High register. For more details, see Broadcast/Multicast Packet Duplication.

NOTE

DCS/DMA CH NO field is per DMA Channel control and therefore, multiple DMA channels can be selected for routing the packet.

43.4.4.6.13 Pre-emption Support

The MAC supports the frame pre-emption feature when the Flexible Rx parser feature is enabled. The minimum packet size for Frame Preemption is 64 bytes. This means that:

- The Rx parser can operate on 64 bytes of Express traffic.
- The Rx parser can operate on 128 bytes of pre-emptible traffic, as a 64 bytes packet of pre-emptible traffic might be interrupted by Express traffic before that packet is fully received, so the parser needs to have 64 bytes on either side of the express traffic in order to capture the full 64-bytes of a pre-empted packet.

43.4.4.6.14 Software Access to the Flexible Receive Parser Memory

Software can read/write into the Rx parser memory, using the following registers via indirect addressing.

- MTL_RXP_Indirect_Acc_Control_Status
- MTL_RXP_Indirect_Acc_data

For more details, see ENET_QOS register descriptions.

43.4.4.6.15 Statistical Counters

IP supports the following statistics counters for flexible receive parser.

1. MTL_RXP_Drop_Cnt
2. DMA_RXP_Error_Cnt

DMA_CH(#i)_Rxp_Accept_cnt (i=0; i<2)

For more details, see ENET_QOS register descriptions.

43.4.4.6.16 Changing the Instruction Table by Software

Software can update the instruction table in the memory in the following ways

- Disable the MAC Receiver and Receive Parser by setting the RE field of the MAC_Configuration register to 0 and FRPE bit of MTL_Operation_mode register to 0. Wait for Rx Parser to become inactive (RXPIA, MTL_RXP_Control_Status register bit[31]).
- (Optional) Program in the Entry address 0 to unconditionally (all MATCH_EN bit 0) skip to OK_INDEX[] (to certain location in the memory) where it is programmed to reject or accept all packets.

43.4.4.6.17 Receive Cut-Through Functionality

In the case of cut-through mode, the Rx controller attempts to transfer the received packets to DMA just after the cut-through threshold amount of packet data is received (RTC field in the MTL_RxQx_Operation_Mode register).

However due to Rx parser's result worst case arrival time can be more than the programmed cut-through threshold. So, the Rx controller delays the packet transfer to DMA accordingly. So, the cut-through functionality is re-defined as follows

The Rx controller attempts to transfer packet to DMA after at least RTC number of bytes have been received and Rx parser results are available.

43.4.4.6.18 Receive Packet Drop Indication

In most cases, the packet drops internally in the Rx Queue. However, back-to-back packets can arrive for the same queue and first packet's Rx parser result is not available when the second packet arrives. In such cases, the packet cannot be flushed internally and forwarded to DMA or the Application Interface and status is indicated accordingly. For more details, see RDES2, bit 16(RXPD) in Receive Normal Descriptor (Write-Back Format) .

43.4.4.6.19 Runt Packet Handling

The Rx parser can perform the operation with 64B minimum size packet. If the runt packet is received (should be indicated by MAC), the Rx parser assumes that it gets dropped in the MAC. This is because, back-to-back runt packets (< 64B) cannot be handled in the Rx parser.

43.4.4.6.20 Uncorrectable ECC Error Handling

When the parser detects an uncorrectable ECC error while parsing the parser memory, the packet is not dropped. Instead, the packet is sent to the application with an Error Indication. The RXPI (RX Parser Incomplete) is set in the packet status along with Error Summary (ES) bit.

For more details, see Receive Normal Descriptor (Write-Back Format) .

43.4.5 IEEE 1588 Timestamp Support

The MAC supports the IEEE 1588 Precision Time Protocol (PTP). The MAC is able to do precise time stamping and capturing of incoming and outgoing frame to enable support of PTP with its associated software. The MAC supports the all clock types defined in IEEE 1588-2008. Typical frequency of PTP timestamp clock is 125 MHz. The MAC supports:

- Taking a timestamp snapshot of all packets or only PTP type packets.
- Taking a timestamp snapshot of only event messages.
- Identifying the PTP message type, version, and PTP payload in packets sent directly over Ethernet and sending the PTP message type status given by the MAC. The PTP status is finally written back to the Receive Descriptor: RDES1 Normal Descriptor (Write-Back Format).
- Measuring sub-second time in decimal or binary format.

43.4.5.1 Delay Request-Response Mechanism

The system or network is classified into the master and slave nodes for distributing the timing and clock information. Following figure shows the process that PTP uses for synchronizing a slave node to a master node by exchanging PTP messages.

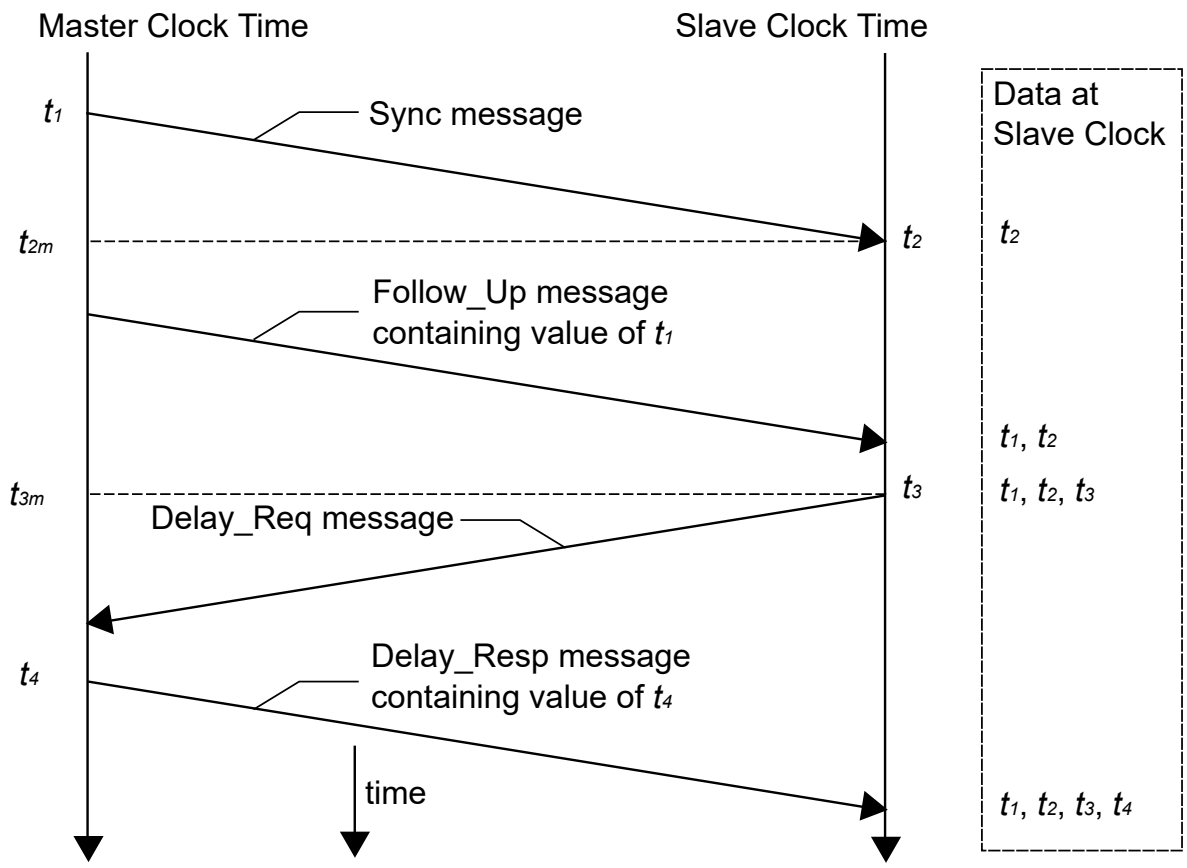


Figure 177. Networked Time Synchronization

As shown in the figure, PTP uses the following process:

1. The master broadcasts the PTP Sync messages to all its nodes. The Sync message contains the reference time information of the master. This message leaves the system of the master at t1. This time must be captured for Ethernet ports at the internal GMII interface.
2. The slave receives the Sync message and also captures the exact time, t2, using its timing reference.
3. The master sends a Follow_up message to the slave, which contains t1 information for later use.
4. The slave sends a Delay_Req message to the master and notes the exact time, t3, at which this packet leaves the internal GMII interface.
5. The master receives the message, capturing the exact time t4, at which the message enters its system.
6. The master sends the t4 information to the slave in the Delay_Resp message.
7. The slave uses the four values of t1, t2, t3, and t4 to synchronize its local timing reference to the timing reference of the master.

Most of the PTP implementation is done in the software above the Ethernet layer. However, the hardware support is required to capture the exact time when specific PTP packets enter or leave the Ethernet port at the internal GMII interface. This timing information must be captured and returned to the software for proper implementation of PTP with high accuracy.

43.4.5.2 Peer-to-Peer PTP Transparent Clock (P2P TC) Message Support

The IEEE 1588-2008 supports peer-to-peer PTP (Pdelay) message in addition to the SYNC, Delay Request, Follow-up, and Delay Response messages. Following figure shows the method to calculate the propagation delay in clocks supporting peer-to-peer path correction.

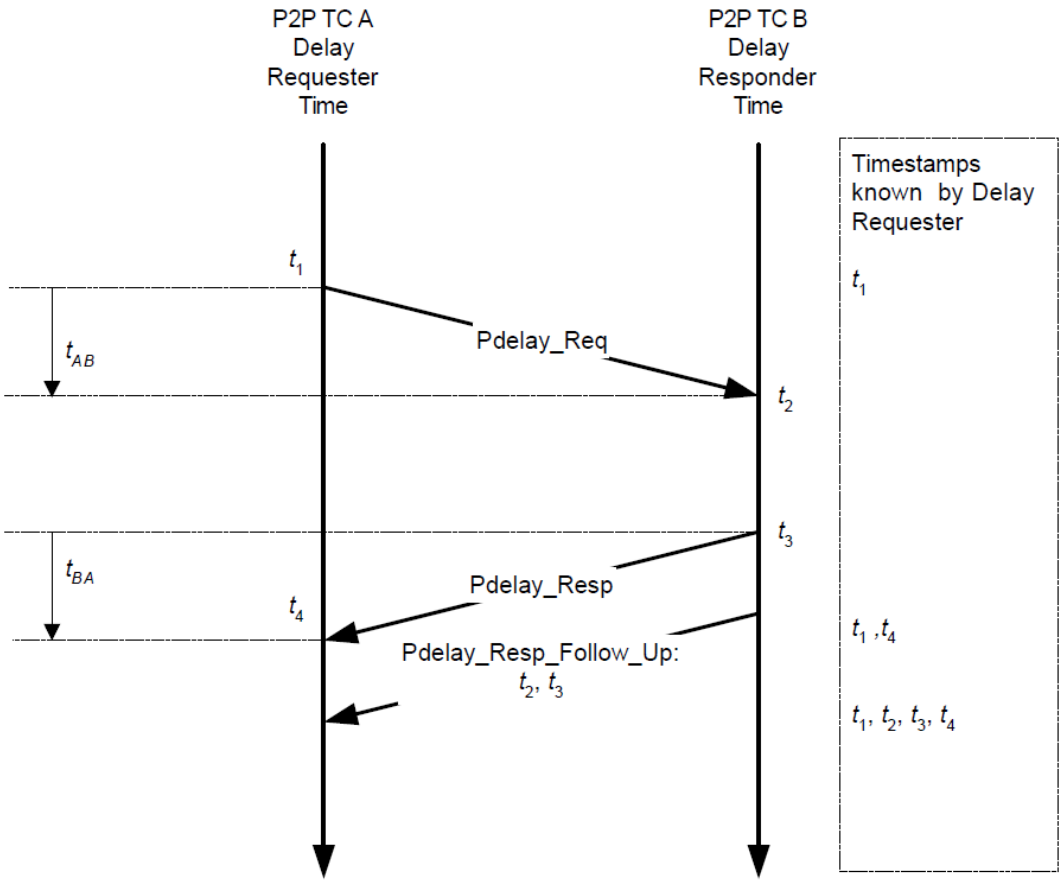


Figure 178. Propagation Delay Calculation in Clocks Supporting Peer-to-Peer Path Correction

As shown in the above figure, the propagation delay is calculated in the following way:

1. Port 1 issues a Pdelay_Req message and generates a timestamp (t1) for the Pdelay_Req message.
2. Port 2 receives the Pdelay_Req message and generates a timestamp (t2) for this message.
3. Port 2 returns a Pdelay_Resp message and generates a timestamp (t3) for this message. To minimize errors because of any frequency offset between the two ports, Port 2 returns the Pdelay_Resp message as quickly as possible after the receipt of the Pdelay_Req message. Port 2 returns any one of the following:
   - Difference between the timestamps t2 and t3 in the Pdelay_Resp message
   - Difference between the timestamps t2 and t3 in the Pdelay_Resp_Follow_Up message
   - Timestamps t2 and t3 in the Pdelay_Resp and Pdelay_Resp_Follow_Up messages, respectively
4. Port 1 generates a timestamp (t4) on receiving the Pdelay_Resp message.
5. Port 1 uses all four timestamps to compute the mean link delay.

43.4.5.3 Timestamp Correction

According to the IEEE 1588 specification, a timestamp must be captured when the PTP message timestamp point (leading edge of the first bit of the octet immediately following the Start Frame Delimiter octet) crosses the boundary between the node and the network. As the MAC takes the timestamp at an internal point far from the actual boundary of the node and network, this captured timestamp is corrected/updated for the ingress/egress path latency (including the delay in the PHY layers). Further correction is done for the inaccuracies/errors introduced due to the clock (internal GMII Tx, Rx clock) being different at the capture point as compared to the PTP clock (clk_ptp_ref_i) that is used to generate the time. The resultant CDC (Clock Domain Crossing) circuits add error depending on the clock period of the internal GMII or MII and PTP clocks.

43.4.5.3.1 Ingress Correction

In the Receive side the timestamp captured at the internal snapshot point is delayed (later in time) as compared to the time at which that packet's SFD bit is received at the port's boundary. Therefore, the captured timestamp must be reduced by the ingress latency and the errors in CDC sampling. This correction value must be determined/calculated by the software and written into the MAC_Timestamp_Ingress_Corr_* registers.

The correction value consists of the following 3 components:

1. External latency in the PHY layer between boundary point and the input of the core.

   If the PHY is compliant to the IEEE 802.3 Clause 45 MMD registers, it has a register indicating the maximum and minimum ingress latency. The software can read these registers and determine the average ingress latency in the PHY. Alternatively (if the PHY does not support these registers), the ingress latency must be determined from its datasheet or timing characteristics.

2. Internal latency from the input of the core to the internal capture point.

   The internal ingress latency can be read from the MAC_Ingress_Timestamp_Latency register. This is a read-only register and gives the latency in scaled Nanoseconds format defined in IEEE 1588 Clause 5.3.2. The latency differs based on the active PHY interface (for example, RMII) and the operating speed. Therefore, the software must read this register after any speed change in the MAC to determine the current internal latency.

3. CDC Synchronization

   The CDC synchronization error is almost equal to 2 times the clock-period of the PTP clock (clk_ptp_ref_i).

The values determined from these 3 components should be added by the software and must be written into the TSIC and TSICSNS fields of the MAC_Timestamp_Ingress_Corr_* registers.

NOTE

The value written to the register must be negative (two's complement as explained below), because it has to be subtracted from the captured timestamp. The MAC Receiver adds the value in this register to the captured timestamp and then gives the resultant value as the timestamp of the received packet.

When TSCTRLSSR bit in MAC_Timestamp_Control register is set, the nanoseconds field of the captured timestamp is in decimal format with a granularity of 1ns. So the Bit31 of TSIC must be set to 1 (for negative value) and bits[30:0] must be written with "10^9 - total ingress_correction_value[nanosecond part]" represented in binary. For example, if the required correction value is -5 ns, then the value is 0xBB9A_C9FB.

When TSCTRLSSR bit in MAC_Timestamp_Control register is reset, the nanoseconds field of the captured timestamp is in binary format with a granularity of ~0.466ns. Therefore, bits[30:0] must be written with "2^31 - total ingress_correction_value" represented in binary with bit[31] = 1.

43.4.5.3.2 Egress Correction

In the Transmit side the timestamp captured at the internal snapshot point is earlier (advanced in time) as compared to the time at which that packet's SFD bit is output at the port's boundary. Therefore, the captured timestamp must be compensated by the egress latency and the errors in CDC sampling. This correction value must be determined/calculated by the software and written into the MAC_Timestamp_Egress_Corr_* registers.

The correction value consists of the following 3 components:

1. External latency in the PHY layer between the output of the core and the boundary of the port and the network

   If the PHY is compliant to the IEEE 802.3 Clause 45 MMD registers, it has a register indicating the maximum and minimum egress latency. The software can read these registers and determine the average egress latency in the PHY. Alternatively (if the PHY does not support these registers), the egress latency must be determined from its datasheet or timing characteristics.

2. Internal latency from the internal capture point and the output of the core

   This internal egress latency can be read from the MAC_Egress_Timestamp_Latency register. This is a read-only register and gives the latency in scaledNanoseconds format defined in IEEE 1588 Clause 5.3.2. The latency will differ based on the active PHY interface (for example, RMII) and the operating speed. Hence the software must read this register after any speed change in the MAC to determine the current internal latency.

3. CDC synchronization error

   The CDC synchronization error value for one-step timestamping would be = (1 * period of clk_ptp_ref_i + 4 * period of clk_tx_i). Otherwise (Two-step timestamping mode), the value = -(2 * period of clk_ptp_ref_i).

43.4.5.3.3 Frequency Range of Reference Timing Clock

The timestamp information is transferred across asynchronous clock domains, from the MAC clock domain to the application/system clock domain. Therefore, a minimum delay is required between two consecutive timestamp captures. This delay is 4 clock cycles of RMII and 3 clock cycles of PTP clocks. If the delay between two timestamp captures is less than this delay, the MAC does not take a timestamp snapshot for the second packet.

43.4.5.3.4 PTP Processing and Control

Following table shows the common message header for the PTP messages. This format is taken from the IEEE 1588-2008.

Table 374. Message Format Defined in IEEE 1588-2008

Bits								Octet	Offset
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
Reserved				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
Reserved								1	5
flagField								2	6
correctionField								8	8
Reserved								4	16
sourcePortIdentity								10	20
sequenceId								2	30
controlField (*)								1	32
logMessageInterval								1	33

(*) – control Field is used in version 1. In version 2, message Type field is used for detecting different message types.

There are some fields in the Ethernet payload that are being use to detect the PTP packet type and control the snapshot to be taken. These fields are given below in table for PTP packets over Ethernet.

Following table provides information about the fields that are matched to control the snapshots for the PTP packets sent over Ethernet for IEEE 1588 version 1 and 2. The octet positions for the tagged packets are offset by 4. This is based on the IEEE 1588-2008, Annex D and the message format.

Table 375. Ethernet PTP Packet Fields Required for Control And Status

Field Matched	Octet Position	Matched Value	Description
MAC Destination Multicast Address[1]	0–5	01-1B-19-00-00-00 01-80-C2-00-00-0E	All PTP messages can use any of the following multicast addresses[2]: • 01-1B-19-00-00-00 • 01-80-C2-00-00-0E[3]
MAC Packet Type	12, 13	0x88F7	PTP Ethernet packet
PTP Control Field (IEEE 1588 version 1)	46	0x00, 0x01, 0x02, 0x03, or 0x04	• 0x00: SYNC • 0x01: Delay_Req • 0x02: Follow_Up • 0x03: Delay_Resp • 0x04: Management
PTP Message Type Field (IEEE 1588 version 2)	14 (nibble)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, or 0xD	• 0x0: SYNC • 0x1: Delay_Req • 0x2: Pdelay_Req • 0x3: Pdelay_Resp • 0x8: Follow_Up • 0x9: Delay_Resp • 0xA: Pdelay_Resp_Follow_Up • 0xB: Announce • 0xC: Signaling • 0xD: Management
PTP Version	15 (nibble)	0x1 or 0x2	• 0x1: Supports PTP version 1 • 0x2: Supports PTP version 2

1. The unicast address match of destination addresses (DA), programmed in MAC address 0 to 31, is used if the TSENMACADDR bit of MAC_Timestamp_Control register is set.
2. IEEE 1588-2008, Annex F
3. The MAC does not consider the PTP version 1 messages with Peer delay multicast address (01-80-C2-00-00-0E) as valid PTP messages.

43.4.5.4 Transmit Path Functions

The MAC captures a timestamp when the Start Packet Delimiter (SFD) of a packet is sent on the internal GMII interface. The packets, for which the timestamps has to be captured can be controlled on per-packet basis. Each Transmit packet can be marked

to indicate whether a timestamp should be captured for it. The MAC does not process the transmitted packets to identify the PTP packets. You can specify the packets by using the control bits in the Transmit descriptor. The MAC returns the timestamp to the software inside the corresponding Transmit descriptor, thus connecting the time-stamp automatically to the specific PTP packet. The 64-bit timestamp information is written to the TDES0 and TDES1 fields. The TDES0 field holds the 32 least significant bits of the timestamp.

43.4.5.5 Receive Path Functions

The MAC can be programmed to capture the timestamp of all packets received on the internal GMII interface or to process packets to identify the valid PTP messages. Use the following options of the MAC_Timestamp_Control register to control the snapshot of the time to be sent to the application:

- Enable snapshot for all packets
- Enable snapshot for IEEE 1588 version 1 or version 2 timestamp
- Enable snapshot for PTP packets transmitted directly over Ethernet
- Enable timestamp snapshot for the received packet for IPv4 or IPv6
- Enable timestamp snapshot only for EVENT messages (SYNC, DELAY_REQ, PDELAY_REQ, or PDELAY_RESP)
- Enable the node to be a master or slave and select the snapshot type

Table 376. Timestamp Snapshot Dependency on Register Bits

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	PTP Messages
00	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp
00	0	1	SYNC
00	1	1	Delay_Req
01	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp,Pdelay_Req, Pdelay_Resp,Pdelay_Resp_Follow_Up
01	0	1	SYNC, Pdelay_Req, Pdelay_Resp
01	1	1	Delay_Req, Pdelay_Req, Pdelay_Resp
10	x	x	SYNC, Delay_Req
11	x	x	Pdelay_Req, Pdelay_Resp

The DMA returns the timestamp to the software inside the corresponding Receive Descriptor. The extended status, containing the timestamp message status and the IPC status, is written in normal descriptor RDES1 and the snapshot of the timestamp is written in RDES0 and RDES1 fields of context descriptor. The RDES0 field holds the 32 least significant bits of the timestamp.

NOTE

For Programming Guidelines for IEEE 1588 Timestamping (System Time Correction), see System Time Correction

43.4.5.6 IEEE 1588 System Time Source

The ENET QOS block supports both external and internal time source for reference timestamping.

- External Time source uses the 64-bit external time reference and clock as input in IP. The clock input is used to synchronize the external timing reference into MAC clock domain. Upper 32-bit indicates the time in seconds while lower 32-bit indicates the time in nanoseconds.
- Internal time source takes only the reference clock input and uses it to internally generate the Reference time (also called the system time) and capture timestamps.
  - UInteger48 seconds Field: The seconds field is the integer portion of the timestamp in units of seconds. It is 48-bits wide.

    For example, 2.000000001 seconds are represented as seconds Field = 0x0000_0000_0002.
  - UInteger32 nano secondsField The nanoseconds field is the fractional portion of the timestamp in units of nanoseconds.

    For example, 2.000000001 seconds are represented as nanoSeconds = 0x0000_0001. The nanoseconds field supports the following two modes:
    - Digital rollover mode: In this mode, the maximum value in the nanoseconds field is 0x3B9A_C9FF, that is, (10e9-1) nanoseconds.
    - Binary rollover mode: In this mode, the nanoseconds field rolls over and increments the seconds field after value 0x7FFF_FFFF. Accuracy is ~0.466 ns per bit.

There is a system time register module which is being used when internal time reference is being used. The 80-bit time is maintained in this module and updated using the input reference clock (clk_ptp_ref_i). This time is the source for taking snapshots (timestamps) of Ethernet packets being transmitted or received at the internal GMII interface.

The system time counter can be initialized or corrected using the coarse correction method. In this method, the initial value or the offset value is written to the Timestamp Update register. For initialization, the system time counter is written with the value in the Timestamp Update register. For system time correction, the offset value is added to or subtracted from the system time.

In the fine correction method, the frequency offset and/or frequency drift of a slave clock (clk_ptp_ref_i) with respect to the master clock (as defined in IEEE 1588-2002) is corrected over a period of time instead of in one clock, as in coarse correction. This helps maintain linear time and does not introduce drastic changes (or a large jitter) in the reference time between PTP Sync message intervals.

NOTE

For Programming Guidelines for IEEE 1588 Timestamping (For Internal Time-stamp Source Configuration). see Initialization Guidelines for System Time Generation

43.4.5.7 IEEE 1588 Higher Word Register

If system time source is internal, then higher word register can be invoked. MAC timestamp is 64-bit wide. The values of the upper 16-bits of the seconds field can be read from the CSR register.

43.4.5.8 Flexible Pulse-Per-Second Output

The ENET QOS block supports the flexibility of programming the start or stop time, width and interval of pulse generated on pulse per second (PPS) output. IP supports 4 such output signals. By default, the block is in fixed pulse per second output mode with an interval of 1 second.

Initially. the start time should be programmed in target time registers "MAC_PPSx_Target_Time_Seconds and MAC_PPSx_Target_Time_Nanoseconds" for all desired/enabled PPS output. Reprogramming the start or stop time must be done after the synchronization of the earlier programmed value. Bit 31 of the MAC_PPS#_Target_Time_Nanoseconds register indicates that the synchronization is complete. If the application programs a start or stop time that has already elapsed, the MAC sets an error status bit indicating the programming error. If enabled, the MAC also sets the Target Time Reached interrupt event. The application can cancel the start or stop request only if the corresponding start or stop time has not elapsed. If the time has elapsed, the cancel command has no effect.

The PPS width and interval are programmed in terms of granularity of system time, that is, number of the units of sub-second increment value. For example, to have a PPS pulse width of 40 ns and interval of 100 ns with PTP reference clock of 50 MHz, you

should program the width and interval to values 2 and 5, respectively. You can achieve smaller granularity by using a faster PTP reference clock. Before giving the command to trigger a pulse or pulse train on the PPS output, you should program or update the interval and width of the PPS signal output.

NOTE

See Programming Guidelines for Flexible Pulse-Per-Second Output

43.4.5.9 Media Clock Generation and Recovery

Media clock recovery and generation needs a dedicated hardware and software to complete the operation. IP generates a reference clock with the help of dedicated hardware which is then multiplied to get the desired media clock. Software extracts the presentation time from the incoming PTP 1722 frames and programs these value into CSR registers so IP can read them for media clock recovery.

As shown in the following figure, the timestamp array and the DMA module are external to IP. IP supports a 32-bit presentation time counter in nanoseconds that wraps at the full32 bit value to match up with 1722 presentation time format and an handshake mechanism with external application(such as DMA) to program the presentation time into CSR register.

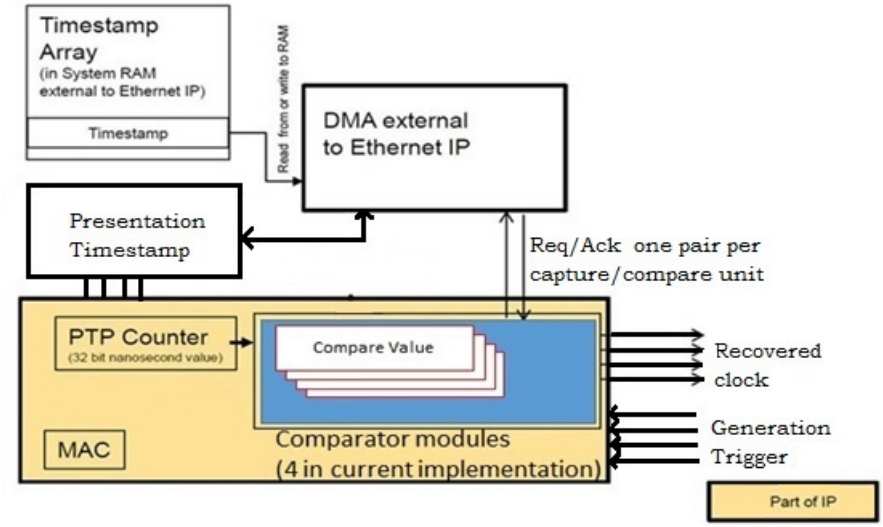


Figure 179. Block Diagram of Media Clock Generation and Recovery

43.4.5.9.1 Presentation Time Counter

Presentation time counter is another view of the PTP System time (1588 timer). For a given PTP System time (PTP[63:32] represents time in seconds and PTP[31:0] represents time in nanoseconds), there exists a corresponding Presentation Time (32-bit value in nanosecond). The Presentation time computed is referred to as Current Presentation Time (CPT).

The Current Presentation Time (CPT) can be viewed as derived from 64bit PTP System Time. PTPNS[63:0] = (PTP[63:32] * 32'd1,000,000,000) + PTP[31:0] where, PTPNS is the PTP system time converted into a 64bit nanosecond value.

Current Presentation Time[31:0] = PTPNS[31:0]

Media Clock Recovery is possible when the System Time is internally generated in IP and CPT is computed in the same way as the Internal PTP System Time generation. Because CPT and System Time values are in nanoseconds, and must be synchronous, the increment cycle of both the timers are same. The timer is updated at the same instance, but the updated value could be different. The software must compute a separate 32-bit update value in nanoseconds for the CPT update. CPT sampled at different edges of a triggering input results in the generation of Media Clock timestamps which are inserted in 1722 based AVBTP packets, to recover the clock at the destination.

43.4.5.9.2 Comparator Modules

As shown in the figure 'Block Diagram of Media Clock Generation and Recovery' the module supports four comparator modules to handle multiple media clocks that might be required. The Comparator modules hand shake with the application to

Program the timestamps into the MAC_PPS(#i)_Target_Time_Seconds register when PTGE field of MAC_Timestamp_Control register is set.

Set the Presentation Time Control bits of the particular instance for recovery mode.

The timestamps received from the application are referred as Target Presentation Time (TPT).

When MCGREN#i field of MAC_PPS_Control register is set to 1, the IP operates in MCGR mode. The comparator module sends up to two requests to the application for TPT write, when PTGE field of MAC_Timestamp_Control register is set and the Presentation time control bits of the particular instance is set for MCGR mode. Subsequent requests are generated each time a Presentation Time match occurs. (CPT transitions from a value less than TPT, to a value greater than or equal to CPT). The first request is asserted when a specific comparator instance is set to MCGR mode with a non-zero Presentation Time Control and an additional request is made when the first data is received into the Comparator. This allows the application to write the next TPT value while IP processes the previous TPT value for a match.

The TPT read from the MAC_PPS(#i)_Target_Time_Second is considered as a future time. A toggle/pulse is generated in the next cycle when a match is detected. When a match is detected, the mcgr_dma_req_o#i request for that comparator is asserted (to obtain next TPT) until the corresponding application acknowledgment is set.

The presentation control bits (PPSCMD#i field of MAC_PPS_Control register) determine the shape of the generated waveform. Also, these bits can be programmed to either toggle or generate a high/low pulse for one PTP clock cycle, upon match.

Media Clock Recovery: A match is when the free-running CPT value matches the received TPT value. A output signal EMAC_TMR signal as shown in the IO mux sheet is asserted (toggle, low pulse, or high pulse) based on the presentation control value programmed.

Media Clock Generation: Based on the presentation control value programmed in MAC_PPS_Control register, the particular comparator captures the presentation time and programs it into MAC_PPS(#i)_Target_Time_Second register. A request is raised to the application to read the captured timestamp. Until the acknowledgment is received from the application, acknowledging that the read operation is complete, no new timestamps are captured.

43.4.5.9.3 Media Clock Generation and Recovery Flow

Following figure shows the media clock generation and recovery flow. For both media clock generation and recovery, set the PTGE field of MAC_Timestamp register for CPT generation and set the MCGREN#i field of MAC_PPS_Control Register for the corresponding instance to be enabled in the MCGR mode.

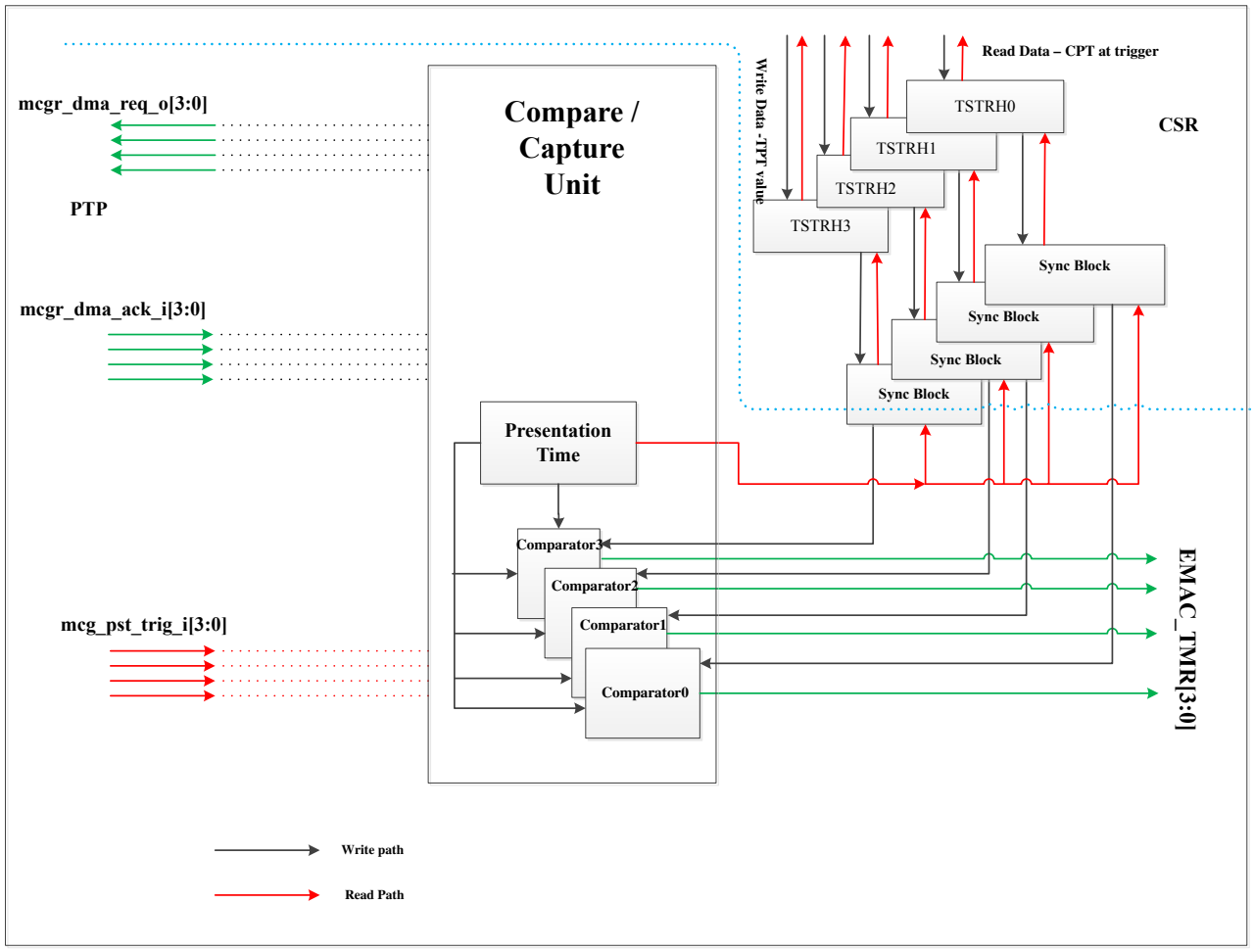


Figure 180. Media Clock Generation and Recovery Flow

NOTE

1. The consecutive triggers to sample the presentation time should always be asserted after a few PTP clock cycles to allow for synchronization delays. (This is not an issue when the input trigger maximum frequency is 8KHz and the PTP clock is running at least at 1MHz)
2. DMA acknowledgment can be both posted as well as non-posted, in the Media Clock Recovery Mode.
3. See "Programming Guidelines for Media Clock Generation and Recovery"

43.4.5.10 One step timestamp

IP supports the one step time stamp feature. It can be enabled for a packet by setting the Bit 20 (OSTC) in the control word.

NOTE

See Programming Guidelines for IEEE 1588 Timestamping .

43.4.5.11 IEEE 1588 Sub Nanoseconds Timestamp

The ENET QOS block supports the ingress and egress correction accuracy in sub-nanoseconds. It is programmed in the TSICSNS and TSECSNS fields of the MAC Timestamp Ingress Correction Subnanosecond and MAC Timestamp Egress

Correction Subnanosecond registers, respectively. In this case, the correction has an unit of nanosecond, multiplied by 2^8. The least significant 8 bits of the value should be programmed in the sub-nanoseconds register.

For example, if the required egress correction is 3.6 nS and TSCTRLSSR bit is set, then TSEC must be 0x3 and TSECSNS must be 0x99 (0.6 * 256). Similarly, if the required ingress correction is -3.6 ns and TSCTRLSSR bit is set, then TSIC must be 0xBB9A_C9FC and TSICSNS must be 0x66 (fractional part of (10^9 - 3.6) * 256).

43.4.6 Multiple Channels and Queues Support

IP supports the five queues and channel on Tx and Rx paths.

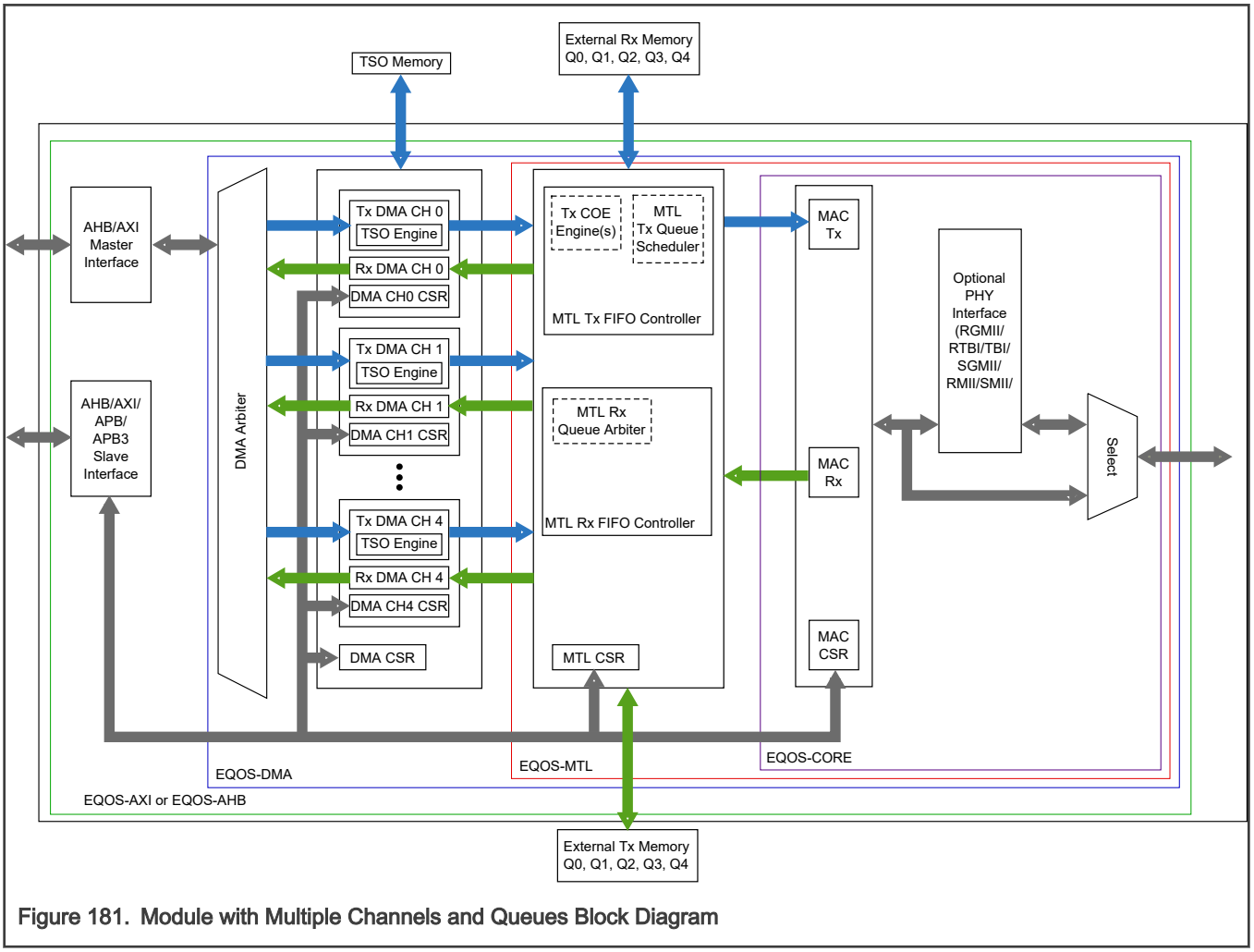


Figure 181. Module with Multiple Channels and Queues Block Diagram

DMA arbiter of IP supports two types of arbitration scheme, fixed priority and weighted round-robin. The DMA arbiter performs the arbitration between the Tx and Rx paths of DMA channels for accessing descriptors and data buffers. The DA bit of the DMA_Mode register specifies the arbitration scheme (fixed or weighted round-robin) between the Tx and Rx DMA of a channel.

Priority between the corresponding Tx DMA and Rx DMA can be set by using TXPR field of the DMA_Mode register. PR field of the DMA_Mode register can be set to specify the weighted priority between the Tx DMA and Rx DMA in round robin arbitration scheme.

Following table provides information about the priority scheme between Tx DMA and Rx DMA.

Table 377. Priority Scheme for Tx DMA and Rx DMA

Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Priority Schemes
x	x	x	0	1	Rx always has priority over Tx
0	0	0	0	0	Tx and Rx have equal priority. Rx gets the access first on simultaneous requests
0	0	1	0	0	Rx has priority over Tx in ratio 2:1
0	1	0	0	0	Rx has priority over Tx in ratio 3:1
0	1	1	0	0	Rx has priority over Tx in ratio 4:1
1	0	0	0	0	Rx has priority over Tx in ratio 5:1
1	0	1	0	0	Rx has priority over Tx in ratio 6:1
1	1	0	0	0	Rx has priority over Tx in ratio 7:1
1	1	1	0	0	Rx has priority over Tx in ratio 8:1
x	x	x	1	1	Tx always has priority over Rx
0	0	0	1	0	Tx and Rx have equal priority. Tx gets the access first on simultaneous requests
0	0	1	1	0	Tx has priority over Rx in ratio 2:1
0	1	0	1	0	Tx has priority over Rx in ratio 3:1
0	1	1	1	0	Tx has priority over Rx in ratio 4:1
1	0	0	1	0	Tx has priority over Rx in ratio 5:1
1	0	1	1	0	Tx has priority over Rx in ratio 6:1
1	1	0	1	0	Tx has priority over Rx in ratio 7:1
1	1	1	1	0	Tx has priority over Rx in ratio 8:1

43.4.6.1 Multiple Queues and Channels Support in the Transmit Path

IP supports the five transmit queues and five channels.

NOTE

See Programming Guidelines for Multi-Channel Multi-Queuing

43.4.6.2 Multiple Queues and Channels Support in the Receive Path

IP supports five receive channels and five queues.

In the Receive direction, the MTL Rx Controller selects the Rx DMA for which it is transferring or reading the data from the Rx FIFO memory. This scheduling is based on the programming done in the respective MTL_RxQ[x]_Control register.

Each Rx DMA indicates when it is ready to transfer data and the size of the burst-length (number of beats) that it has to transfer. The scheduler checks whether sufficient data (of requested burst length) is available to be transferred to these DMAs and then selects the Rx DMA that gets serviced using the programmed priorities.

NOTE

See Programming Guidelines for Multi-Channel Multi-Queuing .

43.4.6.3 Rx Queue to DMA Mapping

The packets in the MTL Rx Queues can be routed to any one of the multiple DMA channels by programming the register MTL_RxQ_DMA_Map0 Register for queues 0 through 3, and MTL_RxQ_DMA_Map1 for queue 4.

The following types of Rx Queue to DMA mapping are possible through programming.

43.4.6.3.1 Static Mapping

In this mode, all of the packets of an Rx Queue are connected to a specific DMA channel. For example, all the packets from Rx Queue 0 can be routed to a DMA channel by programming Q0MDMACH (bit[3:0]) and Q0DDMACH (bit[7] = 0) of the MTL_RxQ_DMA_Map0 Register.

Similarly, packets from other Rx Queues can be routed to any DMA channel by programming register fields corresponding to each Queue.

43.4.6.3.2 Dynamic (Per Packet) Mapping

In this mode, the destination DMA channel of a packet being read from a Rx Queue is not constant but decided independently for each packet. For example, if you set the Q1DDMACH bit of the MTL_RxQ_DMA_Map0 register, the static mapping is disabled for Rx Queue 1 and the value in Q1MDMACH is ignored. The destination DMA channel is decided by the MAC receiver for each packet, depending on the following in decreasing order of priority:

1. L3-L4 filter Based DMA Selection: The TCP/UDP and IP header fields of the received packet are matched against the corresponding values programmed and enabled for comparison in the MAC_L3_L4_Address_Control register. If the match is successful, the DMA channel number programmed in the DMCHN field of the MAC_L3_L4_Ad-dress_Control register is selected as the destination DMA channel number provided DMCHEN bit of the same register is set. If none of the L3-L4 Registers give a comparison match, then the module proceeds to the next step below.
2. Extended VLAN Based DMA Selection: Extended routing is applicable only if the VLAN Filter has passed. Routing is only based on Perfect Filter Result. Each perfect filter has a DMA Channel Enable and a DMA Channel Number field which can be programmed. Routing is applicable for a filter, only if the DMA Channel Enable bit is set. The frame is routed to the smallest Matching Filter's DMA Channel provided it is enabled. If that filter's DMA Channel number is not enabled, the frame gets routed to Channel 0. For example, if a frame's VLAN tag matches filters 7, 3, and 1. Then the MAC checks if Filter 1's DMA Channel Number is enabled through programming. If yes, the frame gets routed to the programmed value; otherwise it gets routed to DMA. When the inverse filter is enabled; is routed to the least mismatched filter's DMA channel number provided it is enabled. If the DMA Channel enable bit is not set, then the frame is routed based on DA based addressing or to Channel 0.

   If Hash Filter is also enabled, it is used to determine the Filter result only. Routing will still depend on the Perfect Filters enabled. If none of the perfect filters are enabled or if all of them are bypassed, then the VLAN Filter based routing will not take place. The frame will be routed via DA Based addressing or to Channel 0. If all the perfect filters give a fail result and the Hash Filter has passed, then the VLAN Filter result is a pass but routing will be based on DA Based Addressing or to Channel 0. Similar behavior is seen when inverse Filtering is enabled as well.
3. Ethernet DA-Based DMA Selection: The DA address of the received packet is compared against the programmed DA values in MAC Address Registers. If the address matches any of the programmed values, the corresponding DCS field (when enabled) determines the destination DMA channel number.

If none of the previous operations is able to make a successful match/decision, then the packet is routed to DMA Channel 0 by default.

43.4.6.3.2.1 Broadcast/Multicast Packet Duplication

Broadcast/Multicast Packet Duplication feature provides a mechanism to send the received Broadcast/Multicast packets to multiple DMA channels.

The DCS field of the of MAC_Address0_High and other optional MAC_Address(#i)_High (i=1 to 2) registers determines the DMA channel number to which the received packet (that matches the MAC Address present in that register) must be routed.

Packet duplication is not supported in other DMA channel routing mechanisms such as Extended VLAN, or L3-L4 based routing.

The Packet Duplication feature is supported only on the highest MTL Queue configured. Therefore, packet duplication for Broadcast/Multicast packets requires that the MCBCQ field of MAC_RxQ_Ctrl1 register is programmed to the highest RxQ present in the configuration.

43.4.6.4 Selection of Tag Priorities Assigned to Rx Queues

The VLAN Tag priorities can be assigned to Rx Queues by programming the PSRQ field in the corresponding MAC_RxQ_Ctrl2 registers. The bit corresponding to the VLAN Tag priority can be set in the PSRQ field for assigning that priority to the Rx Queue. For example, if you want to assign VLAN Tag priority of 3 to Rx Queue 0, set bit [3] in PSRQ field of MAC_RxQ_Ctrl2 register. The VLAN Tag priority assigned to particular Rx Queue must be unique, that is, more than one Rx Queue cannot be assigned the same VLAN Tag priority. However, more than one VLAN Tag priorities can be assigned to same Rx Queue.

The settings in the PSRQ field is used for VLAN Tagged Rx packet routing to Rx Queues as well as for PFC based Tx flow control. The received VLAN Tagged Rx packet is routed to Rx Queue that has the VLAN Tag priority match. In PFC based Tx flow control, PSRQ field corresponding to a particular Rx Queue is used for enabling VLAN Tag priorities in the PFC packet transmitted when corresponding Rx Queue threshold levels are reached.

43.4.6.5 Rx Side Routing from MAC to Queues

IP supports Rx Side Routing from the MAC to Queues. The MAC routes the Rx packets to the Rx Queues based on following packet types in that order

- Unicast/Multicast destination address packets that fail the destination address filter
- Multicast/Broadcast destination address packets that pass the destination address filter
- VLAN Tag Priority field in VLAN Tagged AV data packets
- AV Control packets
- VLAN Tagged IEEE 1588 PTP over Ethernet packets
- Untagged IEEE1588 PTP over Ethernet packets
- VLAN Tag Priority field in VLAN Tagged packets
- Untagged packets

The outer C-VLAN Tagged AV data Rx packets can be routed based on the priorities assigned to Rx Queues through PSRQ field in corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers and the corresponding Rx Queue is enabled for AV through RXQ#EN field in MAC_RxQ_Ctrl0 register. These packets may be single VLAN Tagged with C-VLAN type or Double VLAN Tagged with outer VLAN Tag of C-VLAN type when Double VLAN feature is enabled (EDVLP bit in MAC_VLAN_Tag register is set) with inner C-VLAN Tagged or inner S-VLAN Tagged when SVLAN processing is enabled (ESVL bit in MAC_VLAN_Tag register is set). This type of Rx packet routing is available when AV feature and Multiple Rx Queues are selected in the configuration.

The AV Control Rx packets can be routed based on the Rx Queue number specified in the AVCPQ field in MAC_RxQ_Ctrl1 register and corresponding Rx Queue is enabled for AV through RXQ#EN field in MAC_RxQ_Ctrl0 register. These packets may be single VLAN Tagged with C-VLAN type or Double VLAN Tagged with outer VLAN Tag of C-VLAN type when Double VLAN feature is enabled (EDVLP bit in MAC_VLAN_Tag register is set) with inner C-VLAN Tagged or inner S-VLAN Tagged when SVLAN processing is enabled (ESVL bit in MAC_VLAN_Tag register is set). This type of Rx packet routing is available when AV feature and Multiple Rx Queues are selected in the configuration.

The VLAN Tagged Rx packets can be routed based on the priorities assigned to Rx Queues through PSRQ field in corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers and the corresponding Rx Queue is enabled for DCB/generic through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

The untagged IEEE 1588 PTP over Ethernet Rx packets can be routed based on the Rx Queue number specified in the PTPQ field in MAC_RxQ_Ctrl1 register and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. The VLAN tagged IEEE 1588 PTP over Ethernet Rx packets can be routed based on the priorities assigned to Rx Queues through PSRQ field in corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers or the Rx Queue number specified in the PTPQ field in MAC_RxQ_Ctrl1 register and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This is determined by programming in TPQC field of MAC_RxQ_Ctrl1 register. This type of Rx packet routing is available when IEEE 1588 Timestamp feature support and Multiple Rx Queues are selected in the configuration. The VLAN Tagged IEEE 1588 PTP over Ethernet Rx packets are detected only when 802.1AS mode is disabled (AV8021ASMEN bit in MAC_Timestamp_Control register is set to 0), otherwise this type of Rx packets are routed as generic VLAN Tagged Rx packets.

The multicast/broadcast destination address Rx packets that pass the destination address filter can be routed based on the Rx Queue number specified in the MCBCQ field of MAC_RxQ_Ctrl1 register when enabled through MCBCQEN bit of MAC_RxQ_Ctrl1 register and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

The untagged Rx packets can be routed based on the Rx Queue number specified in the UPQ field of MAC_RxQ_Ctrl1 register and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

The unicast destination address Rx packets that fail the destination address filter can be routed based on the Rx Queue number specified in the UFFQ field of MAC_RxQ_Routing_Ctrl register when enabled through UFFQE bit of MAC_RxQ_Routing_Ctrl register, RA bit of MAC_Packet_Filter register is set to 1 and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

The multicast destination address Rx packets that fail the destination address filter can be routed based on the Rx Queue number specified in the MFFQ field of MAC_RxQ_Routing_Ctrl register when enabled through MFFQE bit of MAC_RxQ_Routing_Ctrl register, RA bit of MAC_Packet_Filter register is set to 1 and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

If the Rx packet cannot be classified in any of the defined packet type for routing, it will be routed through Rx Queue 0.

43.4.6.6 Rx Side Arbitration Between DMA and MTL

IP controller supports Rx side arbitration between the DMA and the MTL.

After completing the current packet processing, the DMA Channel Controller fetches the next descriptor. After the descriptor fetching is complete, the DMA Channel Controller evaluates the amount of data to be to be transferred to the Rx Buffer based on the programmed PBL and Rx Buffer Length. Accordingly, it requests the MTL to transfer the data.

After servicing the current request, the MTL Rx Queue arbitration scheme selects Rx Queue based on the arbitration scheme (RAA field of the MTL_Operation_Mode register) and the weights programmed in the Queue <n> Receive Control Register. The arbitration is done among Queues for which DMA is ready to service. After the Rx Queue is selected, PBL (Programmable Burst Length) amount of data is read out from that Queue and is routed to the Rx DMA Channel based on the Rx Channel selection criteria.

The arbitration takes place when every PBL data transfer is completed and descriptors are ready for the processing from at least one DMA Channel.

43.4.6.7 Tx Side Arbitration between DMA and MTL

IP support Tx side arbitration between the DMA and the MTL. Since the number of Tx DMA channels is always equal to the number of Tx Queues in the MTL, they are always mapped directly. For instance, each Tx DMA pushes data into its respective Tx Queue assigned to it.

The data inside each Tx Queue is stored in packets. Hence, if two DMAs are allowed to transfer data into the same queue, when a Tx DMA starts a packet transfer, the other DMA cannot transfer data unless the previous packet is completely pushed-in. This means that the second DMA remains idle until the first packet is transferred. Hence, each DMA is always connected directly to its corresponding Tx Queue.

43.4.6.8 Time-Sensitive Networking (TSN)

The ENET QOS block supports TSN in 100 Mbps or 1000 Mbps mode, which enables the transfer of time-sensitive traffic over a network. The block implements the IEEE 802.1Qbv-2015 (Enhancements to Scheduling Traffic), IEEE 802.3br (Interspersing Traffic) and IEEE 802.1Qbu (Frame Preemption) TSN protocol.

The Transmit path of Queue 0 supports the strict priority algorithm and is used for best-effort traffic while transmit paths of additional queues support traffic management by using the credit-based shaper algorithm. For a queue, the credit-based shaper algorithm determines that a queue is available for transmission if the following conditions are true:

- The queue contains one or more packets.
- The credit for the queue is positive as per the algorithm.
- AV traffic can be received on all queues. The block can also receive untagged PTP packets in addition to AV traffic on any queue.

43.4.6.8.1 Enhancements to Scheduled Traffic (EST)

The IEEE 802.1Qbv-2015 defines the schedule for each of the queues on every egress port which makes the implementation aware of traffic arrival schedule. This information can be used to block the lower priority traffic from transmission in this time window/slot. This ensures that scheduled traffic is forwarded from sender to receiver through all the network nodes with a deterministic delay.

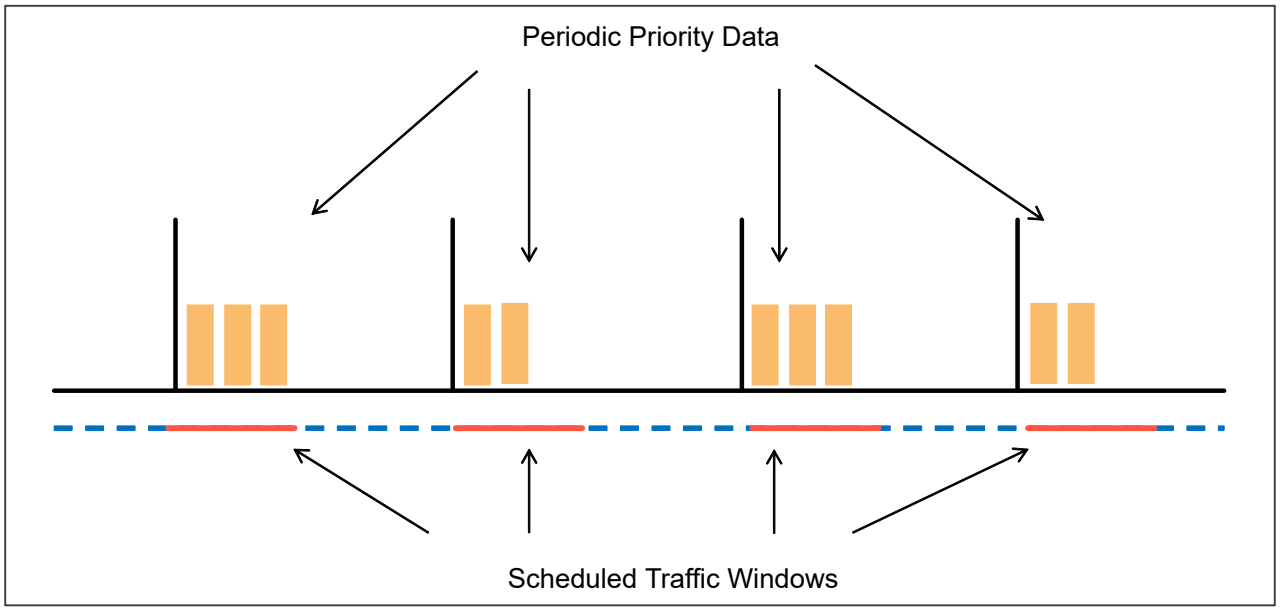


Figure 182. Time Aware Shaper Implementation for Traffic Scheduling

One of the important requirements to achieve a low latency is to ensure there are no interfering frames during the scheduled windows that are reserved for high priority traffic. The use of scheduled traffic imposes limitations while starting a transmission.

As shown in the following figure, if an interfering frame begins transmission just before the start of a reserved time period, it can extend transmission into the reserved window, and potentially interfere with higher priority traffic. Therefore, a guard band whose size is equal to the largest possible interfering frame is required before the window starts.

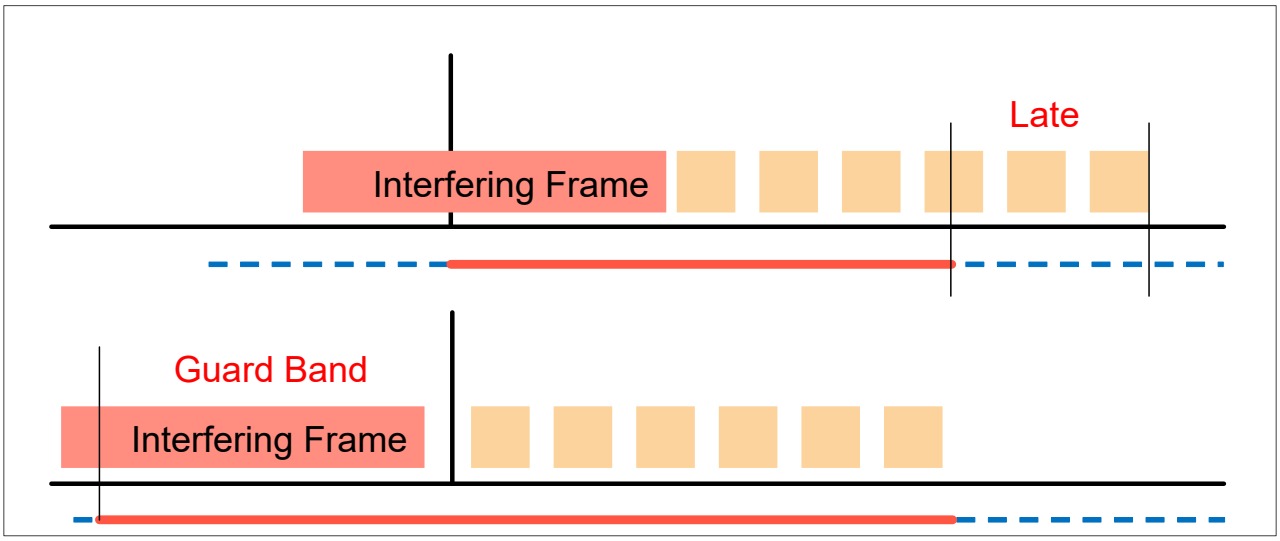


Figure 183. Implementing a Guard Band to Avoid Delays Due to Interfering Frames

A larger guard band equates to a less efficient use of network bandwidth. However, with the implementation of IEEE802.1Qbu (frame preemption), this issue is addressed. Frame preemption breaks the interfering frame into smaller fragments. Therefore, the guard band needs to be only as large as the largest possible interfering fragment instead of the largest possible interfering frame.

During the guard band, only the frames that can complete the transmission of the entire frame before the next gate close event are permitted. This ensures that the high priority traffic can always start at the beginning of the window reserved for it.

43.4.6.8.1.1 Frequently Used Terms in EST

- Gate Control List: The list in the hardware memory that holds the gate controls and the associated time intervals.
- Gate Controls: For a given schedule (row in gate control list) there is a Gate Open (O) and Gate Close (C) state associated with each traffic class. The set of O or C values, whose width is same as configured TCs is called the Gate Controls. Example: CCOCO means TC4=C, TC3=C, TC2=O and so on.
- Time Interval: Time interval (in nanoseconds) is a 16, 20, or 24-bit configurable field in the Gate Control list that indicates the time for which the associated gate controls are valid.
- Base Time Register: Each Gate Control List is associated with a 64-bit Base Time Register that holds the start time (in PTP format) for the list.

43.4.6.8.1.2 Updates to the Transmit Scheduling to Support EST

To support EST, following updates are required to the transmit scheduling:

- Implementation of Gate Control List (GCL)
- Enforcing gate controls in the scheduler
- Accounting for Gate Open (O) duty cycle in the computation of idleSlope (CBS)

43.4.6.8.1.2.1 Implementation of Gate Control List (GCL)

Following figure shows block diagram from the IEEE 802.1Qbv spec that illustrates how the gate control list governs the gate close(C) and open(O) events based on the schedule provided for each event. GCL has the following two parts:

- Time Interval: Defines the time in nano seconds for which the gate controls are valid and should be applied before reading the next gate controls from the list. Configurable width of 16, 20 or 24 to represent a maximum of 64us, 1ms, or 16ms

schedule interval respectively. Supports left shift of the time interval upto 7 bits to be able to apply a multiplication factor from 1 to 128ns (in steps of 2(power)n). The maximum value (post shifting) of this field must be 999,999,999 ns.

- Gate Control: Defines the Open (O) represented by logic 1 or Close (C) represented by logic 0 state for the gate of each TC.

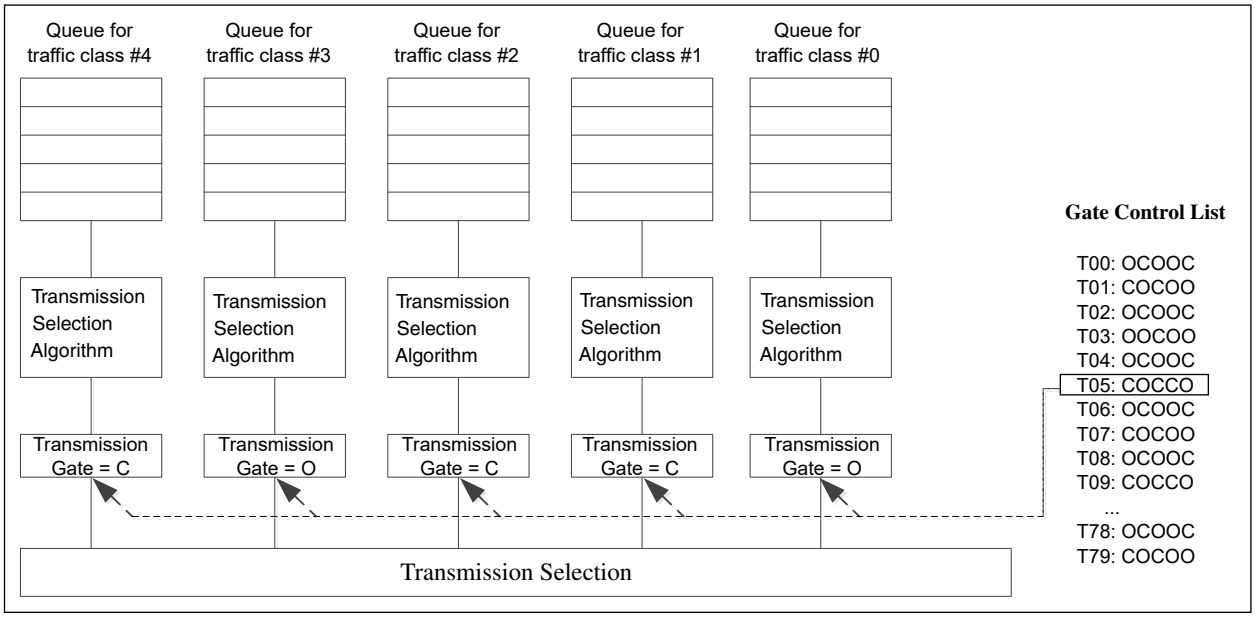


Figure 184. Block Diagram from IEEE 802.1 Qbv Specification - GCL Governing Gate Close and Open Events

The implementation of GCL consists of the following two Gate Control Lists:

- HWOL - Hardware Owned List which is a list for hardware access.
- SWOL - Software Owned List which is a list for software access.

The access to these lists is mutually exclusive. The ownership to the list is set by hardware in the SWOL field of MTL_EST_Status register. Following table provides the implementation details of GCL.

Table 378. External Memory Used for Holding the Two Gate-Control Lists

Gate Control (up to 5 bits)	Time Interval (ns) 16, 20, or 24 bits	
OOCCC	T0	HWOL or SWOL
OOOOC	T1	
CCCCO	T2	
CCCCC	T3	
I	I	
I	I	

Table continues on the next page...

Table 378. External Memory Used for Holding the Two Gate-Control Lists (continued)

Gate Control (up to 5 bits)	Time Interval (ns) 16, 20, or 24 bits	
OCOCO		
OOOOC	T last	
OOOOC	T0	SWOL or HWOL
OOOOC	T1	
OOCCC	T2	
OOOOC	T3	
I	I	
I	I	
OCOCC		
OOCCC	T last	

43.4.6.8.1.2.2 Registers Related to Gate Control List

Following are the set of 4 registers (one for each GCL) related to GCL. These registers are implemented through Indirect Addressing using the GCL_Control and GCL_Data registers.

1. 64-bit Base Time Register (BTR)

2. 40-bit Cycle Time (CTR)

3. m-bit Time Extension (TER)

4. n-bit [Gate Control] List Length (LLR) (n depends of the configured GCL depth)

43.4.6.8.1.2.2.1 Base Time Register (BTR)

Base Time Registers is a 64-bit register that holds the start time to execute the GCL. The format of the BTR is same as the PTP format (upper 32-bits holds time in seconds and lower 32 bits hold time in nano seconds). Once the execution of a given list begins, the implementation can update the value in BTR to indicate the next list execution begin time.

43.4.6.8.1.2.2.2 Cycle Time Register (CTR)

Cycle Time Register is a 40-bit register that holds the time at which the execution of the GCL should be repeated. The Cycle Time Register consists of an 8-bit value in seconds, and a 32-bit value in nano seconds (similar to the PTP time format with truncated seconds register). For a given Gate Control List the start time is "Base Time" + N * "Cycle Time" where N is an integer value representing the iteration number starting with 0 for first iteration. If the Gate Control List execution takes longer than the Cycle time, then the list is truncated at the Cycle Time and the subsequent loop begins at Cycle Time.

43.4.6.8.1.2.2.3 Time Extension Register (TER)

Time Extension Register (TER) is a m-bit (where m = Configured Time Interval width + 7) register that holds the amount of time (in nano seconds) the current Gate Control List can be extended before switching to the new Gate Control List. This will be useful in avoiding small fragments of the current list being executed before switching to a new list.

43.4.6.8.1.2.2.4 List Length Register (LLR)

List Length Register (LLR) is an n-bit register (when n is 7, 8, 9, 10, or 11 for a GCL configured depth of 64, 128, 256, 512, or 1024 respectively) that holds the integer value of the length of the GCL (that is the number of valid rows in each GCL). The processing of the GCL stops after the number of rows read equals to the LLR value.

43.4.6.8.1.2.3 Transmission Gating Implementation

A Bridge or an end station can be enhanced to allow transmission from each TC that is yet to be scheduled relative to a known timescale. To achieve this, a transmission gate is associated with each TC; the state of the transmission gate determines if the queued frames can be selected for transmission.

For a TC, the transmission gate can be in one of the following two states:

1. OPEN:Queued frames are selected for transmission, in accordance with the definition of the transmission selection algorithm associated with the TC.
2. CLOSED: Queued frames are not selected for transmission.

A frame on a traffic class queue is not available for transmission if the transmission gate is in the closed state or if there is insufficient time available to transmit the entirety of that frame before the next gate-close event associated with that queue.

The implementation has visibility to the current schedule of gate controls and the immediate next schedule of the gate controls. So the maximum Gate Open period does not exceed the sum of the two Time Intervals. This is because, a frame is selected for transmission only if the gate is currently Open and the duration of gate open interval is greater than the time taken to transfer the entire frame.

The implementation must know the frame size before the transmission, so that the transmission overruns can be avoided and only the frames that can complete are scheduled at all times.

The implementation adequately compensates for the implementation delays in the data transfer from the buffer to the line by offsetting the current time with all the relevant delays (provided by the CTOV field of MTL_EST_Control register). This ensures that the schedule provided is always accurately implemented at the line.

43.4.6.8.1.3 Idle Slope Computation Updates

When EST is enabled, credit is accumulated only when the gate is open; therefore, the effective data rate of the idleSlope must be increased to reflect the duty cycle for the transmission gate associated with the queue.

The idleSlope is computed based on the GateOpenTime and OperCycleTime values. Program the idleSlope registers (implemented one per CBS enabled TC) based on the following equation. The existing MTL register has sufficient bit width to accommodate the new values for idleSlope.

idleSlope = (operIdleSlope(N) * OperCycle/GateOpenTime)

43.4.6.8.1.4 Operational Details of GCL

Set the switch to software List, the SSWL field of the MTL_EST_Control Register, so that hardware can access the programmed gate control list. The first set of gate controls are applied when the current time is equal to the value in the Base Time Register (BTR) and is held until the programmed "Time Interval" value.

To avoid transmission overruns, one additional gate control event is always read ahead from the list. This enables the GCL to determine the next gate close events (if any) for the TCs that are open.

The scheduling based on the Gate Open state and Time Interval of only the current and subsequent schedule. An internal accumulator is used to add the time intervals when gate controls are applied. BTR + Accumulator holds the time at which the next set of gate controls are to be applied.

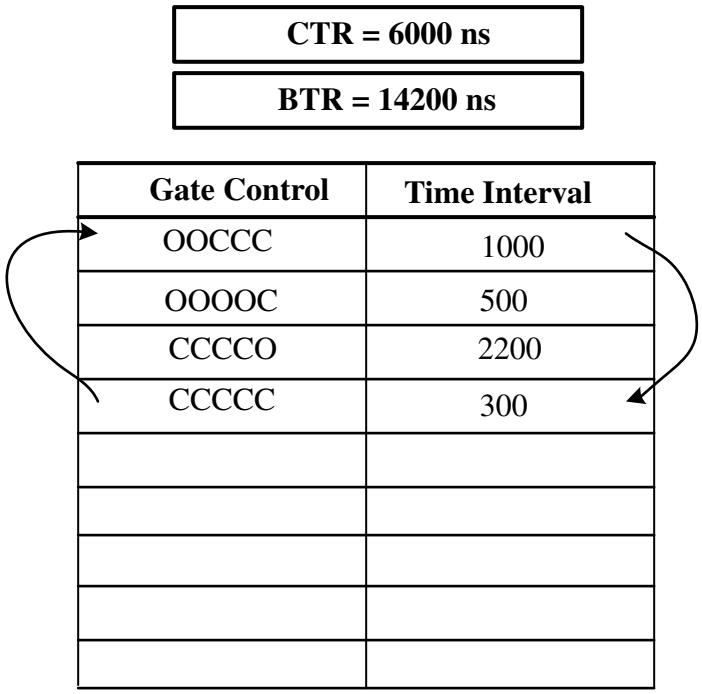


Figure 185. GCL and Associated Registers - BaseTime and CycleTime

The GCL is read progressively from the first row adhering to the schedule. The read operations continue until the List Length (from LLR Register) is reached and the execution of the list restarts at BTR + CTR time. At this point the BTR is incremented by value in CTR to mark the beginning of a new cycle. In the absence of any gate controls, all the gates are deemed to be in Open state, during the execution of the list.

In cases where the execution time of the list is greater than the CycleTime, the list is truncated and the next iteration starts when the current time equals BTR + CTR.

Table 379. GCL and Associated Registers - BTR and CTR

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
14200	OOCCC	1000	14200
15200	OOOOC	1500	14200
15700	CCCCO	3700	14200
17900	CCCCC	4000	14200
18200	OOOOO	0	20200
20200	OOCCC	1000	20200
21200	OOOOC	1500	20200

In the example in table, the execution starts at 14200 and the first set of Gate Controls "OOCCC" are applied immediately. As the time interval is 1000ns the next set of gate controls are applied at 14200 (BTR) + 1000 (Accumulator) = 15200 ns . As there are no

gate controls available after the execution of the last gate control and before the next iteration of the loop, the gates are deemed to be in open state during that time period as depicted at time 18200 in the table.

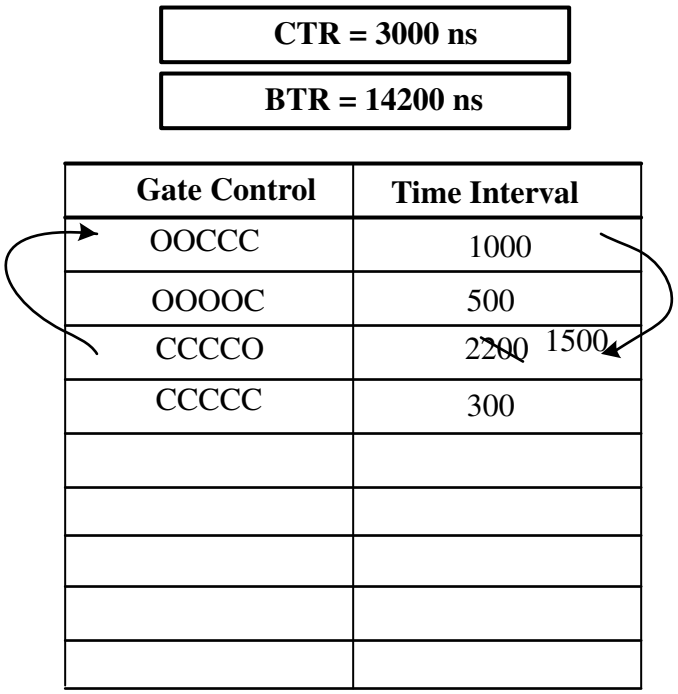


Figure 186. GCL and Associated Registers - BaseTime and CycleTime List Execution Time > CycleTime

As the list execution takes longer than the allocated CycleTime, the list is truncated and the list is started from the BTR+CTR as shown in the following table.

Table 380. GCL and Associated Registers - BTR and CTR, Execution Time > Cycle Time

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
14200	OOCCC	1000	14200
15200	OOOOC	1500	14200
15700	CCCCO	3700	14200
17200	OOCCC	1000	17200
18200	OOOOC	1500	17200
18700	CCCCO	3700	17200

While applying the third set of gate controls, the BTR + CycleTime (17200) < BTR + Accumulator (17900), so the list is truncated and execution switches to a new iteration at 17200.

43.4.6.8.1.5 Installing a New GCL

When a new software programmed GCL is available and must be executed at the new BTR value, the switch to the new GCL can happen in one of the following ways.

- New Base Time aligned with current schedule
- New Base Time unaligned with current schedule

43.4.6.8.1.5.1 New Base Time Aligned with Current Schedule

If the choice of cycle time for the new gating cycle is unchanged from the cycle time for the current gating cycle, and if the BTR chosen for the new gating cycle (new BTR) is an integer multiple of the current cycle time (+ current BTR), then the new gating cycle exactly lines up with the old gating cycle, that is, the cycle start times for the new gating cycle is same as they would have been for the old configuration. This could be considered to be the ideal case and allows the new gating cycle to be installed and executed with no timing issues. The implementation completes the execution of an iteration of the current list and switches to the new list at the beginning of the BaseTime listed in the new list.

If (New BaseTime > = Current Time)

ConfigChangeTime = New BaseTime

Else If (New BaseTime < Current Time)

1. Set the BTRError

2. ConfigChangeTime = (New BaseTime + N* New CycleTime)

where N is the smallest integer for which the relation ConfigChangeTime >= CurrentTime and (N =< 8) is TRUE.

When N > 8 the hardware cannot auto recover and the loop count value in BTRError reporting is set to 1111 requiring the software to reprogram the New Base Time.

Following figure illustrates the installation of the new GCL along with the timelines.

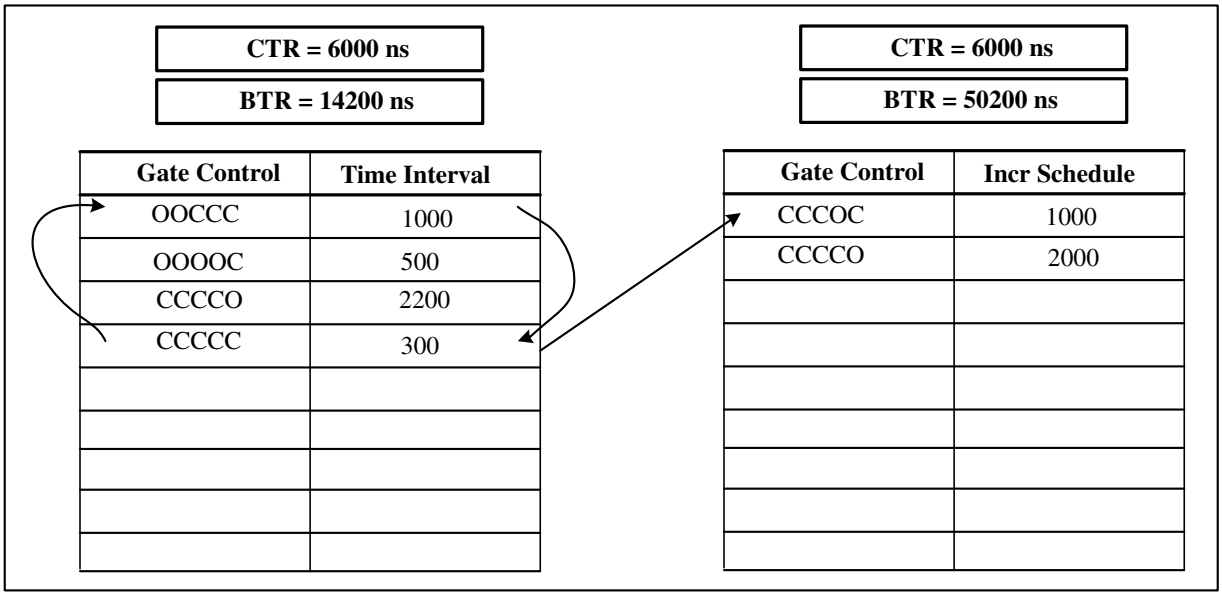


Figure 187. Switching to a New Configuration that is Aligned with the Existing Configuration

In the above example, after the sixth iteration of the first GCL, the BTR values of the old and new GCL are equal. At that point the new GCL is processed as a natural extension to the existing GCL.

Table 381. GCL and Associated Registers - BTR and CTR

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
44200	OOCCC	1000	44200
45200	OOOOC	1500	44200
45700	CCCCO	3700	44200
47900	CCCCC	4000	44200
48200	OOOOO	0	50200
50200	CCCOC	1000	50200
51200	CCCCO	3000	50200
53200	OOOOO	0	56200

43.4.6.8.1.5.2 New Base Time Unaligned with Current Schedule

If new CycleTime differs from current CycleTime or New BaseTime is in the future and is not an integer multiple of current CycleTime, then the old and new cycles do not line up; when new BaseTime is reached (when the new configuration is installed and starts to execute), the last old cycle is normally truncated to start the first new cycle. This could be undesirable if it results in a very short last old cycle; arguably it would be better to simply extend the penultimate old cycle by that small amount, rather than starting a very short cycle. The Cycle Time Extension Register (related to the current config list) allows this extension of the last old cycle to be done in a defined way; if the last complete old cycle ends normally in less than current Cycle Time Extension (TER) ns before the new base time, then the last complete cycle before new BaseTime is reached is extended so that it ends at new BaseTime.

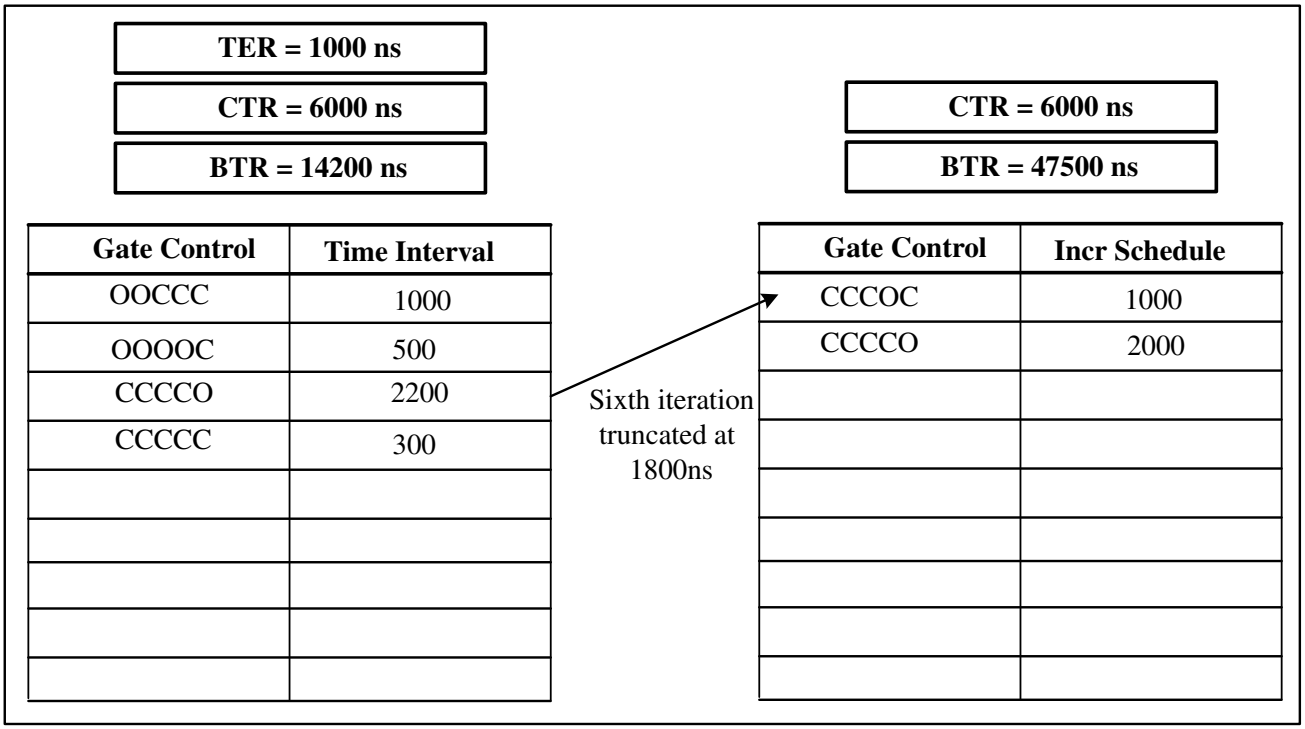


Figure 188. Switching to New List Truncating the Current List

At the end of the fifth iteration the Current time + Cycle time extension (TER) < New BTR so the sixth iteration of current configuration is started. During the sixth iteration of the current list when the new BTR value is smaller than the next schedule in the current list, it switches to the new list.

Table 382. Extending to New List By Truncating the Current List

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
44200	OOCCC	1000	44200
45200	OOOOC	1500	44200
45700	CCCCO	3700	44200
47500	CCCOC	4000	44200
48500	CCCCO	0	50200
50500	OOOOO	1000	50200

Below is an example where the current config list is extended instead of starting a new iteration as the extension time of 800ns is less than the allowed cycle extension time (TER) of 1000ns.

Figure 189. Switching to New List By Extending the Current List

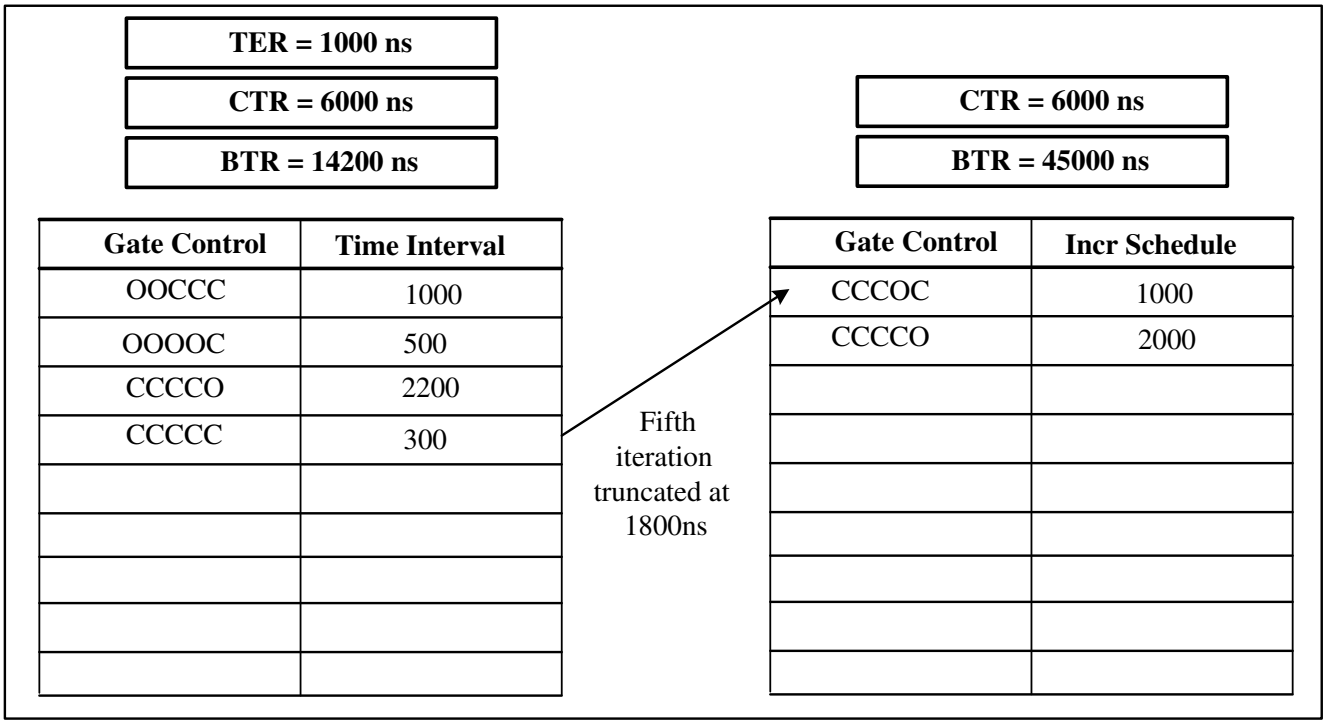


Table 383. Switching to New List By Extending the Current List

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
38200	OOCCC	1000	38200
39200	OOOOC	1500	38200
39700	CCCCO	3700	38200
41900	CCCCC	4000	38200
42200	OOOOO	0	44200
45000	CCCOC	1000	45000
46000	CCCCO	3000	45000
48000	OOOOO	0	51000

43.4.6.8.1.6 Impact of Transmit Scheduling Algorithms on EST

When EST is used in isolation, the Gate Control List manages the final open/close state of the Queues along with the checks carried out by the Transmission Selection Algorithm in MTL. As the Gate Controls operate on a predefined repetitive schedule, it is recommended to use Strict Priority or Credit Based Shaper (CBS) scheduling algorithms.

Other algorithms such as the Weighted Round Robin (WRR), Deficit Weighted Round Robin (DWRR) and Weighted Fair Queuing (WFQ) implement masking of the queues based on the current winning queue. The algorithm is applied only among the group of queues that open simultaneously. To ensure Queues whose gates are "Open" get priority, these algorithms are modified to treat "gate open" queues and "gate closed" queues as separate groups giving priority to gate open queues.

For example, consider 4 queues (Q3, Q2, Q1, Q0) with weights 4:3:2:1; Q3 and Q2 are in Open state in slot one slot, while Q1 and Q0 are in Open state in another slot. In this case, the scheduler works as follows:

1. In the first slot, the Q3 and Q2 are serviced in the ratio of 4:3 for the duration the slot is open.
2. In the second slot, the queues Q1 and Q0 are serviced in the ratio of 2:1.
3. Fresh arbitration is started every time a slot is opened.

In other words, the traffic does not get distributed in the intended ratio of 4:3:2:1; but as two groups with different ratios and only for the duration of the slot when the gates are open continuously

NOTE

See Programming Guidelines for EST

43.4.6.8.2 Frame Preemption (FPE)

Frame preemption breaks the interfering frame into smaller fragments. Therefore, the guard band needs to be only as large as the largest possible interfering fragment instead of the largest possible interfering frame. During the guard band, only the frames that can complete the transmission of the entire frame before the next gate close event are permitted. This ensures that the high priority traffic can always start at the beginning of the window reserved for it.

Preemption allows one or more higher priority (express) frames to interrupt the transmission of a lower priority (preemptable) frame; the preemptable frame transmission is resumed and completed after the express frame transmission is complete. To support frame preemption, the following two abstractions of the MAC are used:

- A preemptable MAC, called pMAC, which carries the preemptable traffic.
- An express MAC, called eMAC, which carries the express traffic.

In the implementation, only the parts of the MAC that hold the state during preemption are replicated and represented as pMAC and eMAC. The MAC uses the following two ways to put the transmission of the preemtable traffic on hold, in the presence of express traffic:

- The MTL scheduler interrupts the preemptable traffic that is currently being transmitted. When the preemption capability is active, the MAC interrupts the transmission and reception of preemptable frames. A preempted fragment can be followed by zero or more express frames, before the continuation fragments. The preemptable frame can be fragmented any number of times, however, the minimum final and non-final fragment length criterion must be met. However, interleaving of more than 1 preemptable packet is not permitted. This implies that if a preemptable packet is fragmented by an express packet, another preemptable packet cannot be transferred until all the remaining fragments of the first preempted packet are transferred.
- The MTL scheduler prevents starting the transmission of preemptable traffic. When the preemption capability is inactive, the pMAC entity is disabled and only express traffic is transmitted or received. Frame preemption feature can be enabled by setting EFPE field of the MAC_FPE_CTRL_STS register.

43.4.6.8.2.1 GCL Modification to Support FPE

In the EST-only configuration, the GCL entry has up to 24 bits of Time Interval and up to 8 high order bits representing the Gate Open/Close state requirements as shown in the following figure.

Gate Control (up to 5 bits)	Time Interval(ns) 16, 20 , or 24 bits
OOCCC	T0
OOOOC	T1
CCCCO	T2
CCCCC	T3
|	|
|	|
OCOCO	T126
OOOOC	T127

Figure 190. Gate Control List When FPE is Disabled

EST only supports SetGate operation, which implies that the gates are set to either open or close at a given time interval. However, when both Frame Preemption (FPE) and EST are enabled, the GCL also supports Set-and-Hold-MAC and Set-and-Release-MAC operations, in addition to the SetGate operation. To enable the support of hold and release operations, the format of the GCL is slightly changed while configuring and enabling the FPE. The first Queue (Q0) is always programmed to carry preemption traffic and therefore it is always Open. The bit corresponding to Q0 is renamed as Release/Hold MAC control. The TX Queues whose packets are preemptable are indicated by setting the PEC field of the MTL_FPE_CTRL_STS register. The GCL bit of the corresponding Queue are ignored and considered as always "Open". So, even if the software writes a "0" ("C"), it is ignored for such queues.

Gate Control (up to 4 bits)	Release/Hold Indication	Time Interval(ns) 16, 20, or 24 bits
CCCO	0	T0
CCCC	0	T1
OCCC	1	T2
COCC	1	T3
CCOC	1	T4
CCCO	0	T5
CCCC	0	T6

Figure 191. Gate Control List When FPE is Enabled

When the Release/Hold bit transitions from a '0' to '1', a Set-and-Hold-MAC operation is performed. This marks the cease of the preemptable traffic. This is achieved by sending a Hold request to MTL "ha" ns in advance (where ha is the time interval mentioned in the Hold Advance (HADV) field of the MTL_FPE_Advance register). When the Release/Hold bit transitions from a '1' to '0' a Set-and-Release-MAC operation is performed. This marks the resuming of the preemptable traffic. This is achieved by sending a Release request to MTL "ra" ns in advance (where ra is the time interval mentioned in the Release Advance (RADV) field of the MTL_FPE_Advance register). The preemptable traffic is blocked for the time duration the Release/Hold bit is set to a '1' in the Gate Control List.

43.4.6.8.2.2 Impact of Preemption on CBS

In Credit Based Shaper (CBS), the definition of "Transmit" is:

- TRUE for the duration of frame transmission from the queue.
- FALSE when frame transmission from the queue is complete.
- When CBS algorithm is used with frame preemption, the value of "Transmit" is TRUE only while the frame is being transmitted by the MAC. If the frame transmission is delayed or interrupted (for instance, a preemptable frame transmission is interrupted to allow the transmission of an express frame from a different queue, or the start of an express frame is delayed because a preemptable frame is being transmitted) the value of "Transmit" is FALSE until transmission of the frame commences or is resumed.

Also, the value of "Transmit" is FALSE during the transmission of any overhead that is a consequence of frame preemption, (for example, when additional frame overhead, such as mCRC or Fragment Count, is added to the preemptable frame.

At any given time, if there are no frames in the queue, and the value of Transmit is FALSE, and credit is positive value, the credit value is set to zero if there is no preemptable frame from the queue for which transmission is in progress but has been interrupted.

43.4.6.8.2.3 mPacket Format

When the preemption capability is active, MAC sends mPackets to the PHY. An mPacket can be one of the following:

1. A express packet
2. A preemptable packet
3. An initial fragment of a preemptable packet
4. A continuation fragment of a preemptable packet

Start mPacket Delimiter (SMD)

The value of the SMD indicates whether the mPacket contains an express packet, the start of a preemptable packet (initial fragment or complete packet), or any of continuation fragments of a preemptable packet. Following table shows the valid SMD values.

Table 384. Possible SMD Values of mPacket

mPacket Type	Notation	Frame Count	Value
verify packet	SMD-V	-	0x07
respond packet	SMD-R	-	0x19
express packet	SMD-E	-	0xD5
preemptable packet start	SMD-S0	0	0xE6
	SMD-S1	1	0x4C

Table continues on the next page...

Table 384. Possible SMD Values of mPacket (continued)

mPacket Type	Notation	Frame Count	Value
	SMD-S2	2	0x7F
	SMD-S3	3	0xB3
continuation fragment	SMD-C0	0	0x61
	SMD-C1	1	0x52
	SMD-C2	2	0x9E
	SMD-C3	3	0x2A

frag_count

A frag_count is a modulo-4 counter that increments for each continuation fragment of the preemptable packet. The frag_count protects against mPacket reassembly errors by enabling detection of the loss of up to 3 packet fragments.

The frag_count field is present only in mPackets with SMD-C notation (continuation fragment). The frag_count is zero in the first continuation fragment of each preemptable packet.

Table 385. Possible frag_count Values

frag_count	Value
0	0xE6
1	0x4C
2	0x7F
3	0xB3

mData

The contents of the packet from the MAC, starting with the first byte after the SFD to the last byte before the FCS are sent in the mData fields of one or more mPackets for that frame. The minimum size of the mData field is 60 bytes.

CRC The CRC field contains a cyclic redundancy check (CRC) and has an indication of the final mPacket of a frame. In the final mPacket of a frame, the CRC field contains the last 4 octets of the MAC frame (the FCS field).

For other mPackets, the CRC field contains an mCRC value. The mCRC is calculated on the octets of the packet from the first octet of the frame (the octet following the SFD of preemption frames) to the last octet of the packet transmitted in that mPacket by performing an XOR of the calculated 32 bit CRC value of the fragment and a value of 0x0000FFFF.

Summary of Packet Formats

- Express Packet: 7bytes of PREAMBLE, SMD-E, Data, and CRC
- Complete Preemptable Packet: 7 bytes of PREAMBLE, <Current Preemptable packet SMD>, Data, CRC
- Initial Fragment (non-final) of Preemptable Packet: 7 bytes of PREAMBLE, <Current Preemptable packet SMD>, Data, mCRC
- Continuation fragments (non-final) of Preemptable packet: 6 bytes of PREAMBLE, <Current Preemptable continuation fragment SMD>, <Current Preemptable continuation fragment FC>, Data, mCRC

- Final fragment of Preemptable packet: 6 bytes of PREAMBLE, <Current Preemptable continuation fragment SMD>, <Current Preemptable continuation fragment FC>, Data, CRC

Table 386. Current and Previous SMD Values

Previous Preemptable packet SMD	Current Preemptable packet SMD
SMD-S0	SMD-S1
SMD-S1	SMD-S2
SMD-S2	SMD-S3
SMD-S3	SMD-S0

Table 387. Current and Previous SMD Values

Previous Preemptable fragment SMD	Previous Preemptable fragment FC	Current Preemptable continuation fragment SMD	Current Preemptable continuation fragment FC
SMD-S0	NA	SMD-C0	FC0
SMD-S1	NA	SMD-C1	FC0
SMD-S2	NA	SMD-C2	FC0
SMD-S3	NA	SMD-C3	FC0
SMD-C0	FC0	SMD-C0	FC1
SMD-C0	FC1	SMD-C0	FC2
SMD-C0	FC2	SMD-C0	FC3
SMD-C0	FC3	SMD-C0	FC0
SMD-C1	FC0	SMD-C1	FC1
SMD-C1	FC1	SMD-C1	FC2
SMD-C1	FC2	SMD-C1	FC3
SMD-C1	FC3	SMD-C1	FC0
SMD-C2	FC0	SMD-C2	FC1
SMD-C2	FC1	SMD-C2	FC2
SMD-C2	FC2	SMD-C2	FC3
SMD-C2	FC3	SMD-C2	FC0
SMD-C3	FC0	SMD-C3	FC1

Table continues on the next page...

Table 387. Current and Previous SMD Values (continued)

Previous Preemptable fragment SMD	Previous Preemptable fragment FC	Current Preemptable continuation fragment SMD	Current Preemptable continuation fragment FC
SMD-C3	FC1	SMD-C3	FC2
SMD-C3	FC2	SMD-C3	FC3
SMD-C3	FC3	SMD-C3	FC0

43.4.6.8.2.4 Transmit Preemption

When FPE is enabled (setting EFPE=1 in MAC_FPE_CTRL_STS register), the MTL preempts a preemptable frame, when a "hold" request is asserted (EST/Qbv configured and enabled) express frames are available for transmission, that is, frame is present in MTL FIFO and is qualified for arbitration, after ensuring that the minimum mPacket mData field size is met. Therefore, preemption occurs only if at least 60 bytes of the preemptable frame have been transmitted and at least 64 bytes (including the frame CRC) remain to be transmitted.

The earliest starting position of preemption is controlled by the AFSZ field of the MTL_ FPE_CTRL_STS Register. Preemption does not occur until at least 64 * (1+ AFSZ) - 4 bytes of the preemptable frame have been sent.

When preemption occurs, all the preemptable queues are blocked and only the express queues are allowed to arbitrate (if more than one express queue has traffic) and transmit.

Continuation fragment of the preempted frame is the first frame to be transmitted after "release" request is asserted (EST/Qbv configured and enabled) and all the express traffic transmission completes.

NOTE

PTP packets should be transmitted as express packets.

MTL communicates the following frame-type information to the MAC using a 2-bit preemption control signal on the MTI interface qualified by SoF and EoF.

- Express Frame
- Preemption Frame (Full or Fragment)
- Continuation Fragment (non-Final or Final)

Table 388. Preemption Control Values on MTI Interface for Various Frame Types

Qualifier	Preemption Control Value	Frame Type
SoF	00	Start Express
SoF	01	Start Preemption
SoF	10	Continuation Fragment
SoF	11	Reserved
EoF	00	End of Frame
EoF	01	End of Fragment

MTL should wait for the previous fragment status to be received before resuming the continuation fragments of a preempted frame. Fragment status is described in detail in the Tx Fragment Status section.

43.4.6.8.2.4.1 MAC Tx Preemption

MAC supports preemption by implementing the functionality needed to generate the mPackets as described in “mPacket Format”.

Based on the Preemption Control value received on the MTI interface (qualified with SoF and EoF), MAC determines the frame type (shown in the above table 'Preemption Control Values on MTI Interface for Various Frame Types') and generates mPackets accordingly.

When the preemption capability is active, MAC replaces the SFD of a preemption packet with an SMD-S value. A 2-bit rolling frame count is encoded in the SMD-S value.

The SMD-E value is the same as the SFD value so the SFD of an express packet does not need to be replaced.

43.4.6.8.2.4.1.1 Tx Fragment Status

MAC sends Tx fragment status to indicate successful transmission of fragmented mPackets.

In case of a transmission error (underflow, jabber, and so on) the frame status is sent (and not a fragment status) with an error indication along with all other relevant status fields. In case of receiving an error status for a transmitted fragment, the MTL drops the remaining fragments and does not send any more continuation fragments.

43.4.6.8.2.5 Receive Preemption

43.4.6.8.2.5.1 MAC Receive Preemption

When FPE is enabled, the MAC Receiver pases the incoming packets and differentiates between Express packets and preemptable packets. An SMD containing an SMD-E indicates express packet, and SMD containing an SMD-S indicates the first mPacket of a preemptable packet.

If an mPacket containing an SMD-S is received when MAC has not completed receiving the previous preempted packet, MAC sets a CRC Error status for the previously received partial packet.

When an SMD-S is detected, MAC records the frame count indicated by the SMD and then begins sending data on the MRI interface.

The MAC checks the last four bytes of the mPacket . If the last four bytes of the mPacket do not match CRC, that indicates the end of the packet with or without a CRC error as per the CRC check result. If the last four octets of the mPacket match, that indicates that the packet was preempted.

An SMD containing an SMD-C indicates an mPacket that continues the data for a preempted packet. Upon receiving an SMD value of SMD-C, MAC checks the following:

1. A preempted packet is in progress
2. The frame count indicated by the SMD matches the frame count of the packet in progress
3. The frag_count value indicates the next fragment count.

If any of the above check fails, the mPacket is discarded and MAC sets a CRC Error status for the partially received packet.

If all the checks pass, the next fragment count is incremented modulo 4.

When a packet is preempted, the MAC saves the state of the partially received packet (filter check status, timestamp, length fields etc.) and will be able to process any received Express packets before the continuation fragment is received.

The MAC Receiver sends a "dummy status" for all the mPacket fragments successfully received and sends the Rx status with the final fragment. If an error is detected during any of the fragments the Rx status is sent and the fragment is marked as final fragment. All subsequent continuation fragments received for this packet are dropped in the MAC.

The MAC communicates the following frame type information to the MTL using a 2-bit preemption control signal.

1. Express Frame
2. Preemption Frame (Full or Fragment)
3. Continuation Fragment (non-Final or Final)

Table 389. Preemption Control Values on MRI Interface for Various Frame Types

Qualifier	Preemption Control Value	Frame Type
SoF	00	Start Express
SoF	01	Start Preemption
SoF	10	Continuation Fragment
SoF	11	Reserved
EoF	00	End of Frame
EoF	01	End of Fragment

43.4.6.8.2.5.1.1 Data Alignment

When a received frame cannot be fragmented on any byte boundary, MAC retains the unaligned bytes of data in the previous fragment and resends them with the next fragment as shown in the following figure.

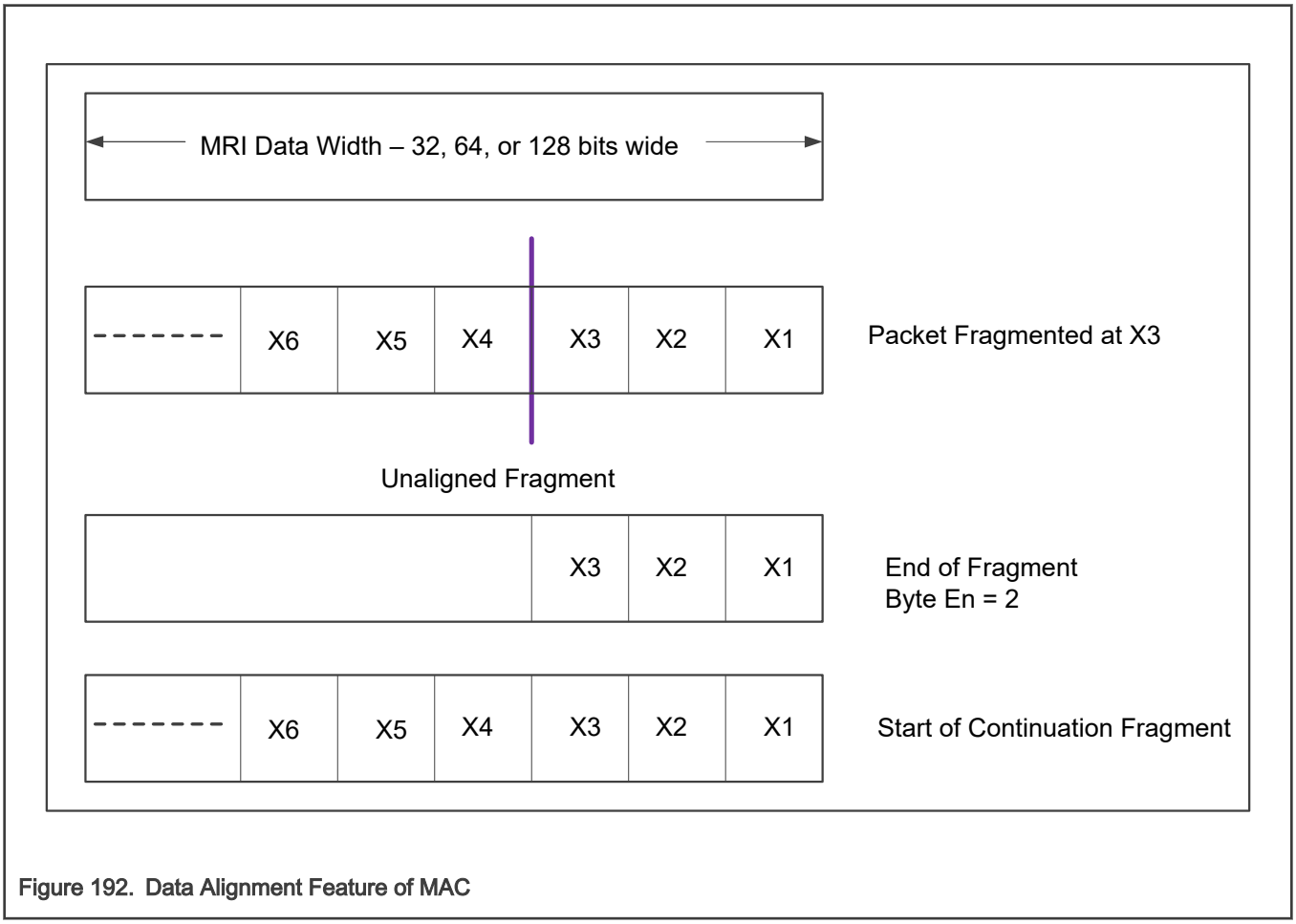


Figure 192. Data Alignment Feature of MAC

43.4.6.8.2.5.1.2 MTL Receive Preemption

The MTL Rx must have at least 2 Rx queues to support FPE function as the preemptable packets and the express packets must be routed to separate Rx queues. The destination Rx queue of a received packet is controlled by MAC_RxQ_Ctrl[n] registers.

Program these registers such that Preemptable traffic and express traffic are not routed to the same Rx queue. As the queue mapping for tagged packets is based on VLAN user-priority field, this implies that priority of preemptable and express packets are mutually exclusive. In other words, packets of a certain priority (traffic class) are either express or preemptable but cannot be both.

For the preemptable frames, in addition to the PSRQ (Priority Selected in Rx queue) based queue routing, a programmable "Frame Preemption Residue Queue" (FPRQ) is supported to route all other Preemption Packets received (Untagged, SA/ DA or VLAN Filter Fails forwarded due to RA being set or VTFE being reset).

Table 390. Preemption Control Values on MRI Interface for Various Frame Types

Preemption Packet Type	Queue Routing
AV Tagged Packets passing the SA/DA and VLAN filters	PSRQ*
DCB/Generic Tagged Packets passing the SA/DA and VLAN filters	PSRQ**
Tagged Packets failing the SA/DA and VLAN filters without setting the Receive All (RA = 0) or setting VTFE = 1	Dropped
All other packets OR when RA = 1	FPRQ

43.4.6.8.2.5.1.3 MTL Receive Arbitration

On the ARI Interface, data is fetched from the MTL Rx FIFO based on the arbitration selected in the MTL_Operation_Mode register. Frame based arbitration can be used only when all the MTL Preemption Queues are operating in Store and Forward mode, else there will be loss of bandwidth on the ARI interface because all the fragments of a preemptable packet is not available. Therefore, any express packets received between the fragments are blocked until all the fragments are received and transferred; this defeats the purpose of express traffic.

When operating in either Threshold (cut through) or Store and forward modes of operation, PBL based arbitration is recommended over Frame based arbitration. In case of PBL based arbitration, the watermark check is always performed and the arbitration/ transfer of data in terms of chunks of "PBL" size of data which is almost similar to the concepts of "fragments". Therefore the express queue packets get blocked for less time as well as the ARI interface transfers the data without loss of efficiency.

43.4.6.8.2.6 Verify and Respond mPackets

When FPE function is present, the MAC can receive and detect the Verify and Respond mPackets, even when FPE is not enabled by software. When MAC detects valid Verify/Respond mPackets, it notifies the software by setting the RVER and RRSP fields of MAC_FPE_CTRL_STS register respectively. Optionally an interrupt can be generated. As such packets have empty (all-zero) data payload, they are dropped inside the MAC and not forwarded to the MRI.

Software can set the SVER and SRSP fields of MAC_FPE_CTRL_STS register to request MAC to transmit Verify and Respond mPackets respectively. Upon successful transmission of these frames, MAC clears the SVER/SRSP bits and sets the TVER & TRSP fields of MAC_FPE_CTRL_STS register. Optionally an interrupt can be generated when these events occur.

43.4.6.8.2.7 Frame Preemption and MMC Counter and Interrupt Registers

The following MMC counters and associated Interrupt registers are instantiated/present in the MAC when Frame Preemption feature is enabled.

Table 391. MMC Counters and Associated Interrupt Registers

Frame Assembly Error Counter	Description	Associated MMC Counter
Frame Assembly Error Counter	A 32-bit counter that provides the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value.	MMC_Rx_Packet_Assembly_Err_Cntr
Frame SMD Error Counter	A 32-bit counter that provides the number of received MAC frames rejected due to arriving with an SMD-C when there was no preceding preempted frame.	MMC_Rx_Packet_SMD_Err_Cntr
Frame Assembly OK Counter	A 32-bit counter that provides the number of MAC frames that were successfully reassembled.	MMC_Rx_Packet_Assembly_Ok_Cntr
MAC Rx Fragment Counter	A 32-bit counter that provides the number of additional mPackets received due to preemption.	MMC_Rx_FPE_Fragment_Cntr
MAC Tx Fragment Counter	A 32-bit counter that provides the number of additional mPackets transmitted due to preemption.	MMC_Tx_FPE_Fragment_Cntr
Hold Request Counter	A 32-bit counter that maintains the count of number of times a hold request is given to MAC.	MMC_Tx_Hold_Req_Cntr

43.4.6.8.2.7.1 Additional Registers Associated With MMC Interrupts

Following are the additional registers associated with MMC interrupts for the MMC error counters:

- MMC_FPE_Tx_Interrupt
- MMC_FPE_Tx_Interrupt_Mask
- MMC_Rx_Interrupt
- MMC_FPE_Rx_Interrupt_Mask

43.4.6.8.3 Time-Based Scheduling

Time-based scheduling feature is suitable for traffic whose periodicity and rate are predictable. To improve the quality-of-service of such traffic,

- The transmit DMA fetches the packet from the host memory for transmission at designated time. This helps the software to setup the Transmit descriptors in advance even before packet is ready/available. It reduces the overhead on the software and avoids constant monitoring of the time and preparing descriptors just in time when the packet is targeted to be transmitted.
- The MAC transmits the packet only at the designated/pre-determined time even if the packets are fetched in advance. This helps in maintaining a constant transmission rate that can be consumed by the receiver station; therefore avoiding congestion and excessive buffering in the network.

The time-based scheduling feature supports fetching and launching an Ethernet packet at (or after) a pre-determined time. The time-based scheduling is supported only in the following modes/configurations:

- Full duplex mode
- Link speed is 100Mbps or higher

The following figure shows the time-based scheduling flow.

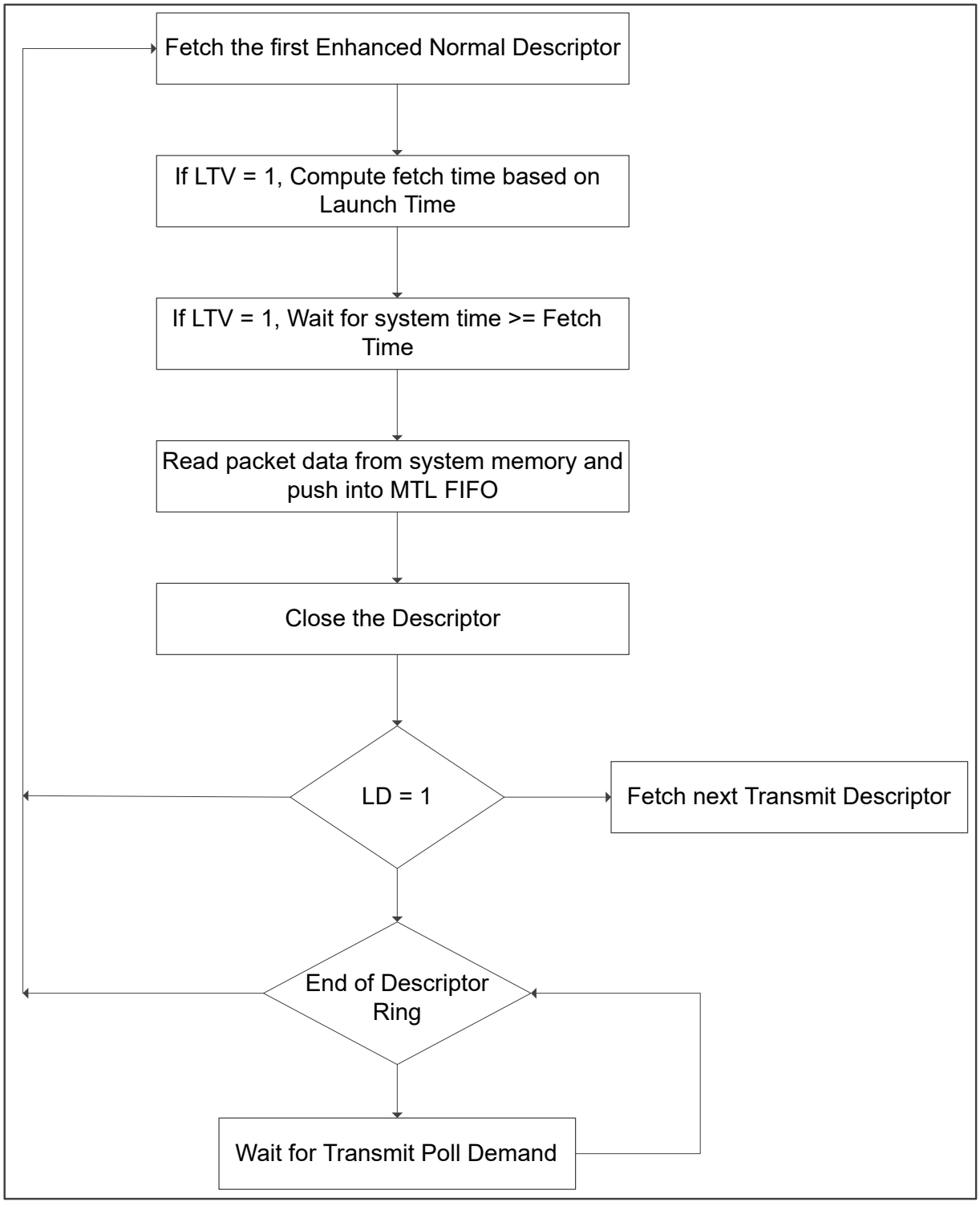


Figure 193. Time-Based Scheduling Flow

Following are the definitions specific to the time-based scheduling feature:

43.4.6.8.3.1 Launch Time

Launch time is the time beyond which the MTL can schedule the packet for transmission. The launch time is reset after the transmission of one frame. Following are the two formats of the launch time.

Normal/Absolute Format:

In this format, the launch time is interpreted in the Normal/Absolute format if the ESTM bit of the MTL_TBS_CTRL register is set to 0.

The launch time is a 32-bit value, where most-significant 8-bits represent the time in seconds and the rest 24 bits represent the time in 256 ns. The launch time is compared against the IEEE 1588 based System/PTP Time (bits[39:8]) and rolls over after 256 seconds.

The maximum value of the lower 32-bits of System Time is 999,999,999 decimal (0x3B9AC9FF) and it wraps to 0 when reaching this value (representing a full second). Therefore, the maximum value of the lower 24 bits of the Launch Time (after multiplying by 256) must be 0x3B9AC9.

As the maximum value of Launch time is 256 seconds,

- Launch Time is greater than current System Time when its value is between System Time[39:8] and System Time[39:8] + 128 sec
- Launch Time is less than current System Time when its value is between "System Time[39:8] + 128sec" and "System Time[39:8] + 256 sec", because this is a modulo 256 computation.

EST/Offset Format

In this format, the Launch time is an offset value relative to the time indicated by the Base Time Register (BTR) of the GCL list provided in the GCL Slot Number (GSN). The value in the BTR is always updated to the start time of the current loop of the GCL.

For each packet, the GSN value and the Launch Time value are specified in the Enhanced Transmit Descriptor in the DMA configurations, or as Control Word in the MTL configurations.

The launch time offset is a 32-bit value; the upper 8-bits represent the time in seconds and the rest 24 bits represent the time in 256 ns, which is added to the BTR value corresponding to GCL Slot Number. The value of the Launch Time Offset must be smaller than the value of the Cycle Time (specified in the CTR register that is implemented when EST is enabled). If the CTR is greater than or equal to 1s, the maximum value of the lower-24 bits of Launch Time offset should be 999,999,999 decimal (0x3B9AC9FF).

In this format, the launch time is a 64-bit value, which is interpreted as, Launch Time = Launch GSN BTR[63:0] + (Launch Time Offset[31:0] << 8), which is compared with the System Time [63:0].

GSN BTR is the Base Time value at which the Launch GSN loop started execution.

GCL Slot Number (GSN) A modulo 16 count of the GCL loop count is implemented and known as GSN. The count is incremented for every new GCL loop; installation of new GCL list does not impact the count. The current GCL slot number can be obtained by reading the CGSN field of the MTL_EST_Status register.

The maximum value of GSN is 15. Therefore GSN values between Current GSN (CGSN field of MTL_EST_Status register) and CGSN+8 represent current or future slots; all other GSN values are interpreted as elapsed slots or past slots. So, for the correct interpretation of time, GSN value must be between CGSN and CGSN+8, for correct interpretation of time.

43.4.6.8.3.2 Launch Expiry Time

Normal/Absolute Mode

In the normal mode, when the LEOV (Launch Expiry Offset Valid bit of MTL_TBS_CTRL register) is set, the Launch Expiry Offset (LEOS field of MTL_TBS_CTRL register) determines the maximum amount of time a frame is eligible for launch, starting from the time the frame becomes eligible for launch.

The Launch Expiry Offset is a 24-bit value defined in 256ns units, with a maximum possible value of 999,999,999 ns (0x3B9AC9FF).

Launch Expiry Time = (Launch Time[39:8] + LEOS[32:8]) * 256ns

The packet with a specific Launch time is considered eligible for transmission when the Launch time is less than the System Time and (if LEOV is set) the System Time is less than the Launch Expiry time.

When the System time is greater than the Launch Expiry time the frame is categorized as expired and will be dropped from the MTL FIFO.

NOTE

- For correct interpretation and meaningful operation, the Fetch, Launch, and Launch Expiry time should never be set to a value larger than Current System Time + 128 sec; such a value is interpreted as time that has already elapsed.
- In full duplex mode, the frames dropped from the MTL FIFO have Error Summary (Bit 15) and Excessive Deferral (bit 3) of TxStatus set.

EST/Offset Mode

In the EST mode of operation, when the LEOV field of the MTL_TBS_CTRL register is set, the launch expiry GSN offset (LEGOS field of the MTL_TBS_CTRL register) and Launch Expiry Offset (LEOS field of MTL_TBS_CTRL register) determine the maximum amount of time a frame remains eligible for launch, starting from the time the frame becomes eligible for launch.

Launch Expiry Offset = (LEGOS: LEOS)

LEGOS holds the GSN offset (multiples of CTR time) and LEOS holds maximum value of CTR (sub CTR values) in ns.

Launch Expiry offset is a 24-bit value defined in the units of 256 ns, with a maximum possible value of the smaller of 999,999,999ns or CTR-1 ns.

The Launch Expiry GSN is computed as follows:

Launch Expiry GSN = (Launch GSN + LEGOS + CCMA) Where, CCMA is the CTR Carry due to Modulo addition. This value is 1 if ((Launch Time Offset + LEOS) << 8) is equal to or greater than CTR.

When CCMA = 1, Launch Expiry Offset = (Launch Time Offset+ LEOS) - CTR

When CCMA = 0,

Launch Expiry Offset = (Launch Time Offset + LEOS).

Launch Expiry Time = Launch Expiry GSN BTR[63:0] + Launch Expiry Time Offset.

When LEOV is not set, Launch Expiry Time is not checked.

When LEOV is set, and

- the system time is greater than the Launch Expiry time, the frame is dropped from MTL FIFO. The frame is considered as expired.
- the launch time is smaller than the System Time and Launch Expiry Time is greater than system time, the frame is considered eligible for launch.

NOTE

- Max value of LEGOS is 7. This implies that when LEOV is set, the frame has a maximum life time of <8 GCL loop iterations after it becomes eligible for Launch.
- The slot number of the First GCL List executed each time after EST is enabled, is zero.

43.4.6.8.3.3 Fetch Time

The fetch time accounts for all possible delays in the DMA fetch operation and ensures that the frame is present in the MTL FIFO before the launch time.

If FTOV field of DMA_TBS_CTRL register is set, then fetch time can be indicated for each packet. If the FTOV filed is not set, the Fetch Time Offset is not valid and the DMA fetches packets without any time constraints.

Normal/Absolute Mode

In the Normal mode fetch time derived/calculated by reducing the time specified in the Fetch Time Offset (FTOS) field of DMA_TBS_CTRL register from the given Launch time.

Normal mode of operation, the Fetch Launch Time is computed as:

Fetch Time[39:8] = (Launch Time[39:8] - FTOS[31:8])

The Fetch time is 32-bits and is compared against System Time[39:8] to determine the Eligible for fetching the frame:

- The Fetch time is defined as "greater" than System Time if the Fetch Time is in the range of "System Time[39:8]" and "System Time[39:8] + 128 sec". The frame is considered as not-eligible for fetch.
- The Fetch time is defined as "smaller" than the System Time if the Fetch time is in the range of "System Time[39:8] + 128 sec" and "System Time[39:8] + 256 sec". The frame is considered as eligible for fetch.

This is a modulo 256 computation.

EST/Offset Mode

In the EST/Offset mode, the Fetch GSN Offset (FGOS field of MTL_TBS_CTRL register) provides the Slot Number offset to be deducted from the Launch GSN. In this case the FTOS value should be the smaller of 999,999,999ns or CTR-1 ns.

If (Launch Time Offset >= FTOS):

Fetch Time Offset = ((Launch Time Offset - FTOS) * 256ns)

CBFS(CTR Borrow for Fetch Subtraction) = 0

If (Launch Time Offset < FTOS) Fetch Time Offset = CTR + ((Launch Time Offset - FTOS) * 256ns)

CBFS (CTR Borrow for Fetch Subtraction) = 1

The Fetch GSN is computed as follows:

Fetch GSN = Launch GSN - FGOS - CBFS

Fetch Time = Fetch GSN BTR[63:0] + Fetch Time Offset

The frame is marked eligible for DMA fetch when the fetch time is smaller than the System Time.

DMA Operations Sequence

Following is the sequence of operation when FTOV = 1:

1. Fetch the first Enhanced Normal Descriptor (FD is set)
2. If LTV is set and Fetch enabled, compute the Fetch Time based on the Launch Time Wait for the System Time to be greater than Fetch Time
3. Read the Frame (Data) from the host memory and transfer to MTL FIFO
4. Close the Normal Descriptor
5. Fetch the next Normal Descriptor (if the previous Descriptor was not the Last)
6. Repeat steps 4 to 6, until the last descriptor of the frame (LD is set)

After the Last Descriptor of a frame, the next enhanced normal descriptor should be programmed with a new Launch Time and with LTV bit set. Otherwise, the subsequent frames will be processed without any time restrictions.

NOTE

See Programming the Launch Time in Time-Based Scheduling

43.4.6.8.4 Credit-Based Shaper Algorithm

The MTL Queue Scheduler (shown in Figure 173) uses the credit-based shaper algorithm to arbitrate the AV traffic in all queues and the legacy Ethernet traffic in Queue 0. You can program the additional queues to use the credit-based shaper algorithm.

The following sections provide information about how you can implement the credit-based shaper algorithm:

- Credit Value
- idleSlopeCredit and sendSlopeCredit Values
- Bandwidth Status

43.4.6.8.4.1 Credit Value

The credit value is part of the credit-based sharper algorithm used by the MTL Queue Scheduler.

The credit value is accumulated every transmit clock cycle, that is, 40 ns for 100 Mbps and 8 ns for 1000 Mbps. The credit to be added or subtracted per cycle can be fractional based on the required idleSlope and sendSlope values, as described in the table below.

Table 392. Credit Value Per Transmit Cycle Example

Mode	Values	Description
100 Mbps	• portTransmitRate = 100 Mbps • idleSlope = 70 Mbps (assuming 70% bandwidth reserved for a higher priority traffic class) • sendSlope = 30 Mbps	• credit = 2.8 bits accumulates per cycle (40 ns) for the higher priority traffic class when best-effort packet is being transmitted. • credit = 1.2 bits drains per cycle (40 ns) when higher priority traffic class packet is being transmitted.

The DMA stores the queue traffic in the respective part of the Tx FIFO based on the slot number in the transmit descriptor (if enabled) or the bandwidth availability on the AMBA AHB application bus.

The credit for a queue builds up only when the packet is available but it cannot be transmitted because the MAC is sending a packet from another queue. The module supports another mode in which the credit can build up in advance for a queue in which no packet is available in respective part of the FIFO. This enables sending a burst of high priority traffic in a queue as soon as the data is available. You can enable this mode with Bit 3 of the CBS Control registers of corresponding queues.

When reset, the accumulated credit parameter in the credit-based shaper algorithm is set to zero if there is positive credit, and there is no packet to transmit in a queue. The credit does not accumulate when there is no packet waiting in a queue and other queues are transmitting. When set, the accumulated credit parameter in the credit-based shaper algorithm is not reset to zero if there is positive credit and no packet to transmit in a queue. The credit accumulates even when there is no packet waiting in a queue and other queues are transmitting.

43.4.6.8.4.2 idleSlopeCredit and sendSlopeCredit Values

The idleSlopeCredit and sendSlopeCredit values are programmable.

The software must program the idleSlopeCredit and sendSlopeCredit values. The programmed values should be the credit accumulated or drained per clock cycle scaled by 1024, such as, 2.8 x 1024 = 2867 and 1.2 x 1024 = 1229. In addition, the software must program the hiCredit and loCredit values, scaled by 1024, to adjust for scaling of the idleSlopeCredit and sendSlopeCredit values. This means that if computed hiCredit and loCredit values are 12,000 bits and 3,036 bits respectively, the values to be programmed in the hiCredit and loCredit registers of the corresponding channel are 12000 x 1024 bits and two's complement of 3036 x 1024, respectively.

43.4.6.8.4.3 Bandwidth Status

The hardware maintains the status of the actual bandwidth consumed by each higher priority queue in the CBS status registers. This enables the software to estimate the average bandwidth consumed by numerically higher traffic classes as compared to the reserved bandwidth.

The CBS status register gives the average number of bits transmitted during the previous programmed slot interval (1, 2, 4, 8, or 16 slots of 125 µs) in a queue. The status register is updated even if the credit-based shaper algorithm is not enabled for a queue. The number of slots over which the average bits transmitted per slot are computed is programmed in Bits[6:4] of the CBS control register of the respective queue. For example, if you have programmed two slots, the average bits are computed over slot numbers 0–1, 2–3, 4–5, and so on.

The value programmed in the idleSlopeCredit register of a queue is proportional to the bandwidth reserved for the queue. The software can allocate any bandwidth that is not used by the higher priority queue to the reserved bandwidth of the lower priority queue.

A lower priority queue, which is using the credit-based shaper algorithm, cannot use the unused reserved bandwidth of any higher priority queue that is using the credit-based shaper algorithm. However, a lower priority queue, which is using the strict-priority algorithm, can use the unused reserved bandwidth of any higher priority queue that uses the credit-based shaper algorithm. For example, Queue 1 and Queue 2 use the credit-based shaper algorithm (with reserved bandwidth of 50% and 25%, respectively) and Queue 0 uses the strict-priority algorithm. If Queue 1 uses only 40% of reserved bandwidth, the remaining 10% is used by Queue 0. Queue 2 cannot exceed the reserved bandwidth of 25%.

43.4.6.9 Queue Modes

Both Tx and Rx queues can handle AV traffic.

Queueing is based on WRR (Weighted Round Robin) or WSP (Weighted Strict Priority) algorithm for generic traffic while it is based on CBS (Credit Base Shaper) and SP (Strict Priority) algorithm for AV traffic.

Whether an Rx Queue is configured for generic or AV-based routing depends on the RXQ#EN field of corresponding queue. Queueing is done based on VLAN tag priority. The VLAN tag priority should match the PSRQ field of the MAC_RXQ_CTRL2 registers. The Rx packets can also be routed to a particular DMA channel based on the DCS field of perfectly-matched MAC Address register.

By default, untagged packets in generic traffic are routed to Receive queue specified in UPQ field of MAC_RXQ_CTRL1 register. Queue 0 is the default value of UPQ field. You can override the default value with any other value, for the UPQ field.

AV control packets (tagged or untagged) are routed based on the AVCPQ field of the MAC_RXQ_CTRL1 register. The PTP over Ethernet packets are routed based on the AVPTPQ field of MAC_RXQ_CTRL1 register.

43.4.6.10 Queue Priorities

Priority of an Rx queue can be programmed in corresponding field of MAC_RxQ_Ctrl2 registers. AV queue (high priority) should be programmed as higher priority than the Best-effort queue (low priority).

43.4.7 TCP/IP Offloading Features

43.4.7.1 Transmit Checksum Offload Engine

The MAC has a Checksum Offload Engine (COE) to support checksum calculation and insertion in the Transmit path, using which, the software can offload the task of checksum insertion to the hardware. In the transmit path MAC calculates the checksum and inserts it in the Tx packet. This feature helps in reducing the load on the software and can improve the overall throughput of the system. This feature is supported for only Q0 queue.

The checksum offload engine module supports two types of checksum calculation and insertion. The checksum engine can be controlled for each packet by setting the CIC bits (TDES3 Bits[17:16]).

Note: The checksum for TCP, UDP, or ICMP is calculated over a complete packet, and then inserted into its corresponding header field. Because of this requirement, when this function is enabled, the Tx FIFO automatically operates in the store-and-forward mode even if IP is configured for Threshold (cut-through) mode.

43.4.7.1.1 IP Header Checksum Engine

In IPv4 datagrams, the integrity of the header fields is indicated by the 16-bit Header Checksum field (the eleventh and twelfth bytes of the IPv4 datagram). The COE detects an IPv4 datagram when the Type field of Ethernet packet has the value 0x0800 and the Version field of IP datagram has the value 0x4. The checksum field of the input packet is ignored during calculation and replaced with the calculated value.

NOTE

IPv6 headers do not have a checksum field. Therefore, the COE does not modify the IPv6 header fields.

The result of this IP header checksum calculation is indicated by the IP Header Error status bit in the Transmit status (TDES3[0]). This status bit is set whenever the values of the Ethernet Type field and the Version field of IP header are not consistent, or when the Ethernet packet does not have enough data, as indicated by the IP header Length field. In other words, this bit is set when an IP header error is asserted under the following circumstances:

- For IPv4 datagrams:
  - The received Ethernet type is 0x0800, but the Version field of IP header is not equal to 0x4.
  - The IPv4 Header Length field indicates a value less than 0x5 (20 bytes).
  - The total packet length is less than the value given in the IPv4 Header Length field.
- For IPv6 datagrams:
  - The Ethernet type is 0x86dd but the IP header Version field is not equal to 0x6.
  - The packet ends before the IPv6 header (40 bytes) or extension header (as given in the corresponding Header Length field in an extension header) is completely received.

43.4.7.1.2 TCP/UDP/ICMP Checksum Engine

The TCP/UDP/ICMP Checksum Engine processes the IPv4 or IPv6 header (including extension headers) and determines whether the encapsulated payload is TCP, UDP, or ICMP. The checksum is calculated for the TCP, UDP, or ICMP payload and inserted into its corresponding field in the header. The Tx COE can work in the following two modes:

- The TCP, UDP, or ICMPv6 pseudo-header is not included in the checksum calculation and is assumed to be present in the Checksum field of the input packet. This engine includes the Checksum field in the checksum calculation, and then replaces the Checksum field with the final calculated checksum.
- The engine ignores the Checksum field, includes the TCP, UDP, or ICMPv6 pseudo-header data into the checksum calculation, and overwrites the checksum field with the final calculated value.

NOTE

For ICMP-over-IPv4 packets, the Checksum field in the ICMP packet must always be 16'h0000 in both modes, because pseudo-headers are not defined for such packets. If it does not equal 16’h0000, an incorrect checksum may be inserted into the packet.

The result of this operation is indicated by the Payload Checksum Error status bit in the Transmit Status vector (TDES3[12]). This engine sets the Payload Checksum Error status bit when:

- It detects that the packet has been forwarded to the MAC Transmitter engine in the store-and-forward mode without the end of packet (EOP) being written to the FIFO.
- when the packet ends before the number of bytes indicated by the Payload Length field in the IP Header is received.

When the packet is longer than the indicated payload length, the COE ignores them as stuff bytes and no error is reported. When this engine detects the first type of error, it does not modify the TCP, UDP, or ICMP header. For the second error type, it still inserts the calculated checksum into the corresponding header field. The following table describes the functions supported by

Transmit Checksum Offload engine based on the packet type. When the MAC does not insert the checksum, it is indicated as "No" in the table.

NOTE

You should not enable checksum insertion for IPv4 or IPv6 packets that are greater than the frame size constraint specified in Description of Transmit Checksum Offload Engine because it may result in incorrect checksum insertion or unexpected behavior.

Table 393. Transmit Checksum Offload Engine Functions for Different Packet Types

Packet Type	Hardware IP headerchecksum insertion	Hardware TCP/ UDPchecksum insertion
Non-IPv4 or IPv6 packet	No	No
IPv4 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in MAC) but less than or equal to the frame size constraint specified in Transmit Checksum Offload Engine.	Yes	Yes
IPv6 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in MAC) but less than or equal to the frame size constraint specified in Transmit Checksum Offload Engine.	Not Applicable	Yes
IPv4 with TCP, UDP, or ICMP	Yes	Yes
IPv4 packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet is an IPv4 fragment	Yes	No
IPv6 packet with the following next header fields in main or extension headers:		
Hop-by-hop options (in IPv6 main header)	Not Applicable	Yes
Hop-by-hop options (in IPv6 extension header)	Not Applicable	No
Destinations options	Not Applicable	Yes
Routing (with segment left 0)	Not Applicable	No
Routing (with segment left > 0)	Not Applicable	No
TCP, UDP, or ICMP	Not Applicable	Yes
Authentication	Not Applicable	No
Any other next header field in main or extension headers	Not Applicable	No
IPv4 packet has TCP header with Options fields	Yes	Yes
IPv4 Tunnels:	nbsp;	nbsp;

Table continues on the next page...

Table 393. Transmit Checksum Offload Engine Functions for Different Packet Types (continued)

Packet Type	Hardware IP headerchecksum insertion	Hardware TCP/ UDPchecksum insertion
IPv4 packet in an IPv4 tunnel	Yes (IPv4 tunnel header)	No
IPv6 packet in an IPv4 tunnel	Yes (IPv4 tunnel header)	No
IPv6 Tunnels:	nbsp;	nbsp;
IPv4 packet in an IPv6 tunnel	Not applicable	No
IPv6 packet in an IPv6 tunnel	Not applicable	No
IPv4 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Yes	Yes
IPv6 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Not applicable	Yes
IPv4 frames with security features (such as encapsulated security payload)	Yes	No
IPv6 frames with security features (such as encapsulated security payload)	Not applicable	No

43.4.7.2 Receive Checksum Offload Engine

IP provides the Checksum Offload Engine that is used to detect any error in an IPv4 or IPv6 packet in the receive path. The MAC verifies the checksum field of the received packet with the internally calculated checksum and provides the status.

The Receive Checksum Offload engine is used to detect errors in IP packets by calculating the header checksum and further matching it with the received header checksum. This engine also identifies a TCP, UDP, or ICMP payload in received IP packets and calculates the checksum of such payloads appropriately.

Here, the IPv4 or IPv6 packets in the received Ethernet packets are detected and processed for data integrity. The MAC receiver identifies IPv4 or IPv6 packets by checking for value 0x0800 or 0x86DD, respectively, in the Type field of the received Ethernet packet. This identification is applicable to single VLAN-tagged packets. It is also applicable to double VLAN-tagged packets when the Enable Double VLAN Processing option is selected and the EDVLP bit of the MAC_VLAN_Tag register is set.

The Rx COE calculates the IPv4 header checksums and checks that they match the received IPv4 header checksums. The result of this operation (pass or fail) is given to the RFC module for insertion into the receive status word. The IP Header Error bit is set for any mismatch between the indicated payload type (Ethernet Type field) and the IP header version, or when the received packet does not have enough bytes, as indicated by the Length field of the IPv4 header (or when fewer than 20 bytes are available in an IPv4 or IPv6 header).

This engine also identifies a TCP, UDP, or ICMP payload in the received IP datagrams (IPv4 or IPv6) and calculates the checksum of such payloads properly, as defined in the TCP, UDP, or ICMP specifications. This engine includes the TCP, UDP, or ICMPv6 pseudo-header bytes for checksum calculation and checks whether the received checksum field matches the calculated value. The result of this operation is given as a Payload Checksum Error bit in the receive status word. This status bit is also set if the length of the TCP, UDP, or ICMP payload does not match the expected payload length given in the IP header.

Following table describes the functions supported by the Rx COE based on the packet type. When the payload of an IP packet is not processed (indicated as "No" in the table), the information (whether the checksum engine is bypassed or not) is given in the receive status.

NOTE

The MAC does not append any payload checksum bytes to the received Ethernet packets.

Table 394. Receive Checksum Offload Engine Functions for Different Packet Types

Packet type	Hardware IP header checksum checking	Hardware TCP/UDP/ICMP checksum checking
Non-IPv4 or IPv6	No	No
IPv4 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in the MAC)	Yes	Yes
IPv6 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in the MAC)	Not Applicable	Yes
IPv4 with TCP, UDP, or ICMP	Yes	Yes
IPv4 header's protocol field contains a protocol other than TCP, UDP, or ICMP	Yes	No
IPv4 packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet is an IPv4 fragment	Yes	No
IPv6 packet with the following next header fields in main or extension headers: • Hop-by-hop options (in IPv6 main header) • Hop-by-hop options (in IPv6 extension header) • Destinations options • Routing (with segment left 0) • Routing (with segment left > 0) • TCP, UDP, or ICMP • Any other next header field in main or extension headers	• Not Applicable • Not Applicable • Not Applicable • Not Applicable • Not Applicable • Not Applicable • Not Applicable	• Yes • No • Yes • Yes • No • Yes • No
IPv4 packet has TCP header with Options fields	Yes	Yes
IPv4 Tunnels:	• Yes (IPv4 tunnel header) • Yes (IPv4 tunnel header)	• No • No

Table continues on the next page...

Table 394. Receive Checksum Offload Engine Functions for Different Packet Types (continued)

Packet type	Hardware IP header checksum checking	Hardware TCP/UDP/ICMP checksum checking
• IPv4 packet in an IPv4 tunnel • IPv6 packet in an IPv4 tunnel		
IPv4 Tunnels: • IPv4 packet in an IPv6 tunnel • IPv6 packet in an IPv6 tunnel	• Not Applicable • Not Applicable	• No • No
IPv4 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled)	Yes	Yes
IPv6 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled)	Not Applicable	Yes
IPv4 frames with security features (such as encapsulated security payload)	Yes	No
IPv6 frames with security features (such as encapsulated security payload)	Not Applicable	No

Set the IPC bit of the MAC_Configuration register for enabling Receive Checksum offload.

43.4.8 MAC Management Counters

IP supports storing the statistics about the received and transmitted packets in registers that are accessible through the application/software. The MMC module maintains a set of registers for gathering statistics on the received and transmitted packets. The MMC counters are free running.

For MMC register details, see registers "MMC_Control" to "MMC_Rx_FPE_Fragment_Cntr" in ENET_QOS register descriptions.

43.4.9 Flow Control

43.4.9.1 Transmit Flow Control

The Transmit Flow Control involves transmitting Pause packets in full-duplex mode and backpressure in half-duplex mode to control the flow of packets from the remote end.

43.4.9.1.1 Flow control in Full duplex mode

IP supports the IEEE802.3x Pause Packets.

Pause packet structure is shown in the following table.

Table 395. Pause Packet Fields

Field	Description
DA	Contains the special multicast address
SA	Contains the MAC address 0
Type	Contains 8808
MAC Control opcode	Contains 0001 for IEEE 802.3x Pause Control packets; 0101 for PFC packets
PT	Contains Pause time specified in the PT field of the MAC_Q#_Tx_Flow_Ctrl register

43.4.9.1.2 Pause Packet Control

When the FCB bit is set, the MAC generates and transmits a single Pause packet. If the FCB bit is set again after the Pause packet transmission is complete, the MAC sends another Pause packet irrespective of whether the pause time is complete or not. To extend the pause or terminate the pause prior to the time specified in the previously-transmitted Pause packet, the application should program the Pause Time register with appropriate value and then again set the FCB bit.

43.4.9.1.3 Flow Control in Half-Duplex Mode

In half-duplex mode, the MAC uses the deferral mechanism for the flow control (backpressure). When the application requests to stop receiving packets, the MAC sends a JAM pattern of 32 bytes when it senses a packet reception, provided the transmit flow control is enabled. This results in a collision and the remote station backs off. If the application requests a packet to be transmitted, it is scheduled and transmitted even when the backpressure is activated. If the backpressure is kept activated for a long time (and more than 16 consecutive collision events occur), the remote stations abort the transmission because of excessive collisions.

Following table describes the flow control in the Tx path for Queue 0 based on the setting of the following bits:

- EHFC bit of MTL_RxQ0_Operation_Mode register
- TFE bit of MAC_Q0_Tx_Flow_Ctrl register
- DM bit of MAC_Configuration register

Flow control is similar for all queues.

Table 396. Tx MAC Flow Control

EHFC	TFE	DM	Description
x	0	x	The MAC transmitter does not perform the flow control or backpressure operation.
0	1	0	The MAC transmitter performs back-pressure when Bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set or the sideband signal sbd_flowctrl_i is 1.
1	1	0	The MAC transmitter performs back-pressure when Bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set or the sideband signal sbd_flowctrl_i is 1. In addition, the MAC Tx performs back-pressure when Rx Queue level crosses the threshold set by Bits[10:8] of MTL_RxQ0_Operation_Mode register.
0	1	1	The MAC transmitter sends the Pause packet when Bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set or the sideband signal sbd_flowctrl_i is 1.

Table continues on the next page...

Table 396. Tx MAC Flow Control (continued)

EHFC	TFE	DM	Description
1	1	1	The MAC transmitter sends the Pause packet when Bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set or the sideband signal sbd_flowctrl_i is 1. In addition, the MAC Tx sends a Pause packet when Rx Queue level crosses the threshold set by Bits[10:8] of MTL_RxQ0_Operation_Mode register.

43.4.9.1.4 Enabling Transmit Flow Control

To independently enable Transmit Flow Control for each Tx queue, set the TFE bit in the MAC_Q#_Tx_Flow_Ctrl register.

43.4.9.2 Receive Flow Control

In the Receive path, the Flow Control is functional only in the full-duplex mode. If any Pause packet is received in the half-duplex mode, the packet is considered as a normal control packet.

NOTE

Receive pause packets should have a frame size of 64 bytes.

The Receive Flow Control is implemented by the MAC based on the bit value of the respective register, and the destination address and different fields of the received packet.

Following table describes the flow control in the Rx path based on the setting of the following bits:

- RFE bit of MAC_RX_FLOW_CTRL register
- DM bit of MAC_CONFIGURATION register

Table 397. Rx MAC Flow Control

RFE	DM	Description
0	x	The MAC receiver does not detect the received Pause packets.
1	0	The MAC receiver does not detect the received Pause packets but recognizes such packets as Control packets.
1	1	The MAC receiver detects or processes the Pause packets and responds to such packets by stopping the MAC transmitter.

43.4.9.2.1 Enabling Receive Flow Control

To enable Pause Flow Control, set the RFE bit in the MAC_Rx_Flow_Ctrl register.

43.4.10 Loopback Mode

The MAC supports Loopback of transmitted packets to its receiver.

The following are some guidelines for using the loopback mode:

- Enable loopback only with the full-duplex mode. In half-duplex mode, the carrier sense signal (crs) or collision (col) signal inputs get sampled which may result into issues such as packet dropping.
- If the loopback mode is enabled without connecting a PHY chip (for example, in FPGA setup), you should externally generate the Tx and Rx clocks and provide these clocks to the MAC.
- Do not loop back big packets. Big packets (>1522 bytes) may get corrupted in the loopback FIFO.

To enable this feature, program the LM bit of the MAC_Configuration register.

You can enable loopback for all PHY interfaces. The data is always looped back through internal asynchronous FIFO on to the internal Receive GMII interface, irrespective of which PHY interface is selected. The loopback data is also passed through the corresponding transmit PHY interface block, on to the Ethernet line.

43.4.11 Descriptors

43.4.11.1 Overview

The DMA in the Ethernet subsystem transfers data based on a linked list of descriptors. The application creates the descriptors in the system memory. The module supports the following two types of descriptors:

- Normal Descriptor: Normal descriptors are used for packet data and to provide control information applicable to the packets to be transmitted.
- Context Descriptor: Context descriptors are used to provide control information applicable to the packet to be transmitted.

Each normal descriptor contains two buffers and two address pointers. These buffers enable the adapter port to be compatible with various types of memory management schemes.

NOTE

There is no limit for the number of descriptors that can be used for a single packet.

43.4.11.2 Descriptor Structure

The module supports the ring structure for DMA descriptor as shown in the following figure.

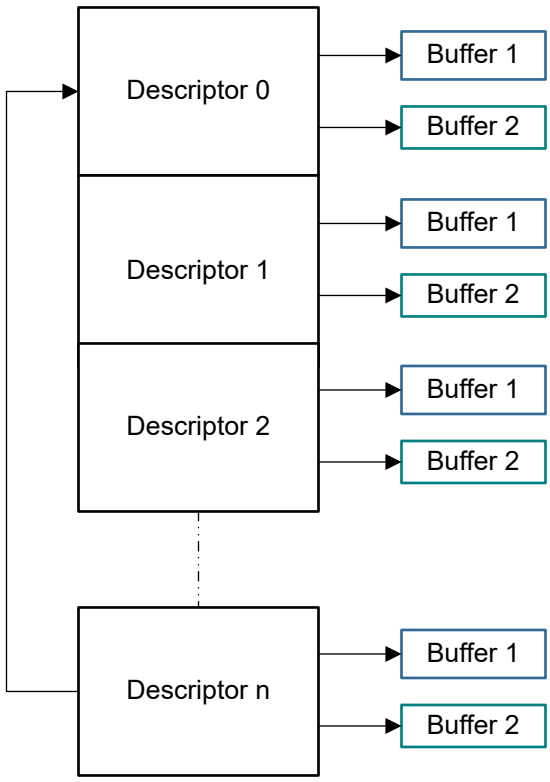


Figure 194. Descriptor Ring Structure

In Ring structure, descriptors are separated by the Word, DWord, or LWord number programmed in the DSL field of the DMA_CH#_Control register. The application needs to program the total ring length, that is, the total number of descriptors in ring span in the following registers of a DMA channel:

- Transmit Descriptor Ring Length Register (DMA_CH#_TxDesc_Ring_Length)
- Channel n Receive Control 2 register (DMA_CH#_RX_CONTROL2)

The Descriptor Tail Pointer Register contains the pointer to the descriptor address (N). The base address and the current descriptor pointer decide the address of the current descriptor that the DMA can process. The descriptors up to one location less than the one indicated by the descriptor tail pointer (N – 1) are owned by the DMA. The DMA continues to process the descriptors until the following condition occurs:

Current Descriptor Pointer == Descriptor Tail Pointer;

The DMA goes into the Suspend mode when this condition occurs. The application must perform a write to the Descriptor Tail pointer register and update the tail pointer so that the following condition is true:

Current Descriptor Pointer < Descriptor Tail Pointer;

The DMA automatically wraps around the base address when the end of ring is reached, as shown in the following figure.

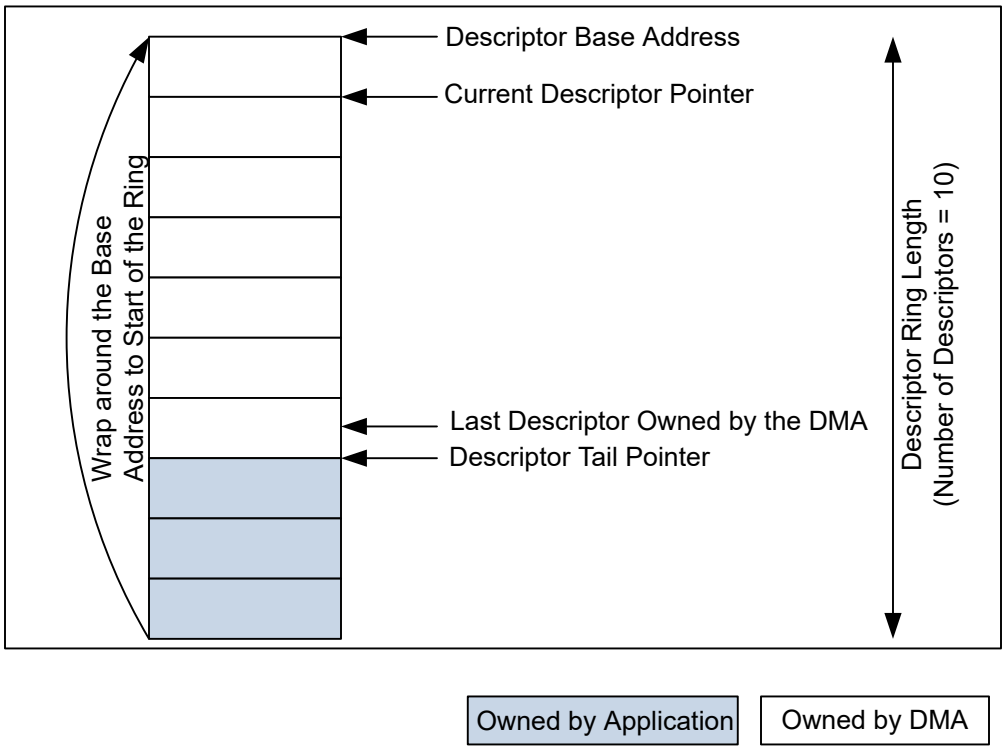


Figure 195. DMA Descriptor Ring

For descriptors owned by the application, the OWN bit of DES3 is reset to 0. For descriptors owned by the DMA, the OWN bit is set to 1. If the application has only one descriptor in the beginning, the application sets the last descriptor address (tail pointer) to Descriptor Base Address + 1. The DMA processes the first descriptor and then waits for the application to advance the tail pointer.

43.4.11.3 Transmit Descriptor

The DMA in the module requires at least one descriptor for a transmit packet. In addition to two buffers, two byte-count buffers, and two address pointers, the transmit descriptor has control fields which can be used to control the MAC operation on per-transmit packet basis. The Transmit Normal descriptor has two formats: Read format and Write-Back format

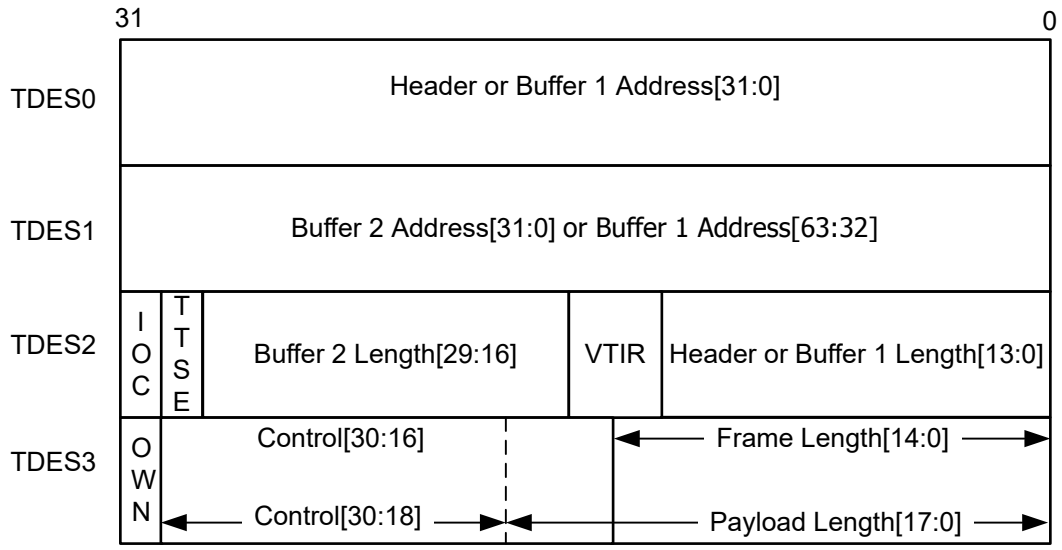
NOTE

TSO Split header is not supported.

43.4.11.3.1 Transmit Normal Descriptor (Read Format)

Following figure shows the Read Format for a Transmit normal descriptor. Table: 'TDES0 Normal Descriptor (Read Format)' through Table: 'TDES3 Normal Descriptor (Read Format)' describe the read format for the Transmit Normal Descriptors: TDES0, TDES1, TDES2, and TDES3.

Figure 196. Transmit Descriptor Read Format



43.4.11.3.1.1 TDES0 Normal Descriptor (Read Format)

Table 398. TDES0 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF1AP	Buffer 1 Address Pointer or TSO Header Address Pointer These bits indicate the physical address of Buffer 1. These bits indicate the TSO Header Address pointer when the following bits are set: • TSE bit of TDES3 • FD bit of TDES3

43.4.11.3.1.2 TDES1 Normal Descriptor (Read Format)

Table 399. TDES1 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF2AP	Buffer 2 or Buffer 1 Address Pointer

Table continues on the next page...

Table 399. TDES1 Normal Descriptor (Read Format)

Bit	Name	Description
		This bit indicates the physical address of Buffer 2 when a descriptor ring structure is used. There is no limitation for the buffer address alignment. In 40- or 48-bit addressing mode, these bits indicate the most-significant 8- or 16- bits of the Buffer 1 Address Pointer.

43.4.11.3.1.3 TDES2 Normal Descriptor (Read Format)

Table 400. TDES2 Normal Descriptor (Read Format)

31	30	29-16	15:14	13:0
IOC	TTSE/ TMWD	B2L	VTIR	HL or B1L

Table 401. TDES2 Normal Descriptor (Read Format)

Bits	Name	Description
31	IOC	Interrupt on Completion This bit controls the setting of TI and ETI status bits in the DMA_CH#_Status register. When ETIC = 1 and TDES2[LD] = 0, this bit sets the ETI bit. When TDES3[LD] = 1, this bit sets the TI status bit.
30	TTSE/ TMWD	Transmit Timestamp Enable or External TSO Memory Write Enable This bit enables the IEEE1588 time stamping for Transmit packet referenced by the descriptor, if TSE bit is not set. If TSE bit is set and external TSO memory is enabled, setting this bit disables external TSO memory writing for this packet.
29:16	B2L	Buffer 2 Length The driver sets this field. When set, this field indicates Buffer 2 length.
15:14	VTIR	VLAN Tag Insertion or Replacement These bits request the MAC to perform VLAN tagging or untagging before transmitting the packets. The application must set the CRC Pad Control bits appropriately when VLAN Tag Insertion, Replacement, or Deletion is enabled for the packet. The following list describes the values of these bits: • 2'b00: Do not add a VLAN tag. • 2'b01: Remove the VLAN tag from the packets before transmission. This option should be used only with the VLAN packets. • 2'b10: Insert a VLAN tag with the tag value programmed in the MAC_VLAN_Incl register or context descriptor. • 2'b11: Replace the VLAN tag in packets with the tag value programmed in the MAC_VLAN_Incl register or context descriptor. This option should be used only with the VLAN packets. These bits are valid when the Enable SA and VLAN Insertion on Tx option is selected while configuring the core.

Table continues on the next page...

Table 401. TDES2 Normal Descriptor (Read Format) (continued)

Bits	Name	Description
13:0	HL or B1L	Header Length or Buffer 1 Length For Header length only bits [9:0] are taken. The size 13:0 is applicable only when interpreting buffer 1 length. If the TCP Segmentation Offload feature is enabled through the TSE bit of TDES3, this field is equal to the header length. When the TSE bit is set in TDES3, the header length includes the length in bytes from Ethernet Source address till the end of the TCP header. The maximum header length supported for TSO feature is 1023 bytes. The maximum header length supported for TSO feature is 1023 bytes. If the TCP Segmentation Offload feature is not enabled, this field is equal to Buffer 1 length.

43.4.11.3.1.4 TDES3 Normal Descriptor (Read Format)

Table 402. TDES3 Normal Descriptor (Read Format)

31	30	29	28	27:26	25:23	22:19	18	17:16	15	14:0
OWN	CTXT	FD	LD	CPC	SAIC	SLOTNUM or THL	TSE	CIC/TPL	TPL	FL/TPL

Table 403. TDES3 Normal Descriptor (Read Format)

Bits	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit after it completes the transfer of data given in the associated buffer(s).
30	CTXT	Context Type This bit should be set to 1'b0 for normal descriptor.
29	FD	First Descriptor When this bit is set, it indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor When this bit is set, it indicates that the buffer contains the last segment of the packet. When this bit is set, the B1L or B2L field should have a non-zero value.
27:26	CPC	CRC Pad Control This field controls the CRC and Pad Insertion for Tx packet. This field is valid only when the first descriptor bit (TDES3[29]) is set. The following list describes the values of Bits[27:26]: • 2'b00: CRC and Pad Insertion The MAC appends the cyclic redundancy check (CRC) at the end of the transmitted packet of length greater than or equal to 60 bytes. The MAC automatically appends padding and CRC to a packet with length less than 60 bytes. • 2'b01: CRC Insertion (Disable Pad Insertion)

Table continues on the next page...

Table 403. TDES3 Normal Descriptor (Read Format) (continued)

Bits	Name	Description
		The MAC appends the CRC at the end of the transmitted packet but it does not append padding. The application should ensure that the padding bytes are present in the packet being transferred from the Transmit Buffer, that is, the packet being transferred from the Transmit Buffer is of length greater than or equal to 60 bytes. • 2'b10: Disable CRC Insertion The MAC does not append the CRC at the end of the transmitted packet. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. • 2'b11: CRC Replacement The MAC replaces the last four bytes of the transmitted packet with recalculated CRC bytes. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. This field is valid only for the first descriptor. Note: When the TSE bit is set, the MAC ignores this field because the CRC and pad insertion is always done for segmentation.
25:23	SAIC	SA Insertion Control These bits request the MAC to add or replace the Source Address field in the Ethernet packet with the value given in the MAC Address 0 or 1 register. The application must set the CRC Pad Control bits appropriately when SA Insertion Control is enabled for the packet. Bit 25 specifies the MAC Address Register (1 or 0) value that is used for Source Address insertion or replacement. The following list describes the values of Bits[24:23]: • 2'b00: Do not include the source address • 2'b01: Include the source address. For reliable transmission, the application must provide frames without source addresses. • 2'b10: Replace the source address. For reliable transmission, the application must provide frames with source addresses. • 2'b11: Reserved These bits are valid in the EQOS-DMA, EQOS-AXI, and EQOS-AHB configurations when the Enable SA and VLAN Insertion on Tx option is selected while configuring the core and when the First Segment control bit (TDES3 [29]) is set. This field is valid only for the first descriptor.
22:19	SLOTNUM or THL	SLOTNUM: Slot Number Control Bits in AV Mode These bits indicate the slot interval in which the data should be fetched from the corresponding buffers addressed by TDES0 or TDES1. When the Transmit descriptor is fetched, the DMA compares the slot number value in this field with the slot interval maintained in the RSN field DMA_CH#_Slot_Function_Control_Status. It fetches the data from the buffers only if a value matches. These bits are valid only for the AV channels. THL: TCP/UDP Header Length

Table continues on the next page...

Table 403. TDES3 Normal Descriptor (Read Format) (continued)

Bits	Name	Description
		If the TSE bit is set, this field contains the length of the TCP/UDP header. The minimum value of this field must be 5 for TCP header. The value must be equal to 2 for UDP header. This field is valid only for the first descriptor.
18	TSE	TSO Split header is not supported. The value of this bit is always zero.
17:16	CIC/TPL	Checksum Insertion Control or TCP Payload Length These bits control the checksum calculation and insertion. The following list describes the bit encoding: • 2'b00: Checksum Insertion Disabled. • 2'b01: Only IP header checksum calculation and insertion are enabled. • 2'b10: IP header checksum and payload checksum calculation and insertion are enabled, but pseudo-header checksum is not calculated in hardware. • 2'b11: IP Header checksum and payload checksum calculation and insertion are enabled, and pseudo-header checksum is calculated in hardware. This field is valid when the Enable Transmit TCP/IP Checksum Offload option is selected and the TSE bit is reset. When the TSE bit is set, this field contains the upper bits [17:16] of the TCP Payload (or IP Payload for UDP fragmentation). This allows the TCP/UDP packet length field to be spanned across TDES3[17:0] to provide 256 KB packet length support. This field is valid only for the first descriptor.
15	TPL	Reserved or TCP Payload Length When the TSE bit is reset, this bit is reserved. When the TSE bit is set, this is Bit 15 of the TCP payload length [17:0]. This field is valid only when the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected while configuring the core.
14:0	FL/TPL	Frame Length or TCP Payload Length This field is equal to the length of the packet to be transmitted in bytes. When the TSE bit is not set, this field is equal to the total length of the packet to be transmitted: Ethernet Header Length + TCP /IP Header Length – Preamble Length – SFD Length + Ethernet Payload Length When the TSE bit is set, this field is equal to the lower 15 bits of the TCP payload length in case of segmentation and IP payload in case of UDP fragmentation. In case of segmentation, this length does not include Ethernet header or TCP/UDP/IP header length. In case of fragmentation, this length does not include Ethernet header and IP header. When DWRR/WFQ algorithm is NOT enabled, value written into this field is not used when TSE = 0.

43.4.11.3.2 Transmit Normal Descriptor (Write-Back Format)

The write-back format of the Transmit Descriptor includes timestamp low, timestamp high, OWN, and Status bits.

The write-back format is applicable only for the last descriptor of the corresponding packet. The LD bit (TDES3[28]) is set in the descriptor where the DMA writes back the status and timestamp information for the corresponding Transmit packet.

Following figure illustrates the write-back format of the Transmit Descriptor.

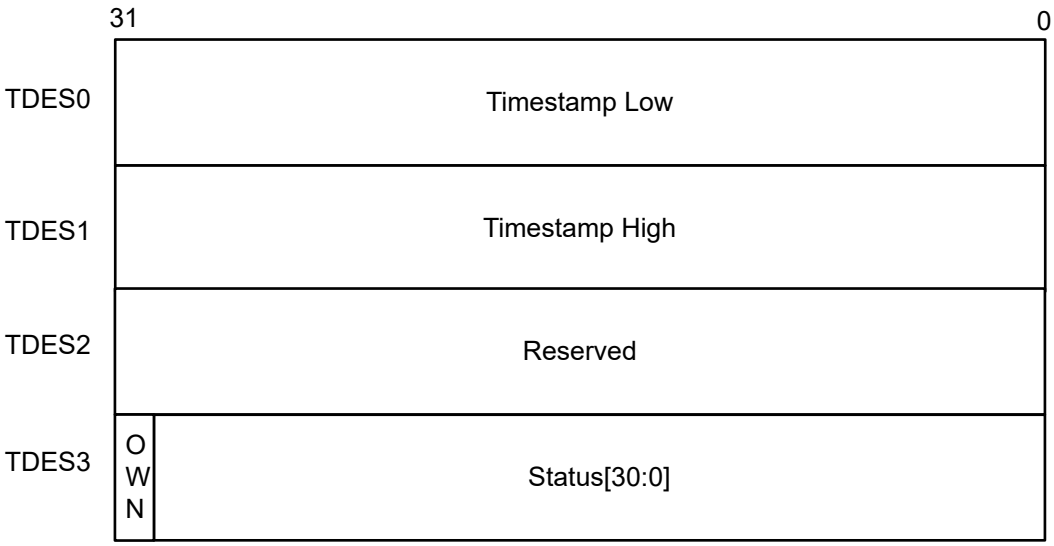


Figure 197. Transmit Descriptor Write-Back Format

TDES0 Normal Descriptor (Write-Back Format)

As described in following table, this format is only applicable to the last descriptor of a packet.

Table 404. TDES0 Normal Descriptor (Write-Back Format)TDES0 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low The DMA updates this field with least significant 32 bits of the timestamp captured for the corresponding Transmit packet. The DMA writes the timestamp only if TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Descriptor bit (LD) in the descriptor is set and the Timestamp status (TTSS) bit is set.

TDES1 Normal Descriptor (Write-Back Format)

As mentioned in the following table, this format is only applicable to the last descriptor of a packet.

Table 405. TDES1 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:0	TTSH	Transmit Packet Timestamp High The DMA updates this field with the most significant 32 bits of the timestamp captured for corresponding transmit packet. The DMA writes the timestamp only if the TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Descriptor bit (LD) in the descriptor is set and Timestamp status (TTSS) bit is set.

TDES2 Normal Descriptor (Write-Back Format)

This format is applicable only to the last descriptor of a packet.

Table 406. TDES2 Normal Descriptor (Write-Back Format)

Bit	Description
31:0	Reserved

TDES3 Normal Descriptor (Write-Back Format)

This format is applicable only to the last descriptor of a packet.

Table 407. TDES3 Normal Descriptor (Write-Back Format)

31	30	29	28	27:24	23	22:18	17	16	15	14	13	12	11	10	9	8	7:4	3	2	1	0
OWN	CTXT	FD	LD	Rsvd	DE	Rsvd	TTSS	EUE	ES	JT	FF	PCE	LoC	NC	LC	EC	CC	ED	UF	DB	IHE

Table 408. TDES3 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the the module DMA owns the descriptor. The DMA clears this bit when it completes the packet transmission. After the write-back is complete, this bit is set to 'b0.
30	CTXT	Context Type This bit should be set to 'b0 for Normal descriptor.
29	FD	First Descriptor This bit indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor This bit is set 'b1 for last descriptor of a packet. The DMA writes the status fields only in the last descriptor of the packet.
27:24	Rsvd	Reserved
23	DE	Descriptor Error When this bit is set, it indicates that the descriptor content is incorrect. The DMA sets this bit during write-back while closing the descriptor. Descriptor Errors can be: • Incorrect sequence from the context descriptor. For example, a location after the first descriptor for a packet. • All 1s • CTXT, LD, and FD bits set to 1.

Table continues on the next page...

Table 408. TDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		Note 1: When Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA will set the TI bit in the DMA_CH#_Status register Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.
22:18	Rsvd	Reserved
17	TTSS	Tx Timestamp Status This status bit indicates that a timestamp has been captured for the corresponding transmit packet. When this bit is set, TDES0 and TDES1 have timestamp values that were captured for the Transmit packet. This field is valid only when the Last Segment control bit (TDES3 [28]) in a descriptor is set. This bit is valid only when IEEE1588 timestamping feature is enabled; otherwise, it is reserved.
16	EUE	ECC Uncorrectable Error Status Indicates the ECC uncorrectable error in the TSO memory. Note: Uncorrectable error in Transmit FIFO memory is reported with (Bit 13) FF = 1. This is because, all such packets are flushed by the module.
15	ES	Error Summary This bit indicates the logical OR of the following bits: • TDES3[0]: IP Header Error • TDES3[14]: Jabber Timeout • TDES3[13]: Packet Flush • TDES3[12]: Payload Checksum Error • TDES3[11]: Loss of Carrier • TDES3[10]: No Carrier • TDES3[9]: Late Collision • TDES3[8]: Excessive Collision • TDES3[3]: Excessive Deferral • TDES3[2]: Underflow Error This bit is also set when EUE (bit 16) is set.
14	JT	Jabber Timeout This bit indicates that the MAC transmitter has experienced a jabber time-out. This bit is set only when the JD bit of the MAC_Configuration register is not set.
13	FF	Packet Flushed This bit indicates that the DMA or MTL flushed the packet because of a software flush command given by the CPU.

Table continues on the next page...

Table 408. TDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
12	PCE	Payload Checksum Error This bit indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either because of insufficient bytes, as indicated by the Payload Length field of the IP Header or the MTL starting to forward the packet to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet packet being transmitted to avoid deadlock, the MTL starts forwarding the packet when the FIFO is full, even in the store-and-forward mode. This error can also occur when Bus Error is detected during packet transfer. When the Full Checksum Offload engine is not enabled, this bit is reserved.
11	LoC	Loss of Carrier This bit indicates that Loss of Carrier occurred during packet transmission (that is, the gmii_crs_i signal was inactive for one or more transmit clock periods during packet transmission). This is valid only for the packets transmitted without collision and when the MAC operates in the half-duplex mode.
10	NC	No Carrier This bit indicates that the carrier sense signal form the PHY was not asserted during transmission.
9	LC	Late Collision This bit indicates that packet transmission was aborted because a collision occurred after the collision window (512 byte times including Preamble and Carrier Extension in GMII mode). This bit is not valid if Underflow Error is set.
8	EC	Excessive Collision This bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the MAC_Configuration register, this bit is set after first collision and the transmission of the packet is aborted.
7:4	CC	Collision Count This 4-bit counter value indicates the number of collisions occurred before the packet was transmitted. The count is not valid when the EC bit is set.
3	ED	Excessive Deferral This bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000 Mbps mode or Jumbo Packet enabled mode) if DC bit is set in the MAC_Configuration register. When TBS is enabled in full duplex mode and this bit is set, it indicates that the frame has been dropped after the expiry time has reached.
2	UF	Underflow Error This bit indicates that the MAC aborted the packet because the data arrived late from the system memory. The underflow error can occur because of either of the following conditions:

Table continues on the next page...

Table 408. TDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• The DMA encountered an empty Transmit Buffer while transmitting the packet • The application filled the MTL Tx FIFO slower than the MAC transmit rate The transmission process enters the suspended state and sets the underflow bit corresponding to a queue in the MTL_Interrupt_Status register.
1	DB	Deferred Bit This bit indicates that the MAC deferred before transmitting because of presence of carrier. This bit is valid only in the half-duplex mode.
0	IHE	IP Header Error When IP Header Error is set, this bit indicates that the Checksum Offload engine detected an IP header error. This bit is valid only when Tx Checksum Offload is enabled. Otherwise, it is reserved. If COE detects an IP header error, it still inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload. In full duplex mode, when EST/Qbv is enabled and this bit is set, it indicates the frame drop status due to Frame Size error or Schedule Error.

43.4.11.3.3 Transmit Context Descriptor

The context descriptor is used to provide the timestamps for one-step timestamp correction, VLAN Tag ID for VLAN insertion feature.

The Transmit Context descriptor can be provided any time before a packet descriptor. The context is valid for the current packet and subsequent packets. The context descriptor is used to provide the timestamps for one-step timestamp correction and VLAN Tag ID for VLAN insertion feature. Write back is done on a context descriptor only to reset the OWN bit.

NOTE

The VLAN Tag IDs and MSS values, provided by the application in a context descriptor with their corresponding Valid bits set, are stored internally by the DMA. When the outer or inner VLAN tag is provided with the Valid bit set, the DMA always passes the last valid VLAN tag to the MTL. The application cannot invalidate the valid VLAN tag stored by the DMA. The VLAN tag is inserted or replaced based on the control inputs provided for the packet.

The Inner VLAN Tag Control input is used only for the next packet that immediately follows the context descriptor. The application must provide a context descriptor before the normal descriptor of each packet for which the DMA should use the inner VLAN Tag control input.

The following figure shows the format of the Transmit Context descriptor.

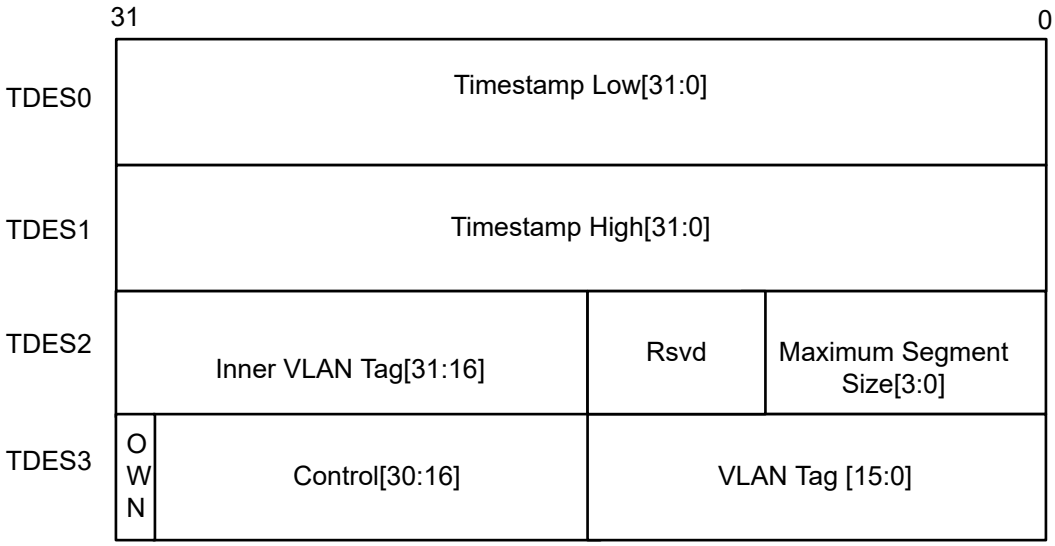


Figure 198. Transmit Context Descriptor Format

43.4.11.3.3.1 TDES0 Context Descriptor

Following table describes the format of the TDES0 Context descriptor.

Table 409. TDES0 Context Descriptor

Bit	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low For one-step correction, the driver can provide the lower 32 bits of timestamp in this descriptor word. The DMA uses this value as the low word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

43.4.11.3.3.2 TDES1 Context Descriptor

Following table describes the format of the TDES1 Context descriptor.

Table 410. TDES1 Context Descriptor

Bit	Name	Description
31:0	TTSH	Transmit Packet Timestamp High For one-step correction, the driver can provide the upper 32 bits of timestamp in this descriptor. The DMA uses this value as the high word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

43.4.11.3.3.3 TDES2 Context Descriptor

Following table explains the format of the TDES2 Context descriptor.

Table 411. TDES2 Context Descriptor

Bit	Name	Description
31:16	IVT	Inner VLAN Tag When the IVLTV bit of TDES3 context descriptor is set and the TCMSSV and OSTC bits of TDES3 context descriptor are reset, TDES2[31:16] contains the inner VLAN Tag to be inserted in the subsequent Transmit packets.
15:14	Rsvd	Reserved
13:0	MSS	TSO Split header is not supported. The value of this bit is always zero.

43.4.11.3.3.4 TDES3 Context Descriptor

Table 412. TDES3 Context Descriptor

31	30	29:28	27	26	25:24	23	22:18	17	16	15:0
OWN	CTXT	Rsvd	OSTC	TCMSSV	Rsvd	CDE	Rsvd	IVLTV	VLTV	VT

Follwoing table details the format of the TDES3 Context descriptor.

Table 413. TDES3 Context Descriptor

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the module DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit immediately after the read.
30	CTXT	Context Type This bit should be set to 1'b1 for Context descriptor.
29:28	Rsvd	Reserved
27	OSTC	One-Step Timestamp Correction Enable When this bit is set, the DMA performs a one-step timestamp correction with reference to the timestamp values provided in TDES0 and TDES1.
26	TCMSSV	One-Step Timestamp Correction Input or MSS Valid When this bit and the OSTC bit are set, it indicates that the Timestamp Correction input provided in TDES0 and TDES1 is valid. When the OSTC bit is reset and this bit and the TSE bit of TDES3 are set in subsequent normal descriptor, it indicates that the MSS input in TDES2 is valid.
25:24	Rsvd	Reserved
23	CDE	Context Descriptor Error

Table continues on the next page...

Table 413. TDES3 Context Descriptor (continued)

Bit	Name	Description
		When this bit is set, it indicates that the descriptor content is incorrect.The DMA sets this bit during write-back while closing the context descriptor. Descriptor Errors can be: • Incorrect sequence from the context descriptor. For example, a location before the first descriptor for a packet. • All 1s. • CD, LD, and FD bits set to 1. Note 1: When Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA will set the TI bit in the DMA_CH#_Status register Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.
22:20	Rsvd	Reserved
19:18	IVTIR	Inner VLAN Tag Insert or Replace When this bit is set, these bits request the MAC to perform Inner VLAN tagging or un-tagging before transmitting the packets. If the packet is modified for VLAN tags, the MAC automatically recalculates and replaces the CRC bytes. The following list describes the values of these bits: • 2'b00: Do not add the inner VLAN tag. • 2'b01: Remove the inner VLAN tag from the packets before transmission. This option should be used only with the VLAN frames. • 2'b10: Insert an inner VLAN tag with the tag value programmed in the MAC_In-ner_VLAN_Incl register or context descriptor. • 2'b11: Replace the inner VLAN tag in packets with the tag value programmed in the MAC_Inner_VLAN_Incl register or context descriptor. This option should be used only with the VLAN frames. These bits are valid when the Enable SA and VLAN Insertion on Tx and Enable Double VLAN Processing options are selected.
17	IVLTV	Inner VLAN Tag Valid When this bit is set, it indicates that the IVT field of TDES2 is valid.
16	VLTV	VLAN Tag Valid When this bit is set, it indicates that the VT field of TDES3 is valid.
15:0	VT	VLAN Tag This field contains the VLAN Tag to be inserted or replaced in the packet. This field is used as VLAN Tag only when the VLTI bit of the MAC_VLAN_Incl register is reset.

43.4.11.4 Receive Descriptor

The DMA in the module attempts to read a descriptor only if the Tail Pointer is different from the Base Pointer or current pointer. It is recommended to have a descriptor ring with a length that can accommodate at least two complete packets received by the MAC.

Otherwise, the performance of the DMA is impacted greatly because of the unavailability of the descriptors. In such situations, the RxFIFO in MTL becomes full and starts dropping packets.

The following Receive Descriptors are present:

- Normal descriptors
- Context descriptors

All RX descriptors are prepared by the software and given to the DMA as "Normal" Descriptors with the content as shown in Receive Normal Descriptor (Read Format). The DMA reads this descriptor and after transferring a received packet (or part of) to the buffers indicated by the descriptor, the Rx DMA will close the descriptor with the corresponding packet status. The format of this status is given in the "Receive Normal Descriptor (Write-Back Format)".

For some packets, the normal descriptor bits are not enough to write the complete status. For such packets, the RX DMA writes the extended status to the next descriptor (without processing or using the Buffers/ Pointers embedded in that descriptor). The format and content of the descriptor write back is described in "Receive Context Descriptor".

43.4.11.4.1 Receive Normal Descriptor (Read Format)

The read format for a Receive Normal descriptor is made up of a header or Buffer 1 address, reserved field, payload or Buffer 2 or Next Descriptor address, a 30-bit reserved filed, OWN bit, and an interrupt bit.

Following figure shows the Read Format for a Receive Normal Descriptor.

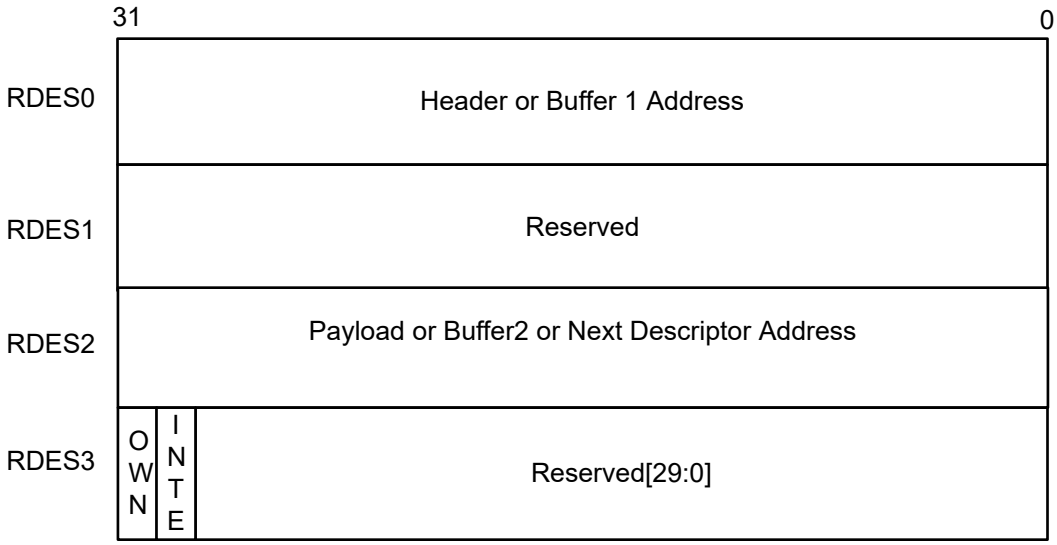


Figure 199. Receive Normal Descriptor Read Format

NOTE

In the Receive Descriptor (Read Format), if the Buffer Address field is all 0s, the module does not transfer data to that buffer and skips to the next buffer or next descriptor.

43.4.11.4.1.1 RDES0 Normal Descriptor (Read Format)

Following table explains the read format of the RDES0 Normal Descriptor.

Table 414. RDES0 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF1AP	Header or Buffer 1 Address Pointer When the SPH bit of Control register of a channel is reset, these bits indicate the physical address of Buffer 1. When the SPH bit is set, these bits indicate the physical address of Header Buffer where the Rx DMA writes the L2/L3/L4 header bytes of the received packet. The application can program a byte-aligned address for this buffer which means that the LS bits of this field can be non-zero. However, while transferring the start of packet, the DMA performs a Write operation with RDES0[1:0] (or RDES0[2:0]/[3:0] in case of 64-/128-bit configuration) as zero. However, the packet data is shifted as per actual offset as given by buffer address pointer. If the address pointer points to a buffer where the middle or last part of the packet is stored, the DMA ignores the offset address and writes to the full location as indicated by the data-width.

43.4.11.4.1.2 RDES1 Normal Descriptor (Read Format)

Following table describes the read format of the RDES1 Normal Descriptor.

Table 415. RDES1 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	Reserved or BUF1AP	In 64-bit addressing mode, this field contains the most-significant 32 bits of the Buffer 1 Address Pointer. Otherwise, this field is reserved.

43.4.11.4.1.3 RDES2 Normal Descriptor (Read Format)

Following table provides the read format of the RDES2 Normal Descriptor.

Table 416. RDES2 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF2AP	Buffer 2 Address Pointer These bits indicate the physical address of Buffer 2. When the SPH bit of the DMA_CH#_Control register is set, the buffer address pointer must be bus width-aligned, that is, RDES2[3:0, 2:0, or 1:0] = 0 corresponding to 128, 64, or 32 bus width. LSBs are ignored internally. When the SPH bit of the DMA_CH#_Control register is reset, there is no limitations on the RDES2 value. However, the RxDMA uses the LS Bits of the pointer address only while transferring the start bytes of a packet. If the BUF2AP is giving the address of a buffer in which the middle or last part of a packet is stored, the DMA ignores BUF2AP[3:0 or 2:0 or 1:0] (corresponding to 128- or 64- or 32-bit data-bus) and writes to the complete location.

43.4.11.4.1.4 RDES3 Normal Descriptor (Read Format)

Following table describes the read format of the RDES3 Normal Descriptor.

31	30	19-26	25	24	23:0
OWN	IOC	Rsvd	BUF2V	BUF1V	Rsvd

Table 417. RDES3 Normal Descriptor (Read Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the module DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: • The DMA completes the packet reception • The buffers associated with the descriptor are full
30	IOC	Interrupt Enabled on Completion When this bit is set, an interrupt is issued to the application when the DMA closes this descriptor.
29:26	Rsvd	Reserved
25	BUF2V	Buffer 2 Address Valid When this bit is set, it indicates to the DMA that the buffer 2 address specified in RDES2 is valid. The application must set this bit so that the DMA can use the address, to which the Buffer 2 address in RDES2 is pointing, to write received packet data.
24	BUF1V	Buffer 1 Address Valid When set, this indicates to the DMA that the buffer 1 address specified in RDES1 is valid. The application must set this value if the address pointed to by Buffer 1 address in RDES1 can be used by the DMA to write received packet data.
23:0	Rsvd	Reserved

43.4.11.4.2 Receive Normal Descriptor (Write-Back Format)

The following figure illustrates the write-back format for a Receive Normal descriptor.

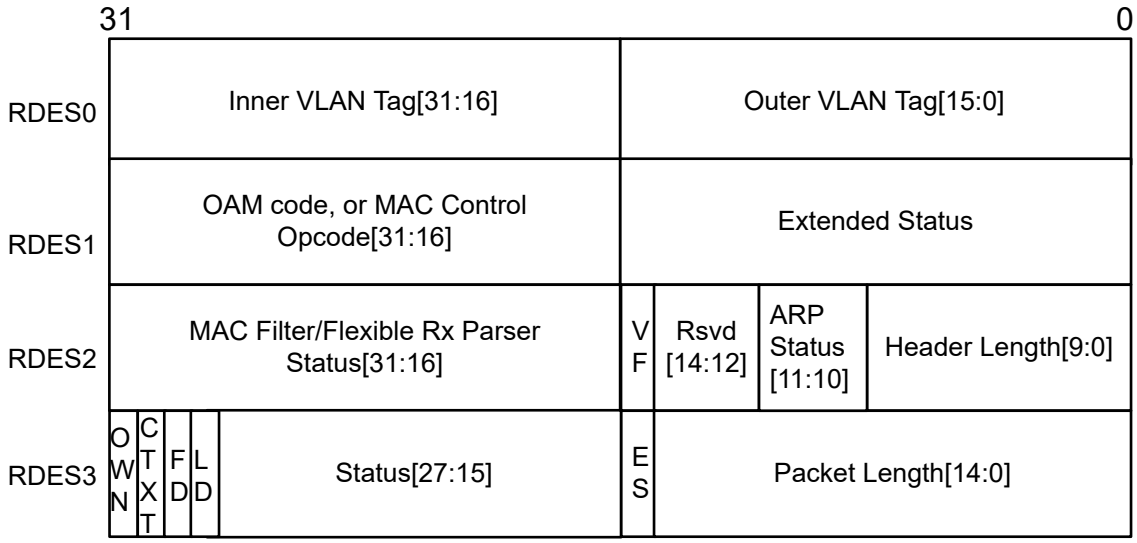


Figure 200. Receive Normal Descriptor (Write-Back Format)

NOTE

When Flexible RX Parser is enabled, RDES2[31:16] indicates Parser status, and not the MAC filter status. MAC filter status is not available when Flexible RX parser is enabled.

43.4.11.4.2.1 RDES0 Normal Descriptor (Write-Back Format)

Following table describes the write-back format for the RDES0 Normal Descriptor.

Table 418. RDES0 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:16	IVT	Inner VLAN Tag This field contains the Inner VLAN tag of the received packet if the RS0V bit of RDES3 is set. This is valid only when Double VLAN tag processing and VLAN tag stripping are enabled.
15:0	OVT	Outer VLAN Tag This field contains the Outer VLAN tag of the received packet if the RS0V bit of RDES3 is set.

43.4.11.4.2.2 RDES1 Normal Descriptor (Write-Back Format)

The Status fields in write-back format are valid only for the last descriptor (RDES3[28] is set). Following table provides the details of the write-back format for RDES1 Normal Descriptor.

31:16	15	14	13	12	11:8	7	6	5	4	3	2:0
OPC	TD	TSA	PV	PFT	PMT	IPCE	IPCB	IPV6	IPV4	IPHE	PT

Table 419. RDES1 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:16	OPC	OAM Sub-Type Code, or MAC Control Packet opcode OAM Sub-Type Code If Bits[18:16] of RDES3 are set to 3'b111, this field contains the OAM sub-type and code fields. MAC Control Packet opcode The bits[15:8] of RDES3 contains the subtype and bits [7:0] contains the code.
15	TD	Timestamp Dropped This bit indicates that the timestamp was captured for this packet but it got dropped in the MTL Rx FIFO because of overflow. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
14	TSA	Timestamp Available When Timestamp is present, this bit indicates that the timestamp value is available in a context descriptor word 2 (RDES2) and word 1(RDES1). This is valid only when the Last Descriptor bit (RDES3 [28]) is set. The context descriptor is written in the next descriptor just after the last normal descriptor for a packet.
13	PV	PTP Version This bit indicates that the received PTP message has the IEEE 1588 version 2 format. When this bit is reset, it indicates the IEEE 1588 version 1 format. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
12	PFT	PTP Packet Type This bit indicates that the PTP message is sent directly over Ethernet. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
11:8	PMT	PTP Message Type These bits are encoded to give the type of the message received: • 0000: No PTP message received • 0001: SYNC (all clock types) • 0010: Follow_Up (all clock types) • 0011: Delay_Req (all clock types) • 0100: Delay_Resp (all clock types) • 0101: Pdelay_Req (in peer-to-peer transparent clock) • 0110: Pdelay_Resp (in peer-to-peer transparent clock) • 0111: Pdelay_Resp_Follow_Up (in peer-to-peer transparent clock) • 1000: Announce • 1001: Management • 1010: Signaling

Table continues on the next page...

Table 419. RDES1 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• 1011–1110: Reserved • 1111: PTP packet with Reserved message type These bits are available only when you select the Timestamp feature.
7	IPCE	IP Payload Error When this bit is set, it indicates either of the following: • The 16-bit IP payload checksum (that is, the TCP, UDP, or ICMP checksum) calculated by the MAC does not match the corresponding checksum field in the received segment. • The TCP, UDP, or ICMP segment length does not match the payload length value in the IP Header field. • The TCP, UDP, or ICMP segment length is less than minimum allowed segment length for TCP, UDP, or ICMP. Bit 15 (ES) of RDES3 is not set when this bit is set.
6	IPCB	IP Checksum Bypassed This bit indicates that the checksum offload engine is bypassed. This bit is available when you select the Enable Receive TCP/IP Checksum Check feature.
5	IPV6	IPv6 header Present This bit indicates that an IPV6 header is detected. When the Enable Split Header Feature option is selected and the SPH bit of Control Register of a channel is set, the IPV6 header is available in the header buffer area to which RDES0 is pointing.
4	IPV4	IPV4 Header Present This bit indicates that an IPV4 header is detected. When the SPH bit of RDES3 is set, the IPV4 header is available in the header buffer area to which RDES0 is pointing.
3	IPHE	IP Header Error When this bit is set, it indicates either of the following: • The 16-bit IPv4 header checksum calculated by the MAC does not match the received checksum bytes. • The IP datagram version is not consistent with the Ethernet Type value. • Ethernet packet does not have the expected number of IP header bytes. This bit is valid when either Bit 5 or Bit 4 is set. This bit is available when you select the Enable Receive TCP/IP Checksum Check feature.
2:0	PT	Payload Type These bits indicate the type of payload encapsulated in the IP datagram processed by the Receive Checksum Offload Engine (COE): • 3'b000: Unknown type or IP/AV payload not processed • 3'b001: UDP • 3'b010: TCP

Table continues on the next page...

Table 419. RDES1 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• 3'b011: ICMP • 3'b110: AV Tagged Data Packet • 3'b111: AV Tagged Control Packet • 3'b101: AV Untagged Control Packet • 3'b100: IGMP if IPV4 Header Present bit is set else DCB (LLDP) Control Packet If the COE does not process the payload of an IP datagram because there is an IP header error or fragmented IP, it sets these bits to 3'b000.

43.4.11.4.2.3 RDES2 Normal Descriptor (Write-Back Format)

31:29	28	27	26:19	18	17	16	15	14	13:11	10	9:0
L3L4FM	L4FM	L3FM	MADRM	HF	DAF	SAF	OTS	ITS	Rsvd	ARPNR	HL

Table 420. RDES2 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:29	L3L4FM	Layer 3 and Layer 4 Filter Number Matched These bits indicate the number of the Layer 3 and Layer 4 Filter that matched the received packet: • 000: Filter 0 • 001: Filter 1 • 010: Filter 2 • 011: Filter 3 • 100: Filter 4 • 101: Filter 5 • 110: Filter 6 • 111: Filter 7 This field is valid only when Bit 28 or Bit 27 is set high. When more than one filter matches, these bits give the number of lowest filter. Note: This status is not available when Flexible RX Parser is enabled.
28	L4FM	Layer 4 Filter Match When this bit is set, it indicates that the received packet matches one of the enabled Layer 4 Port Number fields. This status is given only when one of the following conditions is true: • Layer 3 fields are not enabled and all enabled Layer 4 fields match • All enabled Layer 3 and Layer 4 filter fields match When more than one filter matches, this bit gives the layer 4 filter status of filter indicated by Bits[31:29].

Table continues on the next page...

Table 420. RDES2 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		Note: This status is not available when Flexible RX Parser is enabled.
27	L3FM	Layer 3 Filter Match When this bit is set, it indicates that the received packet matches one of the enabled Layer 3 IP Address fields. This status is given only when one of the following conditions is true: • All enabled Layer 3 fields match and all enabled Layer 4 fields are bypassed • All enabled filter fields match When more than one filter matches, this bit gives the layer 3 filter status of filter indicated by Bits[31:29]. Note: This status is not available when Flexible RX Parser is enabled.
26:19	MADRM	MAC Address Match or Hash Value When the HF bit is reset, this field contains the MAC address register number that matched the Destination address of the received packet. This field is valid only if the DAF bit is reset. When the HF bit is set, this field contains the hash value computed by the MAC. A packet passes the hash filter when the bit corresponding to the hash value is set in the hash filter register. Note: This status is not available when Flexible RX Parser is enabled.
18	HF	Hash Filter Status When this bit is set, it indicates that the packet passed the MAC address hash filter. Bits[26:19] indicate the hash value. Note: This status is not available when Flexible RX Parser is enabled.
17	DAF/RXPI	Destination Address Filter Fail When Flexible RX Parser is disabled, and this bit is set, it indicates that the packet failed the DA Filter in the MAC. When Flexible RX Parser is enabled, this bit is set to indicate that the packet parsing is incomplete (RXPI) due to ECC error. Note: When this bit is set, ES bit of RDES3 is also set.
16	SAF/RXPD	SA Address Filter Fail When Flexible RX Parser is disabled, and this bit is set, it indicates that the packet failed the SA Filter in the MAC. When Flexible RX Parser is enabled, this bit is set to indicate that the packet is dropped (RXPD) by the parser. Note: When this bit is set, ES bit of RDES3 is also set.
15	OTS	VLAN Filter Status When set, this bit indicates that the VLAN Tag of the received packed passed the VLAN filter. This bit is redefined as Outer VLAN Tag Filter Status (OTS). This bit is valid for both Single and Double VLAN Tagged frames

Table continues on the next page...

Table 420. RDES2 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
14	ITS	Inner VLAN Tag Filter Status (ITS) This bit is valid only for Double VLAN Tagged frames, when Double VLAN Processing is enabled. For more information, see Filter Status.
13:11	Rsvd	Reserved
10	ARPNR	ARP Reply Not Generated When this bit is set, it indicates that the MAC did not generate the ARP Reply for received ARP Request packet. This bit is set when the MAC is busy transmitting ARP reply to earlier ARP request (only one ARP request is processed at a time). This bit is reserved when the Enable IPv4 ARP Offload option is not selected.
9:0	HL	L3/L4 Header Length This field contains the length of the header of the packet split by the MAC at L3 or L4 header boundary as identified by the MAC receiver. This field is valid only when the first descriptor bit is set (FD = 1). The header data is written to the Buffer 1 address of corresponding descriptor. If header length is zero, this field is not valid. It implies that the MAC did not identify and split the header. This field is valid when the Enable Split Header Feature option is selected.

43.4.11.4.2.4 RDES3 Normal Descriptor (Write-Back Format)

Following table describes the write-back format for the RDES3 Normal Descriptor.

31	30	29	28	27	26	25	24	23	22	21	20	19	18:16	15	14:0
OWN	CTXT	FD	LD	RS2V	RS1V	RS0V	CE	GP	RWT	OE	RE	DE	LT	ES	PL

Table 421. RDES3 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the module DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: • The DMA completes the packet reception • The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor When this bit is set, it indicates that the current descriptor is a context type descriptor. The DMA writes 1'b0 to this bit for normal receive descriptor. When CTXT and FD bits are used together, {CTXT, FD}

Table continues on the next page...

Table 421. RDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• 2'b00: Intermediate Descriptor • 2'b01: First Descriptor • 2'b10: Reserved • 2'b11: Descriptor Error (due to all 1s) Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data Receive DMA will set the CDE bit in DMA_CH#_Status register but not the RI bit even when IOC is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.
29	FD	First Descriptor When this bit is set, it indicates that this descriptor contains the first buffer of the packet. If the size of the first buffer is 0, the second buffer contains the beginning of the packet. If the size of the second buffer is also 0, the next descriptor contains the beginning of the packet. See the CTXT bit description for details of using the CTXT bit and FD bit together.
28	LD	Last Descriptor When this bit is set, it indicates that the buffers to which this descriptor is pointing are the last buffers of the packet.
27	RS2V	Receive Status RDES2 Valid When this bit is set, it indicates that the status in RDES2 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
26	RS1V	Receive Status RDES1 Valid When this bit is set, it indicates that the status in RDES1 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
25	RS0V	Receive Status RDES0 Valid When this bit is set, it indicates that the status in RDES0 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
24	CE	CRC Error When this bit is set, it indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received packet. This field is valid only when the LD bit of RDES3 is set.
23	GP	Giant Packet When this bit is set, it indicates that the packet length exceeds the specified maximum Ethernet size of 1518, 1522, or 2000 bytes (9018 or 9022 bytes if jumbo packet enable is set). Note: Giant packet indicates only the packet length. It does not cause any packet truncation.
22	RWT	Receive Watchdog Timeout

Table continues on the next page...

Table 421. RDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		When this bit is set, it indicates that the Receive Watchdog Timer has expired while receiving the current packet. The current packet is truncated after watchdog timeout.
21	OE	Overflow Error When this bit is set, it indicates that the received packet is damaged because of buffer overflow in Rx FIFO. Note: This bit is set only when the DMA transfers a partial packet to the application. This happens only when the Rx FIFO is operating in the threshold mode. In the store-and-forward mode, all partial packets are dropped completely in Rx FIFO.
20	RE	Receive Error When this bit is set, it indicates that the gmii_rxer_i signal is asserted while the gmii_rxdv_i signal is asserted during packet reception. This error also includes carrier extension error in the GMII and half-duplex mode. Error can be of less or no extension, or error (rxd!= 0f) during extension.
19	DE	Dribble Bit Error When this bit is set, it indicates that the received packet has a non-integer multiple of bytes (odd nibbles). This bit is valid only in the MII Mode.
18:16	LT	Length/Type Field This field indicates if the packet received is a length packet or a type packet. The encoding of the 3 bits is as follows: • 3'b000: The packet is a length packet • 3'b001: The packet is a type packet. • 3'b011: The packet is a ARP Request packet type • 3'b100: The packet is a type packet with VLAN Tag • 3'b101: The packet is a type packet with Double VLAN Tag • 3'b110: The packet is a MAC Control packet type • 3'b111: The packet is a OAM packet type • 3'b010: Reserved
15	ES	Error Summary When this bit is set, it indicates the logical OR of the following bits: • RDES3[24]: CRC Error • RDES3[19]: Dribble Error • RDES3[20]: Receive Error • RDES3[22]: Watchdog Timeout • RDES3[21]: Overflow Error • RDES3[23]: Giant Packet • RDES2[17]: Destination Address Filter Fail, when Flexible RX Parser is enabled

Table continues on the next page...

Table 421. RDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• RDES2[16]: SA Address Filter Fail, when Flexible RX Parser is enabled • RDES2[13:11]: Receive Parser Overflow Errors (3'b010) This field is valid only when the LD bit of RDES3 is set.
14:0	PL	Packet Length These bits indicate the byte length of the received packet that was transferred to system memory (including CRC). This field is valid when the LD bit of RDES3 is set and Overflow Error bits are reset. The packet length also includes the two bytes appended to the Ethernet packet when IP checksum calculation is enabled and the received packet is not a MAC control packet. This field is valid when the LD bit of RDES3 is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current packet.

43.4.11.4.3 Receive Context Descriptor

This descriptor is read-only for the application. Only the DMA can write to this descriptor. The context descriptor provides information about the extended status related to the last received packet. The Bit 30 of RDES3 indicates the context type descriptor.

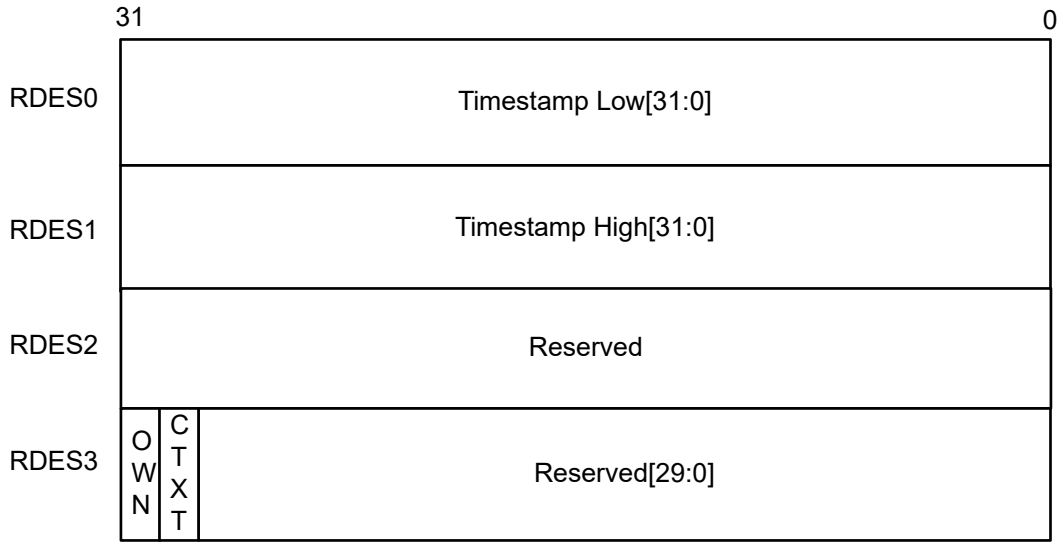


Figure 201. Receive Context Descriptor Format

43.4.11.4.3.1 RDES0 Context Descriptor

Table 422. RDES0 Context Descriptor

Bit	Name	Description
31:0	RTSL	Receive Packet Timestamp Low The DMA updates this field with least significant 32 bits of the timestamp captured for corresponding Receive packet. When this field and the RTSH field of RDES1 show all-ones value, the timestamp must be considered as corrupt.

43.4.11.4.3.2 RDES1 Context Descriptor

Table 423. RDES1 Context Descriptor

Bit	Field	Description
31:0	RTSH	Receive Packet Timestamp High The DMA updates this field with most significant 32 bits of the timestamp captured for corresponding receive packet. When this field and the RTSL field of RDES0 show all-ones value, the timestamp must be considered as corrupt.

43.4.11.4.3.3 RDES2 Context Descriptor

Table 424. RDES2 Context Descriptor

Bit	Description
31:0	Reserved

43.4.11.4.3.4 RDES3 Context Descriptor

31	30	29	28:0
OWN	CTXT	DE	Rsvd

Table 425. RDES3 Context Descriptor

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: • The DMA completes the packet reception • The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor

Table continues on the next page...

Table 425. RDES3 Context Descriptor (continued)

Bit	Name	Description
		When this bit is set, it indicates that the current descriptor is a context descriptor. The DMA writes 1'b1 to this bit for context descriptor. DMA writes 2’b11 to indicate a descriptor error due to all 1s. When CTXT and DE bits are used together, {CTXT, DE} • 2’b00: Reserved • 2’b01: Reserved • 2’b10: Context Descriptor • 2’b11: Descriptor Error Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data Receive DMA will set the CDE bit in DMA_CH#_Status register but not the RI bit even when IOC is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.
29	DE	Descriptor Error See the CTXT bit description for details of using theDE bit along with CTXT bit.
28:0	Rsvd	Reserved

43.4.11.5 Enhanced Descriptor for Time-Based Scheduling

The time-based scheduling feature needs 32 bytes Enhanced Descriptors to be enabled on all the DMA channels that intend to use the feature (by setting the EDSE bit of DMA_CH(#i)_TX_Control register).

The structure of 32-byte Descriptor for the Context and the Normal Descriptor in Read and Write formats are described in the following sections.

43.4.11.5.1 Enhanced Normal Descriptor - Read (32-Bit Mode)

The fields present in the first 16 bytes of the Enhanced Descriptor format of normal descriptor are as follows:

- LTV

  Launch Time Valid indicates the Launch Time (LT) and GSN fields present in the descriptor are valid. The LTV must be set only if the FD bit of the descriptor is not.

- GSN

  This field indicates the GCL slot number associated with the packet.

- Launch Time (LT)

  This field indicates the Launch Time associated with the packet.

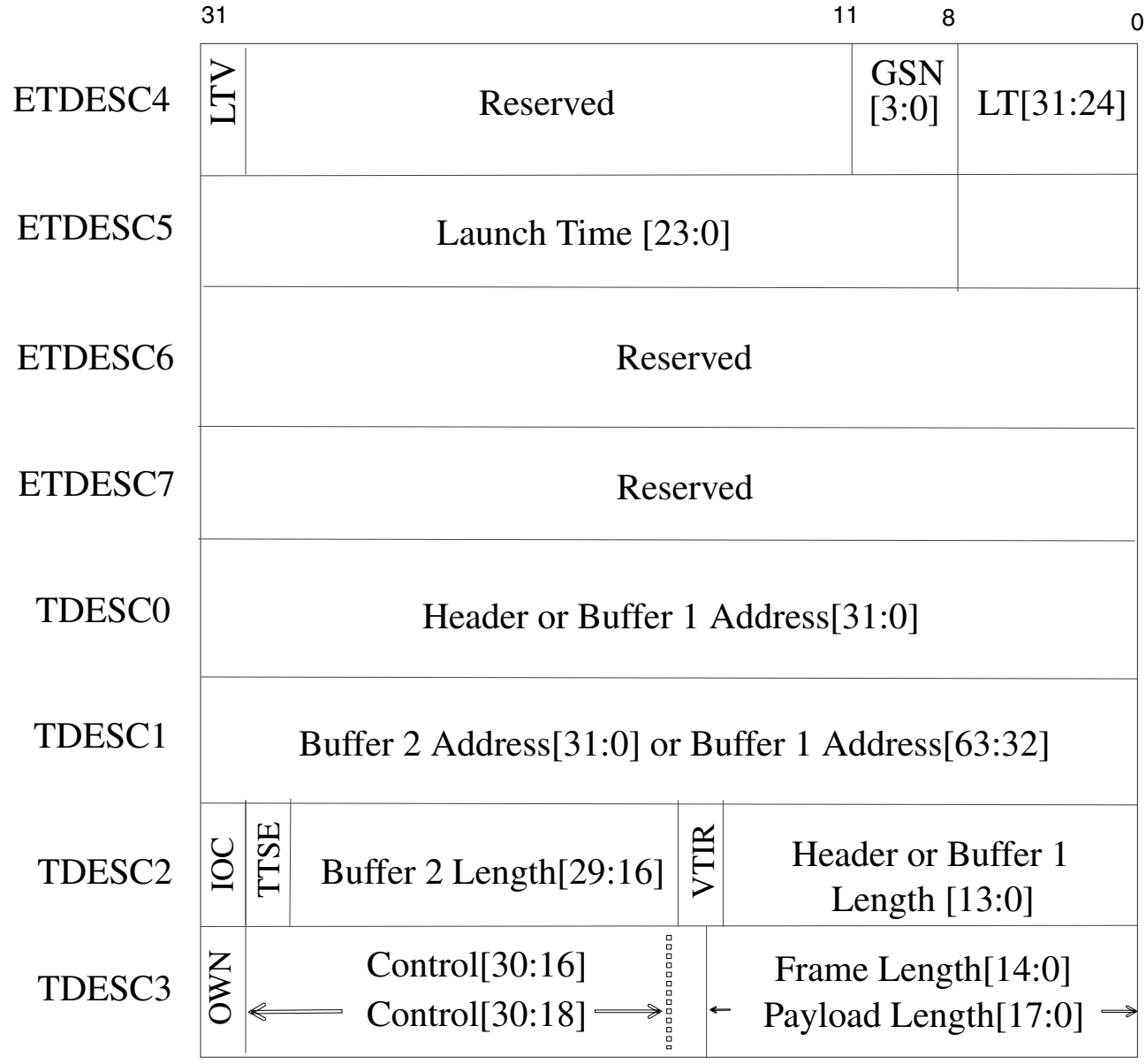


Figure 202. Enhanced Normal Descriptor - Read (32-Bit Mode)

43.4.11.5.2 Enhanced Normal Descriptor (Write, 32-Bit Mode)

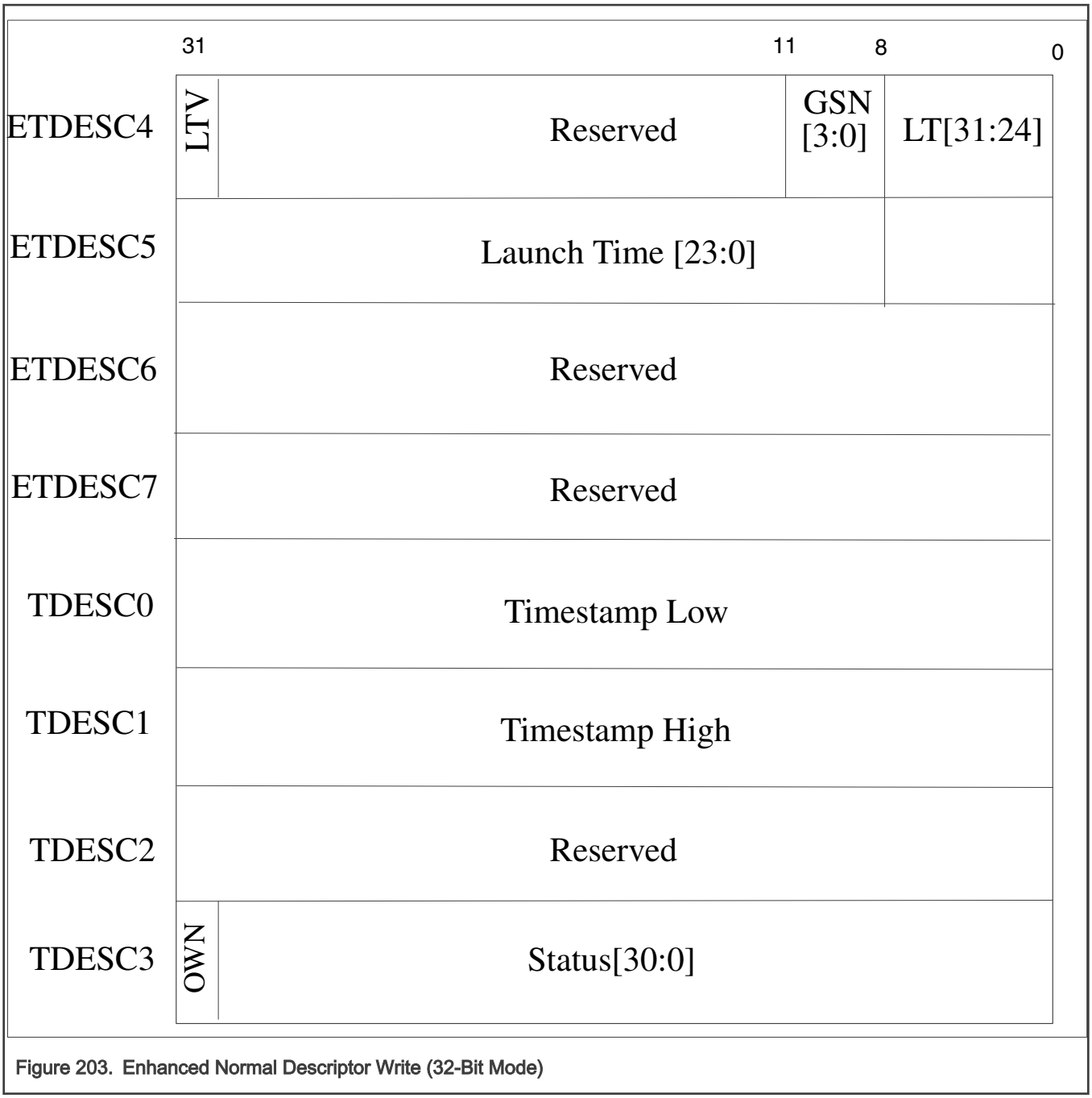


Figure 203. Enhanced Normal Descriptor Write (32-Bit Mode)

NOTE

- All the enhanced descriptor examples, assumes a 32-bit data width configurations. However, the same definitions are valid in 64 and 128 bit configurations.
- In both the write back formats, no modifications can be done to the extended 16 bytes. The rest of the 16 bytes (TDESC0 to TDESC3) are written back as per the previous 16 bytes Descriptor format.
- Fetch time when enabled overrides the AV slot function
- When an unaligned new GCL list (with a different CTR) is installed it is recommended not to have traffic during the installation of the new list. Any traffic during the switching of the lists might have unpredictable behavior regarding Fetch, Launch, and Launch expiry as the CTR and BTR values get updated while the frame is being processed

43.4.11.5.3 Enhanced Context Descriptor (Read, 32-bit Mode)

The first 16 bytes of the Enhanced context descriptor (ETDESC4 to ETDESC7) are reserved and must be zeros. The fields present in last bytes of the Descriptor (TDESC0 to TDESC3) are same as the context descriptor (TDESC0 to TDESC3) in 16 byte format.

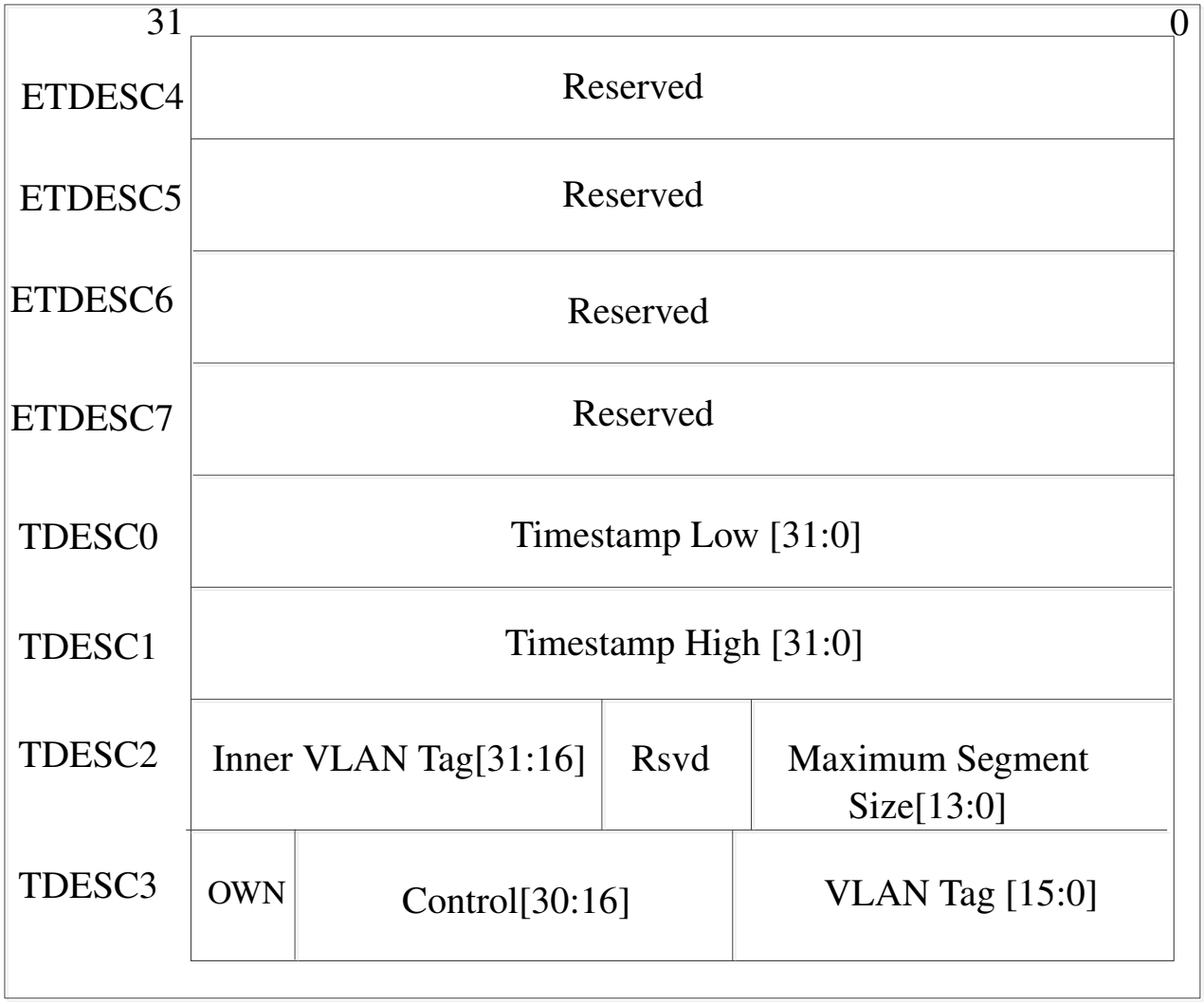


Figure 204. Enhanced Context Descriptor - Read

43.4.11.5.4 Enhanced Context Descriptor (Write, 32 Bit Mode)

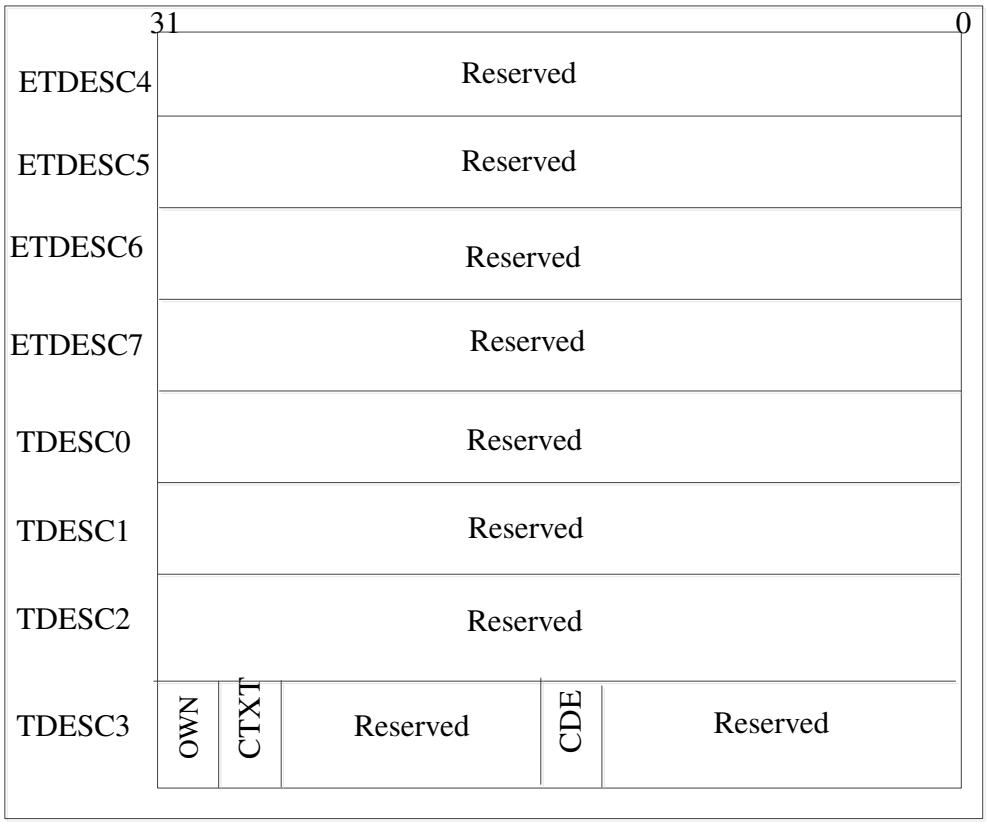


Figure 205. Enhanced Context Descriptor - Write

43.4.12 Power management and low power modes

This section describes the power management (PMT) mechanisms supported by the MAC. PMT supports the reception of network (remote) wake-up frames and magic packet frames. PMT does not perform the clock gate function, but generates interrupts for wake-up frames and magic packets received by the MAC. The PMT block sits on the receiver path of the MAC and is enabled with remote wake-up frame enable and magic packet enable. These enables are in the PMT control and status register and are programmed by the application.

When the power-down mode is enabled in the PMT, then all received frames are dropped by the Ethernet core and they are not forwarded to the application. The Ethernet comes out of the power-down mode only when either a magic packet or a remote wake-up frame is received and the corresponding detection is enabled.

43.4.12.1 Energy Efficient Ethernet

Energy Efficient Ethernet (EEE) is an operational mode that enables the IEEE 802.3 MAC sublayer along with a family of physical layers to operate in the Low-Power Idle (LPI) mode. The EEE operational mode supports the IEEE 802.3 MAC operation at 100 Mbps. The Ethernet module supports the IEEE 802.3az-2010 for EEE. The LPI mode allows power saving by switching off the parts of the communication device functionality when there is no data to be transmitted and received. The systems on both sides

of the link can disable some functionality to save power during periods of low-link utilization. The MAC controls whether the system should enter or exit the LPI mode and communicates this to the PHY. The EEE specifies the capabilities negotiation methods that the link partners can use to determine whether EEE is supported, and then select the set of parameters that are common to both devices.

43.4.12.1.1 Transmit path functions

In the transmit path, the software must set the LPIEN bit of the MAC LPI control status register to indicate to the MAC to stop transmission and initiate the LPI protocol. The MAC completes the transmission in progress, generates its transmission status, and starts transmitting the LPI pattern instead of the IDLE pattern if the link status has been up continuously for a period specified in the LPI LS TIMER(LST) field of MAC LPI timers control register. The PHY link status bit of the LPI control and status register indicates the link status of the PHY.

According to the Energy Efficient Ethernet standard (IEEE 802.3az-2010), the PHY must not stop the transmit clock during the LPI state in the MII (10 or 100) mode.

To make the PHY enter the LPI state, the MAC performs the following tasks:

1. De-asserts transmit enable (TX_EN).
2. Asserts transmit error (TX_ER). Sets TXD[3:0] to 0x1 The MAC maintains the same state of the TX_EN, TX_ER, and TXD signals for the entire duration during which the PHY remains in the LPI state.
3. Updates the status (TLPIEN bit of MAC LPI control status register and generates an interrupt. To bring the PHY out of the LPI state, that is, when the software resets the LPIEN bit, the MAC performs the following tasks:
4. Stops transmitting the LPI pattern and start transmitting the IDLE pattern
5. Starts the LPI TW TIMER
6. Updates the LPI exit status (TLPIEX bit of the MAC LPI control status register and generates an interrupt

The MAC cannot start the transmission until the wake-up time specified for the PHY expires. The auto-negotiated wake-up interval is programmed in the TWT field of the MAC LPI Timers control register.

43.4.12.1.2 Automated entry/exit of LPI mode in TX path

The MAC transmitter can be programmed to enter and exit LPI IDLE mode automatically based on whether it is IDLE for a specific period or has a packet to transfer. These modes are enabled and controlled by the MAC LPI control status register.

When the LPITXA (bit 19) and LPITXEN (bit 16) of the MAC LPI control status register are set, the MAC transmitter enters the LPI IDLE state when the MAC transmit path (including the MTL layers and DMA layers) are idle. The MAC transmitter exits the LPI IDLE state and clears the LPITXEN bit as soon as any of the functions in the TX path (DMA, MTL, or MAC) becomes non-idle due to initiation of a packet transfer. In addition to the above, when bit 20 (LPIATE) is also set, the MAC transmitter enters LPI IDLE state only when the transmit path remains in an idle state (no activity) for the time indicated by the value in MAC LPI Entry Timer. In this mode also, the MAC transmitter exits the LPI IDLE state as soon as any of the functions becomes non-idle. However, the LPITXEN bit is not cleared but remains active so that reentry to LPI IDLE state is possible without any software intervention when the MAC becomes idle again.

When both LPIATE and LPITXA bits are cleared, you can directly control the entry and exit of the LPI IDLE state by programming the LPITXEN bit.

43.4.12.1.3 Receive path functions

In the receive path, when the PHY receives the signals from the link partner to enter into the LPI state, the PHY and MAC perform the following tasks:

1. The PHY asserts RX_ER.
2. The PHY sets RXD[7:0] to 0x01.
3. The PHY de-asserts RX_DV. The PHY maintains the same state of the RX_ER, RXD, and RX_DV signals for the entire duration during which it remains in the LPI state.

4. The MAC updates the RLPIEN bit of the MAC LPI control status register and immediately generates an interrupt.

When the PHY receives signals from the link partner to exit the LPI state, the PHY and MAC perform the following tasks:

1. The PHY de-asserts RX_ER and returns to a normal inter-packet state.
2. The MAC updates the RLPIEX bit of the MAC LPI control status register and generates an interrupt immediately. The sideband signal (synchronous to Rx clock) is also asserted.

43.4.12.1.4 LPI timers

The transmitter maintains the following two timers that are loaded with the respective values from the MAC_LPI_Timers_Control register:

- LPI LS Timer
  - The LPI LS TIMER counts, in milliseconds, the time expired since the link status is up. You can enable the monitoring of the LUD bit of the MAC_PHYIF_Control_Status register by setting the PLSEN bit of the MAC_LPI_Control_Status register. The link status is indicated to the MAC by the LNKSTS bit of the MAC_PHYIF_Control_Status register or the value programmed by the software in the PLS bit of the MAC_LPI_Control_Status register. If the link status is not available in the MAC_PHYIF_Control_Status, the software should get the PHY link status by reading the PHY register and accordingly update the PLS bit. This timer is cleared every time the link goes down. It starts to increment when the link is up again and continues to increment until the value of the timer becomes equal to the terminal count. Once the terminal count is reached, the timer remains at the same value as long as the link is up. The terminal count is the value programmed in bits 25:16 of the MAC_LPI_Timers_Control register. The GMII interface does not assert the LPI pattern unless the terminal count is reached. This ensures a minimum time for which no LPI pattern is asserted after a link is established with the remote station. This period is defined as 1 second in the IEEE 802.3-az-2010. The LPI LS TIMER is 10-bit wide. Therefore, the software can program up to 1,023 ms.
- LPI TW Timer
  - The LPI TW TIMER counts, in microseconds, the time expired since the de-assertion of LPI. The terminal count should be programmed in bits 15:0 of MAC register 53 (LPI timers control register). The terminal count of the timer is the value of resolved transmit TW that is the auto-negotiated time after which the MAC can resume the normal transmit operation. After exiting the LPI mode, the MAC resumes its normal operation after the TW timer reaches the terminal count. The MAC supports the LPI TW TIMER in units of a microsecond. The LPI TW TIMER is 16-bits wide. Therefore, the software can program up to 65,535 µs.

43.4.12.1.5 LPI interrupt

The MAC generates the LPI interrupt when the Tx or Rx side enters or exits the LPI state. The low-power interrupt is asserted when the LPI interrupt status is set. The LPI interrupt can be cleared by reading the MAC LPI control status register.

43.4.12.2 Power management and wake-up

The power management (PMT) block supports the reception of magic packets as wake up packets, as well as the reception of network (remote) wake-up packets as wake up packets.

43.4.12.2.1 Magic packet detection

The magic packet frame is based on a method that uses Advanced Micro Device’s magic packet technology to power up the sleeping device on the network. The MAC receives a specific packet of information, called a magic packet, addressed to the node on the network.

Only magic packets that are addressed to the device or a broadcast address are checked to determine whether they meet the wake-up requirements. Magic packets that pass the address filtering (unicast or broadcast)are checked to determine whether they meet the remote wake-on-LAN data format of 6 bytes of all ones followed by a MAC address appearing 16 times.

The application enables magic packet wake-up by writing a 1 to bit 1 of the PMT control and status register. The PMT block constantly monitors each frame addressed to the node for a specific magic packet pattern. Each frame received is checked for

a 0xFFFF FFFF FFFF pattern following the destination and source address field. The PMT block then checks the frame for 16 repetitions of the MAC address without any breaks or interruptions. In case of a break in the 16 repetitions of the address, the 0xFFFF FFFF FFFF pattern is scanned for again in the incoming frame. The 16 repetitions can be anywhere in the frame but must be preceded by the synchronization stream (0xFFFF FFFF FFFF). The device also accepts a multicast frame, as long as the 16 duplications of the MAC address are detected.

If the MAC address of a node is 0x0011 2233 4455, then the MAC scans for the data sequence:

- 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
- 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
- 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
- 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
- ...CRC

Magic packet detection is updated in the PMT control and status register for magic packet received. A PMT interrupt to the application triggers a read to the PMT CSR to determine whether a magic packet frame is received. When you enable the power-down mode in the PMT block, the MAC drops all received packets and does not forward any packet to the MTL Rx FIFO or the application. The MAC comes out of the power-down mode only when a magic packet is received and the corresponding detection is enabled.

43.4.12.2.2 Remote wake-up detection

The PMT block supports the reception of network (remote) wake-up packets as wake-up packets.

When the MAC is in sleep mode and the remote wake-up bit is enabled in PMT control and status register (0x00C0), normal operation is resumed after receiving a remote wake-up frame. The application writes all eight wake-up filter registers by performing a sequential write to address (0x00C4). The application enables remote wake-up by writing a 1 to bit 2 of the PMT control and status register.

PMT supports 16 programmable filters that allow support of different receive frame patterns. If the incoming frame passes the address filtering of filter command, and if filter CRC-16 matches the incoming examined pattern, then the wake-up frame is received.

Filter_offset (minimum value 12, which refers to the 13th byte of the frame) determines the offset from which the frame is to be examined. Filter byte mask determines which bytes of the frame must be examined. The 31st bit of the byte mask must be set to zero.

The remote wake-up CRC block determines the CRC value that is compared with Filter CRC-16. The wake-up frame is checked only for length error, FCS error, dribble bit error, MII error, collision, and to ensure that it is not a runt frame. Even if the wake-up frame is more than 512 bytes long, if the frame has a valid CRC value, it is considered valid. Wake-up frame detection is updated in the PMT control and status register for every remote wake-up frame received. A PMT interrupt to the application triggers a read to the PMT control and status register to determine the reception of a wake-up frame.

43.4.12.3 System considerations during power-down

MAC neither gates nor stops clocks when power-down mode is enabled. Power saving by clock gating must be done outside the core by the application. The receive data path must be clocked with ENET_RX_CLK during power-down mode because it is involved in magic packet/wake-on-LAN frame detection. However, the transmit path and the application path clocks can be gated off during deep-sleep mode.

The PMT interrupt is asserted when a valid wake-up frame is received. This signal is generated in the receive clock domain.

The recommended power-down and wake-up sequences are as follows.

1. Disable the transmit DMA and wait for any previous frame transmissions to complete. These transmissions can be detected when transmit interrupt (see DMA channel status register bit TI) is received.
2. Disable the MAC transmitter and MAC receiver by clearing the appropriate bits in the MAC configuration register.

3. Wait until the receive DMA empties all the frames from the Rx FIFO. This can be done by reading the appropriate bits of Debug registers in the DMA and MTL CSR space.
4. Enable power-down mode by appropriately configuring the PMT registers.
5. Enable the MAC receiver and enter power-down mode.
6. Gate the application and transmit clock inputs to the core (and other relevant clocks in the system) to reduce power and enter sleep mode.
7. On receiving a valid wake-up frame, the MAC asserts the PMT interrupt signal and exits the power-down mode.
8. On receiving the interrupt, the system must enable the application and transmit clock inputs to the core.
9. Read the PMT status register to clear the interrupt, then enable the other modules in the system and resume normal operation.

43.4.13 Clocks

The table found here describes the clock sources for ENET_QOS. Please see the chip-specific clocking section for clock setting, configuration and gating information.

Table 426. ENET_QOS Clocks

Clock name	Description
aclk_i	Bus clock
clk_csr_i	CSR clock, same source as aclk_i
ptp_ref_i	PTP clock

43.4.14 Interrupts

This module supports interrupt coalescing which reduces the number of interrupts generated by IP and reduces the CPU load. Various interrupt event gets captured in status register. If interrupt enable bit of an interrupt is set then corresponding interrupt will be generated based on event status in status registers. INTM (interrupt mode) bit decides that interrupt signal is a level signal or pulse signal.

Interrupt output signals are shown below which are synchronous to control and Status register (CSR) clock.

- Sbd_intr_o – common interrupt
- Sbd_perch_tx_intr_o[max_dma_ch] – Interrupt per Tx channel
- Sbd_perch_rx_intr_o [max_dma_ch] – Interrupt per Rx channel

NOTE

max_dma_ch is equal to number of Tx/Rx queues which is 2.

The sbd_intr_o common interrupt is level signal. When sbd_intr_o interrupt is asserted, then corresponding interrupt event source can be found in DMA_Interrupt status register. DMA_Interrupt status register is read only register which contains the interrupt event source bits corresponding to each DMA channel (Tx and Rx queue pair), MAC transaction layer and MAC blocks. Software must then read the following registers corresponding to the bits that are set.

- MAC_Interrupt_Status register
- MTL_Interrupt_Status register
- DMA_CH(n)_Status register

sbd_intr_o interrupt gets de-asserted only when all the enabled interrupt events are cleared in their respective status registers and correspondingly all the bits in the DMA_Interrupt_Status register are cleared.

The DMA_CH[n]_Status register captures all the interrupt events of that Tx DMA and Rx DMA channel pair. The DMA_CH[n]_Interrupt_Enable register contains the corresponding enable bits for each of the interrupt event. There are two groups of interrupts in the DMA channel namely Normal and Abnormal interrupts. They are indicated by bits[15:14] of DMA_CH[n]_Status register respectively. The normal group is for events that happen during the normal transfer of packets (TI, RI, TBU) while the abnormal interrupt events are for error events

Interrupts are not queued. If the same interrupt event occurs again before the driver responds to the previous one, no additional interrupts are generated.

The common sbd_intr_o output signal is asserted for the transfer complete interrupts only when the corresponding interrupts are enabled in DMA_CH[n]_Interrupt_Enable register.

This module also supports the following per Channel Transfer Complete interrupt signals.

- sbd_perch_tx_intr_o[max_dma_ch] (Transmit Per Channel Interrupts)
- sbd_perch_rx_intr_o[max_dma_ch] (Receive Per Channel Interrupts)

The behavior of the RI/TI/ sbd_perch_tx_intr_o[]/sbd_perch_rx_intr_o[] changes depending on the settings of INTM field bit[17:16] in DMA_Mode. Following table explains the Transfer Complete Interrupt behavior.

Table 427. Transfer Complete Interrupt Behavior

Interrupt Mode	Behavior of the sbd_perch_tx_intr_o[] and sbd_perch_rx_intr_o[]	Behavior of TI/RI and sbd_intr_o
INTM=1	These signals reflect the value of corresponding TI/RI bits in DMA_CH[n]_Status register when the corresponding interrupt enable is set. Hence they are level signals and are cleared by the application by writing 1’b1 to the RI/TI status bits. This signal will not be asserted when the corresponding interrupt enable bit is not set.	The TI/RI is set as explained above. However, the sbd_intr_o signal and NIS status bit are not asserted for any RI/TI events.
INTM=2	In this mode, RI/TI interrupts are queued. These signals reflect the value of corresponding TI/RI bits inDMA_CH[n]_Status register when the corresponding interrupt enable is set. They are level signals and cleared by software by writing 1’b1 to the RI/TI status bits. However, it is set again if another TI/RI event(s) is detected before the TI/RI bits are cleared for the previous event.	The RI/TI status bits are set whenever the Transfer Complete event is detected and gets reset whenever software driver clears those bits by writing 1. However, if another Transfer Complete event is detected before it is cleared (serviced) by the software, then the module automatically set these status bits again. However, the sbd_intr_o signal is not generated based on TI/RI

43.4.14.1 PMT interrupts

The PMT interrupt signal is asserted when a valid remote wake-up packet is received. When software resets the PWRDWN bit in the remote wake-Up packet detection register, the MAC comes out of the power-down mode, but this event does not generate the PMT interrupt.

When you enable the power-down mode in the PMT block, the MAC drops all received packets and does not forward any packet to the MTL Rx FIFO or the application. The MAC comes out of the power-down mode only when a remote wake-up packet is received and the corresponding detection is enabled. The PMT block is available in the receive path of MAC.

43.5 External signals

NOTE

This section and all its sub-sections are Synopsys Proprietary. Used with permission.

Table 428. External signals

Port name	Interface	I/O	Description
1588_EVENT_AUX_IN	RMII	I	Auxiliary Timestamp Trigger When this signal goes high (rising edge), a snapshot of the system time is taken and stored in the auxiliary timestamp FIFO.
1588_EVENT_IN	RMII	I	Media Clock Generation Trigger Input Trigger (mcg_pst_trig_i) Input to ENET_QOS to capture presentation time. Based on presentation control value of PPS Control (MAC_PPS_CONTROL), this signal can be pulse or level signal.
1588_EVENT_OUT	RMII	O	Pulse Per Second Output (ptp_pps_o) This signal is high based on the PPS mode selected in PPS Control (MAC_PPS_CONTROL). When PPS is programmed in Media Clock Recovery mode, this port indicates the recovered clock. See Programming Guidelines for Flexible Pulse-Per-Second Output for details.
TX_CLK	RGMII	O for (RGMII)	RGMII: The transmit clock signal is always provided by the MAC on the TX_CLK line. Source-synchronous clocking is used: the clock signal that is output (by either the PHY or the MAC) is synchronous with the data signals.
RX_CLK	RGMII	I	RGMII: Received clock signal (recovered from incoming received data)
REF_CLK1	RMII	I/O	This 50Mhz clock used by the RMII interface. When RMII mode is selected, RX_CLK (25Mhz or 2.5Mhz) should be derived from this clock.
TX_EN	RMII, RGMII	O	The MAC drives this signal. This signal has multiple functions depending on the selected PHY interface. • RMII: When high, indicates that valid data is being transmitted on the phy_txd_o bus. Synchronous to REF_CLK.

Table continues on the next page...

Table 428. External signals (continued)

Port name	Interface	I/O	Description
			• RGMII: Contains TX_EN on the rising edge of TX_CLK, and a logical derivative of TX_EN and TX_ER (TX_EN XOR TX_ER) on the falling edge of TX_CLK.
TX_DATA	RMII, RGMII	O	This is a group of transmit data signals driven by the MAC. It has multiple functions depending on the selected PHY interface. • RMII: Bits[1:0] provide the RMII transmit data. The data is valid only when the TX_EN signal is high. Synchronous to REF_CLK. • RGMII: Bits[3:0] provide serial output Ethernet data
RX_DATA	RMII, RGMII	I	This is a group of data signals received from the PHY. It has multiple functions depending on the selected PHY interface. • RMII: Bits [1:0] provide the RMII receive data. The data is valid only when the RX_EN signal is high. Synchronous to REF_CLK. • RGMII: Bits [3:0] contains the Ethernet input data transferred from the PHY to the media-access controller.
MDC	RMII, RGMII	O	Management Data Clock. The MAC provides timing reference for the MDIO or MII through this periodic clock. This clock is generated from the application clock through a clock divider controlled by MAC_MDIO_ADDRESS[CR].
MDIO	RMII, RGMII	I/O	Transfers control information between the external PHY and the MAC. Data is synchronous to MDC. This signal is an input after reset.
RX_EN	RGMII	I	The PHY drives this signal. It has multiple functions depending on the selected PHY interface. • RMII: Contains the CRS and data valid information of the Receive interface. Synchronous to REF_CLK. In RGMII mode, contains receive data valid on the rising edge of RX_CLK, and a logical derivative of receive data valid and RX_ER (receive data valid XOR RX_ER) on the falling edge of RX_CLK.

43.6 Initialization

NOTE

This section and all its sub-sections are Synopsys Proprietary. Used with permission.

This section provides the instructions for initializing the DMA or MAC registers in the proper sequence.

43.6.1 Initializing DMA

Complete the following steps to initialize the DMA:

1. Provide a software reset. This resets all of the MAC internal registers and logic (bit-0 of DMA_Mode).

2. Wait for the completion of the reset process (poll bit 0 of the DMA_MODE, which is only cleared after the reset operation is completed).
3. Program the following fields to initialize the DMA_SYSBUS_MODE register:
    a. AAL
    b. Fixed burst or undefined burst
    c. Burst mode values in case of AHB bus interface, OSR_LMT in case of AXI bus interface.
    d. If fixed length value is enabled, select the maximum burst length possible on the AXI Bus (bits [7:1])
4. Create a descriptor list for transmit and receive. In addition, ensure that the descriptors are owned by DMA (set bit 31 of descriptor TDES3/RDES3). For more information about descriptors, see Descriptors
5. Program the Transmit and Receive Ring length registers (DMA_CH(#i)_TXDESC_RING_LENGTH (for i = 0; i <= 1) and DMA_CH#_RX_CONTROL2 (for i = 0; i <= 1)). The ring length programmed must be at least 4.

    NOTE

    The descriptor address from the start to the end of the ring must not cross the 4GB boundary.

6. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (DMA_CH(#i)_TXDESC_LIST_ADDRESS (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1), DMA_CH(#i)_RXDESC_LIST_ADDRESS (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)). Also, program transmit and receive tail pointer registers indicating to the DMA about the available descriptors (DMA_CH(#i)_TXDESC_TAIL_POINTER (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) and DMA_CH(#i)_RXDESC_TAIL_POINTER (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1))

    NOTE

    For 40-bit or 48-bit addressing mode, program the higher address List registers (DMA_CH[n]_TXDESC_LIST_HADDRESS, DMA_CH[n]_RXDESC_LIST_HADDRESS).

    The tailpointer registers must be advanced to the location immediately after the descriptors that are set, for the DMA to know that additional descriptors are available.

7. Program the settings of the following registers for the parameters like maximum burst-length (PBL) initiated by DMA, descriptor skip lengths, OSP in case of TxDMA, RBSZ in case of RxDMA, and so on:
    - DMA_CH(#i)_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)
    - DMA_CH(#i)_TX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)
    - DMA_CH(#i)_RX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)
8. Enable the interrupts by programming the DMA_CH(#i)_Interrupt_Enable (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register.
9. Start the Receive and Transmit DMAs by setting SR (bit 0) of the DMA_CH(#i)_RX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1) and ST (bit 0) of the DMA_CH(#i)_TX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register.
10. Repeat steps 4 to 9 for all the Tx DMA and Rx DMA channels selected in the hardware.

43.6.2 Initializing MTL Registers

The Transaction Layer (MTL) registers must be initialized to establish the transmit and receive operating modes and commands.

Complete the following steps to initialize the MTL registers:

1. Program the Tx Scheduling (SCHALG) and Receive Arbitration Algorithm (RAA) fields in MTL_Operation_Mode to initialize the MTL operation in case of multiple Tx and Rx queues.
2. Program the Receive Queue to DMA mapping in MTL_RxQ_DMA_Map0 and MTL_RxQ_DMA_Map1 registers.
3. Program the following fields to initialize the mode of operation in the MTL_TxQ0_Operation_Mode register.

a. Transmit Store And Forward (TSF) or Transmit Threshold Control (TTC) in case of threshold mode
b. Transmit Queue Enable (TXQEN) to value 2'b10 to enable Transmit Queue0
c. Transmit Queue Size (TQS)

4. Program the following fields to initialize the mode of operation in the MTL_RxQ0_Operation_Mode register:
   a. Receive Store and Forward (RSF) or RTC in case of Threshold mode
   b. Flow Control Activation and De-activation thresholds for MTL Receive FIFO (RFA and RFD)
   c. Error Packet and undersized good Packet forwarding enable (FEP and FUP)
   d. Receive Queue Size (RQS)
5. Repeat previous two steps for all MTL Tx and Rx queues selected in the configuration.

43.6.3 Initializing MAC

The MAC configuration registers establish the operating mode of the MAC. These registers must be initialized before initializing the DMA.

The following MAC Initialization operations can be performed after DMA initialization. If the MAC initialization is completed before the DMA is configured, enable the MAC receiver (last step in the following sequence) only after the DMA is active. Otherwise, received frames fill the Rx FIFO and overflow.

1. Provide the MAC address registers: MAC_ADDRESS0_HIGH and MAC_ADDRESS0_LOW. If more than one MAC address is enabled in your configuration, program the MAC addresses appropriately.
2. Program the following fields to set the appropriate filters for the incoming frames in the MAC_PACKET_FILTER register:
   a. Receive All
   b. Promiscuous mode
   c. Hash or Perfect Filter
   d. Unicast, multicast, broadcast, and control frames filter settings
3. Program the following fields for proper flow control in the MAC_Q0_TX_FLOW_CTRL register:
   a. Pause time and other Pause frame control bits
   b. Transmit Flow control bits
   c. Flow Control Busy
4. Program the MAC_INTERRUPT_ENABLE register, as required, and if applicable, for your configuration.
5. Program the appropriate fields in the MAC_CONFIGURATION register. For ex: Inter-packet gap while transmission and jabber disable.
6. Set bit 0 and 1 in MAC_CONFIGURATION register to start the MAC transmitter and receiver.

43.7 Application Information

NOTE

This section and all its sub-sections are Synopsys Proprietary. Used with permission.

43.7.1 Performing Normal Receive and Transmit Operation

During normal operation of the module, normal and transmit interrupts are read, descriptors polled, the DMA is suspended (if it does not own descriptors), and values of current host transmitter or receiver descriptor pointers are read for debugging.

For normal operation, complete the following steps:

1. For normal transmit and receive interrupts, read the interrupt status. Then, poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
2. Set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
3. If the descriptors are not owned by the DMA (or no descriptor is available), the DMA goes into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and writing the descriptor tail pointer to Tx/Rx tail pointer register (DMA_CH[n]_TXDESC_TAIL_POINTER and DMA_CH[n]_RXDESC_TAIL_POINTER).
4. The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (DMA_CH[n]_CURRENT_APP_TXDESC and DMA_CH[n]_CURRENT_APP_RXDESC).
5. The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (Register DMA_CH[n]_CURRENT_APP_TXBUFFER and DMA_CH[n]_CURRENT_APP_RXBUFFER).

43.7.2 Stopping and Starting Transmission

Complete the following steps to pause the transmission.

NOTE

The steps are provided for Channel 0.

1. Disable the Transmit DMA (if applicable) by clearing Bit 0 (ST) of DMA_CH(#i)_TX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) Register.
2. Wait for any previous frame transmissions to complete. You can check this by reading the appropriate bits of MTL_TXQ0_DEBUG Register (TRCSTS is not 01 and TXQSTS=0).
3. Disable the MAC transmitter and MAC receiver by clearing Bit (RE) and Bit 1(TE) of the MAC_Config-uration Register.
4. Disable the Receive DMA (if applicable), after making sure that the data in the Rx FIFO is transferred to the system memory (by reading the appropriate bits of MTL_TXQ0_DEBUG Register, PRXQ=0 and RXQSTS=00).
5. Make sure that both Tx Queue and Rx Queue are empty (TXQSTS is 0 in MTL_TXQ0_DEBUG Register and RXQSTS is 0 in MTL_RXQ0_DEBUG Register).
6. To restart the operation, first start the DMAs, and then enable the MAC Transmitter and Receiver.

NOTE

- Do not change the configuration (such as duplex mode, speed, port, or loop back) when the MAC is actively transmitting or receiving. These parameters are changed by software only when the MAC transmitter and receiver are not active.
- Similarly, do not change the DMA-related configuration when Transmit and Receive DMA are active.

43.7.3 Programming Guidelines for Switching to New Descriptor List in RxDMA

Switching to a new descriptor list is different in the Rx DMA compared to the Tx DMA. Switching to a new descriptor list is permitted when the RxDMA is in SUSPEND state, as clarified by the following points:

- Generally, RxDMA prepares the descriptors in advance.
- If the RxDMA goes to SUSPEND due to descriptors not being available, a major failure occurs (software is not able to free the filled-up descriptors/buffers). If this issue is not rectified immediately, frames will be lost because of an RxFIFO Overflow. Therefore, the software is allowed to create a new descriptor list and program the RxDMA to start using it immediately, without going into STOP state.

43.7.4 Programming Guidelines for Multi-Channel Multi-Queuing

43.7.4.1 Transmit

1. Program the Transmit queue size in the TQS field of MTL_TXQ[n]_OPERATION_MODE register. Based on the value programmed in the TQS field, the size of the queue is determined.

   In the Transmit operation, the number of channels is equal to the number of the queues. Due to this reason, the Channel-to-Queue mapping is fixed.

2. For a queue to be used, the queue must be enabled in TXQEN in the corresponding MTL_TXQ[n]_OPERATION_MODE Register. In DMA configurations, the ST bit of DMA_CH[n]_TX_CONTROL Register and corresponding TXQEN in MTL_TXQ[n]_OPERATION_MODE Register needs to be enabled.
3. The scheduling method needs to be programmed in SCHALG of MTL_OPERATION_MODE register.
4. Program the MTL_TXQ[n]_Quantum_Weight for DCB queue as per the selected algorithm. In case of CBS algorithm in AVB queues, the MTL_TXQ[n]_ETS_CONTROL, MTL_TXQ[n]_SENDSLOPECREDIT, MTL_TXQ[n]_HICREDIT and MTL_TXQ[n]_LOCREDIT registers also need to be programmed as required.
5. If DCB is enabled and PFC function is required, program MAC_TXQ_PRTY_MAP0 Register to assign a fixed priority to the queue. This priority assigned is used for determining if the corresponding queue should stop transmitting packet based on the received PFC packet.

43.7.4.2 Receive

1. Program the Receive queue size in the RQS field of MTL_RXQ[n]_OPERATION_MODE Register. Based on the value programmed in RQS field, the size of the queue is determined.
2. Enable the Receive Queues 0 to 4 in the fields RXQ0EN to RXQ4EN in MAC_RXQ_CTRL0 Register for AV or DCB. In DMA configurations, SR bit of statically or dynamically mapped DMA_CH[n]_RX_CONTROL Register and corresponding RXQ[n]_EN in MAC_RXQ_CTRL0 Register needs to be enabled.
3. The MAC routes the Rx packets to the Rx Queues based on following packet types:
   a. AV PTP Packets: Based on the programming of AVPTPQ in MAC_RXQ_CTRL1 Register.
   b. AV Untagged Control packets: Based on the programming of AVCPQ in MAC_RXQ_CTRL1 Register.
   c. Data Center Bridging (DCB) related Link Layer Discovery Protocol (LLDP) packets. Program DCBCPQ in MAC_RXQ_CTRL1 Register to indicate to MAC which queue should get the DCB packets.
   d. VLAN Tag Priority field in VLAN Tagged packets: Program PSRQ4-0 of the MAC_RXQ_CTRL2 and MAC_RXQ_CTRL3 Register for the routing of tagged packets based on the USP (user Priority) field of the received packets to the Rx Queues 0 to 4.
   e. The AV tagged control and data packets are also routed based on PSRQ field of the MAC_RXQ_CTRL2 and MAC_RXQ_CTRL3 registers.

   NOTE

   The priorities set in PSRQ4-0 should be unique.

4. If multiple RX DMA channels are enabled, the following programming should be done for proper arbitration and mapping:
   a. Program the RAA field of MTL_OPERATION_MODE register to select the arbitration algorithm to decide which RxQ is read out from the RxFIFO memory.
   b. Program the MTL_RXQ[n]_CONTROL to decide the weights and the packet arbitration for each RxQ.
   c. If static mapping is programmed in MTL_RXQ_DMA_MAP[n] register (RXQ[n]DADMACH is reset to 0), bits RXQx2DMA and others need to be programmed to select the channel for which each queue is mapped.
   d. Set RXQ[n]DADMACH bit in MTL_RXQ_DMA_MAP0 Register to select dynamic mapping of packets in each RxQueue.

e. In dynamic channel mapping, the routing of a packet to a specific RxDMA channel is decided by the value of DCS field in the lowest MAC Address Register.

43.7.4.2.1 Programming Guidelines for Recovering from DMA Channel Failure

When the DMA channel issues a bus error, follow these steps to recover from the failure.

Recovering from the Receive DMA Channel Failure

1. Set the RPF bit to 1. This flushes all the packets one after the other.

   This step is optional. However, setting this bit prevents HOL (head-of-line) blocking in the Rx queues when packets sent to the RXDMA are stopped due to bus error.
2. Reprogram the specific Registers of the DMA channel.
3. Start the DMA channel.

Recovering from the Transmit DMA Channel Failure

1. Stop the specific DMA channel, even if it is in active state.
2. Flush the corresponding MTL queue.
3. Reprogram the specific Registers of the DMA channel.
4. Start the DMA channel.

NOTE

Due to the known limitations in the design, reprogramming the DMA channel registers might not always be successful in recovering from a Bus Error. If the module is not fully functional after reprogramming the DMA, as a workaround, issue a Soft Reset to recover from the Bus Error.

43.7.5 Programming Guidelines for IEEE 1588 Timestamping

43.7.5.1 Initialization Guidelines for System Time Generation

You can enable the timestamp feature by setting Bit 0 of the MAC_TIMESTAMP_CONTROL Register. However, it is essential that the timestamp counter be initialized after this bit is set. Complete the following steps during the module initialization:

1. Mask the Timestamp Trigger interrupt by clearing the bit 16 of MAC_INTERRUPT_ENABLE Register.
2. Set Bit 0 of MAC_TIMESTAMP_CONTROL Register to enable timestamping.
3. Program MAC_SUB_SECOND_INCREMENT Register based on the PTP clock frequency.
4. If you are using the Fine Correction approach, program MAC_Timestamp_Addend and set Bit 5 of MAC_TIMESTAMP_CONTROL Register.
5. Poll the MAC_TIMESTAMP_CONTROL Register until Bit 5 is cleared.
6. Program Bit 1 of MAC_TIMESTAMP_CONTROL Register to select the Fine Update method (if required).
7. Program MAC_SYSTEM_TIME_SECONDS_UPDATE Register and MAC_SYSTEM_TIME_NANOSECONDS_UPDATE Register with the appropriate time value.
8. Set Bit 2 in MAC_TIMESTAMP_CONTROL Register.
9. The timestamp counter starts operation as soon as it is initialized with the value written in the Timestamp Update registers. If one-step timestamping is enabled
   a. To enable one-step timestamping, program Bit 27 of the TDES3 Context Descriptor.
   b. Program registers MAC_TIMESTAMP_INGRESS_ASYM_CORR and MAC_TIMESTAMP_EGRESS_ASYM_CORR to update the correction field in PDelay_Req PTP messages.
10. Enable the MAC receiver and transmitter for proper timestamping.

NOTE

If timestamp operation is disabled by clearing Bit 0 of MAC_TIMESTAMP_CONTROL Register, repeat all these steps to restart the timestamp operation.

43.7.5.2 System Time Correction

43.7.5.2.1 Coarse Correction Method

To synchronize or update the system time in one process (coarse correction method):

1. Set the offset (positive or negative) in the Timestamp Update registers(MAC_SYSTEM_TIME_SECONDS_UPDATE and MAC_SYSTEM_TIME_NANOSECONDS_UPDATE).
2. Set Bit 3 (TSUPDT) of MAC_TIMESTAMP_CONTROL REGISTER.

The value in the Timestamp Update registers is added to or subtracted from the system time when the TSUPDT bit is cleared.

43.7.5.2.2 Fine Correction Method

To synchronize or update the system time to reduce system-time jitter (fine correction method):

1. With the help of the algorithm, calculate the rate by which you want to make the system time increments slower or faster.
2. Update the MAC_TIMESTAMP_ADDEND with the new value and set Bit 5 of the MAC_TIMESTAMP_CONTROL Register.
3. Wait for the time for which you want the new value of the Addend register to be active. You can do this by enabling the Timestamp Trigger interrupt after the system time reaches the target value.
4. Program the required target time in MAC_PPS[n]_TARGET_TIME_SECONDS Register and MAC_PPS[n]_TARGET_TIME_NANOSECONDS Register.
5. Enable the Timestamp interrupt in bit 12 of MAC_INTERRUPT_ENABLE register.
6. Set bit 4 in Register MAC_Timestamp_Control.
7. When this trigger causes an interrupt, read MAC_INTERRUPT_STATUS Register.
8. Reprogram MAC_TIMESTAMP_ADDEND Register with the old value and set bit 5 again.

43.7.6 Programming Guidelines for AV Feature

After you enable the AV feature in the module controller, there are several programming tasks to be followed:

- Initializing the DMA for QOS-AHB configurations only
- Enabling slot number checking
- Enabling average bits per slot reporting
- Disabling flow control for AV-enabled queues (transmit and receive flow control)

43.7.6.1 Initializing the DMA in Audio Video Feature

The first step in programming the AV feature in a QOS-AHB configuration is to initialize the DMA.

Use this initialization sequence for QOS-AHB/QOS-AXI configurations with AV feature.

1. Provide a software reset to reset all QOS internal registers and logic (bit 0 in DMA_MODE register).
2. Wait for the completion of the reset process. Poll bit 0 of the DMA_MODE register), which is cleared only after the reset operation is completed.
3. Program the fields to initialize the DMA register by setting the values in DMA_MODE register.

4. Create a proper descriptor list for transmit and receive. In addition, ensure that the DMA owns the Transmit and Receive descriptors. When OSF mode is used, at least two TX descriptors are required.

   For more information about descriptors, see Descriptors .

5. Make sure that your software creates three or more different transmit or receive descriptors in the list before reusing any of the descriptors.
6. Program the Transmit and Receive Ring length registers (DMA_CH(#i)_TXDESC_RING_LENGTH (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) and DMA_CH#_RX_CONTROL2 (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)). The ring length programmed must be at least 4.
7. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (DMA_CH(#i)_TXDESC_LIST_ADDRESS (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1), DMA_CH(#i)_TXDESC_LIST_ADDRESS (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)). In addition, you must program the Transmit and Receive tail pointer registers indicating to the DMA about the available descriptors (DMA_CH(#i)_TXDESC_TAIL_POINTER (for i = 0; i <= DWC_EQOS_NUM_D-MA_TX_CH-1) and DMA_CH(#i)_RXDESC_TAIL_POINTER (for i = 0; i <= DWC_EQOS_NUM_D-MA_RX_CH-1)).
8. Program the following fields to initialize the mode of operation in the MTL_TXQ0_OPERATION_MODE register:
   a. Transmit Store And Forward (TSF)
   b. Transmit Threshold Control (TTC)
   c. Transmit Queue Enable (TXQEN) to value 2‘b10 to enable Transmit Queue0
   d. Transmit Queue Size (TQS)
9. Enable the interrupts by programming the DMA_CH(#i)_INTERRUPT_ENABLE (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register.
10. Repeat steps 4 through 9 for all additional channels of AV feature.
11. Program the CBS control register, idleSlope, sendSlope, hiCredit, and loCredit registers of the AV Queues.
12. Start the Receive and Transmit DMA by setting bit 0 of the DMA_CH(#i)_TX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register and bit 0 of the DMA_CH(#i)_RX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1) register.

43.7.6.2 Enabling Slot Number Checking

You can enable slot number checking to specify the intervals at which the DMA channels mapped to AV Queues fetch the frames from the AHB/AXI system bus.

These steps should be completed after step 11 and before step 12 of "Initializing DMA".

You can use the slot number check feature to specify the intervals at which the DMA Channels mapped to AV Queues fetch the frames from the AHB/AXI system bus. This feature is useful for a uniform and periodic transfer of the AV traffic from the host memory. The feature is available only when you enable timestamping and program the MAC_SUB_SECOND_INCREMENT register. To enable the slot number checking:

1. Enable timestamping by following the steps described in "Initialization Guidelines for System Time Generation".
2. Make sure that the SLOTNUM field (bits 22:19) of TDES3 Normal Descriptor (Read Format) contains a valid slot number. You can read the current reference slot number from the DMA_CH(#i)_Slot_Function_Control_Status (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1).
3. Set Bit 0 (ESC) of the Slot Function Control and Status register of a channel to enable the slot number checking.

43.7.6.3 Enabling Average Bits Per Slot Reporting

You can enable the reporting of the average bits that are transmitted in a slot.

The CBS Status register of the additional AV channels provides information about the average bits that are transmitted in a slot. The software can asynchronously read this register to retrieve information about the average bits transmitted per slot. To enable average bits per slot reporting:

1. Enable timestamping by following the steps described in the Initialization Guidelines for System Time Generation .
2. Program Bits [6:4], SLC, of the MTL_TXQ[n]_ETS_CONTROL register of a channel with the number of slots over which the average transmitted bits per slot need to be computed.
3. Enable Bit 9(ABPSSIE) of the MTL_Q[n]_INTERRUPT_CONTROL_STATUS register of a channel to generate the average bits per slot interrupt.

NOTE

- The frequency of this interrupt depends on the value programmed in step 2. For example, when you program value 0 in the SLC field, the interrupt is generated at every 125 microsecond.
- When not required, you can disable this interrupt to stop the interrupt flooding.

4. Read Bits [16:0], ABS, from the MTL_TXQ[n]_ETS_STATUS register of a channel on each interrupt.

NOTE

The software can read the ABS bits in polling mode even if the ABPSIE bit is not enabled. When high, bit 1 (ABPSIS) of the MTL_TXQ[n]_ETS_STATUS register indicates that a new value is updated in the ABS field.

43.7.6.4 Disabling Flow Control for AV Enabled Queues

43.7.6.4.1 Transmit Flow

Program the EHFC (Enable Hardware Flow Control) bit of corresponding Rx Queue's MTL_RXQ[n]_OPERATION_MODE register to 0.

43.7.6.4.2 Receive Flow Control

Program the PSTQ[n] field corresponding to AV enabled Tx Queue in MAC_TXQ_PRTY_MAP0/1 register to 0.

43.7.6.5 Programming Guidelines for Flexible Pulse-Per-Second Output

After you enable the Flexible Pulse-Per-Second Output feature in the module controller, you can perform the following tasks:

- Generating Single Pulse on PPS
- Generating Next Pulse on PPS
- Generating a Pulse Train on PPS
- Generating an Interrupt without Affecting the PPS

43.7.6.5.1 Generating Single Pulse on PPS

To generate single pulse on PPS:

1. Program 11 or 10 (for interrupt) in Bits [6:5], TRGTMODSEL, of MAC_PPS_CONTROL Register.
   This instructs the MAC to use the Target Time registers (MAC_PPS0_TARGET_TIME_SECONDS and MAC_PPS0_TARGET_TIME_NANOSECONDS) for start time of PPS signal output.
2. Program the start time value in the Target Time registers (MAC_PPS0_TARGET_TIME_SECONDS and MAC_PPS0_TARGET_TIME_NANOSECONDS).
3. Program the width of the PPS signal output in MAC_PPS(#i)_WIDTH (for i = 0; i <= 3) Register.
4. Program Bits [3:0], PPSCMD, of MAC_PPS_CONTROL to 0001. This instructs the MAC to generate single pulse on the PPS signal output at the time programmed in the Target Time registers.

43.7.6.5.2 Generating Next Pulse on PPS

When the PPSCMD is executed (PPSCMD bits = 0), you can cancel the pulse generation by giving the Cancel Start Command (PPSCMD=0011) before the programmed start time elapses. You can also program the behavior of the next pulse in advance.

To program the next pulse:

1. Program the start time for the next pulse in the Target Time registers. This time should be more than the time at which the falling edge occurs for the previous pulse.
2. Program the width of the next PPS signal output in MAC_PPS(#i)_WIDTH (for i = 0; i <= DWC_EQOS_PPS_OUT_NUM-1).
3. Program Bits [3:0], PPSCMD, of MAC_PPS_CONTROL to generate a single pulse after the time at which the previous pulse is de-asserted. This instructs the MAC to generate single pulse on the PPS signal output, at the time programmed in Target Time registers.

   If you give this command before the previous pulse becomes low, then the new command overwrites the previous command and the QOS may generate only 1 extended pulse.

43.7.6.5.3 Generating a Pulse Train on PPS

To generate a pulse train on PPS:

1. Program 11 or 10 (for interrupt) in Bits [6:5], TRGTMODSEL, of MAC_PPS_Control Register. This instructs the MAC to use the Target Time registers for start time of the PPS signal output.
2. Program the start time value in the Target Time registers.
3. Program the interval value between the train of pulses on the PPS signal output in MAC_PPS(#i)_INTERVAL (for i = 0; i <= DWC_EQOS_PPS_OUT_NUM-1) Register.
4. Program the width of the PPS signal output in MAC_PPS(#i)_WIDTH (for i = 0; i <= DWC_EQOS_PPS_OUT_NUM-1) Register.
5. Program Bits[3:0], PPSCMD, of MAC_PPS_CONTROL Register to 0010. This instructs the MAC to generate train of pulses on the PPS signal output with start time programmed in Target Time registers.

   By default, the PPS pulse train is free-running unless stopped by ‘STOP Pulse train at time’ or ‘STOP Pulse Train immediately’ commands.
6. Program the stop value in the Target Time registers. Ensure that Bit 31 (TSTRBUSY) of MAC_PPS(#i)_TARGET_TIME_NANOSECONDS (for i = 0; i <= DWC_EQOS_PPS_OUT_NUM-1) Register is reset before programming the Target Time registers again.
7. Program the PPSCMD field (bit 3:0) of MAC_PPS_CONTROL to 0100. This stops the train of pulses on PPS signal output after the programmed stop time specified in Step 6 elapses.

   You can stop the pulse train at any time by programming 0101 in the PPSCMD field. Similarly, you can cancel the Stop Pulse train command (given in Step 7) by programming 0110 in the PPSCMD field before the time (programmed in Step 6) elapses. You can cancel the pulse train generation by programming 0011 in the PPSCMD field before the programmed start time (in Step 2) elapses.

43.7.6.5.4 Generating an Interrupt without Affecting the PPS

You can use bits[6:5], TRGTMODSEL, of the MAC_PPS_CONTROL Register to program the Target Time registers to do any one of the following:

- Generate only interrupts.
- Generate interrupts and the PPS start and stop time.
- Generate only PPS start and stop time.

To program the Target Time registers to generate only interrupt events:

1. Program 00 (for interrupt) in Bits [6:5], TRGTMODSEL, of MAC_PPS_CONTROL Register. This instructs the MAC to use the Target Time registers for target time interrupt.
2. Program a target time value in the Target Time registers. This instructs the MAC to generate an interrupt when the target time elapses.

   If Bits [6:5], TRGTMODSEL, are changed (for example, to control the PPS), then the interrupt generation is overwritten with the new mode and new programmed Target Time register value.

43.7.6.6 Programming Guidelines for VLAN Filtering on Receive

To program VLAN filtering on receive:

1. Program MAC_VLAN_TAG register for the following bit to select the filtering method:
   a. ETV: Enable 12-Bit VLAN Tag Comparison or 16-bit VLAN Tag comparison.
   b. VTHM: VLAN Tag Hash Table Match Enable.
   c. ERIVLT: Enable inner VLAN Tag or outer VLAN Tag (to enable the inner or outer VLAN Tag filtering, Double VLAN Processing should enabled by setting EDVLP)
   d. ERSVLM: Enable Receive S-VLAN Match or C-VLAN match (for S-VLAN processing to be enabled, set ESVL)
   e. DOVLTC: Ignores VLAN Type for Tag Match
   f. VTIM: to enable VLAN Tag Inverse Match instead of the normal VLAN Tag matching
2. Program VL of MAC_VLAN_Tag Register for the 12-bit or 16-bit VLAN tag.
3. If Hash filtering of VLAN tag is enabled, program MAC_VLAN_HASH_TABLE Register. When ETV bit is reset, upper 4 bits of the calculated CRC-32 of VLAN Tag are inverted and used to index the content of MAC_VLAN_Hash_Table register. When ETV bit is set, upper 4 bits of calculated CRC-32 of VLAN Tag are used to index the content of MAC_VLAN_HASH_TABLE register. For example, when ETV bit is set, a hash value of 4b'1000 selects Bit 8 of the VLAN Hash table. When ETV bit is reset a hash value of 4'b1000 selects Bit 7 of the VLAN Hash table.

43.7.6.7 Programming Guidelines for Extended VLAN Filtering and Routing on Receive

For the indirect access of the per VLAN Tag registers, follow these steps.

- Write
  1. Write the required data into the MAC VLAN Tag Data register.
  2. Program the VLAN Tag Control Register's OFS field with the required Filter Register's offset and command type to the CT field. For a write command, set this bit to 0.
  3. Write 1 to the OB field and wait till the OB bit is reset to do the next write. This will guarantee that the appropriate VLAN Tag Filter Register has been programmed.
- Read
  1. Program the VLAN Tag Control Register's OFS Field with the required register's offset and command type to the CT field. For a read command, set this bit to 1.
  2. Write 1 to the OB field and wait till the OB bit is reset. The appropriate VLAN Tag Filter register's value is available in the MAC VLAN Tag Data Register.

43.7.6.8 Programming Sequence for Queue/Channel Based VLAN Inclusion Register

To program the queue/channel-based VLAN inclusion register:

NOTE

Indirect VLAN include registers are not accessed while setting the CBTI bit.

1. Set the CBTI bit of MAC_VLAN_INCL register to 1, to enable queue/channel based VLAN Tag insertion on all transmitted packets. This bit must be set before any indirect access to the queue/channel specific MAC_VLAN_INCL(#i) register.
2. Program the VLAN Tag and VLAN Type to be inserted in packets from a particular queue/channel in VLT and CSVL fields of MAC_VLAN_INCL register; corresponding offset address in ADDR field (0 for queue/channel 0, 1 for queue/channel 1, and so on) must be set. Set the RDWR bit to 0 to indicate write access. The write to byte 0 (byte 3 in Big Endian mode) of MAC_VLAN_INCL register initiates access to indirect access MAC_VLAN_INCL(#i) register.
3. The BUSY bit of MAC_VLAN_INCL register is set by the module to indicate the progress of access to indirect access MAC_VLAN_INCL(#i) register. On completion of the access, the BUSY bit is cleared. The application must not attempt subsequent access to MAC_VLAN_INCL(#i) register when the BUSY bit is 1.
4. Repeat step 2 and step 3 to program VLAN Tag and VLAN Type to be inserted in packets from the remaining queues/channels. The application must ensure that the required VLAN Tag and VLAN Type for all the queues/channels are programmed; otherwise unintended VLAN Tag and VLAN Type might be inserted.

43.7.7 Programming Guidelines for EST

Program the gate control values and time intervals in the Software Owned Gate Control List (SWOL) along with the other EST related registers that are described in ENET_QOS register descriptions to appropriate values. The following sub-sections provide step by step details for programming the GCL and the other EST related registers.

43.7.7.1 Programming the GCL and GCL Linked Registers

To program the Gate Control List (GCL) and the four other registers implemented per GCL:

1. The GCL and the four other GCL-linked registers should be accessed via indirect addressing using the MTL_EST_GCL_CONTROL and MTL_EST_GCL_DATA registers. The SWOL bit of the MTL_EST_STATUS register indicates if GCL0 or GCL1 is owned by software.
2. To program the GCL, write the 32-bit write data to MTL_EST_GCL_DATA register. Then program TL_EST_GCL_CONTROL register to write the write address and other control information.
3. In the MTL_EST_GCL_DATA register, Write Data consists of up to 8 bits (5 configurable, 3 reserved) of Gate Controls and up to 24 bits (configurable) of Time Interval. Gate Close is indicated by programming a '0' and Gate Open is indicated by programming a '1'. For a 4-TC and 20-bit Time Interval Configuration, the data width is 24-bits and the remaining 8-bits are reserved/read-only. The data should be written in the following format.

   {8'h0, TC3, TC2, TC1, TC0, 20-bit Time Interval} where TCx = 0 or 1.
4. MTL_EST_GCL_CONTROL register: Program the SRWO bit to 1 (to start a Write Op) and program the address and R/W fields appropriately.
5. Poll and check for the SRWO bit to be cleared by hardware to indicate the completion of the previous operation before initiating a new R/W operation via the same indirect addressing mode.
6. Repeat steps 3, 4, 5 until the programming of the GCL is completed.
7. Using the same indirect addressing method described above, program the BTR, CTR, TER and LLR registers. Set the GCRR bit in the MTL_EST_GCL_CONTROL register appropriately. The GCRR bit interprets the address field as belonging to these registers (instead of the GCL).

After programming of the GCL and the related registers, program the MTL_EST_CONTROL register to allow hardware to own and process the GCL.When the List Length (as indicated in LLR) is 1, the associated Time Interval should be smaller than the value of the Cycle Time Register. Otherwise, an error is reported (as detailed in the Error handling section) as a single set of gate controls add no value in the TSN context.

NOTE

The time unit in all the GCL related registers is seconds and nanoseconds. In cases where internally generated PTP System Time is used, the nanoseconds field must be programmed to use the Digital Rollover mode. (TSCTRLSSR field of MAC_TIMESTAMP_CONTROL must be set to value '1')

43.7.7.2 Programming the EST Registers

After completing the steps mentioned in "Programming the GCL and GCL Linked Registers" , program the MTL_EST_CONTROL register.

1. Set the Current Time Offset Value and Time Internal Left Shift fields of the MTL_EST_CONTROL register appropriately. Also, set the Enable EST (EEST) and Switch to SWOL (SWOL) bits.

   NOTE

   The CTOV recommendations for GMII for SPRAM configurations are:

   - 96 * Tx clock period, for 32 and 64 bit data width configurations.
   - 128 * Tx clock period for 128 bit data width configurations.

   The CTOV recommendations for GMII for non-SPRAM configurations are:

   - 30 * Tx clock period, when SA/VLAN insertion is enabled.
   - 22 * Tx clock period, when SA/VLAN insertion is not enabled.

2. This will enable the hardware to own and process the new GCL and make a switch to the new GCL at the BTR value. The hardware provides an interrupt (if enabled) when the switch to the new list happens.
3. Appropriate action has to be taken by software to address any other interrupts (explained in the Inter-rupts section) received during the hardware execution of the GCL.

Set the SSWL (Switch to Software Owned List) bit in the MTL_EST_CONTROL register to handoff to hardware. Software is not allowed to write to the GCL and GCL linked registers when the SSWL bit is set, because the new GCL might be in use by the hardware.

The SSWL is reset/cleared by the hardware when it successfully switches to the new list. The hardware also flips the SWOL bit to indicate the new GCL that the software owns.

To install a new GCL, program the GCL it owns (indicated by SWOL bit) as described in "Programming the GCL and GCL Linked Registers" and then program the MTL_EST_CONTROL register as described above. Ensure that the new BTR is set to an appropriate value to avoid BTRError that might need software intervention in some cases.

To avoid transmission overruns, the packet length (frame size) information should be available at all times. Therefore, in the DMA configurations, program the packet length in the first descriptor of every Tx frame. Similarly, in the MTL configuration, provide the packet length in the control word.

NOTE

To avoid transmission overruns, the Packet Length provided in the Transmit Descriptor should account for the SA and VLAN insertion, if applicable, that is, Packet Length should represent the actual packet length on the Ethernet line.

43.7.8 Programming the Launch Time in Time-Based Scheduling

The Launch Time is programmed in the Enhanced Normal Transmit Descriptors in DMA configurations and is driven as a control word in MTL configurations as follows:

The OSTC and Launch Time features are mutually exclusive and should not be used together; in case of a conflict (if OSTC = LTV = 1 in MTL configuration), priority is given to OSTC and the Launch Time is ignored.

In DMA configuration, if a context descriptor is received with a valid OSTC values immediately before receiving a first normal descriptor with LTV bit set, then the LTV is ignored.

43.7.9 Programming Guidelines for Media Clock Generation and Recovery

43.7.9.1 Programming Guidelines for Media Clock Generation

1. To set the PPS instance in Media Clock generation mode, program the appropriate Presentation Time Control (supported generation modes "1001-1011") to PPSCTRL_PPSCMD (for 0th instance)/PPSCMD#i (for 1,2,3 instances) of MAC_PPS_CONTROL register.
2. Based on the selected PPS instance, application must drive the appropriate trigger signal to the corresponding mcg_pst_trig_i[#i].
3. Based on the programmed mode, DUT captures the timestamp and programs it in the MAC_PPS(#i)_TARGET_TIME_SECONDS register. Then, mcgr_dma_req_o[#i] is set.
4. For every request generated by the DUT, the module reads the corresponding MAC_PPS(#i)_TARGET_TIME_SECONDS register and acknowledge the read request by asserting the corresponding mcgr_dma_ack_i[#i].

43.7.9.2 Programming Guidelines for Media Clock Recovery

1. Enable the Current Presentation Time (CPT) Counter by setting the PTGE field of MAC_TIMESTAMP_CONTROL register. In addition to programming the initialization values for the System Time, update the MAC_PRESN_TIME_UPDT register with the equivalent Presentation Time init value. Only then the TSINIT field of MAC_TIMESTAMP_CONTROL register is set.
2. The Increment values used for the System time is also used for the Current Presentation time. This is because, the increment value is in Sub-seconds and Sub-nanoseconds.
3. When an update is needed, along with programming the update values for the System Time, the MAC_PRESN_TIME_UPDT register should be updated with the equivalent Presentation Time update value (32-bit ns). Finally, TSUPDT field of MAC_TIMESTAMP_CONTROL register must be set for the update to happen.
4. Program the MCGREN#i of MAC_PPS_CONTROL register to enable the PPS instance to operate in MCGR mode.
5. To set the PPS instance in Media Clock recovery mode, program the appropriate Presentation Time Control (supported recovery modes "0001 - 0011") to PPSCTRL_PPSCMD (for 0th instance)/PPSCMD#i (for 1,2,3 instances) of MAC_PPS_Control register.
6. After PPS instance is set in recovery mode, corresponding mcgr_dma_req_o[#i] is set by the DUT. For every request generated by the DUT, program the target presentation time into the MAC_PPS(#i)_TARGET_TIME_SECONDS register and acknowledge the request by asserting the corresponding mcgr_d-ma_ack_i[#i]. Acknowledgment can be given before/after programming the Target presentation time.
7. Observe the recovered media clock on the corresponding ptp_pps_o[#i].

43.7.10 Programming Guidelines for ECC Protection for Memories

1. Enable the ECC feature for the respective memory by setting the appropriate bit in MTL_ECC_CONTROL register.
2. A correctable interrupt (sbd_sfty_ce_intr_o) or an uncorrectable interrupt (sbd_sfty_ue_intr_o) are generated to indicate to the application if any correctable/uncorrectable/address errors are detected. Appropriate status is indicated in the DMA/MTL_ECC_INTERRUPT_STATUS register.
3. Debug mode is provided for each memory to specify errors.

NOTE

Enable the ECC feature before the traffic is online (or after the reset), otherwise false interrupts might be triggered.

43.7.10.1 Programming Guidelines for ECC Hardware Error Injection (Debug Mode)

1. Enable the ECC error injection feature for MTL TX/RX and DMA TSO, by setting the appropriate bit in the MTL_DBG_CTL register.

2. Enable the ECC error injection feature for the EST memory by setting the appropriate bit in the EST_GCL_CONTROL register.
3. Enable the ECC error injection for RX Parser memory by setting the appropriate bit in the MTL_RXP_INDIRECT_ACC_CONTROL_STATUS register. To access Rx Parser memory in debug mode, disable the Rx Parser feature by setting the FRPE bit of MTL_OPERATION_MODE register to 0.
4. Where multiple CSR writes are required, for writing single data word into the memory, the application should ensure that all the CSR writes corresponding to one memory write should have the same value for the error injection control word.

43.7.11 Programming Guidelines for On-Chip Datapath Parity Protection

Follow the below steps for on-chip datapath parity protection:

1. Set the EDPP bit of MTL_DPP_CONTROL register to enable the generation of parity and detection of parity Mismatch on datapath.
2. An uncorrectable interrupt (sbd_sfty_ue_intr_o) is generated to indicate to the application whether any parity Mismatches are detected and the appropriate status will be indicated in DMA/MTL_SAFETY_INTERRUPT_STATUS and MAC_DPP_FSM_INTERRUPT_STATUS register.
3. Debug mode is supported for each parity generator, to insert error.

NOTE

To avoid false safety interrupts, enable the datapath parity protection before the start of Rx or Tx traffic.

43.7.12 Programming Guidelines for FSM Parity and Timeout

To program the FSM parity and timeout:

1. Set the PRTYEN and TMOUTEN fields of MAC_FSM_CONTROL register to enable the FSM parity and timeout feature respectively.
2. For parity error detection, force the FSMs to enter into a state with even number of 1s and wait for the safety uncorrectable interrupt to be set with FSMPES of MAC_DPP_FSM_INTERRUPT_STATUS.
3. For Error Injection mode in FSM parity, select the appropriate clock domain for which parity error injection should be set using the [23:16] bits of MAC_FSM_CONTROL register. Wait for the safety uncorrectable interrupt to be set with FSMPES of MAC_DPP_FSM_INTERRUPT_STATUS.
4. For FSM Timeouts, program the large and normal mode values indicated by LTMRMD and NTMRMD of MAC_FSM_ACT_TIMER register.
5. Select large or normal mode tic generation per clock domain by using the [31:24] bits of MAC_FSM_CONTROL Register.
6. Set the TMR field of the MAC_ACT_TIMER field with the appropriate number of CSR clock cycles used to generate 1-microsecond tic.
7. Force the FSM in a particular clock domain to be in an active state for Timeout duration (for example, T-2T). Check for the safety interrupt and the status bit for the corresponding clock domain getting set in [15:8] bits of MAC_DPP_FSM_INTERRUPT_STATUS register.
8. For Error Injection mode in FSM timeout, select the appropriate clock domain for which timeout error injection must be set using the [15:8] bits of MAC_FSM_CONTROL register. Wait for the safety uncorrectable interrupt along with the corresponding error status set in MAC_DPP_FSM_INTERRUPT_STATUS.
9. For application/CSR interface timeout, only normal mode tic generation is used. Based on the timeouts of the configured interface, appropriate fields (MSTTES, SLVTES) of MAC_DPP_FSM_INTERRUPT_STATUS are set along with the safety interrupt.

43.8 Memory Map and register definition

NOTE

This section and all its sub-sections are Synopsys Proprietary. Used with permission.

This section includes the ENET module memory map and detailed descriptions of all registers.

43.8.1 ENET_QOS register descriptions

43.8.1.1 ENET_QOS memory map

ENET_QOS_TSN base address: 428A_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	MAC Configuration Register (MAC_CONFIGURATION)	32	RW	0000_0000
4	MAC Extended Configuration Register (MAC_EXT_CONFIGURATION)	32	RW	0000_0000
8	MAC Packet Filter (MAC_PACKET_FILTER)	32	RW	0000_0000
C	Watchdog Timeout (MAC_WATCHDOG_TIMEOUT)	32	RW	0000_0000
10	MAC Hash Table Register 0 (MAC_HASH_TABLE_REG0)	32	RW	0000_0000
14	MAC Hash Table Register 1 (MAC_HASH_TABLE_REG1)	32	RW	0000_0000
50	MAC VLAN Tag Control (MAC_VLAN_TAG_CTRL)	32	RW	0000_0000
54	MAC VLAN Tag Data (MAC_VLAN_TAG_DATA)	32	RW	0000_0000
58	MAC VLAN Hash Table (MAC_VLAN_HASH_TABLE)	32	RW	0000_0000
60	VLAN Tag Inclusion or Replacement (MAC_VLAN_INCL)	32	RW	0000_0000
64	MAC Inner VLAN Tag Inclusion or Replacement (MAC_INNER_VLAN_INCL)	32	RW	0000_0000
70	MAC Q0 Tx Flow Control (MAC_Q0_TX_FLOW_CTRL)	32	RW	0000_0000
74	MAC Q1 Tx Flow Control (MAC_Q1_TX_FLOW_CTRL)	32	RW	0000_0000
78	MAC Q2 Tx Flow Control (MAC_Q2_TX_FLOW_CTRL)	32	RW	0000_0000
7C	MAC Q3 Tx Flow Control (MAC_Q3_TX_FLOW_CTRL)	32	RW	0000_0000
80	MAC Q4 Tx Flow Control (MAC_Q4_TX_FLOW_CTRL)	32	RW	0000_0000
90	MAC Rx Flow Control (MAC_RX_FLOW_CTRL)	32	RW	0000_0000
94	Receive Queue Control 4 (MAC_RXQ_CTRL4)	32	RW	0000_0000
98	Transmit Queue Priority Mapping 0 (MAC_TXQ_PRTY_MAP0)	32	RW	0000_0000
9C	Transmit Queue Priority Mapping 1 (MAC_TXQ_PRTY_MAP1)	32	RW	0000_0000
A0	Receive Queue Control 0 (MAC_RXQ_CTRL0)	32	RW	0000_0000
A4	Receive Queue Control 1 (MAC_RXQ_CTRL1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
A8	Receive Queue Control 2 (MAC_RXQ_CTRL2)	32	RW	0000_0000
AC	Receive Queue Control 3 (MAC_RXQ_CTRL3)	32	RW	0000_0000
B0	Interrupt Status (MAC_INTERRUPT_STATUS)	32	R	0000_0000
B4	Interrupt Enable (MAC_INTERRUPT_ENABLE)	32	RW	0000_0000
B8	Receive Transmit Status (MAC_RX_TX_STATUS)	32	R	0000_0000
C0	PMT Control and Status (MAC_PMT_CONTROL_STATUS)	32	RW	0000_0000
C4	Remote Wakeup Filter (MAC_RWK_PACKET_FILTER)	32	RW	0000_0000
D0	LPI Control and Status (MAC_LPI_CONTROL_STATUS)	32	RW	0000_0000
D4	LPI Timers Control (MAC_LPI_TIMERS_CONTROL)	32	RW	03E8_0000
D8	Tx LPI Entry Timer Control (MAC_LPI_ENTRY_TIMER)	32	RW	0000_0000
DC	One-microsecond Reference Timer (MAC_ONEUS_TIC_COUNTER)	32	RW	0000_0063
F8	PHY Interface Control and Status (MAC_PHYIF_CONTROL_STATUS)	32	RW	0000_0000
110	MAC Version (MAC_VERSION)	32	R	0000_1052
114	MAC Debug (MAC_DEBUG)	32	R	0000_0000
11C	Optional Features or Functions 0 (MAC_HW_FEATURE0)	32	R	0EFD_71F7
120	Optional Features or Functions 1 (MAC_HW_FEATURE1)	32	R	4199_39A6
124	Optional Features or Functions 2 (MAC_HW_FEATURE2)	32	R	4410_4104
128	Optional Features or Functions 3 (MAC_HW_FEATURE3)	32	R	0C39_5632
200	MDIO Address (MAC_MDIO_ADDRESS)	32	RW	0000_0000
204	MAC MDIO Data (MAC_MDIO_DATA)	32	RW	0000_0000
230	CSR Software Control (MAC_CSR_SW_CTRL)	32	RW	0000_0000
234	Frame Preemption Control (MAC_FPE_CTRL_STS)	32	RW	0000_0000
240	32-bit Binary Rollover Equivalent Time (MAC_PRESN_TIME_NS)	32	R	0000_0000
244	MAC 1722 Presentation Time (MAC_PRESN_TIME_UPDT)	32	RW	0000_0000
300	MAC Address0 High (MAC_ADDRESS0_HIGH)	32	RW	8000_FFFF
304	MAC Address0 Low (MAC_ADDRESS0_LOW)	32	RW	FFFF_FFFF
308	MAC Address1 High (MAC_ADDRESS1_HIGH)	32	RW	0000_FFFF
30C	MAC Address1 Low (MAC_ADDRESS1_LOW)	32	RW	FFFF_FFFF
310	MAC Address2 High (MAC_ADDRESS2_HIGH)	32	RW	0000_FFFF
314	MAC Address2 Low (MAC_ADDRESS2_LOW)	32	RW	FFFF_FFFF

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
318	MAC Address3 High (MAC_ADDRESS3_HIGH)	32	RW	0000_FFFF
31C	MAC Address3 Low (MAC_ADDRESS3_LOW)	32	RW	FFFF_FFFF
320	MAC Address4 High (MAC_ADDRESS4_HIGH)	32	RW	0000_FFFF
324	MAC Address4 Low (MAC_ADDRESS4_LOW)	32	RW	FFFF_FFFF
328	MAC Address5 High (MAC_ADDRESS5_HIGH)	32	RW	0000_FFFF
32C	MAC Address5 Low (MAC_ADDRESS5_LOW)	32	RW	FFFF_FFFF
330	MAC Address6 High (MAC_ADDRESS6_HIGH)	32	RW	0000_FFFF
334	MAC Address6 Low (MAC_ADDRESS6_LOW)	32	RW	FFFF_FFFF
338	MAC Address7 High (MAC_ADDRESS7_HIGH)	32	RW	0000_FFFF
33C	MAC Address7 Low (MAC_ADDRESS7_LOW)	32	RW	FFFF_FFFF
340	MAC Address8 High (MAC_ADDRESS8_HIGH)	32	RW	0000_FFFF
344	MAC Address8 Low (MAC_ADDRESS8_LOW)	32	RW	FFFF_FFFF
348	MAC Address9 High (MAC_ADDRESS9_HIGH)	32	RW	0000_FFFF
34C	MAC Address9 Low (MAC_ADDRESS9_LOW)	32	RW	FFFF_FFFF
350	MAC Address10 High (MAC_ADDRESS10_HIGH)	32	RW	0000_FFFF
354	MAC Address10 Low (MAC_ADDRESS10_LOW)	32	RW	FFFF_FFFF
358	MAC Address11 High (MAC_ADDRESS11_HIGH)	32	RW	0000_FFFF
35C	MAC Address11 Low (MAC_ADDRESS11_LOW)	32	RW	FFFF_FFFF
360	MAC Address12 High (MAC_ADDRESS12_HIGH)	32	RW	0000_FFFF
364	MAC Address12 Low (MAC_ADDRESS12_LOW)	32	RW	FFFF_FFFF
368	MAC Address13 High (MAC_ADDRESS13_HIGH)	32	RW	0000_FFFF
36C	MAC Address13 Low (MAC_ADDRESS13_LOW)	32	RW	FFFF_FFFF
370	MAC Address14 High (MAC_ADDRESS14_HIGH)	32	RW	0000_FFFF
374	MAC Address14 Low (MAC_ADDRESS14_LOW)	32	RW	FFFF_FFFF
378	MAC Address15 High (MAC_ADDRESS15_HIGH)	32	RW	0000_FFFF
37C	MAC Address15 Low (MAC_ADDRESS15_LOW)	32	RW	FFFF_FFFF
380	MAC Address16 High (MAC_ADDRESS16_HIGH)	32	RW	0000_FFFF
384	MAC Address16 Low (MAC_ADDRESS16_LOW)	32	RW	FFFF_FFFF
388	MAC Address17 High (MAC_ADDRESS17_HIGH)	32	RW	0000_FFFF
38C	MAC Address17 Low (MAC_ADDRESS17_LOW)	32	RW	FFFF_FFFF
390	MAC Address18 High (MAC_ADDRESS18_HIGH)	32	RW	0000_FFFF

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
394	MAC Address18 Low (MAC_ADDRESS18_LOW)	32	RW	FFFF_FFFF
398	MAC Address19 High (MAC_ADDRESS19_HIGH)	32	RW	0000_FFFF
39C	MAC Address19 Low (MAC_ADDRESS19_LOW)	32	RW	FFFF_FFFF
3A0	MAC Address20 High (MAC_ADDRESS20_HIGH)	32	RW	0000_FFFF
3A4	MAC Address20 Low (MAC_ADDRESS20_LOW)	32	RW	FFFF_FFFF
3A8	MAC Address21 High (MAC_ADDRESS21_HIGH)	32	RW	0000_FFFF
3AC	MAC Address21 Low (MAC_ADDRESS21_LOW)	32	RW	FFFF_FFFF
3B0	MAC Address22 High (MAC_ADDRESS22_HIGH)	32	RW	0000_FFFF
3B4	MAC Address22 Low (MAC_ADDRESS22_LOW)	32	RW	FFFF_FFFF
3B8	MAC Address23 High (MAC_ADDRESS23_HIGH)	32	RW	0000_FFFF
3BC	MAC Address23 Low (MAC_ADDRESS23_LOW)	32	RW	FFFF_FFFF
3C0	MAC Address24 High (MAC_ADDRESS24_HIGH)	32	RW	0000_FFFF
3C4	MAC Address24 Low (MAC_ADDRESS24_LOW)	32	RW	FFFF_FFFF
3C8	MAC Address25 High (MAC_ADDRESS25_HIGH)	32	RW	0000_FFFF
3CC	MAC Address25 Low (MAC_ADDRESS25_LOW)	32	RW	FFFF_FFFF
3D0	MAC Address26 High (MAC_ADDRESS26_HIGH)	32	RW	0000_FFFF
3D4	MAC Address26 Low (MAC_ADDRESS26_LOW)	32	RW	FFFF_FFFF
3D8	MAC Address27 High (MAC_ADDRESS27_HIGH)	32	RW	0000_FFFF
3DC	MAC Address27 Low (MAC_ADDRESS27_LOW)	32	RW	FFFF_FFFF
3E0	MAC Address28 High (MAC_ADDRESS28_HIGH)	32	RW	0000_FFFF
3E4	MAC Address28 Low (MAC_ADDRESS28_LOW)	32	RW	FFFF_FFFF
3E8	MAC Address29 High (MAC_ADDRESS29_HIGH)	32	RW	0000_FFFF
3EC	MAC Address29 Low (MAC_ADDRESS29_LOW)	32	RW	FFFF_FFFF
3F0	MAC Address30 High (MAC_ADDRESS30_HIGH)	32	RW	0000_FFFF
3F4	MAC Address30 Low (MAC_ADDRESS30_LOW)	32	RW	FFFF_FFFF
3F8	MAC Address31 High (MAC_ADDRESS31_HIGH)	32	RW	0000_FFFF
3FC	MAC Address31 Low (MAC_ADDRESS31_LOW)	32	RW	FFFF_FFFF
400	MAC Address32 High (MAC_ADDRESS32_HIGH)	32	RW	0000_FFFF
404	MAC Address32 Low (MAC_ADDRESS32_LOW)	32	RW	FFFF_FFFF
408	MAC Address33 High (MAC_ADDRESS33_HIGH)	32	RW	0000_FFFF
40C	MAC Address33 Low (MAC_ADDRESS33_LOW)	32	RW	FFFF_FFFF

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
410	MAC Address34 High (MAC_ADDRESS34_HIGH)	32	RW	0000_FFFF
414	MAC Address34 Low (MAC_ADDRESS34_LOW)	32	RW	FFFF_FFFF
418	MAC Address35 High (MAC_ADDRESS35_HIGH)	32	RW	0000_FFFF
41C	MAC Address35 Low (MAC_ADDRESS35_LOW)	32	RW	FFFF_FFFF
420	MAC Address36 High (MAC_ADDRESS36_HIGH)	32	RW	0000_FFFF
424	MAC Address36 Low (MAC_ADDRESS36_LOW)	32	RW	FFFF_FFFF
428	MAC Address37 High (MAC_ADDRESS37_HIGH)	32	RW	0000_FFFF
42C	MAC Address37 Low (MAC_ADDRESS37_LOW)	32	RW	FFFF_FFFF
430	MAC Address38 High (MAC_ADDRESS38_HIGH)	32	RW	0000_FFFF
434	MAC Address38 Low (MAC_ADDRESS38_LOW)	32	RW	FFFF_FFFF
438	MAC Address39 High (MAC_ADDRESS39_HIGH)	32	RW	0000_FFFF
43C	MAC Address39 Low (MAC_ADDRESS39_LOW)	32	RW	FFFF_FFFF
440	MAC Address40 High (MAC_ADDRESS40_HIGH)	32	RW	0000_FFFF
444	MAC Address40 Low (MAC_ADDRESS40_LOW)	32	RW	FFFF_FFFF
448	MAC Address41 High (MAC_ADDRESS41_HIGH)	32	RW	0000_FFFF
44C	MAC Address41 Low (MAC_ADDRESS41_LOW)	32	RW	FFFF_FFFF
450	MAC Address42 High (MAC_ADDRESS42_HIGH)	32	RW	0000_FFFF
454	MAC Address42 Low (MAC_ADDRESS42_LOW)	32	RW	FFFF_FFFF
458	MAC Address43 High (MAC_ADDRESS43_HIGH)	32	RW	0000_FFFF
45C	MAC Address43 Low (MAC_ADDRESS43_LOW)	32	RW	FFFF_FFFF
460	MAC Address44 High (MAC_ADDRESS44_HIGH)	32	RW	0000_FFFF
464	MAC Address44 Low (MAC_ADDRESS44_LOW)	32	RW	FFFF_FFFF
468	MAC Address45 High (MAC_ADDRESS45_HIGH)	32	RW	0000_FFFF
46C	MAC Address45 Low (MAC_ADDRESS45_LOW)	32	RW	FFFF_FFFF
470	MAC Address46 High (MAC_ADDRESS46_HIGH)	32	RW	0000_FFFF
474	MAC Address46 Low (MAC_ADDRESS46_LOW)	32	RW	FFFF_FFFF
478	MAC Address47 High (MAC_ADDRESS47_HIGH)	32	RW	0000_FFFF
47C	MAC Address47 Low (MAC_ADDRESS47_LOW)	32	RW	FFFF_FFFF
480	MAC Address48 High (MAC_ADDRESS48_HIGH)	32	RW	0000_FFFF
484	MAC Address48 Low (MAC_ADDRESS48_LOW)	32	RW	FFFF_FFFF
488	MAC Address49 High (MAC_ADDRESS49_HIGH)	32	RW	0000_FFFF

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
48C	MAC Address49 Low (MAC_ADDRESS49_LOW)	32	RW	FFFF_FFFF
490	MAC Address50 High (MAC_ADDRESS50_HIGH)	32	RW	0000_FFFF
494	MAC Address50 Low (MAC_ADDRESS50_LOW)	32	RW	FFFF_FFFF
498	MAC Address51 High (MAC_ADDRESS51_HIGH)	32	RW	0000_FFFF
49C	MAC Address51 Low (MAC_ADDRESS51_LOW)	32	RW	FFFF_FFFF
4A0	MAC Address52 High (MAC_ADDRESS52_HIGH)	32	RW	0000_FFFF
4A4	MAC Address52 Low (MAC_ADDRESS52_LOW)	32	RW	FFFF_FFFF
4A8	MAC Address53 High (MAC_ADDRESS53_HIGH)	32	RW	0000_FFFF
4AC	MAC Address53 Low (MAC_ADDRESS53_LOW)	32	RW	FFFF_FFFF
4B0	MAC Address54 High (MAC_ADDRESS54_HIGH)	32	RW	0000_FFFF
4B4	MAC Address54 Low (MAC_ADDRESS54_LOW)	32	RW	FFFF_FFFF
4B8	MAC Address55 High (MAC_ADDRESS55_HIGH)	32	RW	0000_FFFF
4BC	MAC Address55 Low (MAC_ADDRESS55_LOW)	32	RW	FFFF_FFFF
4C0	MAC Address56 High (MAC_ADDRESS56_HIGH)	32	RW	0000_FFFF
4C4	MAC Address56 Low (MAC_ADDRESS56_LOW)	32	RW	FFFF_FFFF
4C8	MAC Address57 High (MAC_ADDRESS57_HIGH)	32	RW	0000_FFFF
4CC	MAC Address57 Low (MAC_ADDRESS57_LOW)	32	RW	FFFF_FFFF
4D0	MAC Address58 High (MAC_ADDRESS58_HIGH)	32	RW	0000_FFFF
4D4	MAC Address58 Low (MAC_ADDRESS58_LOW)	32	RW	FFFF_FFFF
4D8	MAC Address59 High (MAC_ADDRESS59_HIGH)	32	RW	0000_FFFF
4DC	MAC Address59 Low (MAC_ADDRESS59_LOW)	32	RW	FFFF_FFFF
4E0	MAC Address60 High (MAC_ADDRESS60_HIGH)	32	RW	0000_FFFF
4E4	MAC Address60 Low (MAC_ADDRESS60_LOW)	32	RW	FFFF_FFFF
4E8	MAC Address61 High (MAC_ADDRESS61_HIGH)	32	RW	0000_FFFF
4EC	MAC Address61 Low (MAC_ADDRESS61_LOW)	32	RW	FFFF_FFFF
4F0	MAC Address62 High (MAC_ADDRESS62_HIGH)	32	RW	0000_FFFF
4F4	MAC Address62 Low (MAC_ADDRESS62_LOW)	32	RW	FFFF_FFFF
4F8	MAC Address63 High (MAC_ADDRESS63_HIGH)	32	RW	0000_FFFF
4FC	MAC Address63 Low (MAC_ADDRESS63_LOW)	32	RW	FFFF_FFFF
700	MMC Control (MAC_MMC_CONTROL)	32	RW	0000_0000
704	MMC Rx Interrupt (MAC_MMC_RX_INTERRUPT)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
708	MMC Tx Interrupt (MAC_MMC_TX_INTERRUPT)	32	R	0000_0000
70C	MMC Rx Interrupt Mask (MAC_MMC_RX_INTERRUPT_MASK)	32	RW	0000_0000
710	MMC Tx Interrupt Mask (MAC_MMC_TX_INTERRUPT_MASK)	32	RW	0000_0000
714	Tx Octet Count Good and Bad (MAC_TX_OCTET_COUNT_GOOD_BAD)	32	R	0000_0000
718	Tx Packet Count Good and Bad (MAC_TX_PACKET_COUNT_GOOD_BAD)	32	R	0000_0000
71C	Tx Broadcast Packets Good (MAC_TX_BROADCAST_PACKETS_GOOD)	32	R	0000_0000
720	Tx Multicast Packets Good (MAC_TX_MULTICAST_PACKETS_GOOD)	32	R	0000_0000
724	Tx Good and Bad 64-Byte Packets (MAC_TX_64OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000
728	Tx Good and Bad 65 to 127-Byte Packets (MAC_TX_65TO127OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000
72C	Tx Good and Bad 128 to 255-Byte Packets (MAC_TX_128TO255OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000
730	Tx Good and Bad 256 to 511-Byte Packets (MAC_TX_256TO511OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000
734	Tx Good and Bad 512 to 1023-Byte Packets (MAC_TX_512TO1023OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000
738	Tx Good and Bad 1024 to Max-Byte Packets (MAC_TX_1024TOMAXOCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000
73C	Good and Bad Unicast Packets Transmitted (MAC_TX_UNICAST_PACKETS_GOOD_BAD)	32	R	0000_0000
740	Good and Bad Multicast Packets Transmitted (MAC_TX_MULTICAST_PACKETS_GOOD_BAD)	32	R	0000_0000
744	Good and Bad Broadcast Packets Transmitted (MAC_TX_BROADCAST_PACKETS_GOOD_BAD)	32	R	0000_0000
748	Tx Packets Aborted By Underflow Error (MAC_TX_UNDERFLOW_ERROR_PACKETS)	32	R	0000_0000
74C	Single Collision Good Packets Transmitted (MAC_TX_SINGLE_COLLISION_GOOD_PACKETS)	32	R	0000_0000
750	Multiple Collision Good Packets Transmitted (MAC_TX_MULTIPLE_COLLISION_GOOD_PACKETS)	32	R	0000_0000
754	Deferred Packets Transmitted (MAC_TX_DEFERRED_PACKETS)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
758	Late Collision Packets Transmitted (MAC_TX_LATE_COLLISION_PACKETS)	32	R	0000_0000
75C	Excessive Collision Packets Transmitted (MAC_TX_EXCESSIVE_COLLISION_PACKETS)	32	R	0000_0000
760	Carrier Error Packets Transmitted (MAC_TX_CARRIER_ERROR_PACKETS)	32	R	0000_0000
764	Bytes Transmitted in Good Packets (MAC_TX_OCTET_COUNT_GOOD)	32	R	0000_0000
768	Good Packets Transmitted (MAC_TX_PACKET_COUNT_GOOD)	32	R	0000_0000
76C	Packets Aborted By Excessive Deferral Error (MAC_TX_EXCESSIVE_DEFERRAL_ERROR)	32	R	0000_0000
770	Pause Packets Transmitted (MAC_TX_PAUSE_PACKETS)	32	R	0000_0000
774	Good VLAN Packets Transmitted (MAC_TX_VLAN_PACKETS_GOOD)	32	R	0000_0000
778	Good Oversize Packets Transmitted (MAC_TX_OSIZE_PACKETS_GOOD)	32	R	0000_0000
780	Good and Bad Packets Received (MAC_RX_PACKETS_COUNT_GOOD_BAD)	32	R	0000_0000
784	Bytes in Good and Bad Packets Received (MAC_RX_OCTET_COUNT_GOOD_BAD)	32	R	0000_0000
788	Bytes in Good Packets Received (MAC_RX_OCTET_COUNT_GOOD)	32	R	0000_0000
78C	Good Broadcast Packets Received (MAC_RX_BROADCAST_PACKETS_GOOD)	32	R	0000_0000
790	Good Multicast Packets Received (MAC_RX_MULTICAST_PACKETS_GOOD)	32	R	0000_0000
794	CRC Error Packets Received (MAC_RX_CRC_ERROR_PACKETS)	32	R	0000_0000
798	Alignment Error Packets Received (MAC_RX_ALIGNMENT_ERROR_PACKETS)	32	R	0000_0000
79C	Runt Error Packets Received (MAC_RX_RUNT_ERROR_PACKETS)	32	R	0000_0000
7A0	Jabber Error Packets Received (MAC_RX_JABBER_ERROR_PACKETS)	32	R	0000_0000
7A4	Good Undersize Packets Received (MAC_RX_UNDERSIZE_PACKETS_GOOD)	32	R	0000_0000
7A8	Good Oversize Packets Received (MAC_RX_OVERSIZE_PACKETS_GOOD)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
7AC	Good and Bad 64-Byte Packets Received (MAC_RX_64OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000
7B0	Good and Bad 64-to-127 Byte Packets Received (MAC_RX_65TO127OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000
7B4	Good and Bad 128-to-255 Byte Packets Received (MAC_RX_128TO255OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000
7B8	Good and Bad 256-to-511 Byte Packets Received (MAC_RX_256TO511OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000
7BC	Good and Bad 512-to-1023 Byte Packets Received (MAC_RX_512TO1023OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000
7C0	Good and Bad 1024-to-Max Byte Packets Received (MAC_RX_1024TOMAXOCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000
7C4	Good Unicast Packets Received (MAC_RX_UNICAST_PACKETS_GOOD)	32	R	0000_0000
7C8	Length Error Packets Received (MAC_RX_LENGTH_ERROR_PACKETS)	32	R	0000_0000
7CC	Out-of-range Type Packets Received (MAC_RX_OUT_OF_RANGE_TYPE_PACKETS)	32	R	0000_0000
7D0	Pause Packets Received (MAC_RX_PAUSE_PACKETS)	32	R	0000_0000
7D4	Missed Packets Due to FIFO Overflow (MAC_RX_FIFO_OVERFLOW_PACKETS)	32	R	0000_0000
7D8	Good and Bad VLAN Packets Received (MAC_RX_VLAN_PACKETS_GOOD_BAD)	32	R	0000_0000
7DC	Watchdog Error Packets Received (MAC_RX_WATCHDOG_ERROR_PACKETS)	32	R	0000_0000
7E0	Receive Error Packets Received (MAC_RX_RECEIVE_ERROR_PACKETS)	32	R	0000_0000
7E4	Good Control Packets Received (MAC_RX_CONTROL_PACKETS_GOOD)	32	R	0000_0000
7EC	Microseconds Tx LPI Asserted (MAC_TX_LPI_USEC_CNTR)	32	R	0000_0000
7F0	Number of Times Tx LPI Asserted (MAC_TX_LPI_TRAN_CNTR)	32	R	0000_0000
7F4	Microseconds Rx LPI Sampled (MAC_RX_LPI_USEC_CNTR)	32	R	0000_0000
7F8	Number of Times Rx LPI Entered (MAC_RX_LPI_TRAN_CNTR)	32	R	0000_0000
800	MMC IPC Receive Interrupt Mask (MAC_MMC_IPC_RX_INTERRUPT_MASK)	32	RW	0000_0000
808	MMC IPC Receive Interrupt (MAC_MMC_IPC_RX_INTERRUPT)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
810	Good IPv4 Datagrams Received (MAC_RXIPV4_GOOD_PACKETS)	32	R	0000_0000
814	IPv4 Datagrams Received with Header Errors (MAC_RXIPV4_HEADER_ERROR_PACKETS)	32	R	0000_0000
818	IPv4 Datagrams Received with No Payload (MAC_RXIPV4_NO_PAYLOAD_PACKETS)	32	R	0000_0000
81C	IPv4 Datagrams Received with Fragmentation (MAC_RXIPV4_FRAGMENTED_PACKETS)	32	R	0000_0000
820	IPv4 Datagrams Received with UDP Checksum Disabled (MAC_RXIPV4_UDP_CHECKSUM_DISABLED_PACKETS)	32	R	0000_0000
824	Good IPv6 Datagrams Received (MAC_RXIPV6_GOOD_PACKETS)	32	R	0000_0000
828	IPv6 Datagrams Received with Header Errors (MAC_RXIPV6_HEADER_ERROR_PACKETS)	32	R	0000_0000
82C	IPv6 Datagrams Received with No Payload (MAC_RXIPV6_NO_PAYLOAD_PACKETS)	32	R	0000_0000
830	IPv6 Datagrams Received with Good UDP (MAC_RXUDP_GOOD_PACKETS)	32	R	0000_0000
834	IPv6 Datagrams Received with UDP Checksum Error (MAC_RXUDP_ERROR_PACKETS)	32	R	0000_0000
838	IPv6 Datagrams Received with Good TCP Payload (MAC_RXTCP_GOOD_PACKETS)	32	R	0000_0000
83C	IPv6 Datagrams Received with TCP Checksum Error (MAC_RXTCP_ERROR_PACKETS)	32	R	0000_0000
840	IPv6 Datagrams Received with Good ICMP Payload (MAC_RXICMP_GOOD_PACKETS)	32	R	0000_0000
844	IPv6 Datagrams Received with ICMP Checksum Error (MAC_RXICMP_ERROR_PACKETS)	32	R	0000_0000
850	Good Bytes Received in IPv4 Datagrams (MAC_RXIPV4_GOOD_OCTETS)	32	R	0000_0000
854	Bytes Received in IPv4 Datagrams with Header Errors (MAC_RXIPV4_HEADER_ERROR_OCTETS)	32	R	0000_0000
858	Bytes Received in IPv4 Datagrams with No Payload (MAC_RXIPV4_NO_PAYLOAD_OCTETS)	32	R	0000_0000
85C	Bytes Received in Fragmented IPv4 Datagrams (MAC_RXIPV4_FRAGMENTED_OCTETS)	32	R	0000_0000
860	Bytes Received with UDP Checksum Disabled (MAC_RXIPV4_UDP_CHECKSUM_DISABLE_OCTETS)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
864	Bytes Received in Good IPv6 Datagrams (MAC_RXIPV6_GOOD_OCTETS)	32	R	0000_0000
868	Bytes Received in IPv6 Datagrams with Data Errors (MAC_RXIPV6_HEADER_ERROR_OCTETS)	32	R	0000_0000
86C	Bytes Received in IPv6 Datagrams with No Payload (MAC_RXIPV6_NO_PAYLOAD_OCTETS)	32	R	0000_0000
870	Bytes Received in Good UDP Segment (MAC_RXUDP_GOOD_OCTETS)	32	R	0000_0000
874	Bytes Received in UDP Segment with Checksum Errors (MAC_RXUDP_ERROR_OCTETS)	32	R	0000_0000
878	Bytes Received in Good TCP Segment (MAC_RXTCP_GOOD_OCTETS)	32	R	0000_0000
87C	Bytes Received in TCP Segment with Checksum Errors (MAC_RXTCP_ERROR_OCTETS)	32	R	0000_0000
880	Bytes Received in Good ICMP Segment (MAC_RXICMP_GOOD_OCTETS)	32	R	0000_0000
884	Bytes Received in ICMP Segment with Checksum Errors (MAC_RXICMP_ERROR_OCTETS)	32	R	0000_0000
8A0	MMC FPE Transmit Interrupt (MAC_MMC_FPE_TX_INTERRUPT)	32	R	0000_0000
8A4	MMC FPE Transmit Mask Interrupt (MAC_MMC_FPE_TX_INTERRUPT_MASK)	32	RW	0000_0000
8A8	MMC FPE Transmitted Fragment Counter (MAC_MMC_TX_FPE_FRAGMENT_CNTR)	32	R	0000_0000
8AC	MMC FPE Transmitted Hold Request Counter (MAC_MMC_TX_HOLD_REQ_CNTR)	32	R	0000_0000
8C0	MMC FPE Receive Interrupt (MAC_MMC_FPE_RX_INTERRUPT)	32	R	0000_0000
8C4	MMC FPE Receive Interrupt Mask (MAC_MMC_FPE_RX_INTERRUPT_MASK)	32	RW	0000_0000
8C8	MMC Receive Packet Reassembly Error Counter (MAC_MMC_RX_PACKET_ASSEMBLY_ERR_CNTR)	32	R	0000_0000
8CC	MMC Receive Packet SMD Error Counter (MAC_MMC_RX_PACKET_SMD_ERR_CNTR)	32	R	0000_0000
8D0	MMC Receive Packet Successful Reassembly Counter (MAC_MMC_RX_PACKET_ASSEMBLY_OK_CNTR)	32	R	0000_0000
8D4	MMC FPE Received Fragment Counter (MAC_MMC_RX_FPE_FRAGMENT_CNTR)	32	R	0000_0000
900	Layer 3 and Layer 4 Control of Filter 0 (MAC_L3_L4_CONTROL0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
904	Layer 4 Address 0 (MAC_LAYER4_ADDRESS0)	32	RW	0000_0000
910	Layer 3 Address 0 Register 0 (MAC_LAYER3_ADDR0_REG0)	32	RW	0000_0000
914	Layer 3 Address 1 Register 0 (MAC_LAYER3_ADDR1_REG0)	32	RW	0000_0000
918	Layer 3 Address 2 Register 0 (MAC_LAYER3_ADDR2_REG0)	32	RW	0000_0000
91C	Layer 3 Address 3 Register 0 (MAC_LAYER3_ADDR3_REG0)	32	RW	0000_0000
930	Layer 3 and Layer 4 Control of Filter 1 (MAC_L3_L4_CONTROL1)	32	RW	0000_0000
934	Layer 4 Address 0 (MAC_LAYER4_ADDRESS1)	32	RW	0000_0000
940	Layer 3 Address 0 Register 1 (MAC_LAYER3_ADDR0_REG1)	32	RW	0000_0000
944	Layer 3 Address 1 Register 1 (MAC_LAYER3_ADDR1_REG1)	32	RW	0000_0000
948	Layer 3 Address 2 Register 1 (MAC_LAYER3_ADDR2_REG1)	32	RW	0000_0000
94C	Layer 3 Address 3 Register 1 (MAC_LAYER3_ADDR3_REG1)	32	RW	0000_0000
960	Layer 3 and Layer 4 Control of Filter 2 (MAC_L3_L4_CONTROL2)	32	RW	0000_0000
964	Layer 4 Address 2 (MAC_LAYER4_ADDRESS2)	32	RW	0000_0000
970	Layer 3 Address 0 Register 2 (MAC_LAYER3_ADDR0_REG2)	32	RW	0000_0000
974	Layer 3 Address 0 Register 2 (MAC_LAYER3_ADDR1_REG2)	32	RW	0000_0000
978	Layer 3 Address 2 Register 2 (MAC_LAYER3_ADDR2_REG2)	32	RW	0000_0000
97C	Layer 3 Address 3 Register 2 (MAC_LAYER3_ADDR3_REG2)	32	RW	0000_0000
990	Layer 3 and Layer 4 Control of Filter 3 (MAC_L3_L4_CONTROL3)	32	RW	0000_0000
994	Layer 4 Address 3 (MAC_LAYER4_ADDRESS3)	32	RW	0000_0000
9A0	Layer 3 Address 0 Register 3 (MAC_LAYER3_ADDR0_REG3)	32	RW	0000_0000
9A4	Layer 3 Address 1 Register 3 (MAC_LAYER3_ADDR1_REG3)	32	RW	0000_0000
9A8	Layer 3 Address 2 Register 3 (MAC_LAYER3_ADDR2_REG3)	32	RW	0000_0000
9AC	Layer 3 Address 3 Register 3 (MAC_LAYER3_ADDR3_REG3)	32	RW	0000_0000
9C0	Layer 3 and Layer 4 Control of Filter 4 (MAC_L3_L4_CONTROL4)	32	RW	0000_0000
9C4	Layer 4 Address 4 (MAC_LAYER4_ADDRESS4)	32	RW	0000_0000
9D0	Layer 3 Address 0 Register 4 (MAC_LAYER3_ADDR0_REG4)	32	RW	0000_0000
9D4	Layer 3 Address 1 Register 4 (MAC_LAYER3_ADDR1_REG4)	32	RW	0000_0000
9D8	Layer 3 Address 2 Register 4 (MAC_LAYER3_ADDR2_REG4)	32	RW	0000_0000
9DC	Layer 3 Address 3 Register 4 (MAC_LAYER3_ADDR3_REG4)	32	RW	0000_0000
9F0	Layer 3 and Layer 4 Control of Filter 5 (MAC_L3_L4_CONTROL5)	32	RW	0000_0000
9F4	Layer 4 Address 5 (MAC_LAYER4_ADDRESS5)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
A00	Layer 3 Address 0 Register 5 (MAC_LAYER3_ADDR0_REG5)	32	RW	0000_0000
A04	Layer 3 Address 1 Register 5 (MAC_LAYER3_ADDR1_REG5)	32	RW	0000_0000
A08	Layer 3 Address 2 Register 5 (MAC_LAYER3_ADDR2_REG5)	32	RW	0000_0000
A0C	Layer 3 Address 3 Register 5 (MAC_LAYER3_ADDR3_REG5)	32	RW	0000_0000
A20	Layer 3 and Layer 4 Control of Filter 6 (MAC_L3_L4_CONTROL6)	32	RW	0000_0000
A24	Layer 4 Address 6 (MAC_LAYER4_ADDRESS6)	32	RW	0000_0000
A30	Layer 3 Address 0 Register 6 (MAC_LAYER3_ADDR0_REG6)	32	RW	0000_0000
A34	Layer 3 Address 1 Register 6 (MAC_LAYER3_ADDR1_REG6)	32	RW	0000_0000
A38	Layer 3 Address 2 Register 6 (MAC_LAYER3_ADDR2_REG6)	32	RW	0000_0000
A3C	Layer 3 Address 3 Register 6 (MAC_LAYER3_ADDR3_REG6)	32	RW	0000_0000
A50	Layer 3 and Layer 4 Control of Filter 0 (MAC_L3_L4_CONTROL7)	32	RW	0000_0000
A54	Layer 4 Address 7 (MAC_LAYER4_ADDRESS7)	32	RW	0000_0000
A60	Layer 3 Address 0 Register 7 (MAC_LAYER3_ADDR0_REG7)	32	RW	0000_0000
A64	Layer 3 Address 1 Register 7 (MAC_LAYER3_ADDR1_REG7)	32	RW	0000_0000
A68	Layer 3 Address 2 Register 7 (MAC_LAYER3_ADDR2_REG7)	32	RW	0000_0000
A6C	Layer 3 Address 3 Register 7 (MAC_LAYER3_ADDR3_REG7)	32	RW	0000_0000
A70	MAC_INDIR_ACCESS_CTRL (MAC_INDIR_ACCESS_CTRL)	32	RW	0000_0000
A74	MAC_INDIR_ACCESS_DATA (MAC_INDIR_ACCESS_DATA)	32	RW	0000_0000
B00	Timestamp Control (MAC_TIMESTAMP_CONTROL)	32	RW	0000_2000
B04	Subsecond Increment (MAC_SUB_SECOND_INCREMENT)	32	RW	0000_0000
B08	System Time Seconds (MAC_SYSTEM_TIME_SECONDS)	32	R	0000_0000
B0C	System Time Nanoseconds (MAC_SYSTEM_TIME_NANOSECONDS)	32	R	0000_0000
B10	System Time Seconds Update (MAC_SYSTEM_TIME_SECONDS_UPDATE)	32	RW	0000_0000
B14	System Time Nanoseconds Update (MAC_SYSTEM_TIME_NANOSECONDS_UPDATE)	32	RW	0000_0000
B18	Timestamp Addend (MAC_TIMESTAMP_ADDEND)	32	RW	0000_0000
B1C	System Time - Higher Word Seconds (MAC_SYSTEM_TIME_HIGHER_WORD_SECONDS)	32	RW	0000_0000
B20	Timestamp Status (MAC_TIMESTAMP_STATUS)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
B30	Transmit Timestamp Status Nanoseconds (MAC_TX_TIMESTAMP_STATUS_NANOSECONDS)	32	R	0000_0000
B34	Transmit Timestamp Status Seconds (MAC_TX_TIMESTAMP_STATUS_SECONDS)	32	R	0000_0000
B40	Auxiliary Timestamp Control (MAC_AUXILIARY_CONTROL)	32	RW	0000_0000
B48	Auxiliary Timestamp Nanoseconds (MAC_AUXILIARY_TIMESTAMP_NANOSECONDS)	32	R	0000_0000
B4C	Auxiliary Timestamp Seconds (MAC_AUXILIARY_TIMESTAMP_SECONDS)	32	R	0000_0000
B50	Timestamp Ingress Asymmetry Correction (MAC_TIMESTAMP_INGRESS_ASYM_CORR)	32	RW	0000_0000
B54	Timestamp Egress Asymmetry Correction (MAC_TIMESTAMP_EGRESS_ASYM_CORR)	32	RW	0000_0000
B58	Timestamp Ingress Correction Nanosecond (MAC_TIMESTAMP_INGRESS_CORR_NANOSECOND)	32	RW	0000_0000
B5C	Timestamp Egress Correction Nanosecond (MAC_TIMESTAMP_EGRESS_CORR_NANOSECOND)	32	RW	0000_0000
B60	Timestamp Ingress Correction Subnanosecond (MAC_TIMESTAMP_INGRESS_CORR_SUBNANOSEC)	32	RW	0000_0000
B64	Timestamp Egress Correction Subnanosecond (MAC_TIMESTAMP_EGRESS_CORR_SUBNANOSEC)	32	RW	0000_0000
B68	Timestamp Ingress Latency (MAC_TIMESTAMP_INGRESS_LATENCY)	32	R	0000_0000
B6C	Timestamp Egress Latency (MAC_TIMESTAMP_EGRESS_LATENCY)	32	R	0000_0000
B70	PPS Control (MAC_PPS_CONTROL)	32	RW	0000_0000
B80	PPS0 Target Time Seconds (MAC_PPS0_TARGET_TIME_SECONDS)	32	RW	0000_0000
B84	PPS0 Target Time Nanoseconds (MAC_PPS0_TARGET_TIME_NANOSECONDS)	32	RW	0000_0000
B88	PPS0 Interval (MAC_PPS0_INTERVAL)	32	RW	0000_0000
B8C	PPS0 Width (MAC_PPS0_WIDTH)	32	RW	0000_0000
B90	PPS1 Target Time Seconds (MAC_PPS1_TARGET_TIME_SECONDS)	32	RW	0000_0000
B94	PPS1 Target Time Nanoseconds (MAC_PPS1_TARGET_TIME_NANOSECONDS)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
B98	PPS1 Interval (MAC_PPS1_INTERVAL)	32	RW	0000_0000
B9C	PPS1 Width (MAC_PPS1_WIDTH)	32	RW	0000_0000
BA0	PPS2 Target Time Seconds (MAC_PPS2_TARGET_TIME_SECONDS)	32	RW	0000_0000
BA4	PPS2 Target Time Nanoseconds (MAC_PPS2_TARGET_TIME_NANOSECONDS)	32	RW	0000_0000
BA8	PPS2 Interval (MAC_PPS2_INTERVAL)	32	RW	0000_0000
BAC	PPS2 Width (MAC_PPS2_WIDTH)	32	RW	0000_0000
BB0	PPS3 Target Time Seconds (MAC_PPS3_TARGET_TIME_SECONDS)	32	RW	0000_0000
BB4	PPS3 Target Time Nanoseconds (MAC_PPS3_TARGET_TIME_NANOSECONDS)	32	RW	0000_0000
BB8	PPS3 Interval (MAC_PPS3_INTERVAL)	32	RW	0000_0000
BBC	PPS3 Width (MAC_PPS3_WIDTH)	32	RW	0000_0000
BC0	PTP Offload Engine Control (MAC_PTO_CONTROL)	32	RW	0000_0000
BC4	Source Port Identity 0 (MAC_SOURCE_PORT_IDENTITY0)	32	RW	0000_0000
BC8	Source Port Identity 1 (MAC_SOURCE_PORT_IDENTITY1)	32	RW	0000_0000
BCC	Source Port Identity 2 (MAC_SOURCE_PORT_IDENTITY2)	32	RW	0000_0000
BD0	Log Message Interval (MAC_LOG_MESSAGE_INTERVAL)	32	RW	0000_0000
C00	MTL Operation Mode (MTL_OPERATION_MODE)	32	RW	0000_0000
C08	FIFO Debug Access Control and Status (MTL_DBG_CTL)	32	RW	0000_0000
C0C	FIFO Debug Status (MTL_DBG_STS)	32	RW	0000_0018
C10	FIFO Debug Data (MTL_FIFO_DEBUG_DATA)	32	RW	0000_0000
C20	MTL Interrupt Status (MTL_INTERRUPT_STATUS)	32	R	0000_0000
C30	Receive Queue and DMA Channel Mapping 0 (MTL_RXQ_DMA_MAP0)	32	RW	0000_0000
C34	Receive Queue and DMA Channel Mapping 1 (MTL_RXQ_DMA_MAP1)	32	RW	0000_0000
C40	Time Based Scheduling Control (MTL_TBS_CTRL)	32	RW	0000_0000
C50	Enhancements to Scheduled Transmission Control (MTL_EST_CONTROL)	32	RW	0000_0000
C54	MTL_EST_EXT_CONTROL (MTL_EST_EXT_CONTROL)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
C58	Enhancements to Scheduled Transmission Status (MTL_EST_STATUS)	32	RW	0000_0000
C60	EST Scheduling Error (MTL_EST_SCH_ERROR)	32	RW	0000_0000
C64	EST Frame Size Error (MTL_EST_FRM_SIZE_ERROR)	32	RW	0000_0000
C68	EST Frame Size Capture (MTL_EST_FRM_SIZE_CAPTURE)	32	R	0000_0000
C70	EST Interrupt Enable (MTL_EST_INTR_ENABLE)	32	RW	0000_0000
C80	EST GCL Control (MTL_EST_GCL_CONTROL)	32	RW	0000_0000
C84	EST GCL Data (MTL_EST_GCL_DATA)	32	RW	0000_0000
C90	Frame Preemption Control and Status (MTL_FPE_CTRL_STS)	32	RW	0000_0000
C94	Frame Preemption Hold and Release Advance (MTL_FPE_ADVANCE)	32	RW	0000_0000
CA0	RXP Control Status (MTL_RXP_CONTROL_STATUS)	32	RW	80FF_00FF
CA4	RXP Interrupt Control Status (MTL_RXP_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000
CA8	RXP Drop Count (MTL_RXP_DROP_CNT)	32	R	0000_0000
CAC	RXP Error Count (MTL_RXP_ERROR_CNT)	32	R	0000_0000
CB0	RXP Indirect Access Control and Status (MTL_RXP_INDIRECT_ACC_CONTROL_STATUS)	32	RW	0000_0000
CB4	RXP Indirect Access Data (MTL_RXP_INDIRECT_ACC_DATA)	32	RW	0000_0000
CB8	MTL_RXP_BYPASS_CNT (MTL_RXP_BYPASS_CNT)	32	R	0000_0000
D00	Queue 0 Transmit Operation Mode (MTL_TXQ0_OPERATION_MODE)	32	RW	0000_0000
D04	Queue 0 Underflow Counter (MTL_TXQ0_UNDERFLOW)	32	R	0000_0000
D08	Queue 0 Transmit Debug (MTL_TXQ0_DEBUG)	32	R	0000_0000
D14	Queue 0 ETS Status (MTL_TXQ0_ETS_STATUS)	32	R	0000_0000
D18	Queue 0 Quantum or Weights (MTL_TXQ0_QUANTUM_WEIGHT)	32	RW	0000_0000
D2C	Queue 0 Interrupt Control Status (MTL_Q0_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000
D30	Queue 0 Receive Operation Mode (MTL_RXQ0_OPERATION_MODE)	32	RW	0000_0000
D34	Queue 0 Missed Packet and Overflow Counter (MTL_RXQ0_MISSED_PACKET_OVERFLOW_CNT)	32	R	0000_0000
D38	Queue 0 Receive Debug (MTL_RXQ0_DEBUG)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
D3C	Queue 0 Receive Control (MTL_RXQ0_CONTROL)	32	RW	0000_0000
D40	Queue 1 Transmit Operation Mode (MTL_TXQ1_OPERATION_MODE)	32	RW	0000_0000
D44	Queue 1 Underflow Counter (MTL_TXQ1_UNDERFLOW)	32	R	0000_0000
D48	Queue 1 Transmit Debug (MTL_TXQ1_DEBUG)	32	R	0000_0000
D50	Queue 1 ETS Control (MTL_TXQ1_ETS_CONTROL)	32	RW	0000_0000
D54	Queue 1 ETS Status (MTL_TXQ1_ETS_STATUS)	32	R	0000_0000
D58	Queue 1 idleSlopeCredit, Quantum or Weights (MTL_TXQ1_QUANTUM_WEIGHT)	32	RW	0000_0000
D5C	Queue 1 sendSlopeCredit (MTL_TXQ1_SENDSLOPECREDIT)	32	RW	0000_0000
D60	Queue 1 hiCredit (MTL_TXQ1_HICREDIT)	32	RW	0000_0000
D64	Queue 1 loCredit (MTL_TXQ1_LOCREDIT)	32	RW	0000_0000
D6C	Queue 1 Interrupt Control Status (MTL_Q1_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000
D70	Queue 1 Receive Operation Mode (MTL_RXQ1_OPERATION_MODE)	32	RW	0000_0000
D74	Queue 1 Missed Packet and Overflow Counter (MTL_RXQ1_MISSED_PACKET_OVERFLOW_CNT)	32	R	0000_0000
D78	Queue 1 Receive Debug (MTL_RXQ1_DEBUG)	32	R	0000_0000
D7C	Queue 1 Receive Control (MTL_RXQ1_CONTROL)	32	RW	0000_0000
D80	Queue 2 Transmit Operation Mode (MTL_TXQ2_OPERATION_MODE)	32	RW	0000_0000
D84	Queue 2 Underflow Counter (MTL_TXQ2_UNDERFLOW)	32	R	0000_0000
D88	Queue 2 Transmit Debug (MTL_TXQ2_DEBUG)	32	R	0000_0000
D90	Queue 2 ETS Control (MTL_TXQ2_ETS_CONTROL)	32	RW	0000_0000
D94	Queue 2 ETS Status (MTL_TXQ2_ETS_STATUS)	32	R	0000_0000
D98	Queue 2 idleSlopeCredit, Quantum or Weights (MTL_TXQ2_QUANTUM_WEIGHT)	32	RW	0000_0000
D9C	Queue 2 sendSlopeCredit (MTL_TXQ2_SENDSLOPECREDIT)	32	RW	0000_0000
DA0	Queue 2 hiCredit (MTL_TXQ2_HICREDIT)	32	RW	0000_0000
DA4	Queue 2 loCredit (MTL_TXQ2_LOCREDIT)	32	RW	0000_0000
DAC	Queue 2 Interrupt Control Status (MTL_Q2_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
DB0	Queue 2 Receive Operation Mode (MTL_RXQ2_OPERATION_MODE)	32	RW	0000_0000
DB4	Queue 2 Missed Packet and Overflow Counter (MTL_RXQ2_MISSED_PACKET_OVERFLOW_CNT)	32	R	0000_0000
DB8	Queue 2 Receive Debug (MTL_RXQ2_DEBUG)	32	R	0000_0000
DBC	Queue 2 Receive Control (MTL_RXQ2_CONTROL)	32	RW	0000_0000
DC0	Queue 3 Transmit Operation Mode (MTL_TXQ3_OPERATION_MODE)	32	RW	0000_0000
DC4	Queue 3 Underflow Counter (MTL_TXQ3_UNDERFLOW)	32	R	0000_0000
DC8	Queue 3 Transmit Debug (MTL_TXQ3_DEBUG)	32	R	0000_0000
DD0	Queue 3 ETS Control (MTL_TXQ3_ETS_CONTROL)	32	RW	0000_0000
DD4	Queue 3 ETS Status (MTL_TXQ3_ETS_STATUS)	32	R	0000_0000
DD8	Queue 3 idleSlopeCredit, Quantum or Weights (MTL_TXQ3_QUANTUM_WEIGHT)	32	RW	0000_0000
DDC	Queue 3 sendSlopeCredit (MTL_TXQ3_SENDSLOPECREDIT)	32	RW	0000_0000
DE0	Queue 3 hiCredit (MTL_TXQ3_HICREDIT)	32	RW	0000_0000
DE4	Queue 3 loCredit (MTL_TXQ3_LOCREDIT)	32	RW	0000_0000
DEC	Queue 3 Interrupt Control Status (MTL_Q3_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000
DF0	Queue 3 Receive Operation Mode (MTL_RXQ3_OPERATION_MODE)	32	RW	0000_0000
DF4	Queue 3 Missed Packet and Overflow Counter (MTL_RXQ3_MISSED_PACKET_OVERFLOW_CNT)	32	R	0000_0000
DF8	Queue 3 Receive Debug (MTL_RXQ3_DEBUG)	32	R	0000_0000
DFC	Queue 3 Receive Control (MTL_RXQ3_CONTROL)	32	RW	0000_0000
E00	Queue 4 Transmit Operation Mode (MTL_TXQ4_OPERATION_MODE)	32	RW	0000_0000
E04	Queue 4 Underflow Counter (MTL_TXQ4_UNDERFLOW)	32	R	0000_0000
E08	Queue 4 Transmit Debug (MTL_TXQ4_DEBUG)	32	R	0000_0000
E10	Queue 4 ETS Control (MTL_TXQ4_ETS_CONTROL)	32	RW	0000_0000
E14	Queue 4 ETS Status (MTL_TXQ4_ETS_STATUS)	32	R	0000_0000
E18	Queue 4 idleSlopeCredit, Quantum or Weights (MTL_TXQ4_QUANTUM_WEIGHT)	32	RW	0000_0000
E1C	Queue 4 sendSlopeCredit (MTL_TXQ4_SENDSLOPECREDIT)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
E20	Queue 4 hiCredit (MTL_TXQ4_HICREDIT)	32	RW	0000_0000
E24	Queue 4 loCredit (MTL_TXQ4_LOCREDIT)	32	RW	0000_0000
E2C	Queue 4 Interrupt Control Status (MTL_Q4_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000
E30	Queue 4 Receive Operation Mode (MTL_RXQ4_OPERATION_MODE)	32	RW	0000_0000
E34	Queue 4 Missed Packet and Overflow Counter (MTL_RXQ4_MISSED_PACKET_OVERFLOW_CNT)	32	R	0000_0000
E38	Queue 4 Receive Debug (MTL_RXQ4_DEBUG)	32	R	0000_0000
E3C	Queue 4 Receive Control (MTL_RXQ4_CONTROL)	32	RW	0000_0000
1000	DMA Bus Mode (DMA_MODE)	32	RW	0000_0000
1004	DMA System Bus Mode (DMA_SYSBUS_MODE)	32	RW	0101_0000
1008	DMA Interrupt Status (DMA_INTERRUPT_STATUS)	32	R	0000_0000
100C	DMA Debug Status 0 (DMA_DEBUG_STATUS0)	32	R	0000_0000
1010	DMA Debug Status 1 (DMA_DEBUG_STATUS1)	32	R	0000_0000
1040	AXI LPI Entry Interval Control (DMA_AXI_LPI_ENTRY_INTERVAL)	32	RW	0000_0000
1050	DMA_TBS_CTRL0 (DMA_TBS_CTRL0)	32	RW	0000_0000
1054	DMA_TBS_CTRL1 (DMA_TBS_CTRL1)	32	RW	0000_0000
1058	DMA_TBS_CTRL2 (DMA_TBS_CTRL2)	32	RW	0000_0000
105C	DMA_TBS_CTRL3 (DMA_TBS_CTRL3)	32	RW	0000_0000
1100	DMA Channel 0 Control (DMA_CH0_CONTROL)	32	RW	0000_0000
1104	DMA Channel 0 Transmit Control (DMA_CH0_TX_CONTROL)	32	RW	0000_0000
1108	DMA Channel 0 Receive Control (DMA_CH0_RX_CONTROL)	32	RW	0000_0000
1114	Channel 0 Tx Descriptor List Address register (DMA_CH0_TXDESC_LIST_ADDRESS)	32	RW	0000_0000
111C	Channel 0 Rx Descriptor List Address register (DMA_CH0_RXDESC_LIST_ADDRESS)	32	RW	0000_0000
1120	Channel 0 Tx Descriptor Tail Pointer (DMA_CH0_TXDESC_TAIL_POINTER)	32	RW	0000_0000
1128	Channel 0 Rx Descriptor Tail Pointer (DMA_CH0_RXDESC_TAIL_POINTER)	32	RW	0000_0000
112C	Channel 0 Tx Descriptor Ring Length (DMA_CH0_TXDESC_RING_LENGTH)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1130	Channel 0 Receive Control 2 register (DMA_CH0_RX_CONTROL2)	32	RW	0000_0000
1134	Channel 0 Interrupt Enable (DMA_CH0_INTERRUPT_ENABLE)	32	RW	0000_0000
1138	Channel 0 Receive Interrupt Watchdog Timer (DMA_CH0_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000
113C	Channel 0 Slot Function Control and Status (DMA_CH0_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0
1144	Channel 0 Current Application Transmit Descriptor (DMA_CH0_CURRENT_APP_TXDESC)	32	R	0000_0000
114C	Channel 0 Current Application Receive Descriptor (DMA_CH0_CURRENT_APP_RXDESC)	32	R	0000_0000
1154	Channel 0 Current Application Transmit Buffer Address (DMA_CH0_CURRENT_APP_TXBUFFER)	32	R	0000_0000
115C	Channel 0 Current Application Receive Buffer Address (DMA_CH0_CURRENT_APP_RXBUFFER)	32	R	0000_0000
1160	DMA Channel 0 Status (DMA_CH0_STATUS)	32	RW	0000_0000
1164	Channel 0 Missed Frame Counter (DMA_CH0_MISS_FRAME_CNT)	32	R	0000_0000
1168	Channel 0 RXP Frames Accepted Counter (DMA_CH0_RXP_ACCEPT_CNT)	32	R	0000_0000
1180	DMA Channel 1 Control (DMA_CH1_CONTROL)	32	RW	0000_0000
1184	DMA Channel 1 Transmit Control (DMA_CH1_TX_CONTROL)	32	RW	0000_0000
1188	DMA Channel 1 Receive Control (DMA_CH1_RX_CONTROL)	32	RW	0000_0000
1194	Channel 1 Tx Descriptor List Address (DMA_CH1_TXDESC_LIST_ADDRESS)	32	RW	0000_0000
119C	Channel 1 Rx Descriptor List Address (DMA_CH1_RXDESC_LIST_ADDRESS)	32	RW	0000_0000
11A0	Channel 1 Tx Descriptor Tail Pointer (DMA_CH1_TXDESC_TAIL_POINTER)	32	RW	0000_0000
11A8	Channel 1 Rx Descriptor Tail Pointer (DMA_CH1_RXDESC_TAIL_POINTER)	32	RW	0000_0000
11AC	Channel 1 Tx Descriptor Ring Length (DMA_CH1_TXDESC_RING_LENGTH)	32	RW	0000_0000
11B0	Channel 1 Receive Control 2 register (DMA_CH1_RX_CONTROL2)	32	RW	0000_0000
11B4	Channel 1 Interrupt Enable (DMA_CH1_INTERRUPT_ENABLE)	32	RW	0000_0000
11B8	Channel 1 Receive Interrupt Watchdog Timer (DMA_CH1_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
11BC	Channel 1 Slot Function Control and Status (DMA_CH1_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0
11C4	Channel 1 Current Application Transmit Descriptor (DMA_CH1_CURRENT_APP_TXDESC)	32	R	0000_0000
11CC	Channel 1 Current Application Receive Descriptor (DMA_CH1_CURRENT_APP_RXDESC)	32	R	0000_0000
11D4	Channel 1 Current Application Transmit Buffer Address (DMA_CH1_CURRENT_APP_TXBUFFER)	32	R	0000_0000
11DC	Channel 1 Current Application Receive Buffer Address (DMA_CH1_CURRENT_APP_RXBUFFER)	32	R	0000_0000
11E0	DMA Channel 1 Status (DMA_CH1_STATUS)	32	RW	0000_0000
11E4	Channel 1 Missed Frame Counter (DMA_CH1_MISS_FRAME_CNT)	32	R	0000_0000
11E8	Channel 1 RXP Frames Accepted Counter (DMA_CH1_RXP_ACCEPT_CNT)	32	R	0000_0000
1200	DMA Channel 2 Control (DMA_CH2_CONTROL)	32	RW	0000_0000
1204	DMA Channel 2 Transmit Control (DMA_CH2_TX_CONTROL)	32	RW	0000_0000
1208	DMA Channel 2 Receive Control (DMA_CH2_RX_CONTROL)	32	RW	0000_0000
1214	Channel 2 Tx Descriptor List Address (DMA_CH2_TXDESC_LIST_ADDRESS)	32	RW	0000_0000
121C	Channel 2 Rx Descriptor List Address (DMA_CH2_RXDESC_LIST_ADDRESS)	32	RW	0000_0000
1220	Channel 2 Tx Descriptor Tail Pointer (DMA_CH2_TXDESC_TAIL_POINTER)	32	RW	0000_0000
1228	Channel 2 Rx Descriptor Tail Pointer (DMA_CH2_RXDESC_TAIL_POINTER)	32	RW	0000_0000
122C	Channel 2 Tx Descriptor Ring Length (DMA_CH2_TXDESC_RING_LENGTH)	32	RW	0000_0000
1230	Channel 2 Receive Control 2 register (DMA_CH2_RX_CONTROL2)	32	RW	0000_0000
1234	Channel 2 Interrupt Enable (DMA_CH2_INTERRUPT_ENABLE)	32	RW	0000_0000
1238	Channel 2 Receive Interrupt Watchdog Timer (DMA_CH2_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000
123C	Channel 2 Slot Function Control and Status (DMA_CH2_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0
1244	Channel 2 Current Application Transmit Descriptor (DMA_CH2_CURRENT_APP_TXDESC)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
124C	Channel 2 Current Application Receive Descriptor (DMA_CH2_CURRENT_APP_RXDESC)	32	R	0000_0000
1254	Channel 2 Current Application Transmit Buffer Address (DMA_CH2_CURRENT_APP_TXBUFFER)	32	R	0000_0000
125C	Channel 2 Current Application Receive Buffer Address (DMA_CH2_CURRENT_APP_RXBUFFER)	32	R	0000_0000
1260	DMA Channel 2 Status (DMA_CH2_STATUS)	32	RW	0000_0000
1264	Channel 2 Missed Frame Counter (DMA_CH2_MISS_FRAME_CNT)	32	R	0000_0000
1268	Channel 2 RXP Frames Accepted Counter (DMA_CH2_RXP_ACCEPT_CNT)	32	R	0000_0000
1280	DMA Channel 3 Control (DMA_CH3_CONTROL)	32	RW	0000_0000
1284	DMA Channel 3 Transmit Control (DMA_CH3_TX_CONTROL)	32	RW	0000_0000
1288	DMA Channel 3 Receive Control (DMA_CH3_RX_CONTROL)	32	RW	0000_0000
1294	Channel 3 Tx Descriptor List Address (DMA_CH3_TXDESC_LIST_ADDRESS)	32	RW	0000_0000
129C	Channel 3 Rx Descriptor List Address (DMA_CH3_RXDESC_LIST_ADDRESS)	32	RW	0000_0000
12A0	Channel 3 Tx Descriptor Tail Pointer (DMA_CH3_TXDESC_TAIL_POINTER)	32	RW	0000_0000
12A8	Channel 3 Rx Descriptor Tail Pointer (DMA_CH3_RXDESC_TAIL_POINTER)	32	RW	0000_0000
12AC	Channel 3 Tx Descriptor Ring Length (DMA_CH3_TXDESC_RING_LENGTH)	32	RW	0000_0000
12B0	Channel 3 Receive Control 2 register (DMA_CH3_RX_CONTROL2)	32	RW	0000_0000
12B4	Channel 3 Interrupt Enable (DMA_CH3_INTERRUPT_ENABLE)	32	RW	0000_0000
12B8	Channel 3 Receive Interrupt Watchdog Time (DMA_CH3_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000
12BC	Channel 3 Slot Function Control and Status (DMA_CH3_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0
12C4	Channel 3 Current Application Transmit Descriptor (DMA_CH3_CURRENT_APP_TXDESC)	32	R	0000_0000
12CC	Channel 3 Current Application Receive Descriptor (DMA_CH3_CURRENT_APP_RXDESC)	32	R	0000_0000
12D4	Channel 3 Current Application Transmit Buffer Address (DMA_CH3_CURRENT_APP_TXBUFFER)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
12DC	Channel 3 Current Application Receive Buffer Address (DMA_CH3_CURRENT_APP_RXBUFFER)	32	R	0000_0000
12E0	DMA Channel 3 Status (DMA_CH3_STATUS)	32	RW	0000_0000
12E4	Channel 3 Missed Frame Counter (DMA_CH3_MISS_FRAME_CNT)	32	R	0000_0000
12E8	Channel 3 RXP Frames Accepted Counter (DMA_CH3_RXP_ACCEPT_CNT)	32	R	0000_0000
1300	Channel 4 Control (DMA_CH4_CONTROL)	32	RW	0000_0000
1304	DMA Channel 4 Transmit Control (DMA_CH4_TX_CONTROL)	32	RW	0000_0000
1308	DMA Channel 4 Receive Control (DMA_CH4_RX_CONTROL)	32	RW	0000_0000
1314	Channel 4 Tx Descriptor List Address (DMA_CH4_TXDESC_LIST_ADDRESS)	32	RW	0000_0000
131C	Channel 4 Rx Descriptor List Address (DMA_CH4_RXDESC_LIST_ADDRESS)	32	RW	0000_0000
1320	Channel 4 Tx Descriptor Tail Pointer (DMA_CH4_TXDESC_TAIL_POINTER)	32	RW	0000_0000
1328	Channel 4 Rx Descriptor Tail Pointer (DMA_CH4_RXDESC_TAIL_POINTER)	32	RW	0000_0000
132C	Channel 4 Tx Descriptor Ring Length (DMA_CH4_TXDESC_RING_LENGTH)	32	RW	0000_0000
1330	DMA Channel 4 Receive Control 2 register (DMA_CH4_RX_CONTROL2)	32	RW	0000_0000
1334	Channel 4 Interrupt Enable (DMA_CH4_INTERRUPT_ENABLE)	32	RW	0000_0000
1338	Channel 4 Receive Interrupt Watchdog Timer (DMA_CH4_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000
133C	Channel 4 Slot Function Control and Status (DMA_CH4_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0
1344	Channel 4 Current Application Transmit Descriptor (DMA_CH4_CURRENT_APP_TXDESC)	32	R	0000_0000
134C	Channel 4 Current Application Receive Descriptor (DMA_CH4_CURRENT_APP_RXDESC)	32	R	0000_0000
1354	Channel 4 Current Application Transmit Buffer Address (DMA_CH4_CURRENT_APP_TXBUFFER)	32	R	0000_0000
135C	Channel 4 Current Application Receive Buffer Address (DMA_CH4_CURRENT_APP_RXBUFFER)	32	R	0000_0000
1360	DMA Channel 4 Status (DMA_CH4_STATUS)	32	RW	0000_0000
1364	Channel 4 Missed Frame Counter (DMA_CH4_MISS_FRAME_CNT)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1368	Channel 4 RXP Frames Accepted Counter (DMA_CH4_RXP_ACCEPT_CNT)	32	R	0000_0000

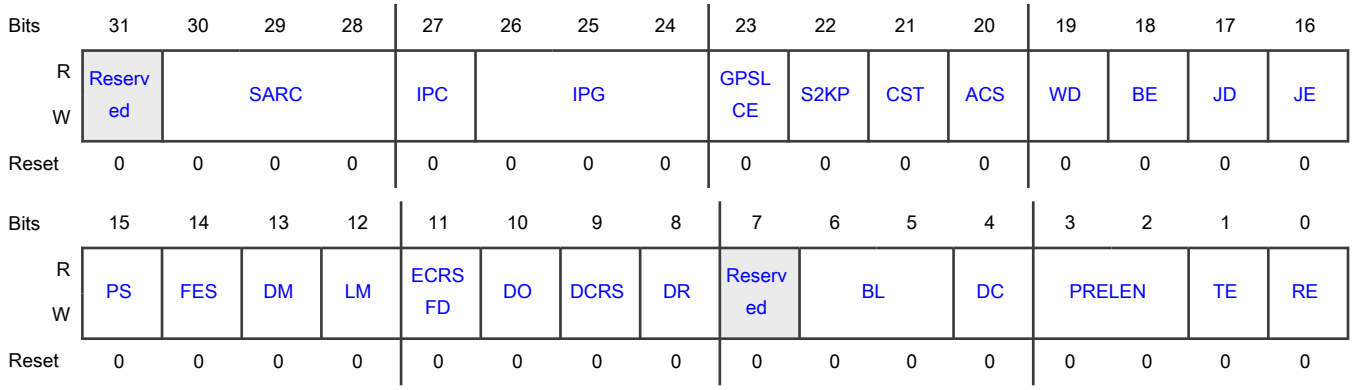
43.8.1.2 MAC Configuration Register (MAC_CONFIGURATION)

The MAC Configuration Register establishes the operating mode of the MAC.

Offset

Register	Offset
MAC_CONFIGURATION	0h

Diagram



Fields

Field	Description
31 —	Reserved
30-28 SARC	Source Address Insertion or Replacement Control This field controls the source address insertion or replacement for all transmitted packets. Bit 30 specifies which MAC Address register (0 or 1) is used for source address insertion or replacement based on the values of Bits[29:28]: 2'b0x: - The mti_sa_ctrl_i and ati_sa_ctrl_i input signals control the SA field generation. 2'b10:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- If Bit 30 is set to 0, the MAC inserts the content of the MAC Address 0 registers in the SA field of all transmitted packets. - If Bit 30 is set to 1 and the Enable MAC Address Register 1 option is selected while configuring the core, the MAC inserts the content of the MAC Address 1 registers in the SA field of all transmitted packets. 2'b11: - If Bit 30 is set to 0, the MAC replaces the content of the MAC Address 0 registers in the SA field of all transmitted packets. - If Bit 30 is set to 1 and the MAC Address Register 1 is enabled, the MAC replaces the content of the MAC Address 1 registers in the SA field of all transmitted packets. Note: - Changes to this field take effect only on the start of a packet. If you write to this register field when a packet is being transmitted, only the subsequent packet can use the updated value, that is, the current packet does not use the updated value. 000 - mti_sa_ctrl_i and ati_sa_ctrl_i input signals control the SA field generation 010 - Contents of MAC Addr-0 inserted in SA field 011 - Contents of MAC Addr-0 replaces SA field 110 - Contents of MAC Addr-1 inserted in SA field 111 - Contents of MAC Addr-1 replaces SA field
27 IPC	Checksum Offload When set, this bit enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking. When this bit is reset, the COE function in the receiver is disabled. The Layer 3 and Layer 4 Packet Filter and Enable Split Header features automatically selects the IPC Full Checksum Offload Engine on the Receive side. When any of these features are enabled, you must set the IPC bit. 0 - IP header/payload checksum checking is disabled 1 - IP header/payload checksum checking is enabled
26-24 IPG	Inter-Packet Gap These bits control the minimum IPG between packets during transmission. Inter-Packet Gap These bits control the minimum IPG between packets during transmission. This range of minimum IPG is valid in full-duplex mode. In the half-duplex mode, the minimum IPG can be configured only for 64-bit times (IPG = 100). Lower values are not considered. When a JAM pattern is being transmitted because of backpressure activation, the MAC does not consider the minimum IPG. The above function (IPG less than 96 bit times) is valid only when EIPGEN bit in MAC_EXT_CONFIGURATION register is reset. When EIPGEN is set, then the minimum IPG

Table continues on the next page...

Table continued from the previous page...

Field	Description
	(greater than 96 bit times) is controlled as per the description given in EIPG field in MAC_EXT_CONFIGURATION register. 000 - 96 bit times IPG 001 - 88 bit times IPG 010 - 80 bit times IPG 011 - 72 bit times IPG 100 - 64 bit times IPG 101 - 56 bit times IPG 110 - 48 bit times IPG 111 - 40 bit times IPG
23 GPSLCE	Giant Packet Size Limit Control Enable When this bit is set, the MAC considers the value in GPSL field in MAC_EXT_CONFIGURATION register to declare a received packet as Giant packet. This field must be programmed to more than 1,518 bytes. Otherwise, the MAC considers 1,518 bytes as giant packet limit. When this bit is reset, the MAC considers a received packet as Giant packet when its size is greater than 1,518 bytes (1522 bytes for tagged packet). The watchdog timeout limit, Jumbo Packet Enable and 2K Packet Enable have higher precedence over this bit, that is the MAC considers a received packet as Giant packet when its size is greater than 9,018 bytes (9,022 bytes for tagged packet) with Jumbo Packet Enabled and greater than 2,000 bytes with 2K Packet Enabled. The watchdog timeout, if enabled, terminates the received packet when watchdog limit is reached. Therefore, the programmed giant packet limit should be less than the watchdog limit to get the giant packet status. 0 - Giant Packet Size Limit Control is disabled 1 - Giant Packet Size Limit Control is enabled
22 S2KP	IEEE 802. IEEE 802.3as Support for 2K Packets When this bit is set, the MAC considers all packets with up to 2,000 bytes length as normal packets. When the JE bit is not set, the MAC considers all received packets of size more than 2K bytes as Giant packets. When this bit is reset and the JE bit is not set, the MAC considers all received packets of size more than 1,518 bytes (1,522 bytes for tagged) as giant packets. For more information about how the setting of this bit and the JE bit impact the Giant packet status, see the Table, Gaint Packet Status based on S2KP and JE Bits. Note: When the JE bit is set, setting this bit has no effect on the giant packet status. 0 - Support upto 2K packet is disabled 1 - Support upto 2K packet is Enabled
21 CST	CRC stripping for Type packets When this bit is set, the last four bytes (FCS) of all packets of Ether type (type field greater than 1,536) are stripped and dropped before forwarding the packet to the application.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	CRC stripping for Type packets When this bit is set, the last four bytes (FCS) of all packets of Ether type (type field greater than 1,536) are stripped and dropped before forwarding the packet to the application. Note: For information about how the settings of the ACS bit and this bit impact the packet length, see the Table, Packet Length based on the CST and ACS Bits. 0 - CRC stripping for Type packets is disabled 1 - CRC stripping for Type packets is enabled
20 ACS	Automatic Pad or CRC Stripping When this bit is set, the MAC strips the Pad or FCS field on the incoming packets only if the value of the length field is less than 1,536 bytes. Automatic Pad or CRC Stripping When this bit is set, the MAC strips the Pad or FCS field on the incoming packets only if the value of the length field is less than 1,536 bytes. All received packets with length field greater than or equal to 1,536 bytes are passed to the application without stripping the Pad or FCS field. When this bit is reset, the MAC passes all incoming packets to the application, without any modification. Note: For information about how the settings of CST bit and this bit impact the packet length, see the Table, Packet Length based on the CST and ACS Bit . 0 - Automatic Pad or CRC Stripping is disabled 1 - Automatic Pad or CRC Stripping is enabled
19 WD	Watchdog Disable When this bit is set, the MAC disables the watchdog timer on the receiver. The MAC can receive packets of up to 16,383 bytes. When this bit is reset, the MAC does not allow more than 2,048 bytes (10,240 if JE is set high) of the packet being received. The MAC cuts off any bytes received after 2,048 bytes. 0 - Watchdog is enabled 1 - Watchdog is disabled
18 BE	Packet Burst Enable When this bit is set, the MAC allows packet bursting during transmission in the GMII half-duplex mode. 0 - Packet Burst is disabled 1 - Packet Burst is enabled
17 JD	Jabber Disable When this bit is set, the MAC disables the jabber timer on the transmitter. The MAC can transfer packets of up to 16,383 bytes. When this bit is reset, if the application sends more than 2,048 bytes of data (10,240 if JE is set high) during transmission, the MAC does not send rest of the bytes in that packet. 0 - Jabber is enabled 1 - Jabber is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
16 JE	Jumbo Packet Enable When this bit is set, the MAC allows jumbo packets of 9,018 bytes (9,022 bytes for VLAN tagged packets) without reporting a giant packet error in the Rx packet status. 0 - Jumbo packet is disabled 1 - Jumbo packet is enabled
15 PS	Port Select This bit selects the Ethernet line speed. This bit, along with Bit 14, selects the exact line speed. In the 10/100 Mbps-only (always 1) or 1000 Mbps-only (always 0) configurations, this bit is read-only (RO) with appropriate value. In default 10/100/1000 Mbps configurations, this bit is read-write (R/W). The mac_speed_o[1] signal reflects the value of this bit. 0 - For 1000 or 2500 Mbps operations 1 - For 10 or 100 Mbps operations
14 FES	Speed This bit selects the speed mode. The mac_speed_o[0] signal reflects the value of this bit. 0 - 10 Mbps when PS bit is 1 and 1 Gbps when PS bit is 0 1 - 100 Mbps when PS bit is 1 and 2.5 Gbps when PS bit is 0
13 DM	Duplex Mode When this bit is set, the MAC operates in the full-duplex mode in which it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in the full-duplex-only configurations. 0 - Half-duplex mode 1 - Full-duplex mode
12 LM	Loopback Mode When this bit is set, the MAC operates in the loopback mode at GMII. The (G)MII Rx clock input (clk_rx_i) is required for the loopback to work properly. This is because the Tx clock is not internally looped back. 0 - Loopback is disabled 1 - Loopback is enabled
11 ECRSFD	Enable Carrier Sense Before Transmission in Full-Duplex Mode When this bit is set, the MAC transmitter checks the CRS signal before packet transmission in the full-duplex mode. The MAC starts the transmission only when the CRS signal is low. When this bit is reset, the MAC transmitter ignores the status of the CRS signal. 0 - ECRSFD is disabled 1 - ECRSFD is enabled
10	Disable Receive Own

Table continues on the next page...

Table continued from the previous page...

Field	Description
DO	When this bit is set, the MAC disables the reception of packets when the gmii_txen_o is asserted in the half-duplex mode. When this bit is reset, the MAC receives all packets given by the PHY. This bit is not applicable in the full-duplex mode. 0 - Enable Receive Own 1 - Disable Receive Own
9 DCRS	Disable Carrier Sense During Transmission When this bit is set, the MAC transmitter ignores the (G)MII CRS signal during packet transmission in the half-duplex mode. As a result, no errors are generated because of Loss of Carrier or No Carrier during transmission. When this bit is reset, the MAC transmitter generates errors because of Carrier Sense. The MAC can even abort the transmission. 0 - Enable Carrier Sense During Transmission 1 - Disable Carrier Sense During Transmission
8 DR	Disable Retry When this bit is set, the MAC attempts only one transmission. When a collision occurs on the GMII interface, the MAC ignores the current packet transmission and reports a Packet Abort with excessive collision error in the Tx packet status. When this bit is reset, the MAC retries based on the settings of the BL field. This bit is applicable only in the half-duplex mode. 0 - Enable Retry 1 - Disable Retry
7 —	Reserved
6-5 BL	Back-Off Limit The back-off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000/2500 Mbps; 512 bit times for 10/100 Mbps) for which the MAC waits before rescheduling a transmission attempt during retries after a collision. n = retransmission attempt. The random integer r takes the value in the range 0 <= r < 2^k This bit is applicable only in the half-duplex mode. 00 - k = min(n,10) 01 - k = min(n,8) 10 - k = min(n,4) 11 - k = min(n,1)
4	Deferral Check

Table continues on the next page...

Table continued from the previous page...

Field	Description
DC	When this bit is set, the deferral check function is enabled in the MAC. The MAC issues a Packet Abort status, along with the excessive deferral error bit set in the Tx packet status, when the Tx state machine is deferred for more than 24,288 bit times in 10 or 100 Mbps mode. If the MAC is configured for 1000/2500 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but it is prevented because of an active carrier sense signal (CRS) on GMII. The defer time is not cumulative. For example, if the transmitter defers for 10,000 bit times because the CRS signal is active and the CRS signal becomes inactive, the transmitter transmits and collision happens. Because of collision, the transmitter needs to back off and then defer again after back off completion. In such a scenario, the deferral timer is reset to 0, and it is restarted. When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive. This bit is applicable only in the half-duplex mode. 0 - Deferral check function is disabled 1 - Deferral check function is enabled
3-2 PRELEN	Preamble Length for Transmit packets These bits control the number of preamble bytes that are added to the beginning of every Tx packet. The preamble reduction occurs only when the MAC is operating in the full-duplex mode. 00 - 7 bytes of preamble 01 - 5 bytes of preamble 10 - 3 bytes of preamble 11 - Reserved
1 TE	Transmitter Enable When this bit is set, the Tx state machine of the MAC is enabled for transmission on the GMII interface. When this bit is reset, the MAC Tx state machine is disabled after it completes the transmission of the current packet. The Tx state machine does not transmit any more packets. 0 - Transmitter is disabled 1 - Transmitter is enabled
0 RE	Receiver Enable When this bit is set, the Rx state machine of the MAC is enabled for receiving packets from the GMII interface. When this bit is reset, the MAC Rx state machine is disabled after it completes the reception of the current packet. The Rx state machine does not receive any more packets from the GMII interface. 0 - Receiver is disabled 1 - Receiver is enabled

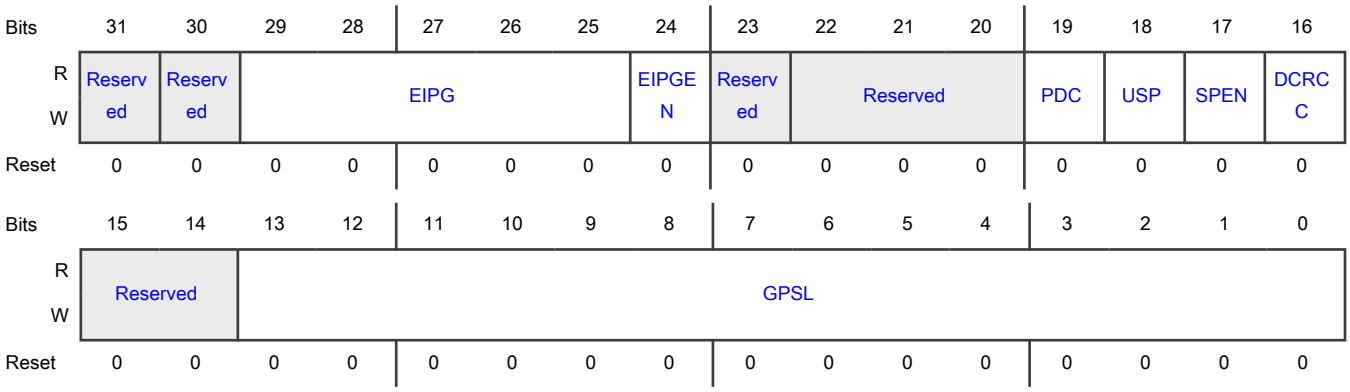
43.8.1.3 MAC Extended Configuration Register (MAC_EXT_CONFIGURATION)

The MAC Extended Configuration Register establishes the operating mode of the MAC.

Offset

Register	Offset
MAC_EXT_CONFIGURATION	4h

Diagram



Fields

Field	Description
31 —	Reserved
30 —	Reserved
29-25 EIPG	Extended Inter-Packet Gap The value in this field is applicable when the EIPGEN bit is set. This field (as Most Significant bits), along with IPG field in CONFIGURATION register, gives the minimum IPG greater than 96 bit times in steps of 8 bit times: {EIPG, IPG} 8'h00 - 104 bit times 8'h01 - 112 bit times 8'h02 - 120 bit times ---------- 8'hFF - 2144 bit times
24 EIPGEN	Extended Inter-Packet Gap Enable When this bit is set, the MAC interprets EIPG field and IPG field in CONFIGURATION register together as minimum IPG greater than 96 bit times in steps of 8 bit times.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is reset, the MAC ignores EIPG field and interprets IPG field in CONFIGURATION register as minimum IPG less than or equal to 96 bit times in steps of 8 bit times. Note: The extended Inter-Packet Gap feature must be enabled when operating in Full-Duplex mode only. There may be undesirable effects on back-pressure function and frame transmission if it is enabled in Half-Duplex mode. 0 - Extended Inter-Packet Gap is disabled 1 - Extended Inter-Packet Gap is enabled
23 —	Reserved
22-20 —	Reserved
19 PDC	Packet Duplication Control When this bit is set, the received packet with Multicast/Broadcast Destination address is routed to multiple Receive DMA Channels. The Receive DMA Channels is identified by the DCS field of MAC_ADDRESS(#i)_HIGH register corresponding to the MAC Address register that matches the Multicast/Broadcast Destination address in the received packet. The DCS field is interpreted to be a one-hot value, each bit corresponding to the Receive DMA Channel. When this bit is reset, the received packet is routed to single Receive DMA Channel. The Receive DMA Channel is identified by the DCS field of ADDRESS(#i)_High register corresponding to the MAC Address register that matches the Destination address in the received packet. The DCS field is interpreted as a binary value. 0 - Packet Duplication Control is disabled 1 - Packet Duplication Control is enabled
18 USP	Unicast Slow Protocol Packet Detect When this bit is set, the MAC detects the Slow Protocol packets with unicast address of the station specified in the MAC_ADDRESS0_HIGH and MAC_ADDRESS0_LOW registers. The MAC also detects the Slow Protocol packets with the Slow Protocols multicast address (01-80-C2-00-00-02). When this bit is reset, the MAC detects only Slow Protocol packets with the Slow Protocol multicast address specified in the IEEE 802.3-2015, Section 5. 0 - Unicast Slow Protocol Packet Detection is disabled 1 - Unicast Slow Protocol Packet Detection is enabled
17 SPEN	Slow Protocol Detection Enable When this bit is set, MAC processes the Slow Protocol packets (Ether Type 0x8809) and provides the Rx status. The MAC discards the Slow Protocol packets with invalid sub-types. When this bit is reset, the MAC forwards all error-free Slow Protocol packets to the application. The MAC considers such packets as normal Type packets. 0 - Slow Protocol Detection is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Slow Protocol Detection is enabled
16 DCRCC	Disable CRC Checking for Received Packets When this bit is set, the MAC receiver does not check the CRC field in the received packets. When this bit is reset, the MAC receiver always checks the CRC field in the received packets. 0 - CRC Checking is enabled 1 - CRC Checking is disabled
15-14 —	Reserved
13-0 GPSL	Giant Packet Size Limit If the received packet size is greater than the value programmed in this field in units of bytes, the MAC declares the received packet as Giant packet. The value programmed in this field must be greater than or equal to 1,518 bytes. Any other programmed value is considered as 1,518 bytes. For VLAN tagged packets, the MAC adds 4 bytes to the programmed value. When the Enable Double VLAN Processing option is selected, the MAC adds 8 bytes to the programmed value for double VLAN tagged packets. The value in this field is applicable when the GPSLCE bit is set in MAC_CONFIGURATION register.

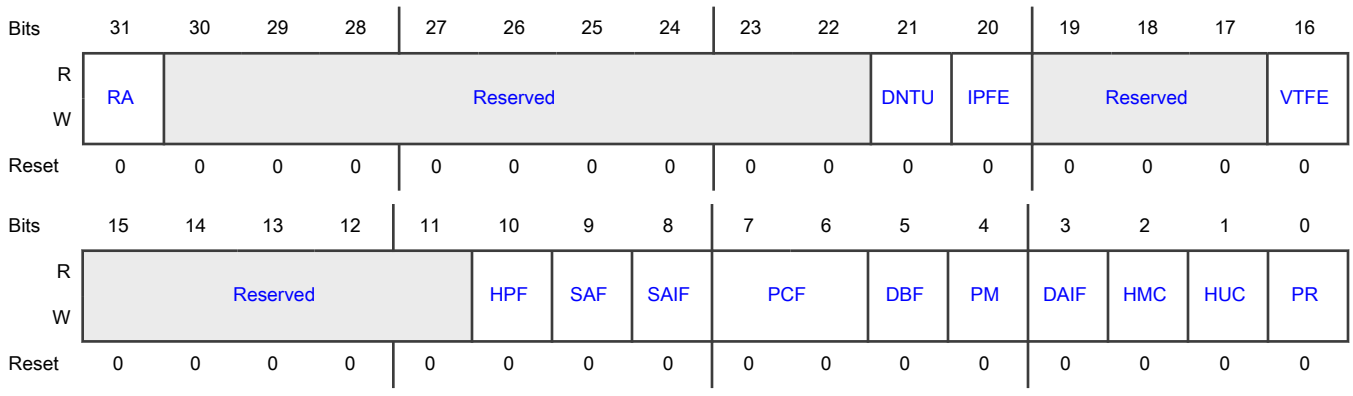
43.8.1.4 MAC Packet Filter (MAC_PACKET_FILTER)

The MAC Packet Filter register contains the filter controls for receiving packets. Some of the controls from this register go to the address check block of the MAC which performs the first level of address filtering. The second level of filtering is performed on the incoming packet based on other controls such as Pass Bad Packets and Pass Control Packets.

Offset

Register	Offset
MAC_PACKET_FILTER	8h

Diagram



Fields

Field	Description
31 RA	Receive All When this bit is set, the MAC Receiver module passes all received packets to the application, irrespective of whether they pass the address filter or not. The result of the SA or DA filtering is updated (pass or fail) in the corresponding bit in the Rx Status Word. When this bit is reset, the Receiver module passes only those packets to the application that pass the SA or DA address filter. 0 - Receive All is disabled 1 - Receive All is enabled
30-22 —	Reserved
21 DNTU	Drop Non-TCP/UDP over IP Packets When this bit is set, the MAC drops the non-TCP or UDP over IP packets. The MAC forward only those packets that are processed by the Layer 4 filter. When this bit is reset, the MAC forwards all non-TCP or UDP over IP packets. 0 - Forward Non-TCP/UDP over IP Packets 1 - Drop Non-TCP/UDP over IP Packets
20 IPFE	Layer 3 and Layer 4 Filter Enable When this bit is set, the MAC drops packets that do not match the enabled Layer 3 and Layer 4 filters. If Layer 3 or Layer 4 filters are not enabled for matching, this bit does not have any effect. When this bit is reset, the MAC forwards all packets irrespective of the match status of the Layer 3 and Layer 4 fields. 0 - Layer 3 and Layer 4 Filters are disabled 1 - Layer 3 and Layer 4 Filters are enabled
19-17 —	Reserved
16 VTFE	VLAN Tag Filter Enable When this bit is set, the MAC drops the VLAN tagged packets that do not match the VLAN Tag. When this bit is reset, the MAC forwards all packets irrespective of the match status of the VLAN Tag. 0 - VLAN Tag Filter is disabled 1 - VLAN Tag Filter is enabled
15-11 —	Reserved
10 HPF	Hash or Perfect Filter

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the address filter passes a packet if it matches either the perfect filtering or hash filtering as set by the HMC or HUC bit. When this bit is reset and the HUC or HMC bit is set, the packet is passed only if it matches the Hash filter. 0 - Hash or Perfect Filter is disabled 1 - Hash or Perfect Filter is enabled
9 SAF	Source Address Filter Enable When this bit is set, the MAC compares the SA field of the received packets with the values programmed in the enabled SA registers. If the comparison fails, the MAC drops the packet. When this bit is reset, the MAC forwards the received packet to the application with updated SAF bit of the Rx Status depending on the SA address comparison. Note: According to the IEEE specification, Bit 47 of the SA is reserved. However, in DWC_ether_qos, the MAC compares all 48 bits. The software driver should take this into consideration while programming the MAC address registers for SA. 0 - SA Filtering is disabled 1 - SA Filtering is enabled
8 SAIF	SA Inverse Filtering When this bit is set, the Address Check block operates in the inverse filtering mode for SA address comparison. If the SA of a packet matches the values programmed in the SA registers, it is marked as failing the SA Address filter. When this bit is reset, if the SA of a packet does not match the values programmed in the SA registers, it is marked as failing the SA Address filter. 0 - SA Inverse Filtering is disabled 1 - SA Inverse Filtering is enabled
7-6 PCF	Pass Control Packets These bits control the forwarding of all control packets (including unicast and multicast Pause packets). 00 - MAC filters all control packets from reaching the application 01 - MAC forwards all control packets except Pause packets to the application even if they fail the Address filter 10 - MAC forwards all control packets to the application even if they fail the Address filter 11 - MAC forwards the control packets that pass the Address filter
5 DBF	Disable Broadcast Packets When this bit is set, the AFM module blocks all incoming broadcast packets. In addition, it overrides all other filter settings. When this bit is reset, the AFM module passes all received broadcast packets. 0 - Enable Broadcast Packets 1 - Disable Broadcast Packets

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 PM	Pass All Multicast When this bit is set, it indicates that all received packets with a multicast destination address (first bit in the destination address field is '1') are passed. When this bit is reset, filtering of multicast packet depends on HMC bit. 0 - Pass All Multicast is disabled 1 - Pass All Multicast is enabled
3 DAIF	DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast packets. When this bit is reset, normal filtering of packets is performed. 0 - DA Inverse Filtering is disabled 1 - DA Inverse Filtering is enabled
2 HMC	Hash Multicast When this bit is set, the MAC performs the destination address filtering of received multicast packets according to the hash table. When this bit is reset, the MAC performs the perfect destination address filtering for multicast packets, that is, it compares the DA field with the values programmed in DA registers. 0 - Hash Multicast is disabled 1 - Hash Multicast is enabled
1 HUC	Hash Unicast When this bit is set, the MAC performs the destination address filtering of unicast packets according to the hash table. When this bit is reset, the MAC performs a perfect destination address filtering for unicast packets, that is, it compares the DA field with the values programmed in DA registers. 0 - Hash Unicast is disabled 1 - Hash Unicast is enabled
0 PR	Promiscuous Mode When this bit is set, the Address Filtering module passes all incoming packets irrespective of the destination or source address. The SA or DA Filter Fails status bits of the Rx Status Word are always cleared when PR is set. 0 - Promiscuous Mode is disabled 1 - Promiscuous Mode is enabled

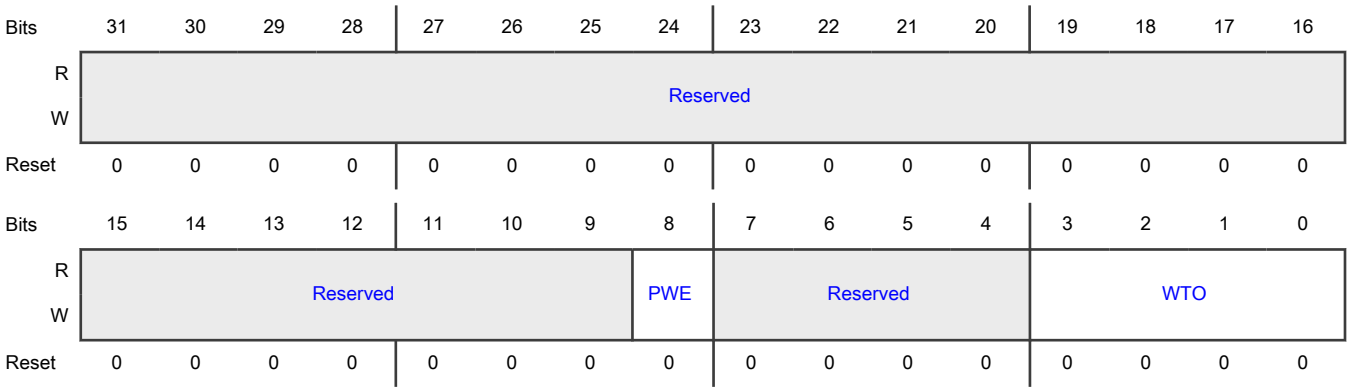
43.8.1.5 Watchdog Timeout (MAC_WATCHDOG_TIMEOUT)

The Watchdog Timeout register controls the watchdog timeout for received packets.

Offset

Register	Offset
MAC_WATCHDOG_TIMEOUT	Ch

Diagram



Fields

Field	Description
31-9 —	Reserved
8 PWE	Programmable Watchdog Enable When this bit is set and the WD bit of the MAC_CONFIGURATION register is reset, the WTO field is used as watchdog timeout for a received packet. When this bit is cleared, the watchdog timeout for a received packet is controlled by setting of WD and JE bits in MAC_CONFIGURATION register. 0 - Programmable Watchdog is disabled 1 - Programmable Watchdog is enabled
7-4 —	Reserved
3-0 WTO	Watchdog Timeout When the PWE bit is set and the WD bit of the MAC_CONFIGURATION register is reset, this field is used as watchdog timeout for a received packet. If the length of a received packet exceeds the value of this field, such packet is terminated and declared as an error packet. Note: When the PWE bit is set, the value in this field should be more than 1,522 (0x05F2). Otherwise, the IEEE 802.3-specified valid tagged packets are declared as error packets and then dropped. 0000 - 2 KB 0001 - 3 KB 0010 - 4 KB

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0011 - 5 KB 0100 - 6 KB 0101 - 7 KB 0110 - 8 KB 0111 - 9 KB 1000 - 10 KB 1001 - 11 KB 1010 - 12 KB 1011 - 13 KB 1100 - 14 KB 1101 - 15 KB 1110 - 16383 Bytes 1111 - Reserved

43.8.1.6 MAC Hash Table Register 0 (MAC_HASH_TABLE_REG0)

The Hash Table Register 0 contains the first 32 bits of the hash table, when the width of the hash table is 128 or 256 bits.

The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper six (seven in 128-bit Hash or eight in 256-bit Hash) bits of the CRC register are used to index the content of the Hash table. The most significant bits determines the register to be used (Hash Table Register X), and the least significant five bits determine the bit within the register. For example, a hash value of 6'b100000 (in 64-bit Hash) selects Bit 0 of the Hash Table Register 1, a value of 7b'1110000 (in 128-bit Hash) selects Bit 16 of the Hash Table Register 3 and a value of 8b'10111111 (in 256-bit Hash) selects Bit 31 of the Hash Table Register 5.

The hash value of the destination address is calculated in the following way:

- Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).

- Perform bitwise reversal for the value obtained in Step 1.

- Take the upper 6 (or 7 or 8) bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1'b1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC_Packet_Filter, all multicast packets are accepted regardless of the multicast hash values.

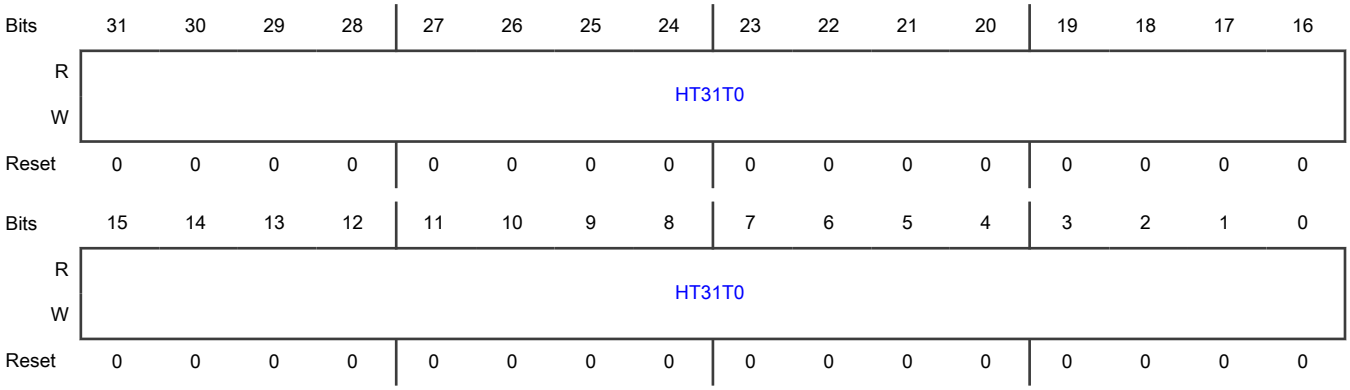
If the Hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Hash Table Register X registers are written.

If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_HASH_TABLE_REG0	10h

Diagram



Fields

Field	Description
31-0 HT31T0	MAC Hash Table First 32 Bits This field contains the first 32 Bits [31:0] of the Hash table.

43.8.1.7 MAC Hash Table Register 1 (MAC_HASH_TABLE_REG1)

The Hash Table Register 1 contains the second 32 bits of the hash table.

The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper six (seven in 128-bit Hash or eight in 256-bit Hash) bits of the CRC register are used to index the content of the Hash table. The most significant bits determines the register to be used (Hash Table Register X), and the least significant five bits determine the bit within the register. For example, a hash value of 6'b100000 (in 64-bit Hash) selects Bit 0 of the Hash Table Register 1, a value of 7b'1110000 (in 128-bit Hash) selects Bit 16 of the Hash Table Register 3 and a value of 8b'10111111 (in 256-bit Hash) selects Bit 31 of the Hash Table Register 5.

The hash value of the destination address is calculated in the following way:

- Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).

- Perform bitwise reversal for the value obtained in Step 1.

- Take the upper 6 (or 7 or 8) bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1'b1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC_Packet_Filter, all multicast packets are accepted regardless of the multicast hash values.

If the Hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Hash Table Register X registers are written.

If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_HASH_TABLE_REG1	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HT63T32															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HT63T32															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 HT63T32	MAC Hash Table Second 32 Bits This field contains the second 32 Bits [63:32] of the Hash table.

43.8.1.8 MAC VLAN Tag Control (MAC_VLAN_TAG_CTRL)

This register is the redefined format of the MAC VLAN Tag Register. It is used for indirect addressing. It contains the address offset, command type and Busy Bit for CSR access of the Per VLAN Tag registers.

Offset

Register	Offset
MAC_VLAN_TAG_CTRL	50h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EIVLRXS	Reserved	EIVLS		ERIVLT	EDVLP	VTHM	EVLRXS	Reserved	EVLS			Reserved		ESVL	VTIM	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								OFS						CT	OB
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 EIVLRXS	Enable Inner VLAN Tag in Rx Status When this bit is set, the MAC provides the inner VLAN Tag in the Rx status. When this bit is reset, the MAC does not provide the inner VLAN Tag in Rx status. 0 - Inner VLAN Tag in Rx status is disabled 1 - Inner VLAN Tag in Rx status is enabled
30 —	Reserved
29-28 EIVLS	Enable Inner VLAN Tag Stripping on Receive This field indicates the stripping operation on inner VLAN Tag in received packet. 00 - Do not strip 01 - Strip if VLAN filter passes 10 - Strip if VLAN filter fails 11 - Always strip
27 ERIVLT	ERIVLT 0 - Inner VLAN tag is disabled 1 - Inner VLAN tag is enabled
26 EDVLP	Enable Double VLAN Processing When this bit is set, the MAC enables processing of up to two VLAN Tags on Tx and Rx (if present). When this bit is reset, the MAC enables processing of up to one VLAN Tag on Tx and Rx (if present). 0 - Double VLAN Processing is disabled 1 - Double VLAN Processing is enabled
25 VTHM	VLAN Tag Hash Table Match Enable When this bit is set, the most significant four bits of CRC of VLAN Tag (ones-complement of most significant four bits of CRC of VLAN Tag when ETV bit is reset) are used to index the content of the MAC_VLAN_Hash_Table register. A value of 1 in the VLAN Hash Table register, corresponding to the index, indicates that the packet matched the VLAN hash table. When the ETV bit is set, the CRC of the 12-bit VLAN Identifier (VID) is used for comparison. When the ETV bit is reset, the ones-complement of the CRC of the 16-bit VLAN tag is used for comparison. When this bit is reset, the VLAN Hash Match operation is not performed. 0 - VLAN Tag Hash Table Match is disabled 1 - VLAN Tag Hash Table Match is enabled
24 EVLRXS	Enable VLAN Tag in Rx status When this bit is set, MAC provides the outer VLAN Tag in the Rx status. When this bit is reset, the MAC does not provide the outer VLAN Tag in Rx status.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - VLAN Tag in Rx status is disabled 1 - VLAN Tag in Rx status is enabled
23 —	Reserved
22-21 EVLS	Enable VLAN Tag Stripping on Receive This field indicates the stripping operation on the outer VLAN Tag in received packet. 00 - Do not strip 01 - Strip if VLAN filter passes 10 - Strip if VLAN filter fails 11 - Always strip
20-19 —	Reserved
18 ESVL	Enable S-VLAN When this bit is set, the MAC transmitter and receiver consider the S-VLAN packets (Type = 0x88A8) as valid VLAN tagged packets. 0 - S-VLAN is disabled 1 - S-VLAN is enabled
17 VTIM	VLAN Tag Inverse Match Enable When this bit is set, this bit enables the VLAN Tag inverse matching. The packets without matching VLAN Tag are marked as matched. When reset, this bit enables the VLAN Tag perfect matching. The packets with matched VLAN Tag are marked as matched. 0 - VLAN Tag Inverse Match is disabled 1 - VLAN Tag Inverse Match is enabled
16-7 —	Reserved
6-2 OFS	Offset This field holds the address offset of the MAC VLAN Tag Filter Register which the application is trying to access. The width of the field depends on the number of MAC VLAN Tag Registers enabled.
1 CT	Command Type This bit indicates if the current register access is a read or a write. When set, it indicate a read operation. When reset, it indicates a write operation. 0 - Write operation 1 - Read operation

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 OB	Operation Busy This bit is set along with a read or write command for initiating the indirect access to per VLAN Tag Filter register. This bit is reset when the read or write command to per VLAN Tag Filter indirect access register is complete. The next indirect register access can be initiated only after this bit is reset. During a write operation, the bit is reset only after the data has been written into the Per VLAN Tag register. During a read operation, the data should be read from the VLAN_TAG_DATA register only after this bit is reset. 0 - Operation Busy is disabled 1 - Operation Busy is enabled

43.8.1.9 MAC VLAN Tag Data (MAC_VLAN_TAG_DATA)

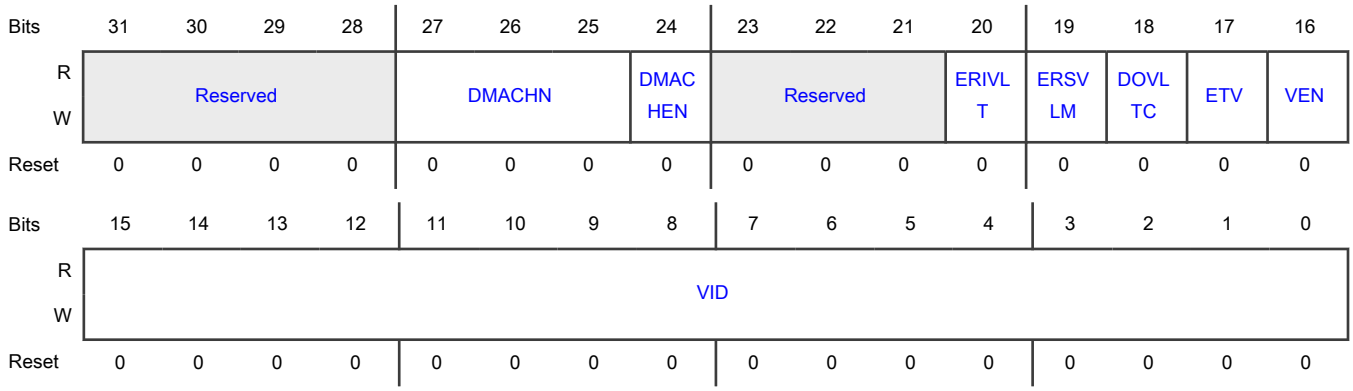
This register holds the read/write data for Indirect Access of the Per VLAN Tag registers. During the read access, this field contains valid read data only after the OB bit is reset.

During the write access, this field should be valid prior to setting the OB bit in the MAC_VLAN_Tag_Ctrl Register.

Offset

Register	Offset
MAC_VLAN_TAG_DATA	54h

Diagram



Fields

Field	Description
31-28 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
27-25 DMACHN	DMA Channel Number The DMA Channel number to which the VLAN Tagged Frame is to be routed if it passes this VLAN Tag Filter is programmed in this field. If the Routing based on VLAN Tag Filter is not necessary, this field need not be programmed.
24 DMACHEN	DMA Channel Number Enable This bit is the Enable for the DMA Channel Number value programmed in the field DMACH. When this bit is reset, the Routing does not occur based on VLAN Filter result. The frame is routed based on DA Based DMA Channel Routing. 0 - DMA Channel Number is disabled 1 - DMA Channel Number is enabled
23-21 —	Reserved
20 ERIVLT	Enable Inner VLAN Tag Comparison This bit is valid only when VLAN Tag Enable of the Filter is set. When this bit and the EDVLP field are set, the MAC receiver enables operation on the inner VLAN Tag (if present). When this bit is reset, the MAC receiver enables operation on the outer VLAN Tag (if present). 0 - Inner VLAN tag comparison is disabled 1 - Inner VLAN tag comparison is enabled
19 ERSVLM	Enable S-VLAN Match for received Frames This bit is valid only when VLAN Tag Enable of the Filter is set. When this bit is set, the MAC receiver enables filtering or matching for S-VLAN (Type = 0x88A8) packets. When this bit is reset, the MAC receiver enables filtering or matching for C-VLAN (Type = 0x8100) packets. 0 - Receive S-VLAN Match is disabled 1 - Receive S-VLAN Match is enabled
18 DOVLTC	Disable VLAN Type Comparison This bit is valid only when VLAN Tag Enable of the Filter is set. When this bit is set, the MAC does not check whether the VLAN Tag specified by the Enable Inner VLAN Tag Comparison bit is of type S-VLAN or C-VLAN. When this bit is reset, the MAC filters or matches the VLAN Tag specified by the Enable Inner VLAN Tag Comparison bit only when VLAN Tag type is similar to the one specified by the Enable S-VLAN Match for received Frames bit. 0 - VLAN type comparison is enabled 1 - VLAN type comparison is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
17 ETV	12bits or 16bits VLAN comparison This bit is valid only when VEN of the Filter is set. When this bit is set, a 12-bit VLAN identifier is used for comparing and filtering instead of the complete 16-bit VLAN tag. Bits [11:0] of VLAN tag are compared with the corresponding field in the received VLAN-tagged packet. 0 - 16 bit VLAN comparison 1 - 12 bit VLAN comparison
16 VEN	VLAN Tag Enable This bit is used to enable or disable the VLAN Tag. When this bit is set, the MAC compares the VLAN Tag of received packet with the VLAN Tag ID. When this bit is reset, no comparison is performed irrespective of the programming of the other fields. 0 - VLAN Tag is disabled 1 - VLAN Tag is enabled
15-0 VID	VLAN Tag ID This field holds the VLAN Tag value which is used by the MAC for perfect comparison. It is valid when VLAN Tag Enable is set.

43.8.1.10 MAC VLAN Hash Table (MAC_VLAN_HASH_TABLE)

When VTHM bit of the MAC_VLAN_TAG register is set, the 16-bit VLAN Hash Table register is used for group address filtering based on the VLAN tag. For hash filtering, the content of the 16-bit VLAN tag or 12-bit VLAN ID (based on the ETV bit of MAC_VLAN_TAG Register) in the incoming packet is passed through the CRC logic. When ETV bit of MAC_VLAN_TAG register is set, the upper four bits of the calculated CRC are used to index the contents of the VLAN Hash table. When ETV bit of MAC_VLAN_TAG register is reset, the ones-complement of upper four bits of the calculated CRC are used to index the contents of the VLAN Hash table. For example, when ETV bit is set a hash value of 4b'1000 selects Bit 8 of the VLAN Hash table, whereas when ETV bit is reset a hash value of 4b'1000 selects Bit 7 of the VLAN Hash table.

The hash value of the destination address is calculated in the following way:

- Calculate the 32-bit CRC for the VLAN tag or ID (For steps to calculate CRC32, see Section 3.2.8 of IEEE 802.3).

- Perform bitwise reversal for the value obtained in step 1.

- Take the upper four bits from the value obtained in step 2.

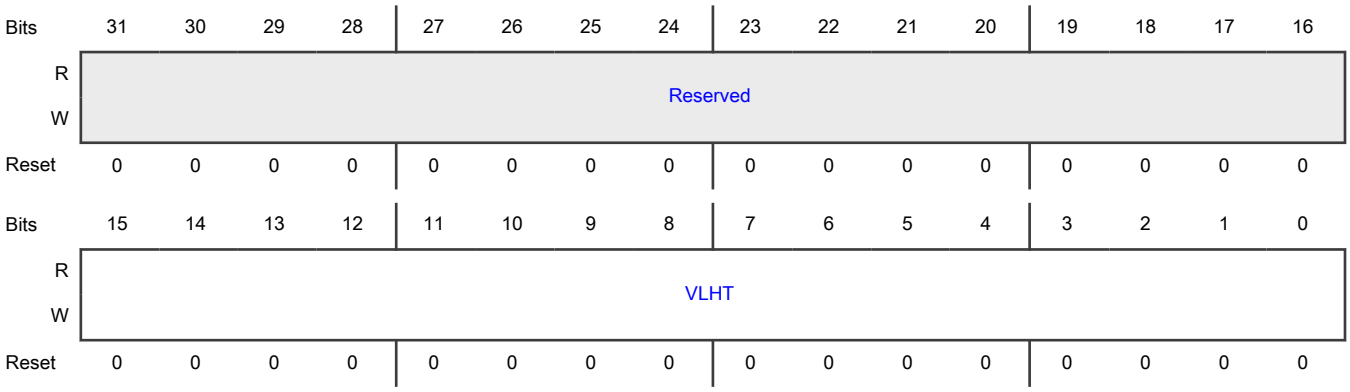
If the VLAN hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[15:8] (in little-endian mode) or Bits[7:0] (in big-endian mode) of this register are written.

- If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_VLAN_HASH_TABLE	58h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 VLHT	VLAN Hash Table This field contains the 16-bit VLAN Hash Table.

43.8.1.11 VLAN Tag Inclusion or Replacement (MAC_VLAN_INCL)

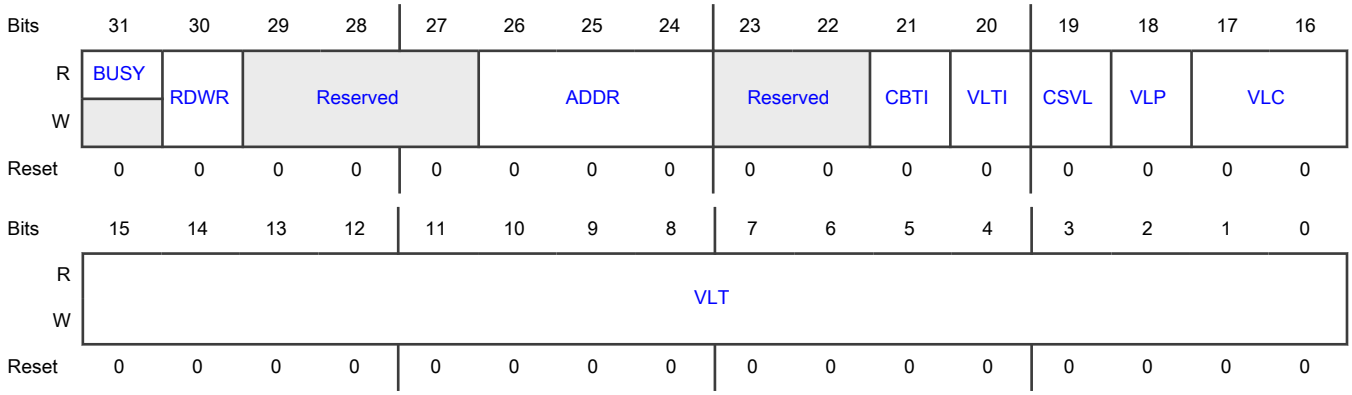
The VLAN Tag Inclusion or Replacement register contains the VLAN tag for insertion or replacement in the Transmit packets.

The VLAN Tag Inclusion or Replacement register contains the VLAN tag for insertion or replacement in the Transmit packets. It also contains the VLAN tag insertion controls.

Offset

Register	Offset
MAC_VLAN_INCL	60h

Diagram



Fields

Field	Description
31 BUSY	Busy This bit indicates the status of the read/write operation of indirect access to the queue/channel specific VLAN inclusion register. For write operation write to a register is complete when this bit is reset. For read operation the read data is valid when the bit is reset. The application must make sure that this bit is reset before attempting subsequent access to this register. 0 - Busy status not detected 1 - Busy status detected
30 RDWR	Read write control This bit controls the read or write operation for indirectly accessing the queue/channel specific VLAN Inclusion register. When set indicates write operation and when reset indicates read operation. This does not have any effect when CBTI is reset. 0 - Read operation of indirect access 1 - Write operation of indirect access
29-27 —	Reserved
26-24 ADDR	Address This field selects one of the queue/channel specific VLAN Inclusion register for read/write access. This does not have any effect when CBTI is reset.
23-22 —	Reserved
21	Channel based tag insertion

Table continues on the next page...

Table continued from the previous page...

Field	Description
CBTI	When this bit is set, outer VLAN tag is inserted for every packets transmitted by the MAC. The tag value is taken from the queue/channel specific VLAN tag register. The VLTI, VLP, VLC, and VLT fields of this register are ignored when this bit is set. When this bit is set, a write operation to byte 3 of this register initiates the read/write access to the indirect register. When reset, outer VLAN operation is based on the setting of VLTI, VLP, VLC and VLT fields of this register. 0 - Channel based tag insertion is disabled 1 - Channel based tag insertion is enabled
20 VLTI	VLAN Tag Input When this bit is set, it indicates that the VLAN tag to be inserted or replaced in Tx packet should be taken from: - The Tx descriptor 0 - VLAN Tag Input is disabled 1 - VLAN Tag Input is enabled
19 CSVL	C-VLAN or S-VLAN When this bit is set, S-VLAN type (0x88A8) is inserted or replaced in the 13th and 14th bytes of transmitted packets. When this bit is reset, C-VLAN type (0x8100) is inserted or replaced in the 13th and 14th bytes of transmitted packets. 0 - C-VLAN type (0x8100) is inserted or replaced 1 - S-VLAN type (0x88A8) is inserted or replaced
18 VLP	VLAN Priority Control When this bit is set, the control bits[17:16] are used for VLAN deletion, insertion, or replacement. When this bit is reset, the mti_vlan_ctrl_i control input is used and bits[17:16] are ignored. 0 - VLAN Priority Control is disabled 1 - VLAN Priority Control is enabled
17-16 VLC	VLAN Tag Control in Transmit Packets - 2'b00: No VLAN tag deletion, insertion, or replacement - 2'b01: VLAN tag deletion The MAC removes the VLAN type (bytes 13 and 14) and VLAN tag (bytes 15 and 16) of all transmitted packets with VLAN tags. VLAN Tag Control in Transmit Packets - 2'b00: No VLAN tag deletion, insertion, or replacement - 2'b01: VLAN tag deletion The MAC removes the VLAN type (bytes 13 and 14) and VLAN tag (bytes 15 and 16) of all transmitted packets with VLAN tags. - 2'b10: VLAN tag insertion The MAC inserts VLT in bytes 15 and 16 of the packet after inserting the Type value (0x8100 or 0x88a8) in bytes 13 and 14. This operation is performed on all transmitted packets, irrespective of whether they already have a VLAN tag. - 2'b11: VLAN tag replacement

Table continues on the next page...

Table continued from the previous page...

Field	Description
	The MAC replaces VLT in bytes 15 and 16 of all VLAN-type transmitted packets (Bytes 13 and 14 are 0x8100 or 0x88a8). Note: Changes to this field take effect only on the start of a packet. If you write this register field when a packet is being transmitted, only the subsequent packet can use the updated value, that is, the current packet does not use the updated value. 00 - No VLAN tag deletion, insertion, or replacement 01 - VLAN tag deletion 10 - VLAN tag insertion 11 - VLAN tag replacement
15-0 VLT	VLAN Tag for Transmit Packets This field contains the value of the VLAN tag to be inserted or replaced. The value must only be changed when the transmit lines are inactive or during the initialization phase. Bits[15:13] are the User Priority field, Bit 12 is the CFI/DEI field, and Bits[11:0] are the VID field in the VLAN tag. The following list describes the bits of this field: - Bits[15:13]: User Priority - Bit 12: Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) - Bits[11:0]: VLAN Identifier (VID) field of VLAN tag

43.8.1.12 MAC Inner VLAN Tag Inclusion or Replacement (MAC_INNER_VLAN_INCL)

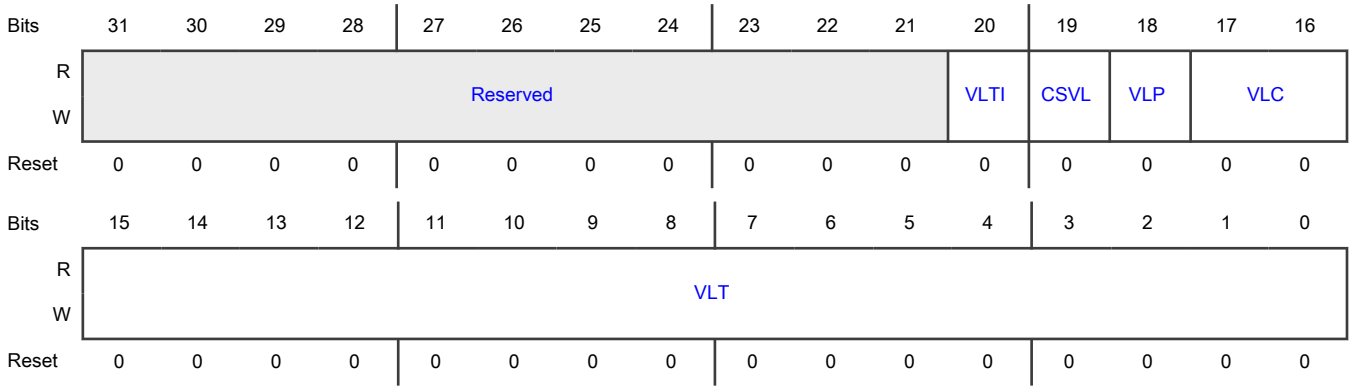
The Inner VLAN Tag Inclusion or Replacement register contains the inner VLAN tag to be inserted or replaced in the Transmit packet.

The Inner VLAN Tag Inclusion or Replacement register contains the inner VLAN tag to be inserted or replaced in the Transmit packet. It also contains the inner VLAN tag insertion controls.

Offset

Register	Offset
MAC_INNER_VLAN_INCL	64h

Diagram



Fields

Field	Description
31-21 —	Reserved
20 VLTI	VLAN Tag Input When this bit is set, it indicates that the VLAN tag to be inserted or replaced in Tx packet should be taken from: - The Tx descriptor 0 - VLAN Tag Input is disabled 1 - VLAN Tag Input is enabled
19 CSVL	C-VLAN or S-VLAN When this bit is set, S-VLAN type (0x88A8) is inserted or replaced in the 13th and 14th bytes of transmitted packets. When this bit is reset, C-VLAN type (0x8100) is inserted or replaced in the 13th and 14th bytes of transmitted packets. 0 - C-VLAN type (0x8100) is inserted 1 - S-VLAN type (0x88A8) is inserted
18 VLP	VLAN Priority Control When this bit is set, the VLC field is used for VLAN deletion, insertion, or replacement. When this bit is reset, the mti_vlan_ctrl_i control input is used and the VLC field is ignored. 0 - VLAN Priority Control is disabled 1 - VLAN Priority Control is enabled
17-16 VLC	VLAN Tag Control in Transmit Packets - 2'b00: No VLAN tag deletion, insertion, or replacement - 2'b01: VLAN tag deletion The MAC removes the VLAN type (bytes 17 and 18) and VLAN tag (bytes 19 and 20) of all transmitted packets with VLAN tags. - 2'b10: VLAN tag insertion

Table continues on the next page...

Table continued from the previous page...

Field	Description
	The MAC inserts VLT in bytes 19 and 20 of the packet after inserting the Type value (0x8100 or 0x88a8) in bytes 17 and 18. This operation is performed on all transmitted packets, irrespective of whether they already have a VLAN tag. - 2'b11: VLAN tag replacement The MAC replaces VLT in bytes 19 and 20 of all VLAN-type transmitted packets (Bytes 17 and 18 are 0x8100 or 0x88a8). Note: Changes to this field take effect only on the start of a packet. If you write this register field when a packet is being transmitted, only the subsequent packet can use the updated value, that is, the current packet does not use the updated value. 00 - No VLAN tag deletion, insertion, or replacement 01 - VLAN tag deletion 10 - VLAN tag insertion 11 - VLAN tag replacement
15-0 VLT	VLAN Tag for Transmit Packets This field contains the value of the VLAN tag to be inserted or replaced. The value must only be changed when the transmit lines are inactive or during the initialization phase. Bits[15:13] are the User Priority field, Bit 12 is the CFI/DEI field, and Bits[11:0] are the VID field in the VLAN tag. The following list describes the bits of this field: - Bits[15:13]: User Priority - Bit 12: Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) - Bits[11:0]: VLAN Identifier (VID) field of VLAN tag

43.8.1.13 MAC Q0 Tx Flow Control (MAC_Q0_TX_FLOW_CTRL)

The Flow Control register controls the generation and reception of the Control (Pause Command) packets by the Flow control module of the MAC.

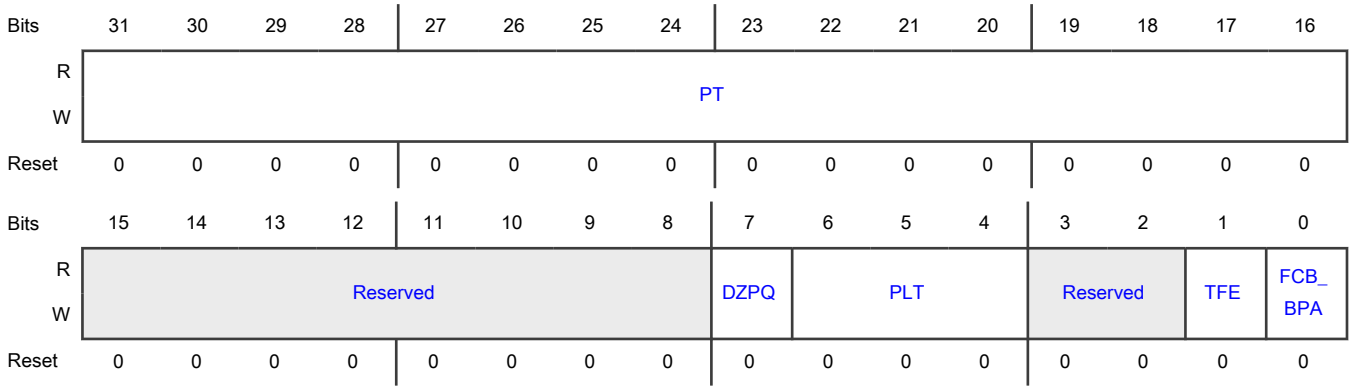
The Flow Control register controls the generation and reception of the Control (Pause Command) packets by the Flow control module of the MAC. A Write to a register with the Busy bit set to 1 triggers the Flow Control block to generate a Pause packet. The fields of the control packet are selected as specified in the 802.3x specification, and the Pause Time value from this register is used in the Pause Time field of the control packet. The Busy bit remains set until the control packet is transferred onto the cable. The application must make sure that the Busy bit is cleared before writing to the register.

When the PFCE bit in the MAC_Q0_RX_FLOW_CTRL register is enabled, this register controls the generation of Priority Flow Control (PFC) frames with priorities mapped according to PSRQ0 in the MAC_Q0_RXQ_CTRL2 register.

Offset

Register	Offset
MAC_Q0_TX_FLOW_CTRL	70h

Diagram



Fields

Field	Description
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0 - Zero-Quanta Pause packet generation is enabled 1 - Zero-Quanta Pause packet generation is disabled
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000 - Pause Time minus 4 Slot Times (PT -4 slot times) 001 - Pause Time minus 28 Slot Times (PT -28 slot times)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	010 - Pause Time minus 36 Slot Times (PT -36 slot times) 011 - Pause Time minus 144 Slot Times (PT -144 slot times) 100 - Pause Time minus 256 Slot Times (PT -256 slot times) 101 - Pause Time minus 512 Slot Times (PT -512 slot times) 110 - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable Full-Duplex Mode: In the full-duplex mode, when this bit is set, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. Half-Duplex Mode: In the half-duplex mode, when this bit is set, the MAC enables the backpressure operation. When this bit is reset, the backpressure feature is disabled. 0 - Transmit Flow Control is disabled 1 - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy or Backpressure Activate This bit initiates a Pause packet in the full-duplex mode and activates the backpressure function in the half-duplex mode if the TFE bit is set. Full-Duplex Mode: In the full-duplex mode, this bit should be read as 1'b0 before writing to this register. To initiate a Pause packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When Pause packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Half-Duplex Mode: When this bit is set (and TFE bit is set) in the half-duplex mode, the MAC asserts the backpressure. During backpressure, when the MAC receives a new packet, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically ORed with the mti_flowctrl_i input signal for the backpressure function. When the MAC is configured for the full-duplex mode, the BPA is automatically disabled. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flow Control Busy or Backpressure Activate is disabled 1 - Flow Control Busy or Backpressure Activate is enabled

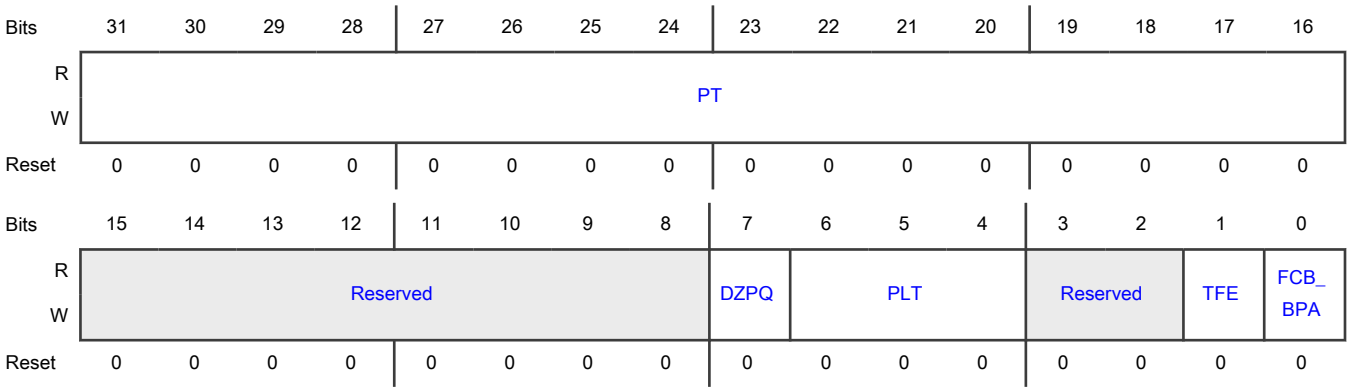
43.8.1.14 MAC Q1 Tx Flow Control (MAC_Q1_TX_FLOW_CTRL)

This register controls the generation of PFC Control packets of priorities mapped as per the PSRQi field in the MAC_Q1_RXQ_CTRL2/MAC_Q1_RXQ_CTRL3 registers.

Offset

Register	Offset
MAC_Q1_TX_FLOW_CTRL	74h

Diagram



Fields

Field	Description
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0 - Zero-Quanta Pause packet generation is enabled 1 - Zero-Quanta Pause packet generation is disabled
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000 - Pause Time minus 4 Slot Times (PT -4 slot times) 001 - Pause Time minus 28 Slot Times (PT -28 slot times) 010 - Pause Time minus 36 Slot Times (PT -36 slot times) 011 - Pause Time minus 144 Slot Times (PT -144 slot times) 100 - Pause Time minus 256 Slot Times (PT -256 slot times) 101 - Pause Time minus 512 Slot Times (PT -512 slot times) 110 - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable When this bit is set in full-duplex mode, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. 0 - Transmit Flow Control is disabled 1 - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy This bit initiates a PFC packet if the TFE bit is set. To initiate a PFC packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When PFC packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flow Control Busy or Backpressure Activate is disabled 1 - Flow Control Busy or Backpressure Activate is enabled

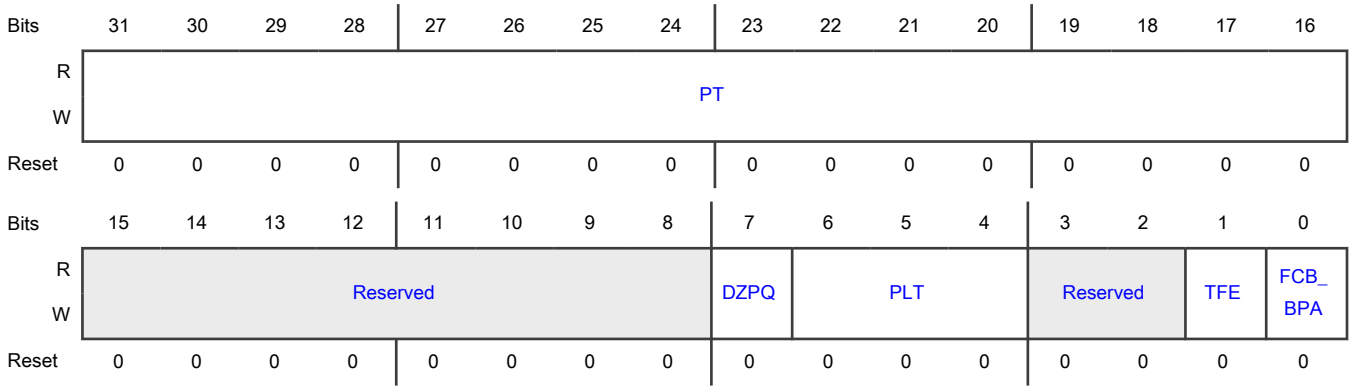
43.8.1.15 MAC Q2 Tx Flow Control (MAC_Q2_TX_FLOW_CTRL)

This register controls the generation of PFC Control packets of priorities mapped as per the PSRQi field in the MAC_Q2_RXQ_CTRL2/MAC_Q2_RXQ_CTRL3 registers.

Offset

Register	Offset
MAC_Q2_TX_FLOW_CTRL	78h

Diagram



Fields

Field	Description
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0 - Zero-Quanta Pause packet generation is enabled 1 - Zero-Quanta Pause packet generation is disabled
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000 - Pause Time minus 4 Slot Times (PT -4 slot times) 001 - Pause Time minus 28 Slot Times (PT -28 slot times)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	010 - Pause Time minus 36 Slot Times (PT -36 slot times) 011 - Pause Time minus 144 Slot Times (PT -144 slot times) 100 - Pause Time minus 256 Slot Times (PT -256 slot times) 101 - Pause Time minus 512 Slot Times (PT -512 slot times) 110 - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable When this bit is set in full-duplex mode, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. 0 - Transmit Flow Control is disabled 1 - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy This bit initiates a PFC packet if the TFE bit is set. To initiate a PFC packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When PFC packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flow Control Busy or Backpressure Activate is disabled 1 - Flow Control Busy or Backpressure Activate is enabled

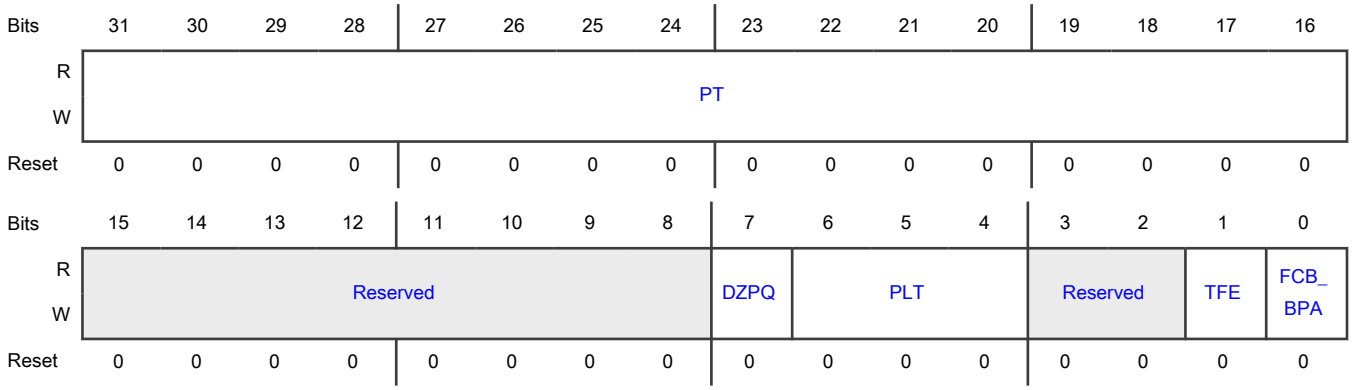
43.8.1.16 MAC Q3 Tx Flow Control (MAC_Q3_TX_FLOW_CTRL)

This register controls the generation of PFC Control packets of priorities mapped as per the PSRQi field in the MAC_Q3_RXQ_CTRL2/MAC_Q3_RXQ_CTRL3 registers.

Offset

Register	Offset
MAC_Q3_TX_FLOW_CTRL	7Ch

Diagram



Fields

Field	Description
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0 - Zero-Quanta Pause packet generation is enabled 1 - Zero-Quanta Pause packet generation is disabled
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000 - Pause Time minus 4 Slot Times (PT -4 slot times) 001 - Pause Time minus 28 Slot Times (PT -28 slot times)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	010 - Pause Time minus 36 Slot Times (PT -36 slot times) 011 - Pause Time minus 144 Slot Times (PT -144 slot times) 100 - Pause Time minus 256 Slot Times (PT -256 slot times) 101 - Pause Time minus 512 Slot Times (PT -512 slot times) 110 - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable When this bit is set in full-duplex mode, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. 0 - Transmit Flow Control is disabled 1 - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy This bit initiates a PFC packet if the TFE bit is set. To initiate a PFC packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When PFC packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flow Control Busy or Backpressure Activate is disabled 1 - Flow Control Busy or Backpressure Activate is enabled

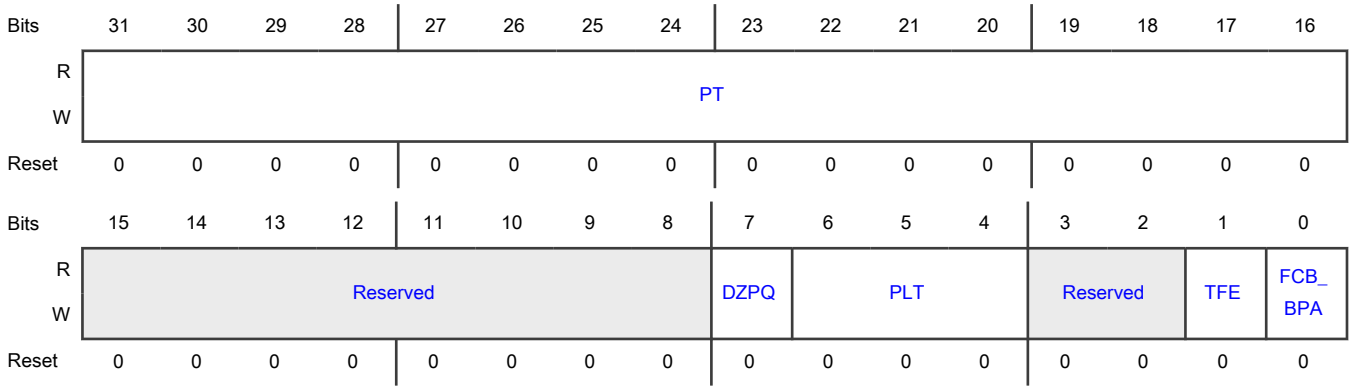
43.8.1.17 MAC Q4 Tx Flow Control (MAC_Q4_TX_FLOW_CTRL)

This register controls the generation of PFC Control packets of priorities mapped as per the PSRQi field in the MAC_Q4_RXQ_CTRL2/MAC_Q4_RXQ_CTRL3 registers.

Offset

Register	Offset
MAC_Q4_TX_FLOW_CTRL	80h

Diagram



Fields

Field	Description
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0 - Zero-Quanta Pause packet generation is enabled 1 - Zero-Quanta Pause packet generation is disabled
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000 - Pause Time minus 4 Slot Times (PT -4 slot times) 001 - Pause Time minus 28 Slot Times (PT -28 slot times)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	010 - Pause Time minus 36 Slot Times (PT -36 slot times) 011 - Pause Time minus 144 Slot Times (PT -144 slot times) 100 - Pause Time minus 256 Slot Times (PT -256 slot times) 101 - Pause Time minus 512 Slot Times (PT -512 slot times) 110 - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable When this bit is set in full-duplex mode, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. 0 - Transmit Flow Control is disabled 1 - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy This bit initiates a PFC packet if the TFE bit is set. To initiate a PFC packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When PFC packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flow Control Busy or Backpressure Activate is disabled 1 - Flow Control Busy or Backpressure Activate is enabled

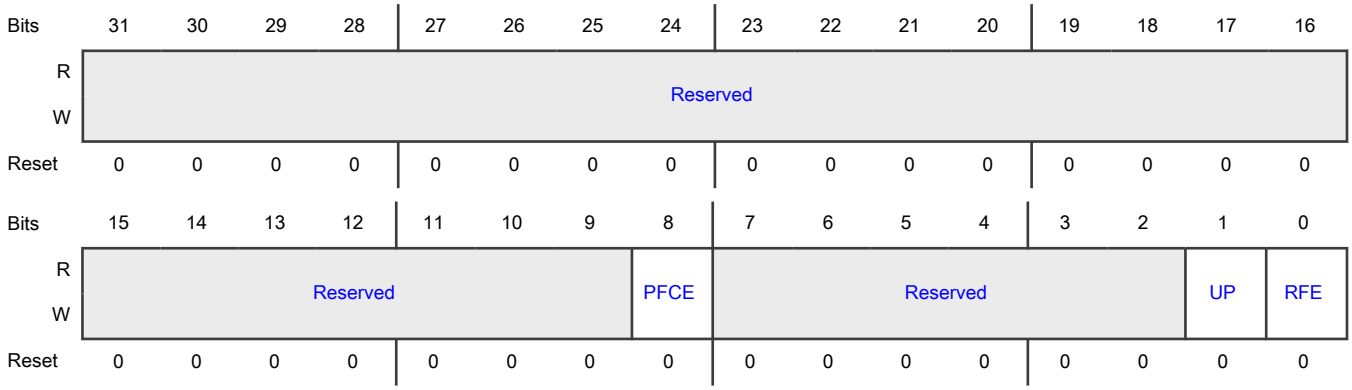
43.8.1.18 MAC Rx Flow Control (MAC_RX_FLOW_CTRL)

The Receive Flow Control register controls the pausing of MAC Transmit based on the received Pause packet.

Offset

Register	Offset
MAC_RX_FLOW_CTRL	90h

Diagram



Fields

Field	Description
31-9 —	Reserved
8 PFCE	Priority Based Flow Control Enable When this bit is set, it enables generation and reception of priority-based flow control (PFC) packets. When this bit is reset, it enables generation and reception of 802.3x Pause control packets. 0 - Priority Based Flow Control is disabled 1 - Priority Based Flow Control is enabled
7-2 —	Reserved
1 UP	Unicast Pause Packet Detect A pause packet is processed when it has the unique multicast address specified in the IEEE 802.3. When this bit is set, the MAC can also detect Pause packets with unicast address of the station. This unicast address should be as specified in MAC_ADDRESS0_HIGH and MAC_ADDRESS0_LOW. When this bit is reset, the MAC only detects Pause packets with unique multicast address. Note: The MAC does not process a Pause packet if the multicast address is different from the unique multicast address. This is also applicable to the received PFC packet when the Priority Flow Control (PFC) is enabled. The unique multicast address (0x01_80_C2_00_00_01) is as specified in IEEE 802.1 Qbb-2011. 0 - Unicast Pause Packet Detect disabled 1 - Unicast Pause Packet Detect enabled
0 RFE	Receive Flow Control Enable When this bit is set and the MAC is operating in full-duplex mode, the MAC decodes the received Pause packet and disables its transmitter for a specified (Pause) time. When this bit is reset or the MAC is operating in half-duplex mode, the decode function of the Pause packet is disabled. When PFC is enabled, flow control is enabled for PFC packets. The MAC decodes the received PFC packet and disables the Transmit queue, with matching priorities, for a duration of received Pause time.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Receive Flow Control is disabled 1 - Receive Flow Control is enabled

43.8.1.19 Receive Queue Control 4 (MAC_RXQ_CTRL4)

The Receive Queue Control 4 register controls the routing of unicast and multicast packets that fail the Destination or Source address filter to the Rx queues.

Offset

Register	Offset
MAC_RXQ_CTRL4	94h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												VFFQ			VFFQE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				MFFQ			MFFQE	Reserved				UFFQ			UFFQE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-20 —	Reserved
19-17 VFFQ	VLAN Tag Filter Fail Packets Queue This field holds the Rx queue number to which the tagged packets failing the Destination or Source Address filter (and UFFQE/MFFQE not enabled) or failing the VLAN tag filter must be routed to. This field is valid only when the VFFQE bit is set.
16 VFFQE	VLAN Tag Filter Fail Packets Queuing Enable When this bit is set, the tagged packets which fail the Destination or Source address filter or fail the VLAN tag filter, are routed to the Rx Queue Number programmed in the VFFQ. When this bit is reset, the tagged packets which fail the Destination or Source address filter or fail the VLAN tag filter are routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - VLAN tag Filter Fail Packets Queuing is disabled 1 - VLAN tag Filter Fail Packets Queuing is enabled
15-12 —	Reserved
11-9 MFFQ	Multicast Address Filter Fail Packets Queue. This field holds the Rx queue number to which the Multicast packets failing the Destination or Source Address filter are routed to. This field is valid only when the MFFQE bit is set.
8 MFFQE	Multicast Address Filter Fail Packets Queuing Enable. When this bit is set, the Multicast packets which fail the Destination or Source address filter is routed to the Rx Queue Number programmed in the MFFQ. When this bit is reset, the Multicast packets which fail the Destination or Source address filter is routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set. 0 - Multicast Address Filter Fail Packets Queuing is disabled 1 - Multicast Address Filter Fail Packets Queuing is enabled
7-4 —	Reserved
3-1 UFFQ	Unicast Address Filter Fail Packets Queue. Unicast Address Filter Fail Packets Queue. This field holds the Rx queue number to which the Unicast packets failing the Destination or Source Address filter are routed to. This field is valid only when the UFFQE bit is set.
0 UFFQE	Unicast Address Filter Fail Packets Queuing Enable. When this bit is set, the Unicast packets which fail the Destination or Source address filter is routed to the Rx Queue Number programmed in the UFFQ. When this bit is reset, the Unicast packets which fail the Destination or Source address filter is routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set. 0 - Unicast Address Filter Fail Packets Queuing is disabled 1 - Unicast Address Filter Fail Packets Queuing is enabled

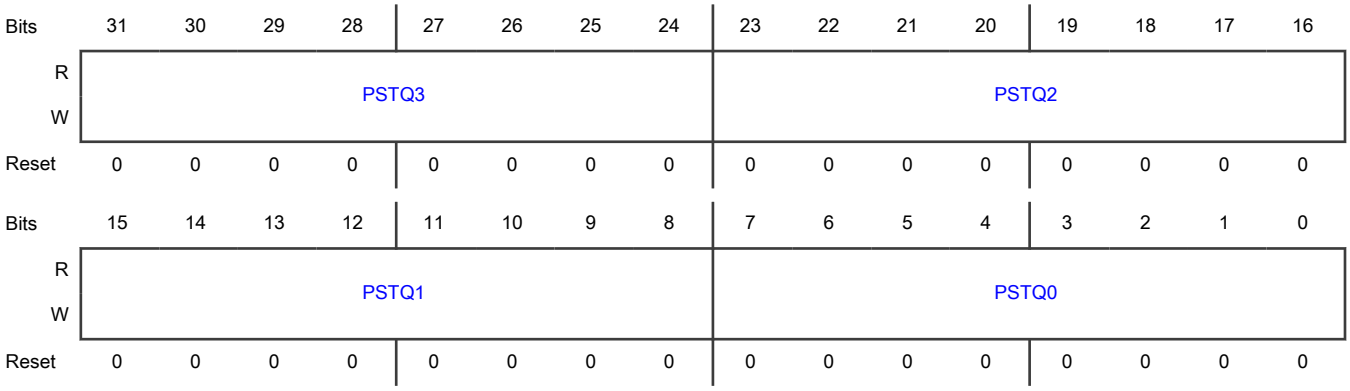
43.8.1.20 Transmit Queue Priority Mapping 0 (MAC_TXQ_PRTY_MAP0)

The Transmit Queue Priority Mapping 0 register contains the priority values assigned to Tx Queue 0 through Tx Queue 3.

Offset

Register	Offset
MAC_TXQ_PRTY_MAP0	98h

Diagram



Fields

Field	Description
31-24 PSTQ3	Priorities Selected in Transmit Queue 3 This bit is similar to the PSTQ0 bit.
23-16 PSTQ2	Priorities Selected in Transmit Queue 2 This bit is similar to the PSTQ0 bit.
15-8 PSTQ1	Priorities Selected in Transmit Queue 1 This bit is similar to the PSTQ0 bit.
7-0 PSTQ0	Priorities Selected in Transmit Queue 0 This field holds the priorities assigned to Tx Queue 0 by the software. This field determines if Tx Queue 0 should be blocked from transmitting specified pause time when a PFC packet is received with priorities matching the priorities programmed in this field. If the content of this field is not mutually exclusive to corresponding fields of other Transmit queues, that is, same priority is mapped to multiple Tx queues, the MAC blocks all queues with matching priority for specified time.

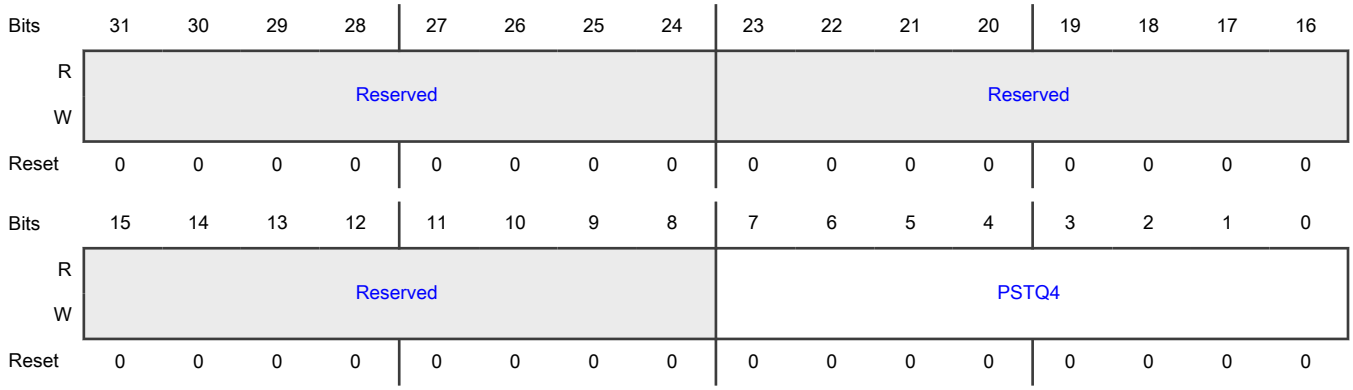
43.8.1.21 Transmit Queue Priority Mapping 1 (MAC_TXQ_PRTY_MAP1)

The Transmit Queue Priority Mapping 1 register contains the priority values assigned to Tx Queue 4.

Offset

Register	Offset
MAC_TXQ_PRTY_MAP1	9Ch

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 —	Reserved
15-8 —	Reserved
7-0 PSTQ4	Priorities Selected in Transmit Queue 4 This field holds the priorities assigned to Tx Queue 4 by the software. This field determines if Tx Queue 4 should be blocked from transmitting specified pause time when a PFC packet is received with priorities matching the priorities programmed in this field. If the content of this field is not mutually exclusive to corresponding fields of other Transmit queues, that is, same priority is mapped to multiple Tx queues, the MAC blocks all queues with matching priority for specified time.

43.8.1.22 Receive Queue Control 0 (MAC_RXQ_CTRL0)

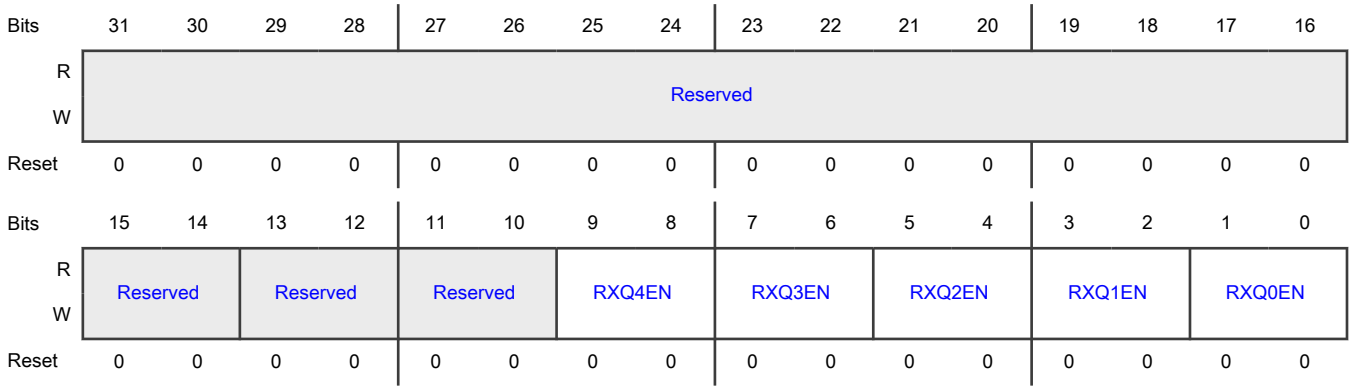
The Receive Queue Control 0 register controls the queue management in the MAC Receiver.

Note: In multiple Rx queues configuration, all the queues are disabled by default. Enable the Rx queue by programming the corresponding field in this register.

Offset

Register	Offset
MAC_RXQ_CTRL0	A0h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-14 —	Reserved
13-12 —	Reserved
11-10 —	Reserved
9-8 RXQ4EN	Receive Queue 4 Enable This field is similar to the RXQ0EN field. 00 - Queue not enabled 01 - Queue enabled for AV 10 - Queue enabled for DCB/Generic 11 - Reserved
7-6 RXQ3EN	Receive Queue 3 Enable This field is similar to the RXQ0EN field. 00 - Queue not enabled 01 - Queue enabled for AV 10 - Queue enabled for DCB/Generic 11 - Reserved
5-4 RXQ2EN	Receive Queue 2 Enable This field is similar to the RXQ0EN field. 00 - Queue not enabled 01 - Queue enabled for AV 10 - Queue enabled for DCB/Generic

Table continues on the next page...

Table continued from the previous page...

Field	Description
	11 - Reserved
3-2 RXQ1EN	Receive Queue 1 Enable This field is similar to the RXQ0EN field. 00 - Queue not enabled 01 - Queue enabled for AV 10 - Queue enabled for DCB/Generic 11 - Reserved
1-0 RXQ0EN	Receive Queue 0 Enable This field indicates whether Rx Queue 0 is enabled for AV or DCB. 00 - Queue not enabled 01 - Queue enabled for AV 10 - Queue enabled for DCB/Generic 11 - Reserved

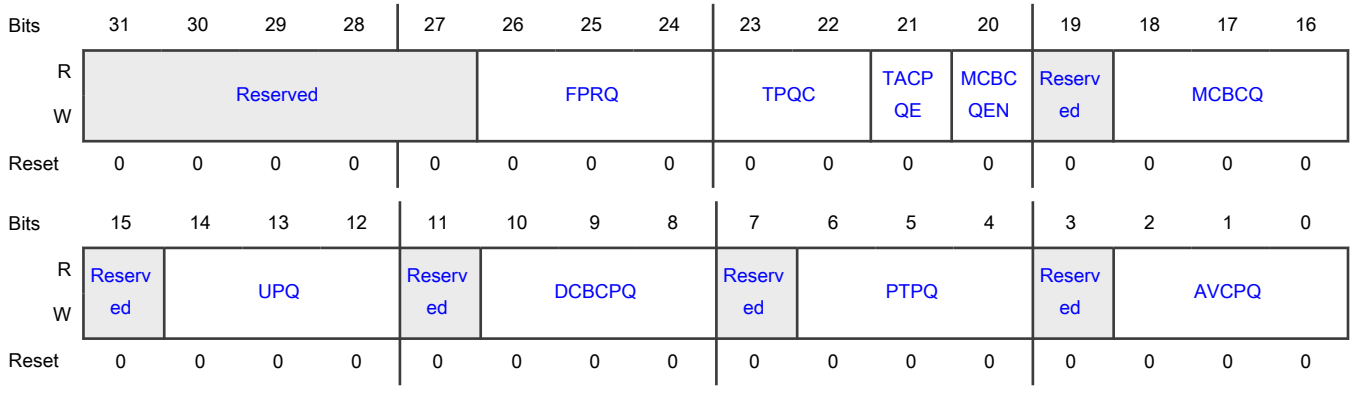
43.8.1.23 Receive Queue Control 1 (MAC_RXQ_CTRL1)

The Receive Queue Control 1 register controls the routing of multicast, broadcast, AV, DCB, and untagged packets to the Rx queues.

Offset

Register	Offset
MAC_RXQ_CTRL1	A4h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-24 FPRQ	Frame Preemption Residue Queue This field holds the Rx queue number to which the residual preemption frames must be forwarded. Preemption frames that are tagged and pass the SA/DA/VLAN filtering are routed based on PSRQ and all other frames are treated as residual frames and is routed to the queue number mentioned in this field. The Queue-0 is used as a default queue for express frames, so this field cannot be programmed to a value 0.
23-22 TPQC	Tagged PTP over Ethernet Packets Queuing Control. This field controls the routing of the VLAN Tagged PTPoE packets. If DWC_EQOS_AV_ENABLE is selected in the configuration, the following programmable options are allowed. - 2'b00: VLAN Tagged PTPoE packets are routed as generic VLAN Tagged packet (based on PSRQ for only non-AV enabled Rx Queues). - 2'b01: VLAN Tagged PTPoE packets are routed to Rx Queue specified by PTPQ field (That Rx Queue can be enabled for AV or non-AV traffic). - 2'b10: VLAN Tagged PTPoE packets are routed to only AV enabled Rx Queues based on PSRQ. - 2'b11: Reserved If DWC_EQOS_AV_ENABLE is not selected in the configuration, the following programmable options are allowed. - 1'b0: VLAN Tagged PTPoE packets are routed as generic VLAN Tagged packet (based on PSRQ for DCB/Generic enabled Rx Queues). - 1'b1: VLAN Tagged PTPoE packets are routed to Rx Queues specified by PTPQ field.
21 TACPQE	Tagged AV Control Packets Queuing Enable. When set, the MAC routes the received Tagged AV Control packets to the Rx queue specified by AVCPQ field. When reset, the MAC routes the received Tagged AV Control packets based on the tag priority matching the PSRQ fields in MAC_RXQ_CTRL2 and MAC_RXQ_CTRL3 registers. 0 - Tagged AV Control Packets Queuing is disabled 1 - Tagged AV Control Packets Queuing is enabled
20 MCBCQEN	Multicast and Broadcast Queue Enable This bit specifies that Multicast or Broadcast packets routing to the Rx Queue is enabled and the Multicast or Broadcast packets must be routed to Rx Queue specified in MCBCQ field. 0 - Multicast and Broadcast Queue is disabled 1 - Multicast and Broadcast Queue is enabled
19 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
18-16 MCBCQ	Multicast and Broadcast Queue This field specifies the Rx Queue onto which Multicast or Broadcast Packets are routed. Any Rx Queue enabled for Generic/DCB/AV traffic can be used to route the Multicast or Broadcast Packets. 000 - Receive Queue 0 001 - Receive Queue 1 010 - Receive Queue 2 011 - Receive Queue 3 100 - Receive Queue 4 101 - Reserved 110 - Reserved 111 - Reserved
15 —	Reserved
14-12 UPQ	Untagged Packet Queue This field indicates the Rx Queue to which Untagged Packets are to be routed. Any Rx Queue enabled for Generic/DCB/AV traffic can be used to route the Untagged Packets. 000 - Receive Queue 0 001 - Receive Queue 1 010 - Receive Queue 2 011 - Receive Queue 3 100 - Receive Queue 4 101 - Reserved 110 - Reserved 111 - Reserved
11 —	Reserved
10-8 DCBCPQ	DCB Control Packets Queue This field specifies the Rx queue on which the received DCB control packets are routed. The DCB data packets are routed based on the PSRQ field of the Transmit Flow Control Register of corresponding queue. 000 - Receive Queue 0 001 - Receive Queue 1 010 - Receive Queue 2

Table continues on the next page...

Table continued from the previous page...

Field	Description
	011 - Receive Queue 3 100 - Receive Queue 4 101 - Reserved 110 - Reserved 111 - Reserved
7 —	Reserved
6-4 PTPQ	PTP Packets Queue This field specifies the Rx queue on which the PTP packets sent over the Ethernet payload (not over IPv4 or IPv6) are routed. When the AV8021ASMEN bit of MAC_TIMESTAMP_CONTROL register is set, only untagged PTP over Ethernet packets are routed on an Rx Queue. If the bit is not set, then based on programming of TPQC field, both tagged and untagged PTPoE packets can be routed to this Rx Queue. 000 - Receive Queue 0 001 - Receive Queue 1 010 - Receive Queue 2 011 - Receive Queue 3 100 - Receive Queue 4 101 - Reserved 110 - Reserved 111 - Reserved
3 —	Reserved
2-0 AVCPQ	AV Untagged Control Packets Queue This field specifies the Receive queue on which the received AV tagged and untagged control packets are routed. The AV tagged (when TACPQE bit is set) and untagged control packets are routed to Receive queue specified by this field. 000 - Receive Queue 0 001 - Receive Queue 1 010 - Receive Queue 2 011 - Receive Queue 3 100 - Receive Queue 4 101 - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
	110 - Reserved 111 - Reserved

43.8.1.24 Receive Queue Control 2 (MAC_RXQ_CTRL2)

This register controls the routing of tagged packets based on the USP (user Priority) field of the received packets to the RxQueues 0 to 3.

Offset

Register	Offset
MAC_RXQ_CTRL2	A8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PSRQ3								PSRQ2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PSRQ1								PSRQ0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 PSRQ3	Priorities Selected in the Receive Queue 3 This field decides the priorities assigned to Rx Queue 3. All packets with priorities that match the values set in this field are routed to Rx Queue 3. For example, if PSRQ3[6, 3] are set, packets with USP field equal to 3 or 6 are routed to Rx Queue 3. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. This field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 3 crosses the flow control threshold settings.
23-16 PSRQ2	Priorities Selected in the Receive Queue 2 This field decides the priorities assigned to Rx Queue 2. All packets with priorities that match the values set in this field are routed to Rx Queue 2.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	For example, if PSRQ2[1, 0] are set, packets with USP field equal to 1 or 0 are routed to Rx Queue 2. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. this field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 2 crosses the flow control threshold settings.
15-8 PSRQ1	Priorities Selected in the Receive Queue 1 This field decides the priorities assigned to Rx Queue 1. All packets with priorities that match the values set in this field are routed to Rx Queue 1. For example, if PSRQ1[4] is set, packets with USP field equal to 4 are routed to Rx Queue 1. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. this field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 1 crosses the flow control threshold settings.
7-0 PSRQ0	Priorities Selected in the Receive Queue 0 This field decides the priorities assigned to Rx Queue 0. All packets with priorities that match the values set in this field are routed to Rx Queue 0. For example, if PSRQ0[5] is set, packets with USP field equal to 5 are routed to Rx Queue 0. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. this field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 0 crosses the flow control threshold settings.

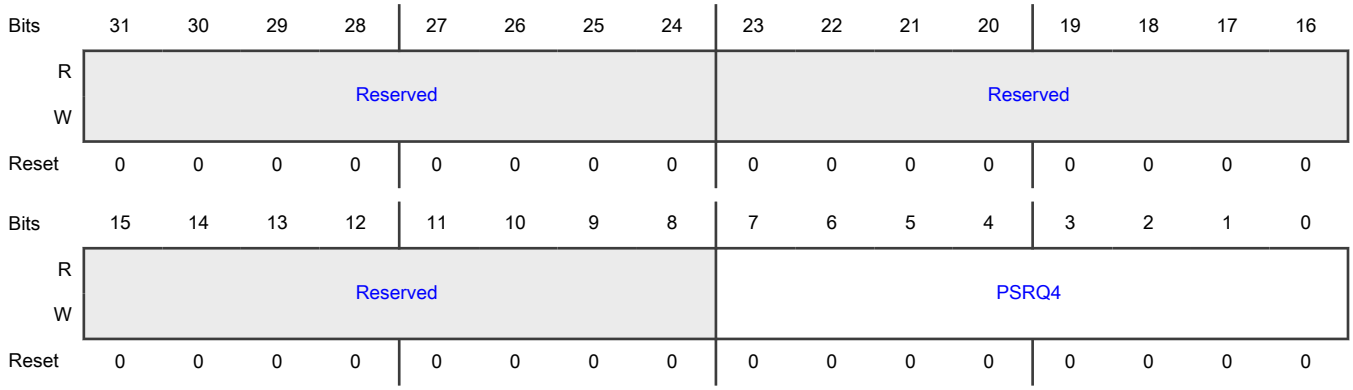
43.8.1.25 Receive Queue Control 3 (MAC_RXQ_CTRL3)

This register controls the routing of tagged packets based on the USP (user Priority) field of the received packets to Rx Queue 4.

Offset

Register	Offset
MAC_RXQ_CTRL3	ACh

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 —	Reserved
15-8 —	Reserved
7-0 PSRQ4	Priorities Selected in the Receive Queue 4 This field decides the priorities assigned to Rx Queue 4. All packets with priorities that match the values set in this field are routed to Rx Queue 4. For example, if PSRQ4[7:4] is set, packets with USP field equal to 7, 6, 5, or 4 are routed to Rx Queue 4. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. This field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 4 crosses the flow control threshold settings.

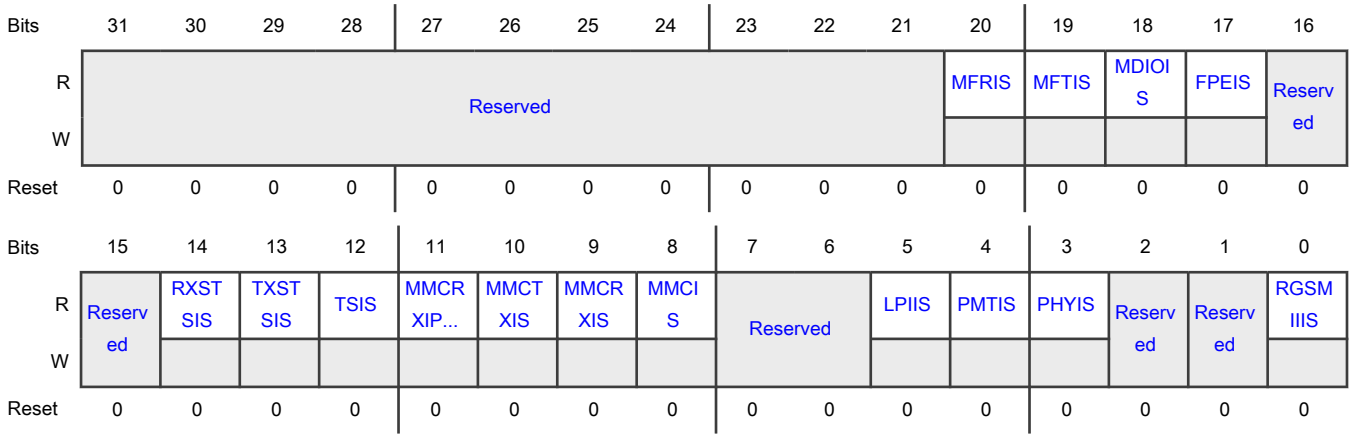
43.8.1.26 Interrupt Status (MAC_INTERRUPT_STATUS)

The Interrupt Status register contains the status of interrupts.

Offset

Register	Offset
MAC_INTERRUPT_STATUS	B0h

Diagram



Fields

Field	Description
31-21 —	Reserved
20 MFRIS	MMC FPE Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC FPE Receive Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option along with FPE support. 0 - MMC FPE Receive Interrupt status not active 1 - MMC FPE Receive Interrupt status active
19 MFTIS	MMC FPE Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC FPE Transmit Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option along with FPE support. 0 - MMC FPE Transmit Interrupt status not active 1 - MMC FPE Transmit Interrupt status active
18 MDIOIS	MDIO Interrupt Status This bit indicates an interrupt event after the completion of MDIO operation. To reset this bit, the application has to read this bit/Write 1 to this bit when RCWE bit of CSR_SW_CTRL register is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - MDIO Interrupt status not active 1 - MDIO Interrupt status active
17 FPEIS	Frame Preemption Interrupt Status

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This bit indicates an interrupt event during the operation of Frame Preemption (Bits[19:16] of MAC_FPE_CTRL_STS register is set). To reset this bit, the application must clear the event in MAC_FPE_CTRL_STS that has caused the Interrupt. 0 - Frame Preemption Interrupt status not active 1 - Frame Preemption Interrupt status active
16 —	Reserved
15 —	Reserved
14 RXSTSIS	Receive Status Interrupt This bit indicates the status of received packets. This bit is set when the RWT bit is set in the MAC_RX_TX_STATUS register. This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of CSR_SW_CTRL register is set) in the MAC_RX_TX_STATUS register. 0 - Receive Interrupt status not active 1 - Receive Interrupt status active
13 TXSTSIS	Transmit Status Interrupt This bit indicates the status of transmitted packets. This bit is set when any of the following bits is set in the MAC_RX_TX_STATUS register: - Excessive Collision (EXCOL) - Late Collision (LCOL) - Excessive Deferral (EXDEF) - Loss of Carrier (LCARR) - No Carrier (NCARR) - Jabber Timeout (TJT) This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of CSR_SW_CTRL register is set) in the MAC_RX_TX_STATUS register. 0 - Transmit Interrupt status not active 1 - Transmit Interrupt status active
12 TSIS	Timestamp Interrupt Status If the Timestamp feature is enabled, this bit is set when any of the following conditions is true: - The system time value is equal to or exceeds the value specified in the Target Time High and Low registers. - There is an overflow in the Seconds register. - The Target Time Error occurred, that is, programmed target time already elapsed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If the Auxiliary Snapshot feature is enabled, this bit is set when the auxiliary snapshot trigger is asserted. In configurations other than EQOS_CORE, when drop transmit status is enabled in MTL, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and MAC_TXTIMESTAMP_STATUS_SECONDS registers. When PTP offload feature is enabled, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and MAC_TX_TIMESTAMP_STATUS_SECONDS registers, for PTO generated Delay Request and Pdelay request packets. This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of CSR_SW_CTRL register is set) in the MAC_TIMESTAMP_STATUS register. 0 - Timestamp Interrupt status not active 1 - Timestamp Interrupt status active
11 MMCRXIPIS	MMC Receive Checksum Offload Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) and Enable Receive TCP/IP Checksum Check options. 0 - MMC Receive Checksum Offload Interrupt status not active 1 - MMC Receive Checksum Offload Interrupt status active
10 MMCTXIS	MMC Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. 0 - MMC Transmit Interrupt status not active 1 - MMC Transmit Interrupt status active
9 MMCRXIS	MMC Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. 0 - MMC Receive Interrupt status not active 1 - MMC Receive Interrupt status active
8 MMCIS	MMC Interrupt Status This bit is set high when Bit 11, Bit 10, or Bit 9 is set high. This bit is cleared only when all these bits are low. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. 0 - MMC Interrupt status not active 1 - MMC Interrupt status active

Table continues on the next page...

Table continued from the previous page...

Field	Description
7-6 —	Reserved
5 LPIIS	LPI Interrupt Status When the Energy Efficient Ethernet feature is enabled, this bit is set for any LPI state entry or exit in the MAC Transmitter or Receiver. This bit is cleared when the corresponding interrupt source bit of MAC_LPI_CONTROL_STATUS register is read (or corresponding interrupt source bit of MAC_LPI_CONTROL_STATUS register is written to 1 when RCWE bit of MAC_CSR_SW_CTRL register is set). 0 - LPI Interrupt status not active 1 - LPI Interrupt status active
4 PMTIS	PMT Interrupt Status This bit is set when a Magic packet or Wake-on-LAN packet is received in the power-down mode (RWKPRCVD and MGKPRCVD bits in MAC_PMT_CONTROL_STATUS register). This bit is cleared when corresponding interrupt source bit are cleared because of a Read operation to the MAC_PMT_CONTROL_STATUS register (or corresponding interrupt source bit of MAC_PMT_CONTROL_STATUS register is written to 1 when RCWE bit of CSR_SW_CTRL register is set). This bit is valid only when you select the Enable Power Management option. 0 - PMT Interrupt status not active 1 - PMT Interrupt status active
3 PHYIS	PHY Interrupt This bit is set when rising edge is detected on the phy_intr_i input. This bit is cleared when this register is read (or this bit is written to 1 when RCWE bit of CSR_SW_CTRL register is set). 0 - PHY Interrupt not detected 1 - PHY Interrupt detected
2 —	Reserved
1 —	Reserved
0 RGSMIIIS	RGMII or SMII Interrupt Status This bit is set because of any change in value of the Link Status of RGMII or SMII interface (LNKSTS bit in MAC_PHYIF_CONTROL_STATUS register). This bit is cleared when the PHYIF_CONTROL_STATUS register is read (or LNKSTS bit of MAC_PHYIF_CONTROL_STATUS register is written to 1 when RCWE bit of MAC_CSR_SW_CTRL register is set). This bit is valid only when you select the optional RGMII or SMII PHY interface. 0 - RGMII or SMII Interrupt Status is not active

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - RGMII or SMII Interrupt Status is active

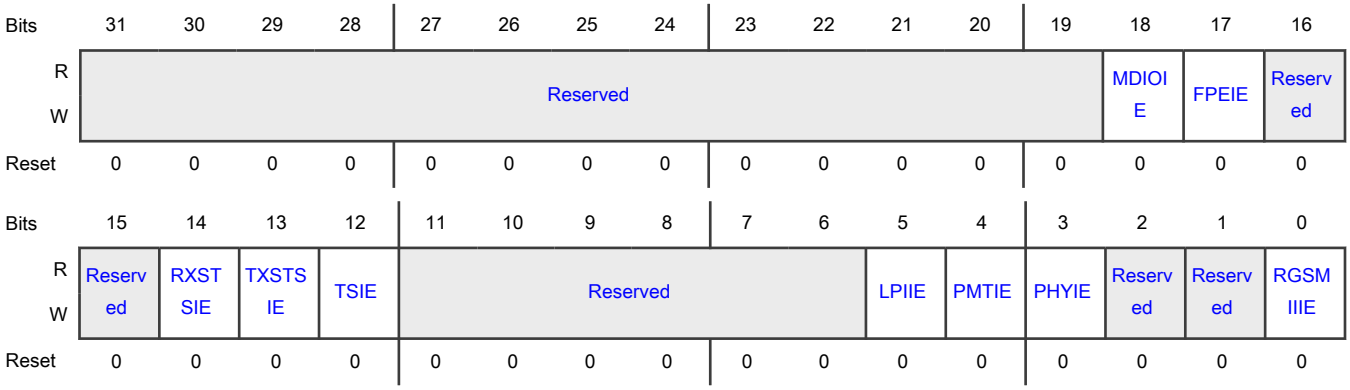
43.8.1.27 Interrupt Enable (MAC_INTERRUPT_ENABLE)

The Interrupt Enable register contains the masks for generating the interrupts.

Offset

Register	Offset
MAC_INTERRUPT_ENABLE	B4h

Diagram



Fields

Field	Description
31-19 —	Reserved
18 MDIOIE	MDIO Interrupt Enable When this bit is set, it enables the assertion of the interrupt when MDIOIS field is set in the MAC_INTERRUPT_STATUS register. 0 - MDIO Interrupt is disabled 1 - MDIO Interrupt is enabled
17 FPEIE	Frame Preemption Interrupt Enable When this bit is set, it enables the assertion of the interrupt when FPEIS field is set in the MAC_INTERRUPT_STATUS. 0 - Frame Preemption Interrupt is disabled 1 - Frame Preemption Interrupt is enabled
16	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
15 —	Reserved
14 RXSTSIE	Receive Status Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[RXSTSIS]. 0 - Receive Status Interrupt is disabled 1 - Receive Status Interrupt is enabled
13 TXSTSIE	Transmit Status Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[TXSTSIS]. 0 - Timestamp Status Interrupt is disabled 1 - Timestamp Status Interrupt is enabled
12 TSIE	Timestamp Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[TSIS]. 0 - Timestamp Interrupt is disabled 1 - Timestamp Interrupt is enabled
11-6 —	Reserved
5 LPIIE	LPI Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[LPIIS]. 0 - LPI Interrupt is disabled 1 - LPI Interrupt is enabled
4 PMTIE	PMT Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[PMTIS]. 0 - PMT Interrupt is disabled 1 - PMT Interrupt is enabled
3 PHYIE	PHY Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[PHYIS]. 0 - PHY Interrupt is disabled 1 - PHY Interrupt is enabled
2 —	Reserved
1 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 RGSMIIIE	RGMII or SMII Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of RGSMIIIS bit in MAC_INTERRUPT_STATUS register. 0 - RGMII or SMII Interrupt is disabled 1 - RGMII or SMII Interrupt is enabled

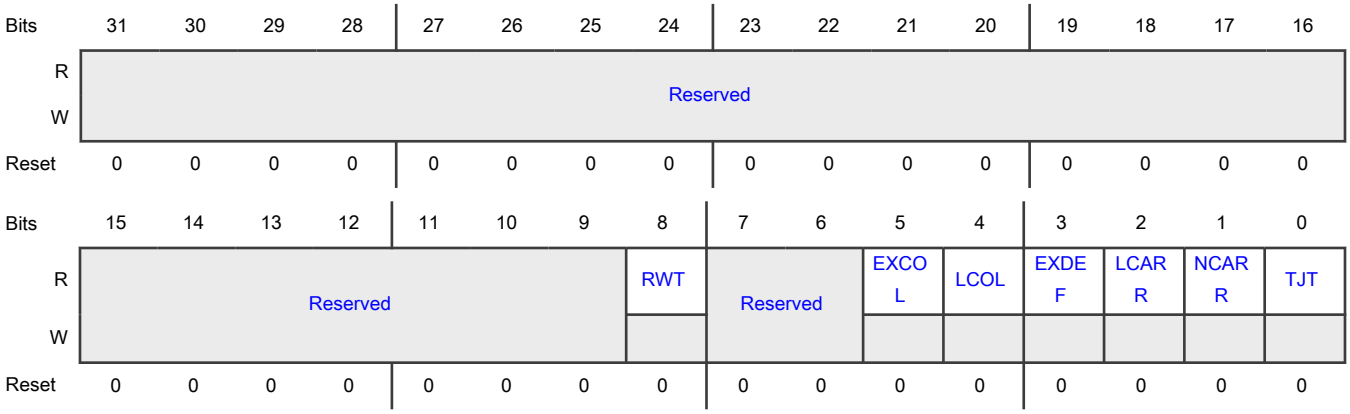
43.8.1.28 Receive Transmit Status (MAC_RX_TX_STATUS)

The Receive Transmit Status register contains the Receive and Transmit Error status.

Offset

Register	Offset
MAC_RX_TX_STATUS	B8h

Diagram



Fields

Field	Description
31-9 —	Reserved
8 RWT	Receive Watchdog Timeout This bit is set when a packet with length greater than 2,048 bytes is received (10, 240 bytes when Jumbo Packet mode is enabled) and the WD bit is reset in the MAC_CONFIGURATION register. Receive Watchdog Timeout This bit is set when a packet with length greater than 2,048 bytes is received (10, 240 bytes when Jumbo Packet mode is enabled) and the WD bit is reset in the MAC_CONFIGURATION register. This

Table continues on the next page...

Table continued from the previous page...

Field	Description
	bit is set when a packet with length greater than 16,383 bytes is received and the WD bit is set in the MAC_CONFIGURATION register. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - No receive watchdog timeout 1 - Receive watchdog timed out
7-6 —	Reserved
5 EXCOL	Excessive Collisions When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the transmission aborted after 16 successive collisions while attempting to transmit the current packet. Excessive Collisions When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the transmission aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the CONFIGURATION register, this bit is set after the first collision and the packet transmission is aborted. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - No collision 1 - Excessive collision is sensed
4 LCOL	Late Collision When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the packet transmission aborted because a collision occurred after the collision window (512 bytes including Preamble and Carrier Extension in GMII mode). Late Collision When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the packet transmission aborted because a collision occurred after the collision window (512 bytes including Preamble and Carrier Extension in GMII mode). This bit is not valid if the Underflow error occurs. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - No collision 1 - Late collision is sensed
3 EXDEF	Excessive Deferral When the DTXSTS bit is set in the MAC_OPERATION_MODE register and the DC bit is set in the MAC_CONFIGURATION register, this bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 in 1000/2500 Mbps mode or when Jumbo packet is enabled). Excessive Deferral

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When the DTXSTS bit is set in the MAC_OPERATION_MODE register and the DC bit is set in the MAC_CONFIGURATION register, this bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 in 1000/2500 Mbps mode or when Jumbo packet is enabled). Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - No Excessive deferral 1 - Excessive deferral
2 LCARR	Loss of Carrier When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the loss of carrier occurred during packet transmission, that is, the phy_crs_i signal was inactive for one or more transmission clock periods during packet transmission. Loss of Carrier When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the loss of carrier occurred during packet transmission, that is, the phy_crs_i signal was inactive for one or more transmission clock periods during packet transmission. This bit is valid only for packets transmitted without collision. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Carrier is present 1 - Loss of carrier
1 NCARR	No Carrier When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the carrier signal from the PHY is not present at the end of preamble transmission. No Carrier When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the carrier signal from the PHY is not present at the end of preamble transmission. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Carrier is present 1 - No carrier
0 TJT	Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired which happens when the packet size exceeds 2,048 bytes (10,240 bytes when the Jumbo packet is enabled) and JD bit is reset in the MAC_CONFIGURATION register. Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired which happens when the packet size exceeds 2,048 bytes (10,240 bytes when the Jumbo packet is enabled) and JD bit is reset in the MAC_CONFIGURATION register. This bit is set when the packet size exceeds 16,383 bytes and the JD bit is set in the MAC_CONFIGURATION register. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - No Transmit Jabber Timeout 1 - Transmit Jabber Timeout occurred

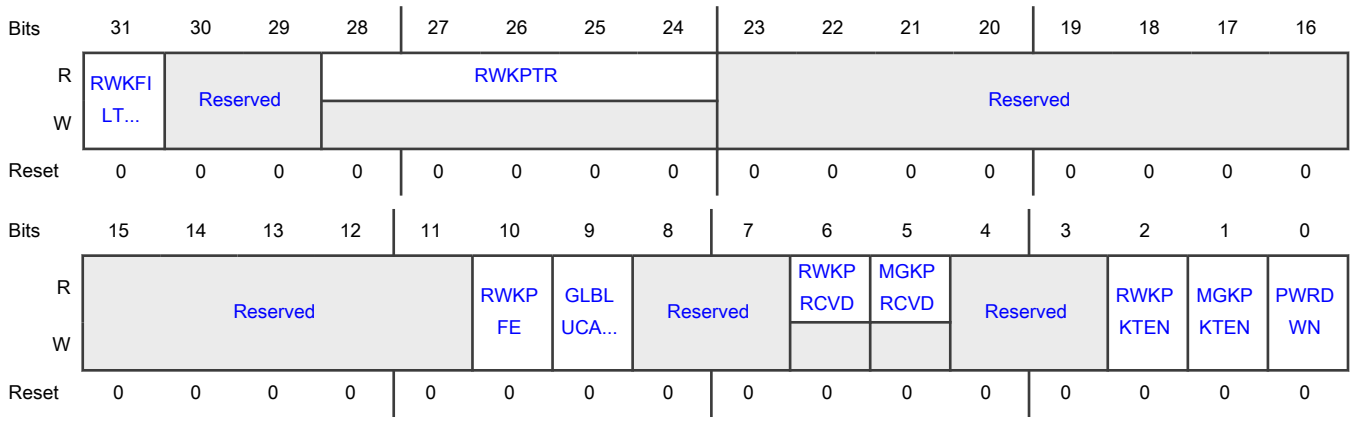
43.8.1.29 PMT Control and Status (MAC_PMT_CONTROL_STATUS)

The PMT Control and Status Register.

Offset

Register	Offset
MAC_PMT_CONTROL_STATUS	C0h

Diagram



Fields

Field	Description
31 RWKFILTRST	Remote Wake-Up Packet Filter Register Pointer Reset When this bit is set, the remote wake-up packet filter register pointer is reset to 3'b000. Remote Wake-Up Packet Filter Register Pointer Reset When this bit is set, the remote wake-up packet filter register pointer is reset to 3'b000. It is automatically cleared after 1 clock cycle. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Remote Wake-Up Packet Filter Register Pointer is not Reset 1 - Remote Wake-Up Packet Filter Register Pointer is Reset
30-29	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
28-24 RWKPTR	Remote Wake-up FIFO Pointer This field gives the current value (0 to 7, 15, or 31 when 4, 8, or 16 Remote Wake-up Packet Filters are selected) of the Remote Wake-up Packet Filter register pointer. Remote Wake-up FIFO Pointer This field gives the current value (0 to 7, 15, or 31 when 4, 8, or 16 Remote Wake-up Packet Filters are selected) of the Remote Wake-up Packet Filter register pointer. When the value of this pointer is equal to maximum for the selected number of Remote Wake-up Packet Filters, the contents of the Remote Wake-up Packet Filter Register are transferred to the clk_rx_i domain when a Write occurs to that register.
23-11 —	Reserved
10 RWKPFE	Remote Wake-up Packet Forwarding Enable When this bit is set along with RWKPKTEN, the MAC receiver drops all received frames until it receives the expected Wake-up frame. Remote Wake-up Packet Forwarding Enable When this bit is set along with RWKPKTEN, the MAC receiver drops all received frames until it receives the expected Wake-up frame. All frames after that event including the received wake-up frame are forwarded to application. This bit is then self-cleared on receiving the wake-up packet. The application can also clear this bit before the expected wake-up frame is received. In such cases, the MAC reverts to the default behavior where packets received are forwarded to the application. This bit must only be set when RWKPKTEN is set high and PWRDWN is set low. The setting of this bit has no effect when PWRDWN is set high. Note: If Magic Packet Enable and Wake-Up Frame Enable are both set along with setting of this bit and Magic Packet is received prior to wake-up frame, this bit is self-cleared on receiving Magic Packet, the received Magic packet is dropped, and all frames after received Magic Packet are forwarded to application. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Remote Wake-up Packet Forwarding is disabled 1 - Remote Wake-up Packet Forwarding is enabled
9 GLBLUCAST	Global Unicast When this bit set, any unicast packet filtered by the MAC (DAF) address recognition is detected as a remote wake-up packet. 0 - Global unicast is disabled 1 - Global unicast is enabled
8-7 —	Reserved
6 RWKPRCVD	Remote Wake-Up Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a remote wake-up packet. Remote Wake-Up Packet Received

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, it indicates that the power management event is generated because of the reception of a remote wake-up packet. This bit is cleared when this register is read. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Remote wake-up packet is received 1 - Remote wake-up packet is received
5 MGKPRCVD	Magic Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a magic packet. Magic Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a magic packet. This bit is cleared when this register is read. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - No Magic packet is received 1 - Magic packet is received
4-3 —	Reserved
2 RWKPKTEN	Remote Wake-Up Packet Enable When this bit is set, a power management event is generated when the MAC receives a remote wake-up packet. 0 - Remote wake-up packet is disabled 1 - Remote wake-up packet is enabled
1 MGKPKTEN	Magic Packet Enable When this bit is set, a power management event is generated when the MAC receives a magic packet. 0 - Magic Packet is disabled 1 - Magic Packet is enabled
0 PWRDWN	Power Down When this bit is set, the MAC receiver drops all received packets until it receives the expected magic packet or remote wake-up packet. Power Down When this bit is set, the MAC receiver drops all received packets until it receives the expected magic packet or remote wake-up packet. This bit is then self-cleared and the power-down mode is disabled. The software can clear this bit before the expected magic packet or remote wake-up packet is received. The packets received by the MAC after this bit is cleared are forwarded to the application. This bit must only be set when the Magic Packet Enable, Global Unicast, or Remote Wake-Up Packet Enable bit is set high. Note: You can gate-off the CSR clock during the power-down mode. However, when the CSR clock is gated-off, you cannot perform any read or write operations on this register. Therefore, the Software cannot clear this bit. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Power down is disabled 1 - Power down is enabled

43.8.1.30 Remote Wakeup Filter (MAC_RWK_PACKET_FILTER)

The Remote Wakeup Filter registers are implemented as 8, 16, or 32 indirect access registers (wkuppktfilter_reg#i) based on whether 4, 8, or 16 Remote Wakeup Filters are selected in the configuration and accessed by application through MAC_RWK_PACKET_FILTER register.

The Remote Wakeup Filter registers are implemented as 8, 16, or 32 indirect access registers (wkuppktfilter_reg#i) based on whether 4, 8, or 16 Remote Wakeup Filters are selected in the configuration and accessed by application through MAC_RWK_PACKET_FILTER register. When the Remote Wakeup Filters are to be programmed, the entire set of wkuppktfilter_reg registers must be written. The wkuppktfilter_reg register is programmed by sequentially writing the eight, sixteen or thirty-two register values in MAC_RWK_PACKET_FILTER register for wkuppktfilter_REG0, wkuppktfilter_REG1, ..., wkuppktfilter_REG31 respectively. The wkuppktfilter_reg register is read in a similar way. The MAC updates the wkuppktfilter_reg register current pointer value in RWKPTR field of MAC_PMT_CONTROL_STATUS register.

The Remote Wakeup Filters are arranged in blocks of 4 filters each and each such block have eight 32-bit wide registers, viz. wkuppktfilter_REG0-7, wkuppktfilter_reg8-15, wkuppktfilter_REG16-23 and wkuppktfilter_REG24-31. The fields of Remote Wakeup Filter are described as follows:

Filter i Byte Mask: The filter i byte mask register defines the bytes of the packet that are examined by filter i (0, 1, 2, 3, .., 15) to determine whether or not a packet is a wake-up packet.

- The MSB (31st bit) must be zero.

- Bit j[30:0] is the byte mask.

- If Bit j (byte number) of the byte mask is set, the CRC block processes the Filter i Offset + j of the incoming packet; otherwise Filter i Offset + j is ignored.

Filter i Command: The 4-bit filter i command controls the filter i operation.

- Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet.

- Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC-16 value. Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2".

- Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. The details are provided below:

-- The And_Previous bit setting is applicable within a set of 4 filters.

-- Setting of And_Previous bit of filter that is not enabled has no effect, that is setting And_Previous bit of lowest number filter in the set of 4 filters has no effect. For example, setting of And_Previous bit of Filter 0 has no effect.

-- If And_Previous bit is set for filter to form AND chained filter, the AND chain breaks at the point any filter is not enabled. For example: If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set) but Filter 1 is not enabled (bit 0 in Filter 1 command is reset), then only Filter 2 result is considered. If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 3 And_Previous bit is set (bit 1 in Filter 3 command is set), but Filter 1 is not enabled (bit 0 in Filter 1 command is reset), then only Filter 2 result ANDed with Filter 3 result is considered. If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 3 And_Previous bit is set (bit 1 in Filter 3 command is set), but Filter 2 is not enabled (bit 0 in Filter 2 command is reset), then since setting of Filter 2 And_Previous bit has no effect only Filter 1 result ORed with Filter 3 result is considered.

-- If filters chained by And_Previous bit setting have complementary programming, then a frame may never pass the AND chained filter. For example, if Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 1 Address_Type bit is set (bit 3 in Filter 1 command is set) indicating multicast detection and Filter 2 Address_Type bit is reset (bit 3 in Filter 2 command is reset) indicating unicast detection or vice versa, a remote wakeup frame does not pass the AND chained filter as a remote wakeup frame cannot be of both unicast and multicast address type.

- Bit 0 is the enable for filter i. If Bit 0 is not set, filter i is disabled.

Filter i Offset: The filter i offset register defines the offset (within the packet) from which the filter i examines the packets.

- This 8-bit pattern-offset is the offset for the filter i first byte to be examined.

- The minimum allowed offset is 12, which refers to the 13th byte of the packet.

- The offset value 0 refers to the first byte of the packet.

Filter i CRC-16: The filter i CRC-16 register contains the CRC-16 value calculated from the pattern and the byte mask programmed in the Remote Wakeup filter register.

- The 16-bit CRC calculation uses the following polynomial: G(x) = x^16 + x^15 + x^2 + 1

- Each mask, used in the hash function calculation, is compared with a 16-bit value associated with that mask. Each filter has the following:

-- 32-bit Mask: Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1, the corresponding byte is taken into the CRC-16 calculation.

-- 8-bit Offset Pointer: Specifies the byte to start the CRC-16 computation. The pointer and the mask are used together to locate the bytes to be used in the CRC-16 calculations.

The structure of the RWK Filter registers is shown in "Figure 10-11 Remote Wake-Up Packet Filter Register".

Offset

Register	Offset
MAC_RWK_PACKET_FILTER	C4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WKUPFRMFTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WKUPFRMFTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 WKUPFRMFTR	RWK Packet Filter This field contains the various controls of RWK Packet filter.

43.8.1.31 LPI Control and Status (MAC_LPI_CONTROL_STATUS)

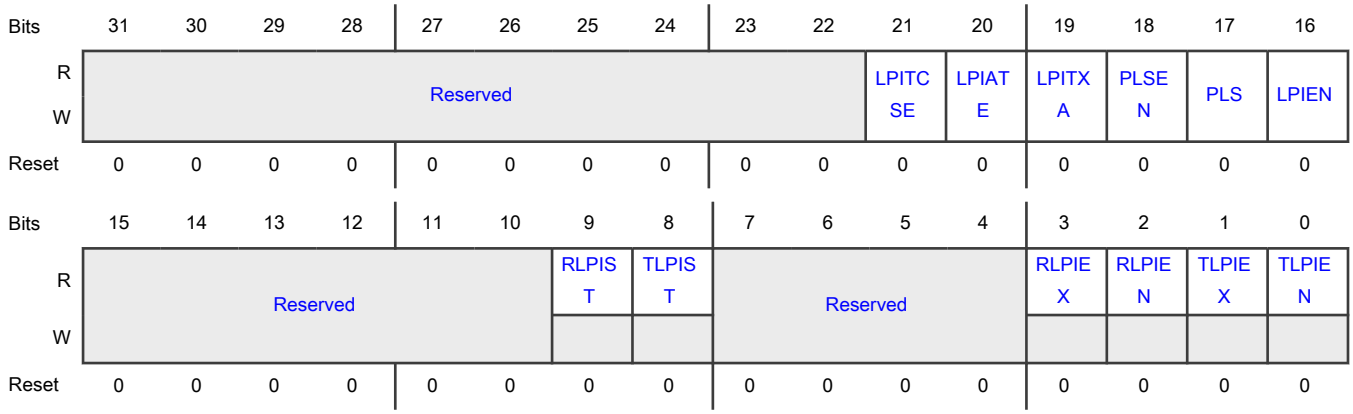
The LPI Control and Status Register controls the LPI functions and provides the LPI interrupt status.

The LPI Control and Status Register controls the LPI functions and provides the LPI interrupt status. The status bits are cleared when this register is read.

Offset

Register	Offset
MAC_LPI_CONTROL_STATUS	D0h

Diagram



Fields

Field	Description
31-22 —	Reserved
21 LPITCSE	LPI Tx Clock Stop Enable When this bit is set, the MAC asserts sbd_tx_clk_gating_ctrl_o signal high after it enters Tx LPI mode to indicate that the Tx clock to MAC can be stopped. LPI Tx Clock Stop Enable When this bit is set, the MAC asserts sbd_tx_clk_gating_ctrl_o signal high after it enters Tx LPI mode to indicate that the Tx clock to MAC can be stopped.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is reset, the MAC does not assert sbd_tx_clk_gating_ctrl_o signal high after it enters Tx LPI mode. If RGMII Interface is selected, the Tx clock is required for transmitting the LPI pattern. The Tx Clock cannot be gated and so the LPITCSE bit cannot be programmed. 0 - LPI Tx Clock Stop is disabled 1 - LPI Tx Clock Stop is enabled
20 LPIATE	LPI Timer Enable This bit controls the automatic entry of the MAC Transmitter into and exit out of the LPI state. LPI Timer Enable This bit controls the automatic entry of the MAC Transmitter into and exit out of the LPI state. When LPIATE, LPITXA and LPIEN bits are set, the MAC Transmitter enters LPI state only when the complete MAC TX data path is IDLE for a period indicated by the MAC_LPI_ENTRY_TIMER register. After entering LPI state, if the data path becomes non-IDLE (due to a new packet being accepted for transmission), the Transmitter exits LPI state but does not clear LPIEN bit. This enables the re-entry into LPI state when it is IDLE again. When LPIATE is 0, the LPI Auto timer is disabled and MAC Transmitter enters LPI state based on the settings of LPITXA and LPIEN bit descriptions. 0 - LPI Timer is disabled 1 - LPI Timer is enabled
19 LPITXA	LPI Tx Automate This bit controls the behavior of the MAC when it is entering or coming out of the LPI mode on the Transmit side. LPI Tx Automate This bit controls the behavior of the MAC when it is entering or coming out of the LPI mode on the Transmit side. This bit is not functional in the EQOS-CORE configurations in which the Tx clock gating is done during the LPI mode. If the LPITXA and LPIEN bits are set to 1, the MAC enters the LPI mode only after all outstanding packets (in the core) and pending packets (in the application interface) have been transmitted. The MAC comes out of the LPI mode when the application sends any packet for transmission or the application issues a Tx FIFO Flush command. In addition, the MAC automatically clears the LPIEN bit when it exits the LPI state. If Tx FIFO Flush is set in the FTQ bit of MAC_TXQ0_OPERATION_MODE register, when the MAC is in the LPI mode, it exits the LPI mode. When this bit is 0, the LPIEN bit directly controls behavior of the MAC when it is entering or coming out of the LPI mode. 0 - LPI Tx Automate is disabled 1 - LPI Tx Automate is enabled
18	PHY Link Status Enable This bit enables the link status received on the RGMII, SGMII, or SMII Receive paths to be used for activating the LPI LS TIMER.

Table continues on the next page...

Table continued from the previous page...

Field	Description
PLSEN	PHY Link Status Enable This bit enables the link status received on the RGMII, SGMII, or SMII Receive paths to be used for activating the LPI LS TIMER. When this bit is set, the MAC uses the link-status bits of the MAC_PHYIF_CONTROL_STATUS register and the PLS bit for the LPI LS Timer trigger. When this bit is reset, the MAC ignores the link-status bits of the MAC_PHYIF_CONTROL_STATUS register and takes only the PLS bit. 0 - PHY Link Status is disabled 1 - PHY Link Status is enabled
17 PLS	PHY Link Status This bit indicates the link status of the PHY. PHY Link Status This bit indicates the link status of the PHY. The MAC Transmitter asserts the LPI pattern only when the link status is up (OKAY) at least for the time indicated by the LPI LS TIMER. When this bit is set, the link is considered to be okay (UP) and when this bit is reset, the link is considered to be down. 0 - link is down 1 - link is okay (UP)
16 LPIEN	LPI Enable When this bit is set, it instructs the MAC Transmitter to enter the LPI state. LPI Enable When this bit is set, it instructs the MAC Transmitter to enter the LPI state. When this bit is reset, it instructs the MAC to exit the LPI state and resume normal transmission. This bit is cleared when the LPITXA bit is set and the MAC exits the LPI state because of the arrival of a new packet for transmission. 0 - LPI state is disabled 1 - LPI state is enabled
15-10 —	Reserved
9 RLPIST	Receive LPI State When this bit is set, it indicates that the MAC is receiving the LPI pattern on the GMII interface. 0 - Receive LPI state not detected 1 - Receive LPI state detected
8 TLPIST	Transmit LPI State When this bit is set, it indicates that the MAC is transmitting the LPI pattern on the GMII interface. 0 - Transmit LPI state not detected 1 - Transmit LPI state detected
7-4	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
3 RLPIEX	Receive LPI Exit When this bit is set, it indicates that the MAC Receiver has stopped receiving the LPI pattern on the GMII interface, exited the LPI state, and resumed the normal reception. Receive LPI Exit When this bit is set, it indicates that the MAC Receiver has stopped receiving the LPI pattern on the GMII interface, exited the LPI state, and resumed the normal reception. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Note: This bit may not be set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than three clock cycles of CSR clock. 0 - Receive LPI exit not detected 1 - Receive LPI exit detected
2 RLPIEN	Receive LPI Entry When this bit is set, it indicates that the MAC Receiver has received an LPI pattern and entered the LPI state. Receive LPI Entry When this bit is set, it indicates that the MAC Receiver has received an LPI pattern and entered the LPI state. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Note: This bit may not be set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than three clock cycles of CSR clock. 0 - Receive LPI entry not detected 1 - Receive LPI entry detected
1 TLPIEX	Transmit LPI Exit When this bit is set, it indicates that the MAC transmitter exited the LPI state after the application cleared the LPIEN bit and the LPI TW Timer has expired. Transmit LPI Exit When this bit is set, it indicates that the MAC transmitter exited the LPI state after the application cleared the LPIEN bit and the LPI TW Timer has expired. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). 0 - Transmit LPI exit not detected 1 - Transmit LPI exit detected
0 TLPIEN	Transmit LPI Entry When this bit is set, it indicates that the MAC Transmitter has entered the LPI state because of the setting of the LPIEN bit. Transmit LPI Entry When this bit is set, it indicates that the MAC Transmitter has entered the LPI state because of the setting of the LPIEN bit. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). 0 - Transmit LPI entry not detected 1 - Transmit LPI entry detected

Table continues on the next page...

Field	Description

43.8.1.32 LPI Timers Control (MAC_LPI_TIMERS_CONTROL)

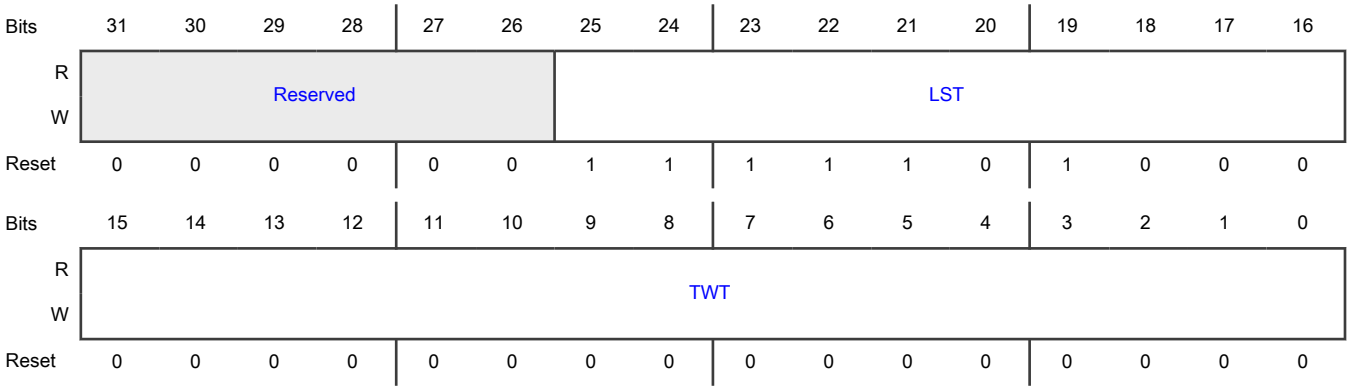
The LPI Timers Control register controls the timeout values in the LPI states.

The LPI Timers Control register controls the timeout values in the LPI states. It specifies the time for which the MAC transmits the LPI pattern and also the time for which the MAC waits before resuming the normal transmission.

Offset

Register	Offset
MAC_LPI_TIMERS_CONTROL	D4h

Diagram



Fields

Field	Description
31-26 —	Reserved
25-16 LST	LPI LS Timer This field specifies the minimum time (in milliseconds) for which the link status from the PHY should be up (OKAY) before the LPI pattern can be transmitted to the PHY. LPI LS Timer This field specifies the minimum time (in milliseconds) for which the link status from the PHY should be up (OKAY) before the LPI pattern can be transmitted to the PHY. The MAC does not transmit the LPI pattern even when the LPIEN bit is set unless the LPI LS Timer reaches the programmed terminal count. The default value of the LPI LS Timer is 1000 (1 sec) as defined in the IEEE standard.
15-0 TWT	LPI TW Timer This field specifies the minimum time (in microseconds) for which the MAC waits after it stops transmitting the LPI pattern to the PHY and before it resumes the normal transmission. LPI TW Timer

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field specifies the minimum time (in microseconds) for which the MAC waits after it stops transmitting the LPI pattern to the PHY and before it resumes the normal transmission. The TLPIEX status bit is set after the expiry of this timer.

43.8.1.33 Tx LPI Entry Timer Control (MAC_LPI_ENTRY_TIMER)

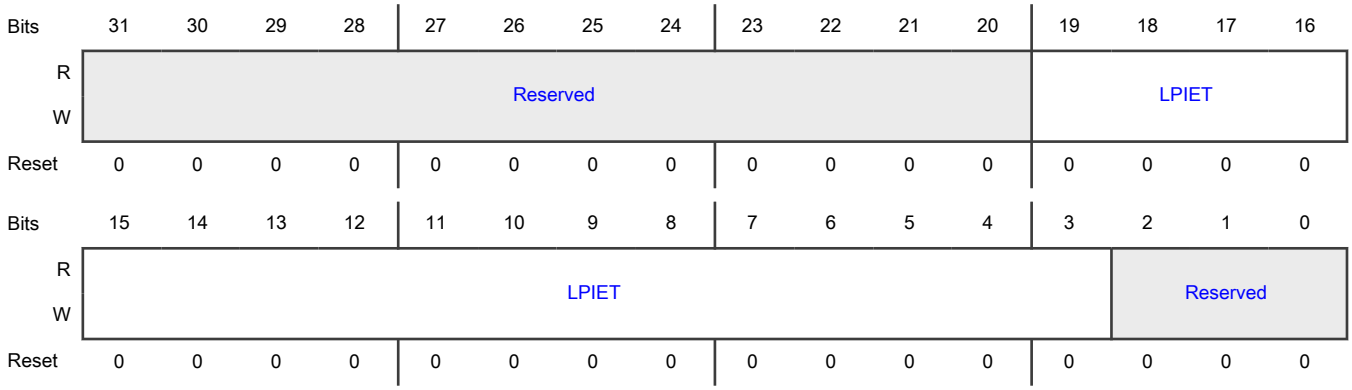
This register controls the Tx LPI entry timer.

This register controls the Tx LPI entry timer. This counter is enabled only when bit[20](LPITE) bit of MAC_LPI_CONTROL_STATUS is set to 1.

Offset

Register	Offset
MAC_LPI_ENTRY_TIMER	D8h

Diagram



Fields

Field	Description
31-20 —	Reserved
19-3 LPIET	LPI Entry Timer This field specifies the time in microseconds the MAC waits to enter LPI mode, after it has transmitted all the frames. LPI Entry Timer This field specifies the time in microseconds the MAC waits to enter LPI mode, after it has transmitted all the frames. This field is valid and used only when LPITE and LPITXA are set to 1. Bits [2:0] are read-only so that the granularity of this timer is in steps of 8 micro-seconds.

Table continues on the next page...

Table continued from the previous page...

Field	Description
2-0 —	Reserved

43.8.1.34 One-microsecond Reference Timer (MAC_ONEUS_TIC_COUNTER)

This register controls the generation of the Reference time (1 microsecond tic) for all the LPI timers. This timer has to be programmed by the software initially.

Offset

Register	Offset
MAC_ONEUS_TIC_COUNTER	DCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				TIC_1US_CNTR											
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1

Fields

Field	Description
31-12 —	Reserved
11-0 TIC_1US_CNTR	1US TIC Counter The application must program this counter so that the number of clock cycles of CSR clock is 1us. 1US TIC Counter The application must program this counter so that the number of clock cycles of CSR clock is 1us. (Subtract 1 from the value before programming). For example if the CSR clock is 100MHz then this field needs to be programmed to value 100 - 1 = 99 (which is 0x63). This is required to generate the 1US events that are used to update some of the EEE related counters.

43.8.1.35 PHY Interface Control and Status (MAC_PHYIF_CONTROL_STATUS)

The PHY Interface Control and Status register indicates the status signals received by the SGMII, RGMII, or SMII interface (selected at reset) from the PHY.

Offset

Register	Offset
MAC_PHYIF_CONTROL_STATUS	F8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										Reserved	Reserved	LNKSTS	LNKSPEED		LNKMOD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											Reserved	Reserved	Reserved	LUD	TC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-22 —	Reserved
21 —	Reserved
20 —	Reserved
19 LNKSTS	Link Status This bit indicates whether the link is up (1'b1) or down (1'b0). 0 - Link down 1 - Link up
18-17 LNKSPEED	Link Speed This bit indicates the current speed of the link. 00 - 2.5 MHz 01 - 25 MHz 10 - 125 MHz

Table continues on the next page...

Table continued from the previous page...

Field	Description
	11 - Reserved
16 LNKMOD	Link Mode This bit indicates the current mode of operation of the link. 0 - Half-duplex mode 1 - Full-duplex mode
15-5 —	Reserved
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 LUD	Link Up or Down This bit indicates whether the link is up or down during transmission of configuration in the RGMII, SGMII, or SMII interface. 0 - Link down 1 - Link up
0 TC	Transmit Configuration in RGMII, SGMII, or SMII When set, this bit enables the transmission of duplex mode, link speed, and link up or down information to the PHY in the RGMII, SMII, or SGMII port. Transmit Configuration in RGMII, SGMII, or SMII When set, this bit enables the transmission of duplex mode, link speed, and link up or down information to the PHY in the RGMII, SMII, or SGMII port. When this bit is reset, no such information is driven to the PHY. The details of this feature are provided in the following sections: - "Reduced Gigabit Media Independent Interface" - "Serial Media Independent Interface" - "Serial Gigabit Media Independent Interface" 0 - Disable Transmit Configuration in RGMII, SGMII, or SMII 1 - Enable Transmit Configuration in RGMII, SGMII, or SMII

43.8.1.36 MAC Version (MAC_VERSION)

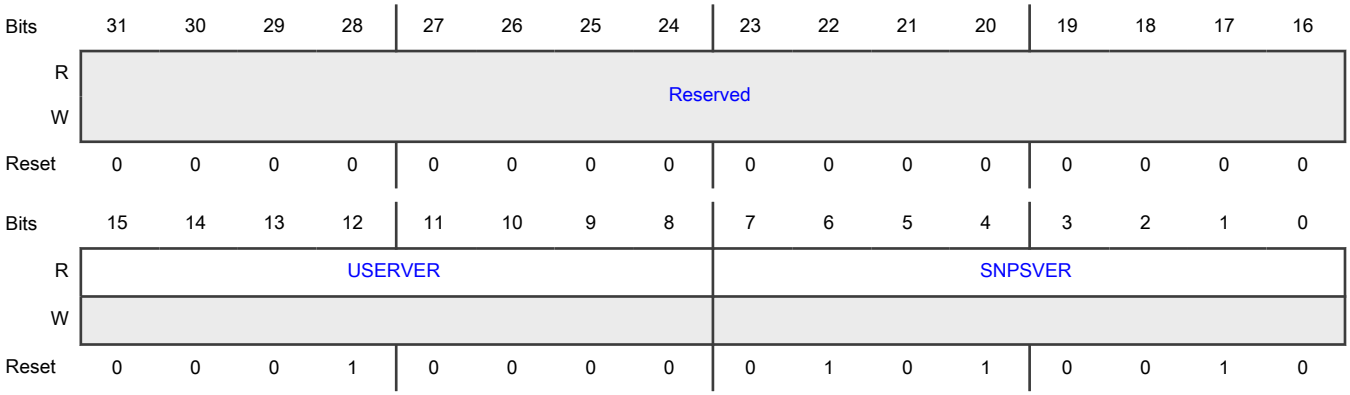
The version register identifies the version of the DWC_ether_qos.

The version register identifies the version of the DWC_ether_qos. This register contains two bytes: one that we uses to identify the core release number, and the other that you set while configuring the core.

Offset

Register	Offset
MAC_VERSION	110h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 USERVER	User-defined Version (8'h10)
7-0 SNPSVER	Synopsys-defined Version

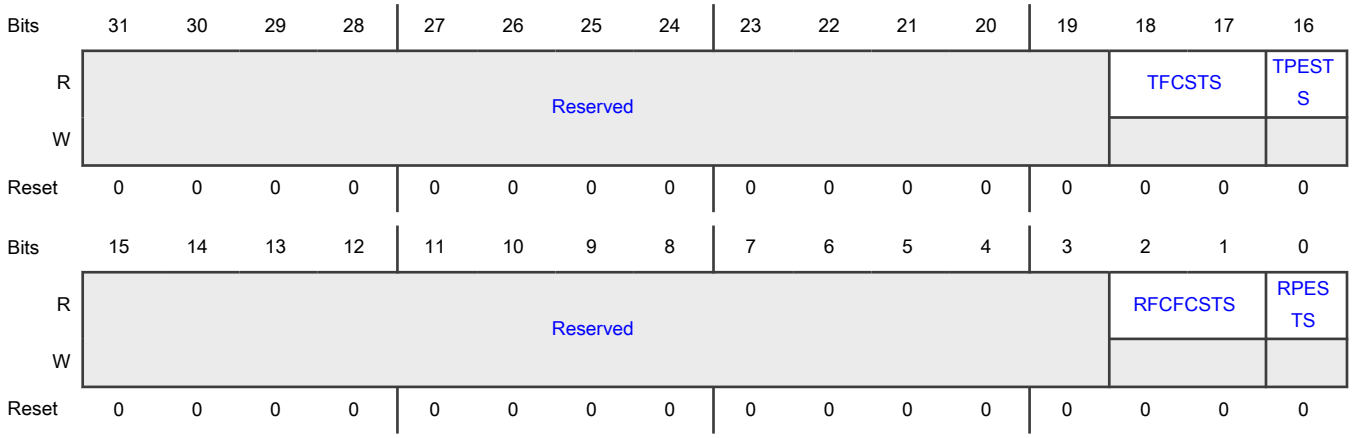
43.8.1.37 MAC Debug (MAC_DEBUG)

The Debug register provides the debug status of various MAC blocks.

Offset

Register	Offset
MAC_DEBUG	114h

Diagram



Fields

Field	Description
31-19 —	Reserved
18-17 TFCSTS	MAC Transmit Packet Controller Status This field indicates the state of the MAC Transmit Packet Controller module. 00 - Idle state 01 - Waiting for one of the following: Status of the previous packet OR IPG or back off period to be over 10 - Generating and transmitting a Pause control packet (in full-duplex mode) 11 - Transferring input packet for transmission
16 TPESTS	MAC GMII or MII Transmit Protocol Engine Status When this bit is set, it indicates that the MAC GMII or MII transmit protocol engine is actively transmitting data, and it is not in the Idle state. 0 - MAC GMII Transmit Protocol Engine Status not detected 1 - MAC GMII Transmit Protocol Engine Status detected
15-3 —	Reserved
2-1 RFCFCSTS	MAC Receive Packet Controller FIFO Status When this bit is set, this field indicates the active state of the small FIFO Read and Write controllers of the MAC Receive Packet Controller module.
0 RPESTS	MAC GMII Receive Protocol Engine Status When this bit is set, it indicates that the MAC GMII receive protocol engine is actively receiving data, and it is not in the Idle state. 0 - MAC GMII Receive Protocol Engine Status not detected 1 - MAC GMII Receive Protocol Engine Status detected

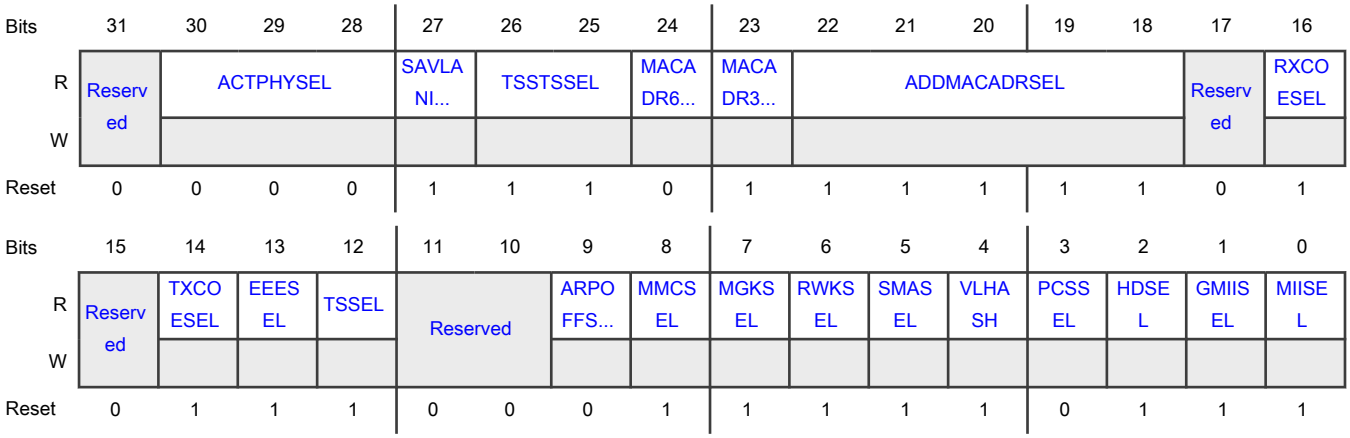
43.8.1.38 Optional Features or Functions 0 (MAC_HW_FEATURE0)

This register indicates the presence of first set of the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

Offset

Register	Offset
MAC_HW_FEATURE0	11Ch

Diagram



Fields

Field	Description
31 —	Reserved
30-28 ACTPHYSEL	Active PHY Selected When you have multiple PHY interfaces in your configuration, this field indicates the sampled value of phy_intf_sel_i during reset de-assertion. 000 - GMII 001 - RGMII 010 - SGMII 011 - TBI 100 - RMII 101 - RTBI 110 - SMII 111 - RevMII
27 SAVLANINS	Source Address or VLAN Insertion Enable This bit is set to 1 when the Enable SA and VLAN Insertion on Tx option is selected 0 - Source Address or VLAN Insertion Enable option is not selected 1 - Source Address or VLAN Insertion Enable option is selected
26-25 TSSTSSEL	Timestamp System Time Source This bit indicates the source of the Timestamp system time: This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	00 - Reserved 01 - Internal 10 - External 11 - Both
24 MACADR64SEL	MAC Addresses 64-127 Selected This bit is set to 1 when the Enable Additional 64 MAC Address Registers (64-127) option is selected 0 - MAC Addresses 64-127 Select option is not selected 1 - MAC Addresses 64-127 Select option is selected
23 MACADR32SEL	MAC Addresses 32-63 Selected This bit is set to 1 when the Enable Additional 32 MAC Address Registers (32-63) option is selected 0 - MAC Addresses 32-63 Select option is not selected 1 - MAC Addresses 32-63 Select option is selected
22-18 ADDMACADRSEL	MAC Addresses 1-31 Selected This bit is set to 1 when the non-zero value is selected for Enable Additional 1-31 MAC Address Registers option
17 —	Reserved
16 RXCOESEL	Receive Checksum Offload Enabled This bit is set to 1 when the Enable Receive TCP/IP Checksum Check option is selected 0 - Receive Checksum Offload Enable option is not selected 1 - Receive Checksum Offload Enable option is selected
15 —	Reserved
14 TXCOESEL	Transmit Checksum Offload Enabled This bit is set to 1 when the Enable Transmit TCP/IP Checksum Insertion option is selected 0 - Transmit Checksum Offload Enable option is not selected 1 - Transmit Checksum Offload Enable option is selected
13 EEESEL	Energy Efficient Ethernet Enabled This bit is set to 1 when the Enable Energy Efficient Ethernet (EEE) option is selected 0 - Energy Efficient Ethernet Enable option is not selected 1 - Energy Efficient Ethernet Enable option is selected
12 TSSEL	IEEE 1588-2008 Timestamp Enabled This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected 0 - IEEE 1588-2008 Timestamp Enable option is not selected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - IEEE 1588-2008 Timestamp Enable option is selected
11-10 —	Reserved
9 ARPOFFSEL	ARP Offload Enabled This bit is set to 1 when the Enable IPv4 ARP Offload option is selected 0 - ARP Offload Enable option is not selected 1 - ARP Offload Enable option is selected
8 MMCSEL	RMON Module Enable This bit is set to 1 when the Enable MAC Management Counters (MMC) option is selected 0 - RMON Module Enable option is not selected 1 - RMON Module Enable option is selected
7 MGKSEL	PMT Magic Packet Enable This bit is set to 1 when the Enable Magic Packet Detection option is selected 0 - PMT Magic Packet Enable option is not selected 1 - PMT Magic Packet Enable option is selected
6 RWKSEL	PMT Remote Wake-up Packet Enable This bit is set to 1 when the Enable Remote Wake-Up Packet Detection option is selected 0 - PMT Remote Wake-up Packet Enable option is not selected 1 - PMT Remote Wake-up Packet Enable option is selected
5 SMASEL	SMA (MDIO) Interface This bit is set to 1 when the Enable Station Management (MDIO Interface) option is selected 0 - SMA (MDIO) Interface not selected 1 - SMA (MDIO) Interface selected
4 VLHASH	VLAN Hash Filter Selected This bit is set to 1 when the Enable VLAN Hash Table Based Filtering option is selected 0 - VLAN Hash Filter not selected 1 - VLAN Hash Filter selected
3 PCSSEL	PCS Registers (TBI, SGMII, or RTBI PHY interface) This bit is set to 1 when the TBI, SGMII, or RTBI PHY interface option is selected 0 - No PCS Registers (TBI, SGMII, or RTBI PHY interface) 1 - PCS Registers (TBI, SGMII, or RTBI PHY interface)
2 HDSEL	Half-duplex Support This bit is set to 1 when the half-duplex mode is selected 0 - No Half-duplex support 1 - Half-duplex support
1	1000 Mbps Support This bit is set to 1 when 1000 Mbps is selected as the Mode of Operation

Table continues on the next page...

Table continued from the previous page...

Field	Description
GMIISEL	0 - No 1000 Mbps support 1 - 1000 Mbps support
0 MIISEL	10 or 100 Mbps Support This bit is set to 1 when 10/100 Mbps is selected as the Mode of Operation 0 - No 10 or 100 Mbps support 1 - 10 or 100 Mbps support

43.8.1.39 Optional Features or Functions 1 (MAC_HW_FEATURE1)

This register indicates the presence of second set of the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

Offset

Register	Offset
MAC_HW_FEATURE1	120h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	L3L4FNUM				Reserved	HASHTBLSZ		POUOST	Reserved	RAVSEL	AVSEL	DBGMEMA	TSOEN	SPHEN	DCBEN
W																
Reset	0	1	0	0	0	0	0	1	1	0	0	1	1	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADDR64		ADVTHWO..	PTOEN	OSTEN	TXFIFOSIZE					SPRAM	RXFIFOSIZE				
W																
Reset	0	0	1	1	1	0	0	1	1	0	1	0	0	1	1	0

Fields

Field	Description
31 —	Reserved
30-27 L3L4FNUM	Total number of L3 or L4 Filters This field indicates the total number of L3 or L4 filters: 0000 - No L3 or L4 Filter 0001 - 1 L3 or L4 Filter

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0010 - 2 L3 or L4 Filters 0011 - 3 L3 or L4 Filters 0100 - 4 L3 or L4 Filters 0101 - 5 L3 or L4 Filters 0110 - 6 L3 or L4 Filters 0111 - 7 L3 or L4 Filters 1000 - 8 L3 or L4 Filters
26 —	Reserved
25-24 HASHTBLSZ	Hash Table Size This field indicates the size of the hash table: 00 - No hash table 01 - 64 10 - 128 11 - 256
23 POUOST	One Step for PTP over UDP/IP Feature Enable This bit is set to 1 when the Enable One step timestamp for PTP over UDP/IP feature is selected. 0 - One Step for PTP over UDP/IP Feature is not selected 1 - One Step for PTP over UDP/IP Feature is selected
22 —	Reserved
21 RAVSEL	Rx Side Only AV Feature Enable This bit is set to 1 when the Enable Audio Video Bridging option on Rx Side Only is selected. 0 - Rx Side Only AV Feature is not selected 1 - Rx Side Only AV Feature is selected
20 AVSEL	AV Feature Enable This bit is set to 1 when the Enable Audio Video Bridging option is selected. 0 - AV Feature is not selected 1 - AV Feature is selected
19 DBGMEMA	DMA Debug Registers Enable This bit is set to 1 when the Debug Mode Enable option is selected 0 - DMA Debug Registers option is not selected 1 - DMA Debug Registers option is selected
18 TSOEN	TCP Segmentation Offload Enable This bit is set to 1 when the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected 0 - TCP Segmentation Offload Feature is not selected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - TCP Segmentation Offload Feature is selected
17 SPHEN	Split Header Feature Enable This bit is set to 1 when the Enable Split Header Structure option is selected 0 - Split Header Feature is not selected 1 - Split Header Feature is selected
16 DCBEN	DCB Feature Enable This bit is set to 1 when the Enable Data Center Bridging option is selected 0 - DCB Feature is not selected 1 - DCB Feature is selected
15-14 ADDR64	Address Width. Address Width. This field indicates the configured address width: 00 - 32 01 - 40 10 - 48 11 - Reserved
13 ADVTHWORD	IEEE 1588 High Word Register Enable This bit is set to 1 when the Add IEEE 1588 Higher Word Register option is selected 0 - IEEE 1588 High Word Register option is not selected 1 - IEEE 1588 High Word Register option is selected
12 PTOEN	PTP Offload Enable This bit is set to 1 when the Enable PTP Timestamp Offload Feature is selected. 0 - PTP Offload feature is not selected 1 - PTP Offload feature is selected
11 OSTEN	One-Step Timestamping Enable This bit is set to 1 when the Enable One-Step Timestamp Feature is selected. 0 - One-Step Timestamping feature is not selected 1 - One-Step Timestamping feature is selected
10-6 TXFIFOSIZE	MTL Transmit FIFO Size This field contains the configured value of MTL Tx FIFO in bytes expressed as Log to base 2 minus 7, that is, Log2(TXFIFO_SIZE) -7: 0_0000 - 128 bytes 0_0001 - 256 bytes 0_0010 - 512 bytes 0_0011 - 1024 bytes 0_0100 - 2048 bytes

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0_0101 - 4096 bytes 0_0110 - 8192 bytes 0_0111 - 16384 bytes 0_1000 - 32 KB 0_1001 - 64 KB 0_1010 - 128 KB 0_1011 - Reserved
5 SPRAM	Single Port RAM Enable This bit is set to 1 when the Use single port RAM Feature is selected. 0 - Single Port RAM feature is not selected 1 - Single Port RAM feature is selected
4-0 RXFIFOSIZE	MTL Receive FIFO Size This field contains the configured value of MTL Rx FIFO in bytes expressed as Log to base 2 minus 7, that is, Log2(RXFIFO_SIZE) -7: 0_0000 - 128 bytes 0_0001 - 256 bytes 0_0010 - 512 bytes 0_0011 - 1024 bytes 0_0100 - 2048 bytes 0_0101 - 4096 bytes 0_0110 - 8192 bytes 0_0111 - 16384 bytes 0_1000 - 32 KB 0_1001 - 64 KB 0_1010 - 128 KB 0_1011 - 256 KB 0_1100 - Reserved

43.8.1.40 Optional Features or Functions 2 (MAC_HW_FEATURE2)

This register indicates the presence of third set of the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

Offset

Register	Offset
MAC_HW_FEATURE2	124h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	AUXSNAPNUM			Reserved	PPSOUTNUM			Reserved		TXCHCNT				Reserved	
W																
Reset	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXCHCNT				Reserved		TXQCNT				Reserved		RXQCNT			
W																
Reset	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0

Fields

Field	Description
31 —	Reserved
30-28 AUXSNAPNUM	Number of Auxiliary Snapshot Inputs This field indicates the number of auxiliary snapshot inputs: 000 - No auxiliary input 001 - 1 auxiliary input 010 - 2 auxiliary input 011 - 3 auxiliary input 100 - 4 auxiliary input 101 - Reserved
27 —	Reserved
26-24 PPSOUTNUM	Number of PPS Outputs This field indicates the number of PPS outputs: 000 - No PPS output 001 - 1 PPS output 010 - 2 PPS output 011 - 3 PPS output 100 - 4 PPS output 101 - Reserved
23-22 —	Reserved
21-18 TXCHCNT	Number of DMA Transmit Channels This field indicates the number of DMA Transmit channels: 0000 - 1 MTL Tx Channel

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0001 - 2 MTL Tx Channels 0010 - 3 MTL Tx Channels 0011 - 4 MTL Tx Channels 0100 - 5 MTL Tx Channels 0101 - Reserved 0110 - Reserved 0111 - Reserved
17-16 —	Reserved
15-12 RXCHCNT	Number of DMA Receive Channels This field indicates the number of DMA Receive channels: 0000 - 1 MTL Rx Channel 0001 - 2 MTL Rx Channels 0010 - 3 MTL Rx Channels 0011 - 4 MTL Rx Channels 0100 - 5 MTL Rx Channels 0101 - Reserved 0110 - Reserved 0111 - Reserved
11-10 —	Reserved
9-6 TXQCNT	Number of MTL Transmit Queues This field indicates the number of MTL Transmit queues: 0000 - 1 MTL Tx Queue 0001 - 2 MTL Tx Queues 0010 - 3 MTL Tx Queues 0011 - 4 MTL Tx Queues 0100 - 5 MTL Tx Queues 0101 - Reserved 0110 - Reserved 0111 - Reserved
5-4 —	Reserved
3-0	Number of MTL Receive Queues This field indicates the number of MTL Receive queues:

Table continues on the next page...

Table continued from the previous page...

Field	Description
RXQCNT	0000 - 1 MTL Rx Queue 0001 - 2 MTL Rx Queues 0010 - 3 MTL Rx Queues 0011 - 4 MTL Rx Queues 0100 - 5 MTL Rx Queues 0101 - Reserved 0110 - Reserved 0111 - Reserved

43.8.1.41 Optional Features or Functions 3 (MAC_HW_FEATURE3)

This register indicates the presence of fourth set the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

Offset

Register	Offset
MAC_HW_FEATURE3	128h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		ASP		TBSSEL	FPESEL	Reserved				ESTWID		ESTDEP			ESTSEL
W																
Reset	0	0	0	0	1	1	0	0	0	0	1	1	1	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	FRPES		FRPBS		FRPSEL	PDUPSEL	Reserved			DVLAN	CBTISEL	Reserved	NRVF		
W																
Reset	0	1	0	1	0	1	1	0	0	0	1	1	0	0	1	0

Fields

Field	Description
31-30 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
29-28 ASP	Automotive Safety Package Following are the encoding for the different Safety features 00 - No Safety features selected 01 - Only "ECC protection for external memory" feature is selected 10 - All the Automotive Safety features are selected without the "Parity Port Enable for external interface" feature 11 - All the Automotive Safety features are selected with the "Parity Port Enable for external interface" feature
27 TBSSEL	Time Based Scheduling Enable This bit is set to 1 when the Time Based Scheduling feature is selected. 0 - Time Based Scheduling Enable feature is not selected 1 - Time Based Scheduling Enable feature is selected
26 FPESEL	Frame Preemption Enable This bit is set to 1 when the Enable Frame preemption feature is selected. 0 - Frame Preemption Enable feature is not selected 1 - Frame Preemption Enable feature is selected
25-22 —	Reserved
21-20 ESTWID	Width of the Time Interval field in the Gate Control List This field indicates the width of the Configured Time Interval Field 00 - Width not configured 01 - 16 10 - 20 11 - 24
19-17 ESTDEP	Depth of the Gate Control List This field indicates the depth of Gate Control list expressed as Log2(DWC_EQOS_EST_DEP)-5 000 - No Depth configured 001 - 64 010 - 128 011 - 256 100 - 512 101 - 1024 110 - Reserved
16 ESTSEL	Enhancements to Scheduling Traffic Enable This bit is set to 1 when the Enable Enhancements to Scheduling Traffic feature is selected. 0 - Enable Enhancements to Scheduling Traffic feature is not selected 1 - Enable Enhancements to Scheduling Traffic feature is selected

Table continues on the next page...

Table continued from the previous page...

Field	Description
15 —	Reserved
14-13 FRPES	Flexible Receive Parser Table Entries size This field indicates the Max Number of Parser Entries supported by Flexible Receive Parser. 00 - 64 Entries 01 - 128 Entries 10 - 256 Entries 11 - Reserved
12-11 FRPBS	Flexible Receive Parser Buffer size This field indicates the supported Max Number of bytes of the packet data to be Parsed by Flexible Receive Parser. 00 - 64 Bytes 01 - 128 Bytes 10 - 256 Bytes 11 - Reserved
10 FRPSEL	Flexible Receive Parser Selected This bit is set to 1 when the Enable Flexible Programmable Receive Parser option is selected. 0 - Flexible Receive Parser feature is not selected 1 - Flexible Receive Parser feature is selected
9 PDUPSEL	Broadcast/Multicast Packet Duplication This bit is set to 1 when the Broadcast/Multicast Packet Duplication feature is selected. 0 - Broadcast/Multicast Packet Duplication feature is not selected 1 - Broadcast/Multicast Packet Duplication feature is selected
8-6 —	Reserved
5 DVLAN	Double VLAN Tag Processing Selected This bit is set to 1 when the Enable Double VLAN Processing Feature is selected. 0 - Double VLAN option is not selected 1 - Double VLAN option is selected
4 CBTISEL	Queue/Channel based VLAN tag insertion on Tx Enable This bit is set to 1 when the Enable Queue/Channel based VLAN tag insertion on Tx Feature is selected. 0 - Enable Queue/Channel based VLAN tag insertion on Tx feature is not selected 1 - Enable Queue/Channel based VLAN tag insertion on Tx feature is selected
3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
2-0 NRVF	Number of Extended VLAN Tag Filters Enabled This field indicates the Number of Extended VLAN Tag Filters selected: 000 - No Extended Rx VLAN Filters 001 - 4 Extended Rx VLAN Filters 010 - 8 Extended Rx VLAN Filters 011 - 16 Extended Rx VLAN Filters 100 - 24 Extended Rx VLAN Filters 101 - 32 Extended Rx VLAN Filters 110 - Reserved

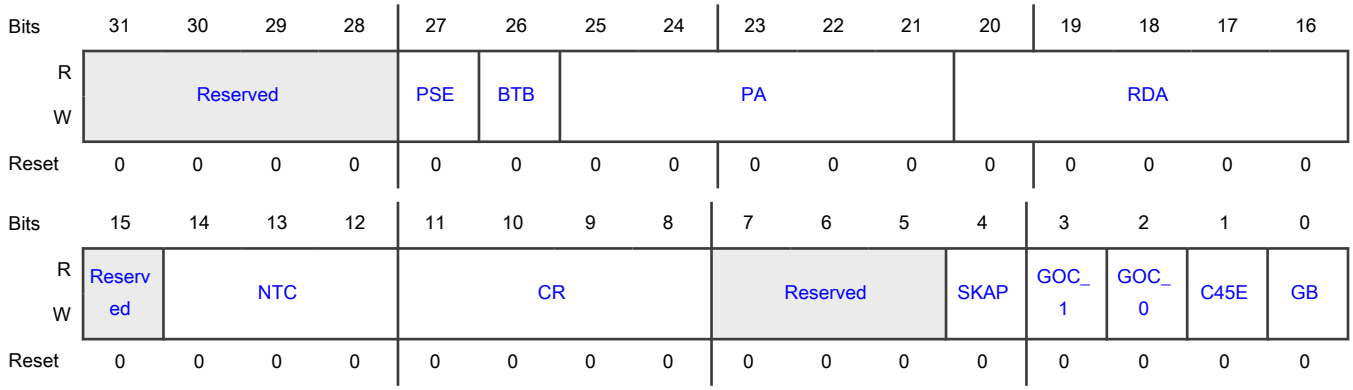
43.8.1.42 MDIO Address (MAC_MDIO_ADDRESS)

The MDIO Address register controls the management cycles to external PHY through a management interface.

Offset

Register	Offset
MAC_MDIO_ADDRESS	200h

Diagram



Fields

Field	Description
31-28 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
27 PSE	Preamble Suppression Enable When this bit is set, the SMA suppresses the 32-bit preamble and transmits MDIO frames with only 1 preamble bit. Preamble Suppression Enable When this bit is set, the SMA suppresses the 32-bit preamble and transmits MDIO frames with only 1 preamble bit. When this bit is 0, the MDIO frame always has 32 bits of preamble as defined in the IEEE specifications. 0 - Preamble Suppression disabled 1 - Preamble Suppression enabled
26 BTB	Back to Back transactions When this bit is set and the NTC has value greater than 0, then the MAC informs the completion of a read or write command at the end of frame transfer (before the trailing clocks are transmitted). Back to Back transactions When this bit is set and the NTC has value greater than 0, then the MAC informs the completion of a read or write command at the end of frame transfer (before the trailing clocks are transmitted). The software can thus initiate the next command which is executed immediately irrespective of the number trailing clocks generated for the previous frame. When this bit is reset, then the read/write command completion (GB is cleared) only after the trailing clocks are generated. In this mode, it is ensured that the NTC is always generated after each frame. This bit must not be set when NTC=0. 0 - Back to Back transactions disabled 1 - Back to Back transactions enabled
25-21 PA	Physical Layer Address This field indicates which Clause 22 PHY devices (out of 32 devices) the MAC is accessing. Physical Layer Address This field indicates which Clause 22 PHY devices (out of 32 devices) the MAC is accessing. This field indicates which Clause 45 capable PHYs (out of 32 PHYs) the MAC is accessing.
20-16 RDA	Register/Device Address These bits select the PHY register in selected Clause 22 PHY device. Register/Device Address These bits select the PHY register in selected Clause 22 PHY device. These bits select the Device (MMD) in selected Clause 45 capable PHY.
15	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
14-12 NTC	Number of Trailing Clocks This field controls the number of trailing clock cycles generated on gmii_mdc_o (MDC) after the end of transmission of MDIO frame. Number of Trailing Clocks This field controls the number of trailing clock cycles generated on gmii_mdc_o (MDC) after the end of transmission of MDIO frame. The valid values can be from 0 to 7. Programming the value to 3'h3 indicates that there are additional three clock cycles on the MDC line after the end of MDIO frame transfer.
11-8 CR	CSR Clock Range The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency used in your design: - 0000: CSR clock = 60-100 MHz; MDC clock = CSR clock/42 - 0001: CSR clock = 100-150 MHz; MDC clock = CSR clock/62 - 0010: CSR clock = 20-35 MHz; MDC clock = CSR clock/16 - 0011: CSR clock = 35-60 MHz; MDC clock = CSR clock/26 - 0100: CSR clock = 150-250 MHz; MDC clock = CSR clock/102 - 0101: CSR clock = 250-300 MHz; MDC clock = CSR clock/124 - 0110: CSR clock = 300-500 MHz; MDC clock = CSR clock/204 - 0111: CSR clock = 500-800 MHz; MDC clock = CSR clock/324 The suggested range of CSR clock frequency applicable for each value (when Bit 11 = 0) ensures that the MDC clock is approximately between 1. CSR Clock Range The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency used in your design: - 0000: CSR clock = 60-100 MHz; MDC clock = CSR clock/42 - 0001: CSR clock = 100-150 MHz; MDC clock = CSR clock/62 - 0010: CSR clock = 20-35 MHz; MDC clock = CSR clock/16 - 0011: CSR clock = 35-60 MHz; MDC clock = CSR clock/26 - 0100: CSR clock = 150-250 MHz; MDC clock = CSR clock/102 - 0101: CSR clock = 250-300 MHz; MDC clock = CSR clock/124 - 0110: CSR clock = 300-500 MHz; MDC clock = CSR clock/204 - 0111: CSR clock = 500-800 MHz; MDC clock = CSR clock/324 The suggested range of CSR clock frequency applicable for each value (when Bit 11 = 0) ensures that the MDC clock is approximately between 1.0 MHz to 2.5 MHz freqency range. When Bit 11 is set, you can achieve a higher frequency of the MDC clock than the frequency limit of 2.5 MHz (specified in the IEEE 802.3) and program a clock divider of lower value. For example, when CSR clock is of 100 MHz frequency and you program these bits as 1010, the resultant MDC clock is of 12.5 MHz which is above the range specified in IEEE 802.3. Program the following values only if the interfacing chips support faster MDC clocks: - 1000: CSR clock/4 - 1001: CSR clock/6 - 1010: CSR clock/8 - 1011: CSR clock/10 - 1100: CSR clock/12

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- 1101: CSR clock/14 - 1110: CSR clock/16 - 1111: CSR clock/18 These bits are not used for accessing RevMII. These bits are read-only if the RevMII interface is selected as single PHY interface.
7-5 —	Reserved
4 SKAP	Skip Address Packet When this bit is set, the SMA does not send the address packets before read, write, or post-read increment address packets. Skip Address Packet When this bit is set, the SMA does not send the address packets before read, write, or post-read increment address packets. This bit is valid only when C45E is set. 0 - Skip Address Packet is disabled 1 - Skip Address Packet is enabled
3 GOC_1	GMII Operation Command 1 This bit is higher bit of the operation command to the PHY or RevMII, GOC_1 and GOC_O is encoded as follows: - 00: Reserved - 01: Write - 10: Post Read Increment Address for Clause 45 PHY - 11: Read When Clause 22 PHY or RevMII is enabled, only Write and Read commands are valid. 0 - GMII Operation Command 1 is disabled 1 - GMII Operation Command 1 is enabled
2 GOC_0	GMII Operation Command 0 This is the lower bit of the operation command to the PHY or RevMII. GMII Operation Command 0 This is the lower bit of the operation command to the PHY or RevMII. When in SMA mode (MDIO master) this bit along with GOC_1 determines the operation to be performed to the PHY. When only RevMII is selected in configuration this bit is read-only and tied to 1. 0 - GMII Operation Command 0 is disabled 1 - GMII Operation Command 0 is enabled
1 C45E	Clause 45 PHY Enable When this bit is set, Clause 45 capable PHY is connected to MDIO. Clause 45 PHY Enable When this bit is set, Clause 45 capable PHY is connected to MDIO. When this bit is reset, Clause 22 capable PHY is connected to MDIO. 0 - Clause 45 PHY is disabled 1 - Clause 45 PHY is enabled
0	GMII Busy The application sets this bit to instruct the SMA to initiate a Read or Write access to the MDIO slave.

Table continues on the next page...

Table continued from the previous page...

Field	Description
GB	GMII Busy The application sets this bit to instruct the SMA to initiate a Read or Write access to the MDIO slave. The MAC clears this bit after the MDIO frame transfer is completed. Hence the software must not write or change any of the fields in MAC_MDIO_ADDRESS and MAC_MDIO_DATA registers as long as this bit is set. For write transfers, the application must first write 16-bit data in the GDI field (and also RA field when C45E is set) in MAC_MDIO_DATA register before setting this bit. When C45E is set, it should also write into the RA field of MAC_MDIO_DATA register before initiating a read transfer. When a read transfer is completed (GB=0), the data read from the PHY register is valid in the GD field of the MAC_MDIO_DATA register. Note: Even if the addressed PHY is not present, there is no change in the functionality of this bit. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - GMII Busy is disabled 1 - GMII Busy is enabled

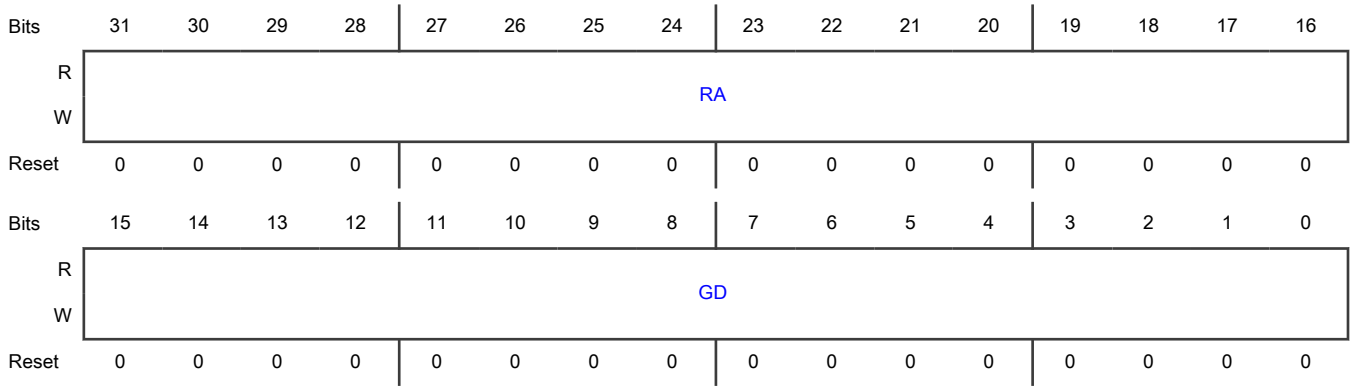
43.8.1.43 MAC MDIO Data (MAC_MDIO_DATA)

The MAC_MDIO_DATA register stores the Write data to be written to the PHY register located at the address specified in MAC_MDIO_ADDRESS. This register also stores the Read data from the PHY register located at the address specified by MAC_MDIO_ADDRESS register.

Offset

Register	Offset
MAC_MDIO_DATA	204h

Diagram



Fields

Field	Description
31-16 RA	Register Address This field is valid only when C45E is set. Register Address This field is valid only when C45E is set. It contains the Register Address in the PHY to which the MDIO frame is intended for.
15-0 GD	GMII Data This field contains the 16-bit data value read from the PHY or RevMII after a Management Read operation or the 16-bit data value to be written to the PHY or RevMII before a Management Write operation.

43.8.1.44 CSR Software Control (MAC_CSR_SW_CTRL)

This register contains SW programmable controls for changing the CSR access response and status bits clearing.

Offset

Register	Offset
MAC_CSR_SW_CTRL	230h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							Reserved	Reserved							RCWE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-9 —	Reserved
8 —	Reserved
7-1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
0 RCWE	Register Clear on Write 1 Enable When this bit is set, the access mode of some register fields changes to Clear on Write 1, the application needs to set that respective bit to 1 to clear it. Register Clear on Write 1 Enable When this bit is set, the access mode of some register fields changes to Clear on Write 1, the application needs to set that respective bit to 1 to clear it. When this bit is reset, the access mode of these register fields remain as Clear on Read. 0 - Register Clear on Write 1 is disabled 1 - Register Clear on Write 1 is enabled

43.8.1.45 Frame Preemption Control (MAC_FPE_CTRL_STS)

This register controls the operation of Frame Preemption.

Offset

Register	Offset
MAC_FPE_CTRL_STS	234h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												TRSP	TVER	RRSP	RVER
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												S1_SET_0	SRSP	SVER	EFPE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-20 —	Reserved
19	Transmitted Respond Frame Set when a Respond mPacket is transmitted (triggered by setting SRSP field).

Table continues on the next page...

Table continued from the previous page...

Field	Description
TRSP	Transmitted Respond Frame Set when a Respond mPacket is transmitted (triggered by setting SRSP field). An interrupt can be generated for this event if FPEIE bit of MAC_INTERRUPT_ENABLE is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Not transmitted Respond Frame 1 - transmitted Respond Frame
18 TVER	Transmitted Verify Frame Set when a Verify mPacket is transmitted (triggered by setting SVER field). Transmitted Verify Frame Set when a Verify mPacket is transmitted (triggered by setting SVER field). An interrupt can be generated for this event if FPEIE bit of MAC_INTERRUPT_ENABLE is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Not transmitted Verify Frame 1 - transmitted Verify Frame
17 RRSP	Received Respond Frame Set when a Respond mPacket is received. Received Respond Frame Set when a Respond mPacket is received. An interrupt can be generated for this event if FPEIE bit of MAC_INTERRUPT_ENABLE is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Not received Respond Frame 1 - Received Respond Frame
16 RVER	Received Verify Frame Set when a Verify mPacket is received. Received Verify Frame Set when a Verify mPacket is received. An interrupt can be generated for this event if FPEIE bit of MAC_INTERRUPT_ENABLE is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Not received Verify Frame 1 - Received Verify Frame
15-4 —	Reserved
3	Reserved, Must be set to "0".

Table continues on the next page...

Table continued from the previous page...

Field	Description
S1_SET_0	Reserved, Must be set to "0". This field is reserved for Synopsys internal use, and must always be set to "0" unless instructed by Synopsys. Setting to "1" might cause unexpected behavior in the IP.
2 SRSP	Send Respond mPacket When set indicates hardware to send a Respond mPacket. Send Respond mPacket When set indicates hardware to send a Respond mPacket. Reset by hardware after sending the Respond mPacket. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Send Respond mPacket is disabled 1 - Send Respond mPacket is enabled
1 SVER	Send Verify mPacket When set indicates hardware to send a verify mPacket. Send Verify mPacket When set indicates hardware to send a verify mPacket. Reset by hardware after sending the Verify mPacket. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Send Verify mPacket is disabled 1 - Send Verify mPacket is enabled
0 EFPE	Enable Tx Frame Preemption When set Frame Preemption Tx functionality is enabled. 0 - Tx Frame Preemption is disabled 1 - Tx Frame Preemption is enabled

43.8.1.46 32-bit Binary Rollover Equivalent Time (MAC_PRESN_TIME_NS)

This register contains the 32-bit binary rollover equivalent time of the PTP System Time in ns

Exists when DWC_EQOS_FLEXI_PPS_OUT_EN is configured

Offset

Register	Offset
MAC_PRESN_TIME_NS	240h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MPTN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MPTN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MPTN	MAC 1722 Presentation Time in ns These bits indicate the value of the 32-bit binary rollover equivalent time of the PTP System Time in ns

43.8.1.47 MAC 1722 Presentation Time (MAC_PRESN_TIME_UPDT)

This field holds the 32-bit value of MAC 1722 Presentation Time in ns, that should be added to the Current Presentation Time Counter value. Init happens when TSINIT is set, and update happens when the TSUPDT bit is set (TSINIT and TSINIT defined in TIMESTAMP_CONTROL register).

Exists when DWC_EQOS_FLEXI_PPS_OUT_EN is configured

Offset

Register	Offset
MAC_PRESN_TIME_UPDT	244h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MPTU															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MPTU															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MPTU	MAC 1722 Presentation Time Update This field holds the init value or the update value for the presentation time. MAC 1722 Presentation Time Update This field holds the init value or the update value for the presentation time. When used for update, this field holds the 32-bit value in ns, that should be added to the Current Presentation Time Counter value. Init happens when TSINIT is set, and update happens when the TSUPDT bit is set (TSINIT and TSINIT defined in MAC_TIMESTAMP_CONTROL register). When ADDSUB field of MAC_SYSTEM_TIME_NANOSECONDS_UPDATE is set, this value is directly used for subtraction

43.8.1.48 MAC Address0 High (MAC_ADDRESS0_HIGH)

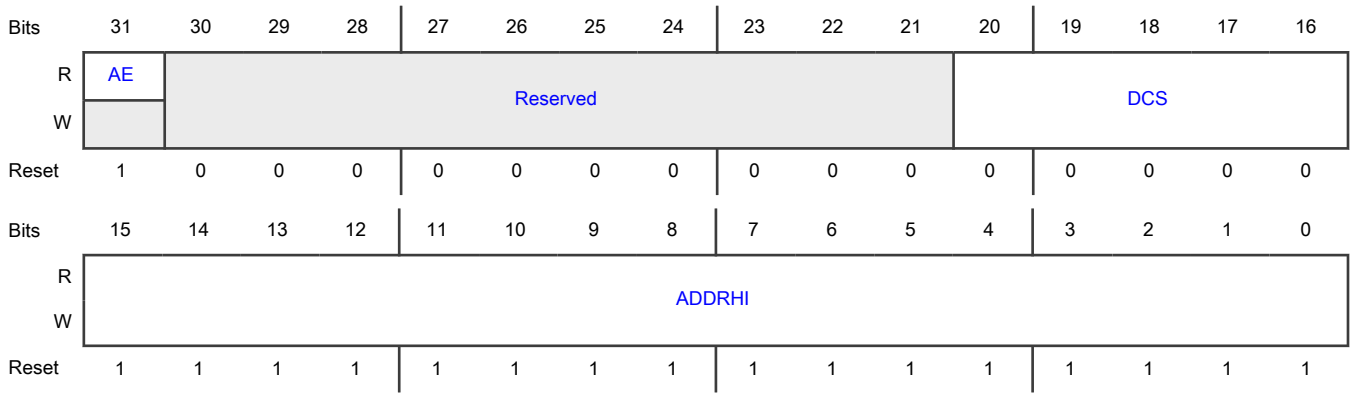
The MAC Address0 High register holds the upper 16 bits of the first 6-byte MAC address of the station. The first DA byte that is received on the (G)MII interface corresponds to the LS byte (Bits [7:0]) of the MAC Address Low register. For example, if 0x112233445566 is received (0x11 in lane 0 of the first column) on the (G)MII as the destination address, then the MacAddress0 Register [47:0] is compared with 0x665544332211.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, then the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address0 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS0_HIGH	300h

Diagram



Fields

Field	Description
31	Address Enable This bit is always set to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Description
AE	0 - INVALID : This bit must be always set to 1 1 - This bit is always set to 1
30-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address0 content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address0 content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address0 content is routed.
15-0 ADDRHI	MAC Address0[47:32] This field contains the upper 16 bits [47:32] of the first 6-byte MAC address. MAC Address0[47:32] This field contains the upper 16 bits [47:32] of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.

43.8.1.49 MAC Address0 Low (MAC_ADDRESS0_LOW)

The MAC Address0 Low register holds the lower 32 bits of the 6-byte first MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS0_LOW	304h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC Address0[31:0] This field contains the lower 32 bits of the first 6-byte MAC address. MAC Address0[31:0] This field contains the lower 32 bits of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.

43.8.1.50 MAC Address1 High (MAC_ADDRESS1_HIGH)

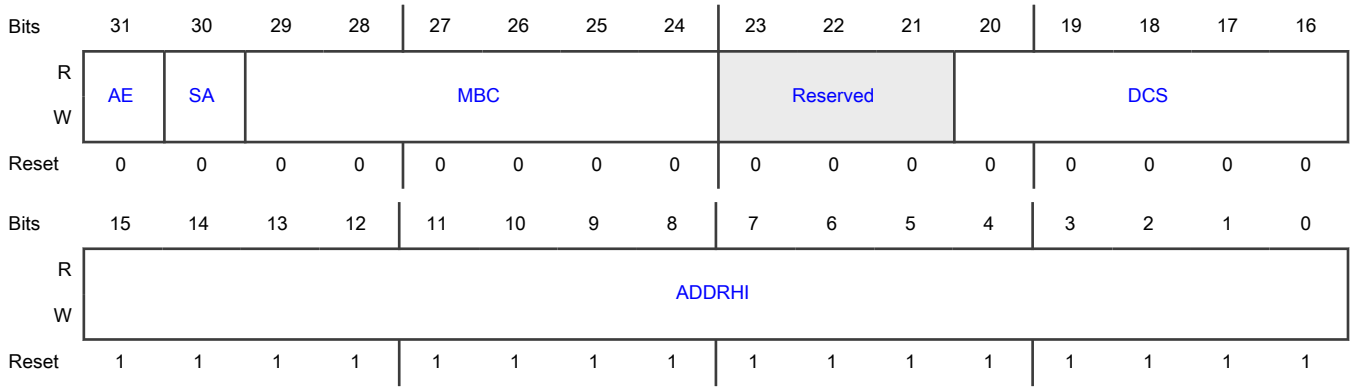
The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS1_HIGH	308h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS1_High[15:8] - Bit 28: ADDRESS1_High[7:0] - Bit 27: ADDRESS1_Low[31:24] - .. - Bit 24: ADDRESS1_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

43.8.1.51 MAC Address1 Low (MAC_ADDRESS1_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS1_LOW	30Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.52 MAC Address2 High (MAC_ADDRESS2_HIGH)

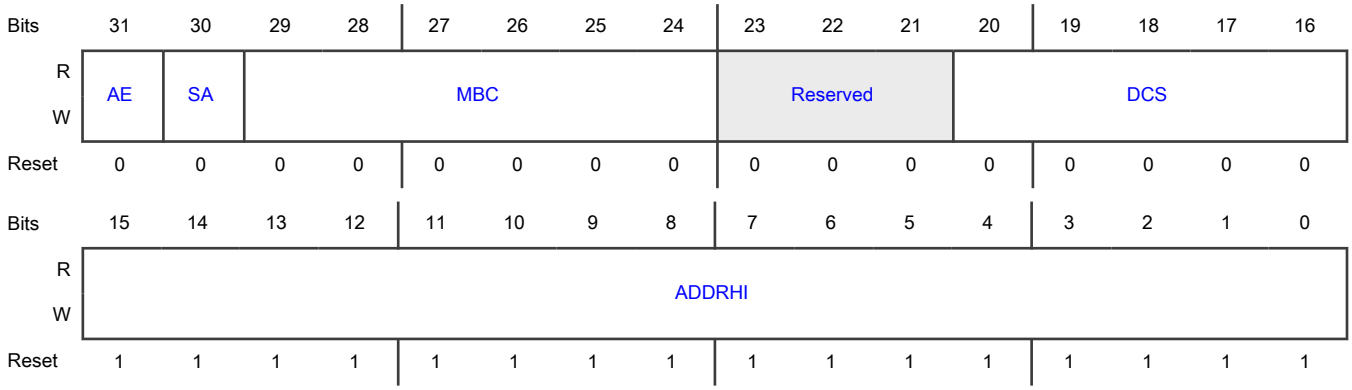
The MAC ADDRESS2 High register holds the upper 16 bits of the third 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS2 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS2_HIGH	310h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the third MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the third MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS2[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS2[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS2[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS2 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS2_High[15:8] - Bit 28: ADDRESS2_High[7:0] - Bit 27: ADDRESS2_Low[31:24] - .. - Bit 24: ADDRESS2_Low[7:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address2 content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address2 content is routed.
15-0 ADDRHI	MAC ADDRESS2 [47:32] This field contains the upper 16 bits[47:32] of the third 6-byte MAC address.

43.8.1.53 MAC Address2 Low (MAC_ADDRESS2_LOW)

The MAC ADDRESS2 Low register holds the lower 32 bits of the third 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS2_LOW	314h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS2 [31:0] This field contains the lower 32 bits of third 6-byte MAC address. MAC ADDRESS2 [31:0] This field contains the lower 32 bits of third 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.54 MAC Address3 High (MAC_ADDRESS3_HIGH)

The MAC ADDRESS3 High register holds the upper 16 bits of the fourth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS3 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS3_HIGH	318h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved				DCS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the fourth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the fourth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS3[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS3[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS3[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS3 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS3_High[15:8] - Bit 28: ADDRESS3_High[7:0] - Bit 27: ADDRESS3_Low[31:24] - .. - Bit 24: ADDRESS3_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address3 content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address3 content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address3 content is routed.
15-0 ADDRHI	MAC ADDRESS3 [47:32] This field contains the upper 16 bits[47:32] of the fourth 6-byte MAC address.

43.8.1.55 MAC Address3 Low (MAC_ADDRESS3_LOW)

The MAC ADDRESS3 Low register holds the lower 32 bits of the fourth 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS3_LOW	31Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS3 [31:0] This field contains the lower 32 bits of fourth 6-byte MAC address. MAC ADDRESS3 [31:0] This field contains the lower 32 bits of fourth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.56 MAC Address4 High (MAC_ADDRESS4_HIGH)

The MAC ADDRESS4 High register holds the upper 16 bits of the fifth 6-byte MAC address of the station.

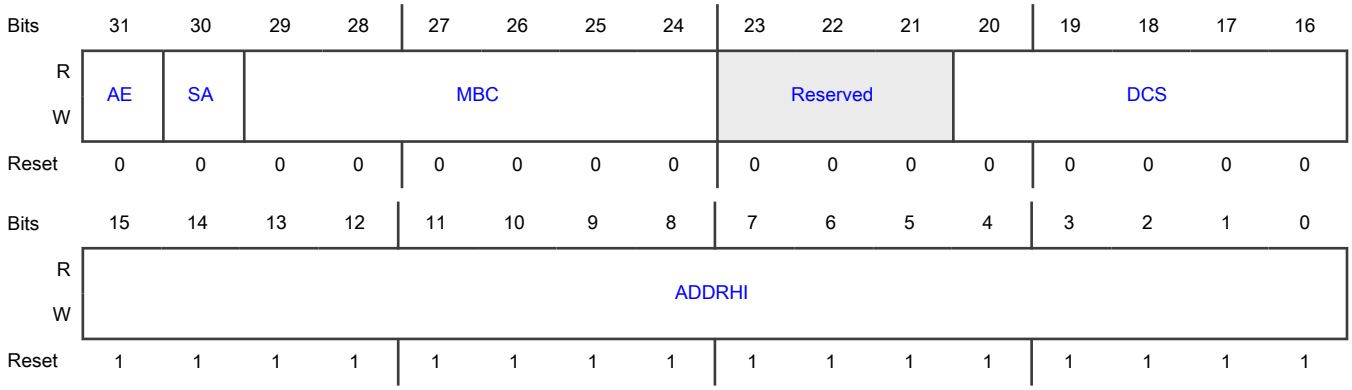
The MAC ADDRESS4 High register holds the upper 16 bits of the fifth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS4 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS4_HIGH	320h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the fifth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the fifth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS4[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS4[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS4[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS4 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS4_High[15:8] - Bit 28: ADDRESS4_High[7:0] - Bit 27: ADDRESS4_Low[31:24] - .. - Bit 24: ADDRESS4_Low[7:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS4 [47:32] This field contains the upper 16 bits[47:32] of the fifth 6-byte MAC address.

43.8.1.57 MAC Address4 Low (MAC_ADDRESS4_LOW)

The MAC ADDRESS4 Low register holds the lower 32 bits of the fifth 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS4_LOW	324h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS4 [31:0] This field contains the lower 32 bits of fifth 6-byte MAC address. MAC ADDRESS4 [31:0] This field contains the lower 32 bits of fifth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.58 MAC Address5 High (MAC_ADDRESS5_HIGH)

The MAC ADDRESS5 High register holds the upper 16 bits of the sixth 6-byte MAC address of the station.

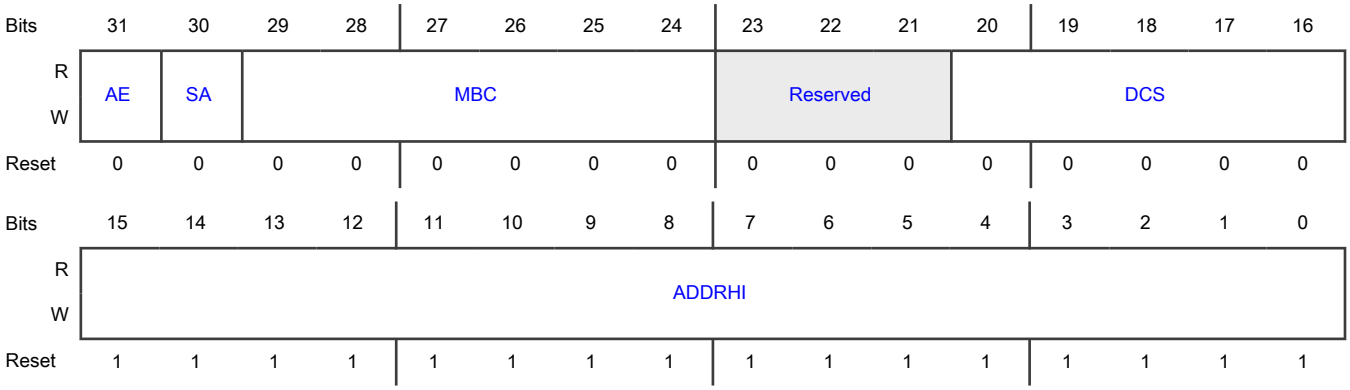
The MAC ADDRESS5 High register holds the upper 16 bits of the sixth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS5 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS5_HIGH	328h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the sixth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the sixth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS5[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS5[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS5[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS5 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS5_High[15:8] - Bit 28: ADDRESS5_High[7:0] - Bit 27: ADDRESS5_Low[31:24] - .. - Bit 24: ADDRESS5_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS5 [47:32] This field contains the upper 16 bits[47:32] of the sixth 6-byte MAC address.

43.8.1.59 MAC Address5 Low (MAC_ADDRESS5_LOW)

The MAC ADDRESS5 Low register holds the lower 32 bits of the sixth 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS5_LOW	32Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS5 [31:0] This field contains the lower 32 bits of sixth 6-byte MAC address. MAC ADDRESS5 [31:0] This field contains the lower 32 bits of sixth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.60 MAC Address6 High (MAC_ADDRESS6_HIGH)

The MAC ADDRESS6 High register holds the upper 16 bits of the seventh 6-byte MAC address of the station.

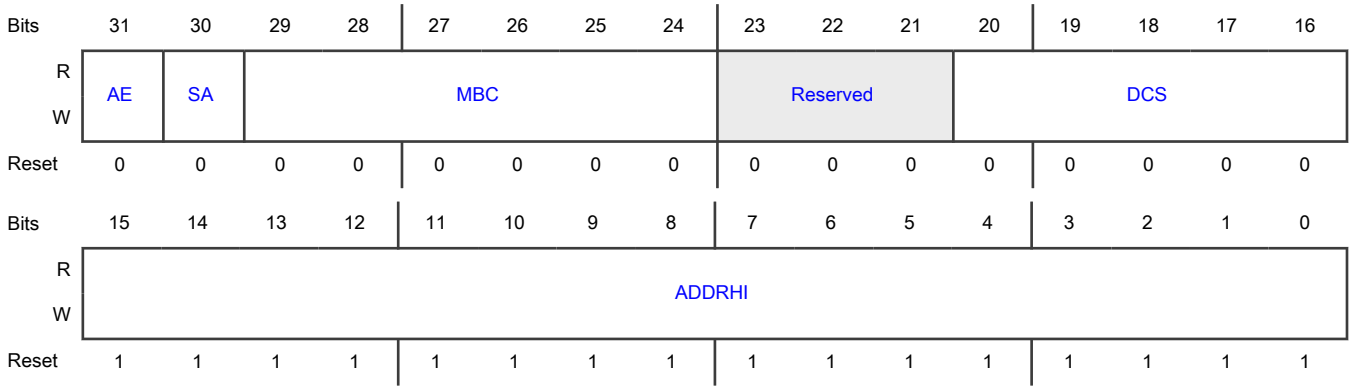
The MAC ADDRESS6 High register holds the upper 16 bits of the seventh 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS6 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS6_HIGH	330h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the seventh MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the seventh MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS6[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS6[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS6[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS6 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS6_High[15:8] - Bit 28: ADDRESS6_High[7:0] - Bit 27: ADDRESS6_Low[31:24] - .. - Bit 24: ADDRESS6_Low[7:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS6 [47:32] This field contains the upper 16 bits[47:32] of the seventh 6-byte MAC address.

43.8.1.61 MAC Address6 Low (MAC_ADDRESS6_LOW)

The MAC ADDRESS6 Low register holds the lower 32 bits of the seventh 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS6_LOW	334h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS6 [31:0] This field contains the lower 32 bits of seventh 6-byte MAC address. MAC ADDRESS6 [31:0] This field contains the lower 32 bits of seventh 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.62 MAC Address7 High (MAC_ADDRESS7_HIGH)

The MAC ADDRESS7 High register holds the upper 16 bits of the eighth 6-byte MAC address of the station.

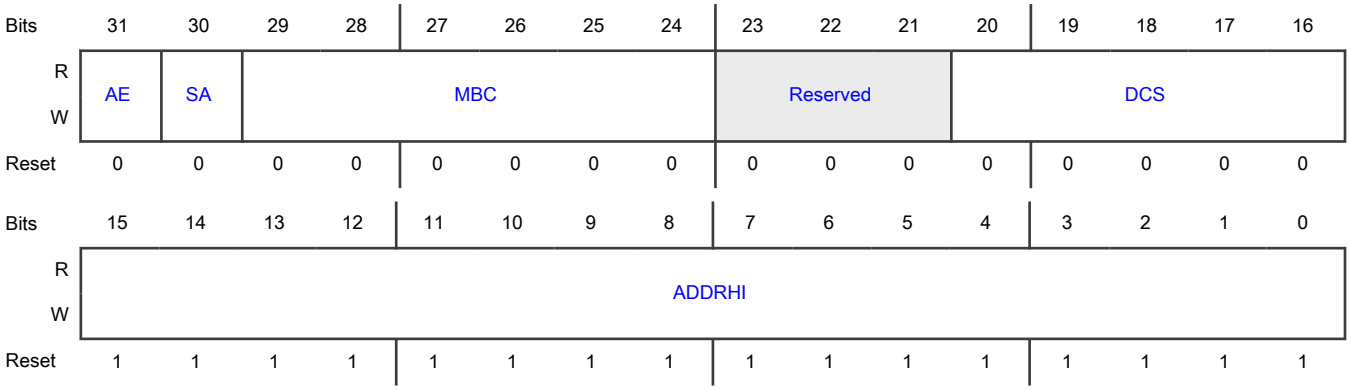
The MAC ADDRESS7 High register holds the upper 16 bits of the eighth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS7 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS7_HIGH	338h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the eighth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the eighth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS7[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS7[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS7[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS7 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS7_High[15:8] - Bit 28: ADDRESS7_High[7:0] - Bit 27: ADDRESS7_Low[31:24] - .. - Bit 24: ADDRESS7_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS7 [47:32] This field contains the upper 16 bits[47:32] of the eighth 6-byte MAC address.

43.8.1.63 MAC Address7 Low (MAC_ADDRESS7_LOW)

The MAC ADDRESS7 Low register holds the lower 32 bits of the eighth 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS7_LOW	33Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS7 [31:0] This field contains the lower 32 bits of eighth 6-byte MAC address. MAC ADDRESS7 [31:0] This field contains the lower 32 bits of eighth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.64 MAC Address8 High (MAC_ADDRESS8_HIGH)

The MAC ADDRESS8 High register holds the upper 16 bits of the ninth 6-byte MAC address of the station.

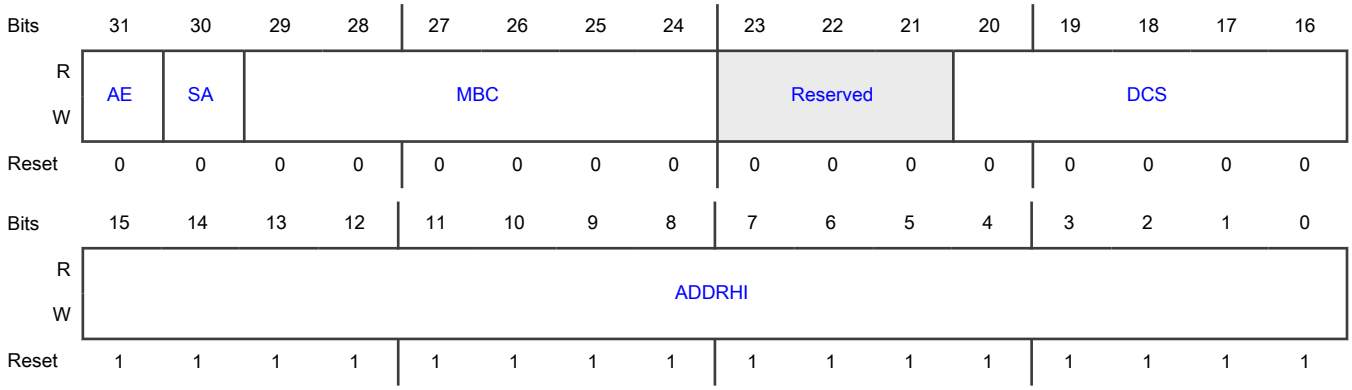
The MAC ADDRESS8 High register holds the upper 16 bits of the ninth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS8 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS8_HIGH	340h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the ninth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the ninth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS8[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS8[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS8[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS8 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS9_High[15:8] - Bit 28: ADDRESS9_High[7:0] - Bit 27: ADDRESS9_Low[31:24] - .. - Bit 24: ADDRESS9_Low[7:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS8 [47:32] This field contains the upper 16 bits[47:32] of the ninth 6-byte MAC address.

43.8.1.65 MAC Address8 Low (MAC_ADDRESS8_LOW)

The MAC ADDRESS8 Low register holds the lower 32 bits of the ninth 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS8_LOW	344h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS8 [31:0] This field contains the lower 32 bits of ninth 6-byte MAC address. MAC ADDRESS8 [31:0] This field contains the lower 32 bits of ninth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.66 MAC Address9 High (MAC_ADDRESS9_HIGH)

The MAC ADDRESS9 High register holds the upper 16 bits of the tenth 6-byte MAC address of the station.

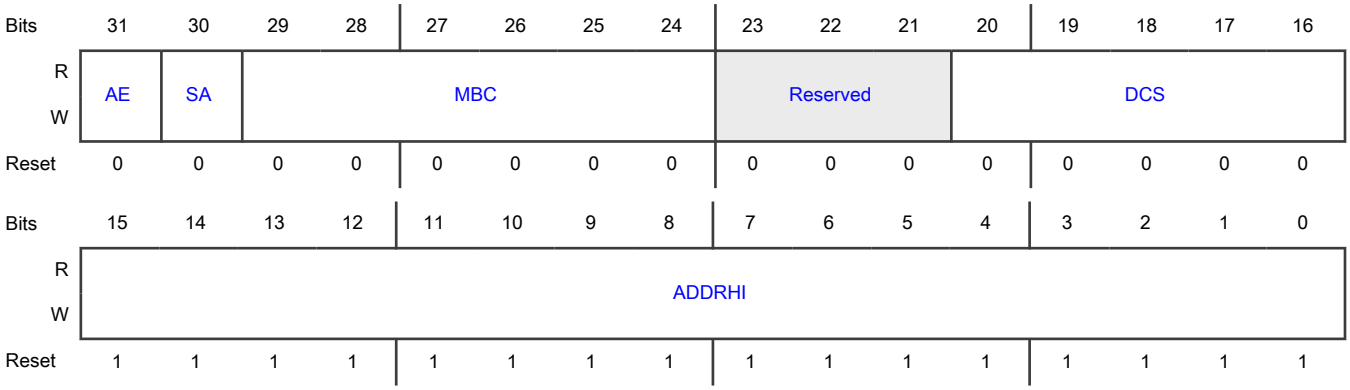
The MAC ADDRESS9 High register holds the upper 16 bits of the tenth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS9 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS9_HIGH	348h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the tenth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the tenth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS9[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS9[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS9[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS9 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS9_High[15:8] - Bit 28: ADDRESS9_High[7:0] - Bit 27: ADDRESS9_Low[31:24] - .. - Bit 24: ADDRESS9_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS9 [47:32] This field contains the upper 16 bits[47:32] of the tenth 6-byte MAC address.

43.8.1.67 MAC Address9 Low (MAC_ADDRESS9_LOW)

The MAC ADDRESS9 Low register holds the lower 32 bits of the tenth 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS9_LOW	34Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS9 [31:0] This field contains the lower 32 bits of tenth 6-byte MAC address. MAC ADDRESS9 [31:0] This field contains the lower 32 bits of tenth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.68 MAC Address10 High (MAC_ADDRESS10_HIGH)

The MAC ADDRESS10 High register holds the upper 16 bits of the eleventh 6-byte MAC address of the station.

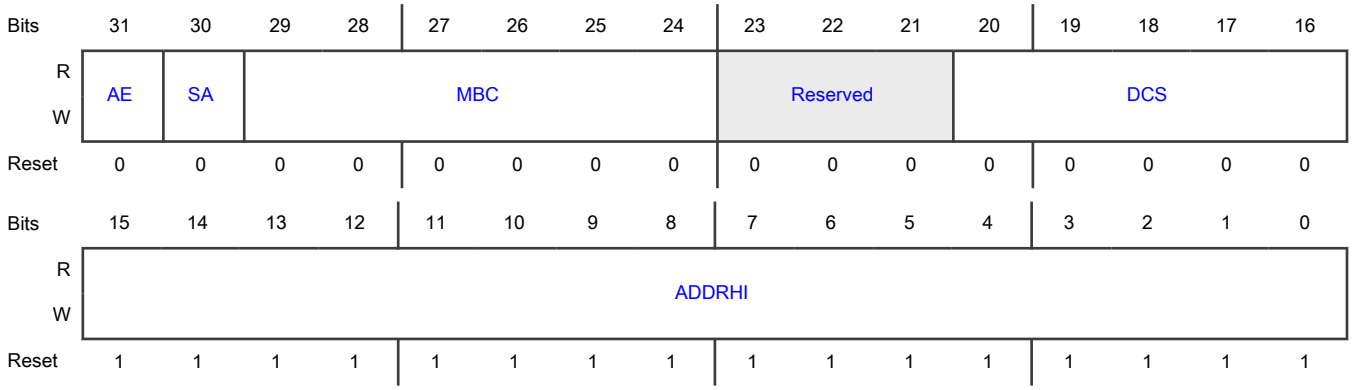
The MAC ADDRESS10 High register holds the upper 16 bits of the eleventh 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS10 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS10_HIGH	350h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the eleventh MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the eleventh MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS10[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS10[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS10[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS10 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS10_High[15:8] - Bit 28: ADDRESS10_High[7:0] - Bit 27: ADDRESS10_Low[31:24] - .. - Bit 24: ADDRESS10_Low[7:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS10 [47:32] This field contains the upper 16 bits[47:32] of the eleventh 6-byte MAC address.

43.8.1.69 MAC Address10 Low (MAC_ADDRESS10_LOW)

The MAC ADDRESS10 Low register holds the lower 32 bits of the eleventh 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS10_LOW	354h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS10 [31:0] This field contains the lower 32 bits of eleventh 6-byte MAC address. MAC ADDRESS10 [31:0] This field contains the lower 32 bits of eleventh 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.70 MAC Address11 High (MAC_ADDRESS11_HIGH)

The MAC ADDRESS11 High register holds the upper 16 bits of the twelfth 6-byte MAC address of the station.

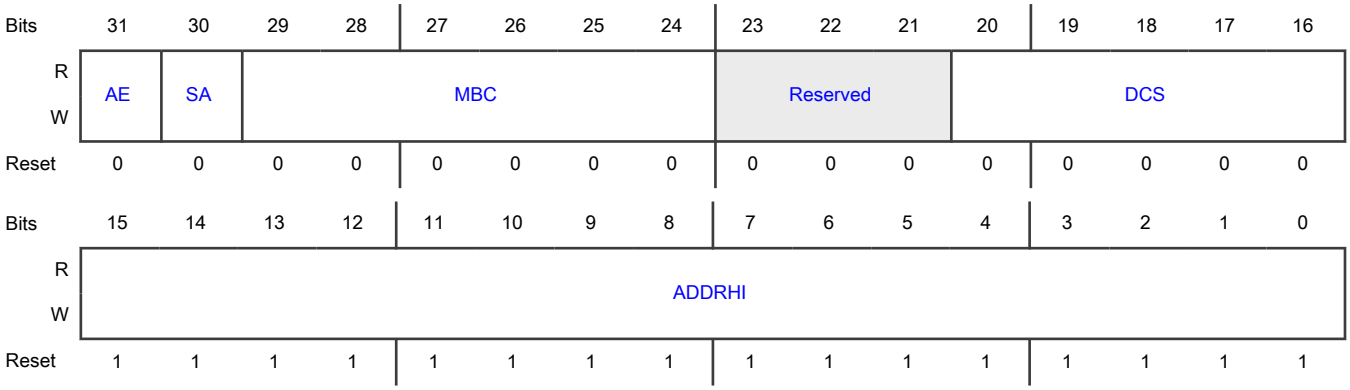
The MAC ADDRESS11 High register holds the upper 16 bits of the twelfth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS11 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS11_HIGH	358h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the twelfth MAC address for perfect filtering. Address Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the address filter module uses the twelfth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS11[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS11[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS11[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS11 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS11_High[15:8] - Bit 28: ADDRESS11_High[7:0] - Bit 27: ADDRESS11_Low[31:24] - .. - Bit 24: ADDRESS11_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-0 ADDRHI	MAC ADDRESS11 [47:32] This field contains the upper 16 bits[47:32] of the twelfth 6-byte MAC address.

43.8.1.71 MAC Address11 Low (MAC_ADDRESS11_LOW)

The MAC ADDRESS11 Low register holds the lower 32 bits of the twelfth 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS11_LOW	35Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS11 [31:0] This field contains the lower 32 bits of twelfth 6-byte MAC address. MAC ADDRESS11 [31:0] This field contains the lower 32 bits of twelfth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.72 MAC Address12 High (MAC_ADDRESS12_HIGH)

The MAC ADDRESS12 High register holds the upper 16 bits of the thirteenth 6-byte MAC address of the station.

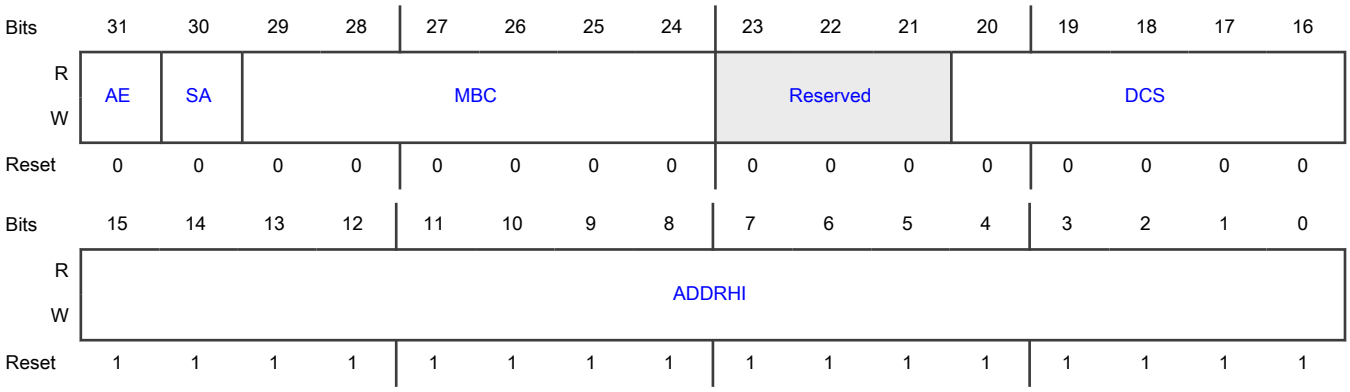
The MAC ADDRESS12 High register holds the upper 16 bits of the thirteenth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS12 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS12_HIGH	360h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the thirteenth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the thirteenth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS12[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS12[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS12[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS12 registers. Each bit controls the masking of the bytes as follows:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- Bit 29: ADDRESS12_High[15:8] - Bit 28: ADDRESS12_High[7:0] - Bit 27: ADDRESS12_Low[31:24] - .. - Bit 24: ADDRESS12_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS12 [47:32] This field contains the upper 16 bits[47:32] of the thirteenth 6-byte MAC address.

43.8.1.73 MAC Address12 Low (MAC_ADDRESS12_LOW)

The MAC ADDRESS12 Low register holds the lower 32 bits of the thirteenth 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS12_LOW	364h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS12 [31:0] This field contains the lower 32 bits of thirteenth 6-byte MAC address. MAC ADDRESS12 [31:0] This field contains the lower 32 bits of thirteenth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.74 MAC Address13 High (MAC_ADDRESS13_HIGH)

The MAC ADDRESS13 High register holds the upper 16 bits of the 14th 6-byte MAC address of the station.

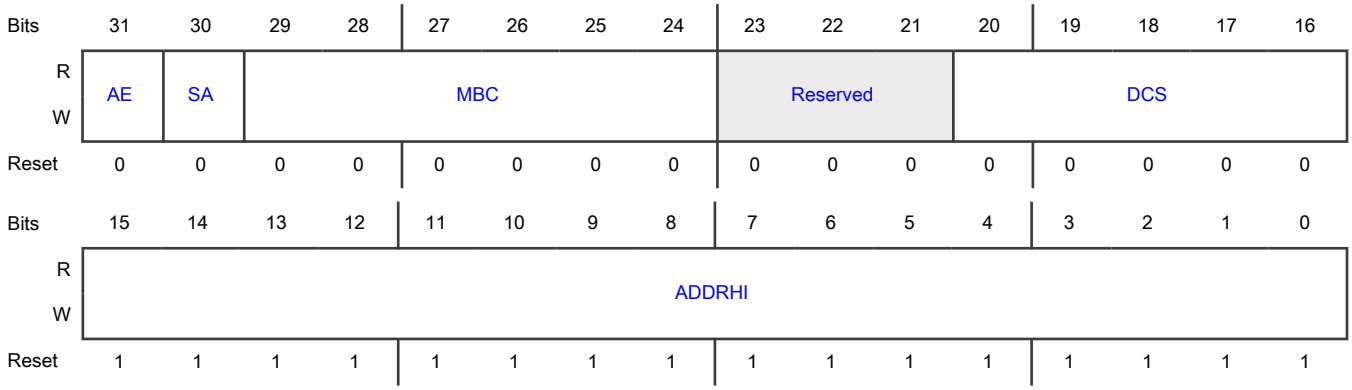
The MAC ADDRESS13 High register holds the upper 16 bits of the 14th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS13 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS13_HIGH	368h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 14th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 14th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS13[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS13[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS13[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS13 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS13_High[15:8] - Bit 28: ADDRESS13_High[7:0] - Bit 27: ADDRESS13_Low[31:24] - .. - Bit 24: ADDRESS13_Low[7:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS13 [47:32] This field contains the upper 16 bits[47:32] of the 14th 6-byte MAC address.

43.8.1.75 MAC Address13 Low (MAC_ADDRESS13_LOW)

The MAC ADDRESS13 Low register holds the lower 32 bits of the 14th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS13_LOW	36Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS13 [31:0] This field contains the lower 32 bits of 14th 6-byte MAC address. MAC ADDRESS13 [31:0] This field contains the lower 32 bits of 14th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.76 MAC Address14 High (MAC_ADDRESS14_HIGH)

The MAC ADDRESS14 High register holds the upper 16 bits of the 15th 6-byte MAC address of the station.

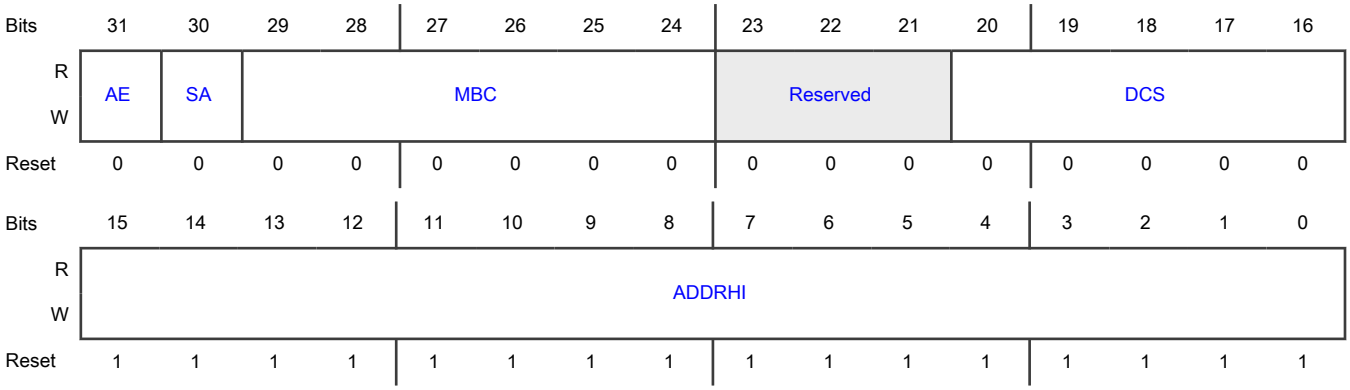
The MAC ADDRESS14 High register holds the upper 16 bits of the 15th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS14 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS14_HIGH	370h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 15th MAC address for perfect filtering. Address Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the address filter module uses the 15th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS14[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS14[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS14[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS14 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS14_High[15:8] - Bit 28: ADDRESS14_High[7:0] - Bit 27: ADDRESS14_Low[31:24] - .. - Bit 24: ADDRESS14_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-0 ADDRHI	MAC ADDRESS14 [47:32] This field contains the upper 16 bits[47:32] of the 15th 6-byte MAC address.

43.8.1.77 MAC Address14 Low (MAC_ADDRESS14_LOW)

The MAC ADDRESS14 Low register holds the lower 32 bits of the 15th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS14_LOW	374h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS14 [31:0] This field contains the lower 32 bits of 15th 6-byte MAC address. MAC ADDRESS14 [31:0] This field contains the lower 32 bits of 15th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.78 MAC Address15 High (MAC_ADDRESS15_HIGH)

The MAC ADDRESS15 High register holds the upper 16 bits of the 16th 6-byte MAC address of the station.

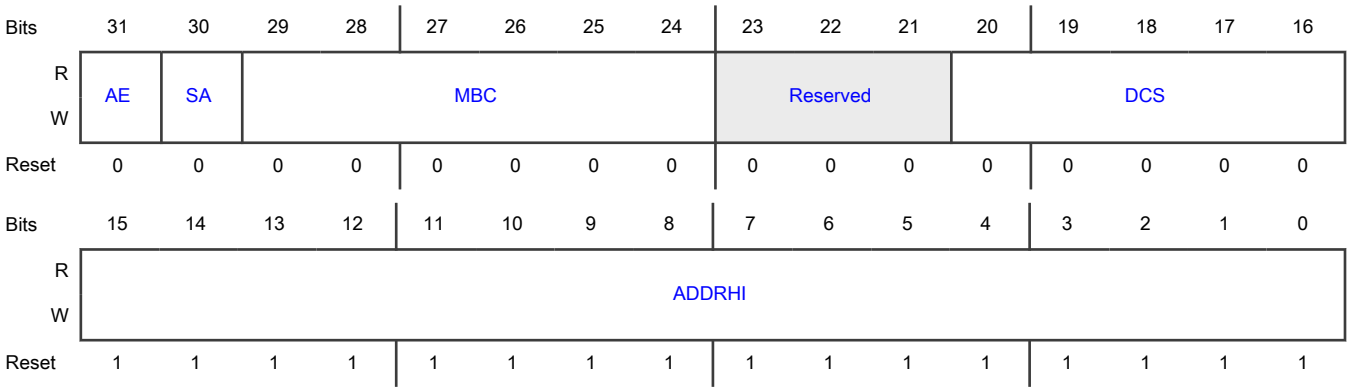
The MAC ADDRESS15 High register holds the upper 16 bits of the 16th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS15 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS15_HIGH	378h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 16th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 16th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS15[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS15[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS15[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS15 registers. Each bit controls the masking of the bytes as follows:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- Bit 29: ADDRESS15_High[15:8] - Bit 28: ADDRESS15_High[7:0] - Bit 27: ADDRESS15_Low[31:24] - .. - Bit 24: ADDRESS15_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS15 [47:32] This field contains the upper 16 bits[47:32] of the 16th 6-byte MAC address.

43.8.1.79 MAC Address15 Low (MAC_ADDRESS15_LOW)

The MAC ADDRESS15 Low register holds the lower 32 bits of the 16th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS15_LOW	37Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS15 [31:0] This field contains the lower 32 bits of 16th 6-byte MAC address. MAC ADDRESS15 [31:0] This field contains the lower 32 bits of 16th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.80 MAC Address16 High (MAC_ADDRESS16_HIGH)

The MAC ADDRESS16 High register holds the upper 16 bits of the 17th 6-byte MAC address of the station.

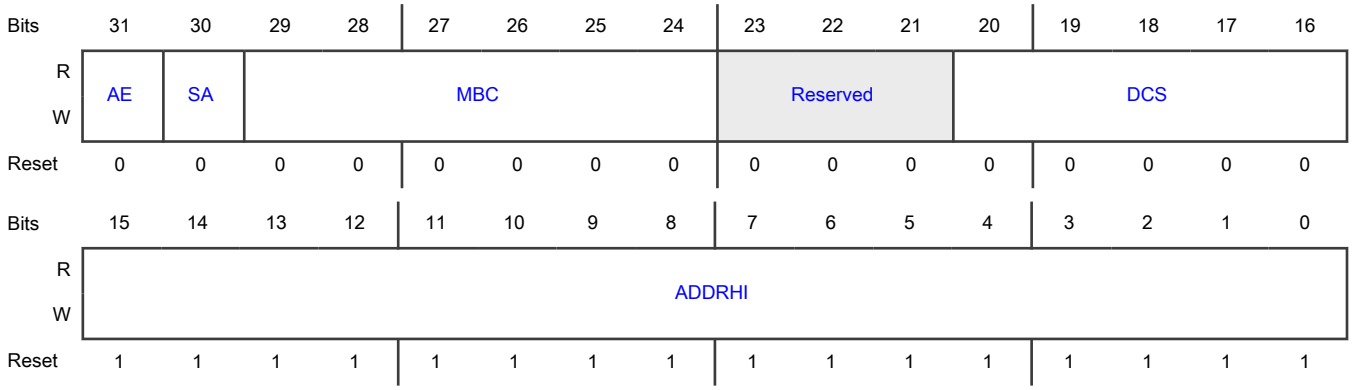
The MAC ADDRESS16 High register holds the upper 16 bits of the 17th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS16 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS16_HIGH	380h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 17th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 17th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS16[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS16[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS16[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS16 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS16_High[15:8] - Bit 28: ADDRESS16_High[7:0] - Bit 27: ADDRESS16_Low[31:24] - .. - Bit 24: ADDRESS16_Low[7:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS16 [47:32] This field contains the upper 16 bits[47:32] of the 17th 6-byte MAC address.

43.8.1.81 MAC Address16 Low (MAC_ADDRESS16_LOW)

The MAC ADDRESS16 Low register holds the lower 32 bits of the 17th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS16_LOW	384h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS16 [31:0] This field contains the lower 32 bits of 17th 6-byte MAC address. MAC ADDRESS16 [31:0] This field contains the lower 32 bits of 17th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.82 MAC Address17 High (MAC_ADDRESS17_HIGH)

The MAC ADDRESS17 High register holds the upper 16 bits of the 18th 6-byte MAC address of the station.

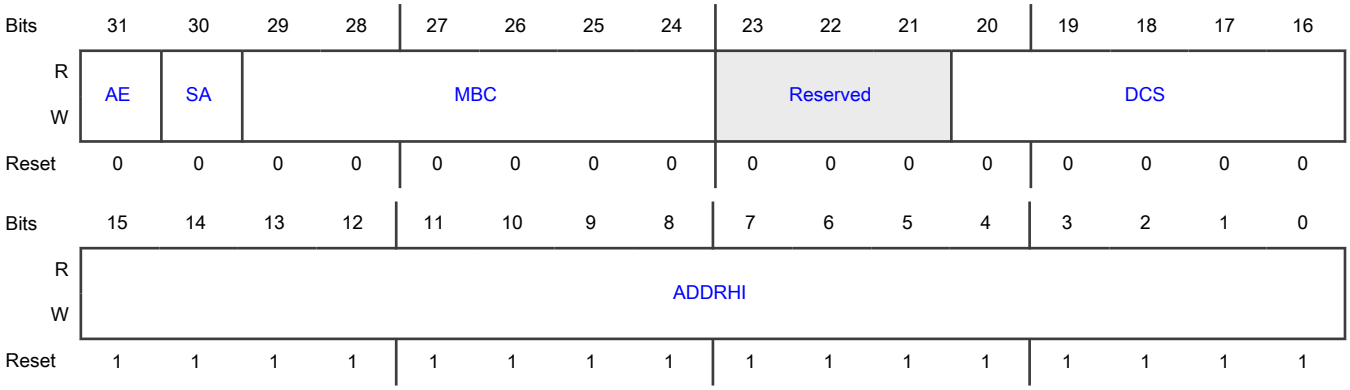
The MAC ADDRESS17 High register holds the upper 16 bits of the 18th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS17 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS17_HIGH	388h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 18th MAC address for perfect filtering. Address Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the address filter module uses the 18th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS17[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS17[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS17[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS17 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS17_High[15:8] - Bit 28: ADDRESS17_High[7:0] - Bit 27: ADDRESS17_Low[31:24] - .. - Bit 24: ADDRESS17_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-0 ADDRHI	MAC ADDRESS17 [47:32] This field contains the upper 16 bits[47:32] of the 18th 6-byte MAC address.

43.8.1.83 MAC Address17 Low (MAC_ADDRESS17_LOW)

The MAC ADDRESS17 Low register holds the lower 32 bits of the 18th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS17_LOW	38Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS17 [31:0] This field contains the lower 32 bits of 18th 6-byte MAC address. MAC ADDRESS17 [31:0] This field contains the lower 32 bits of 18th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.84 MAC Address18 High (MAC_ADDRESS18_HIGH)

The MAC ADDRESS18 High register holds the upper 16 bits of the 19th 6-byte MAC address of the station.

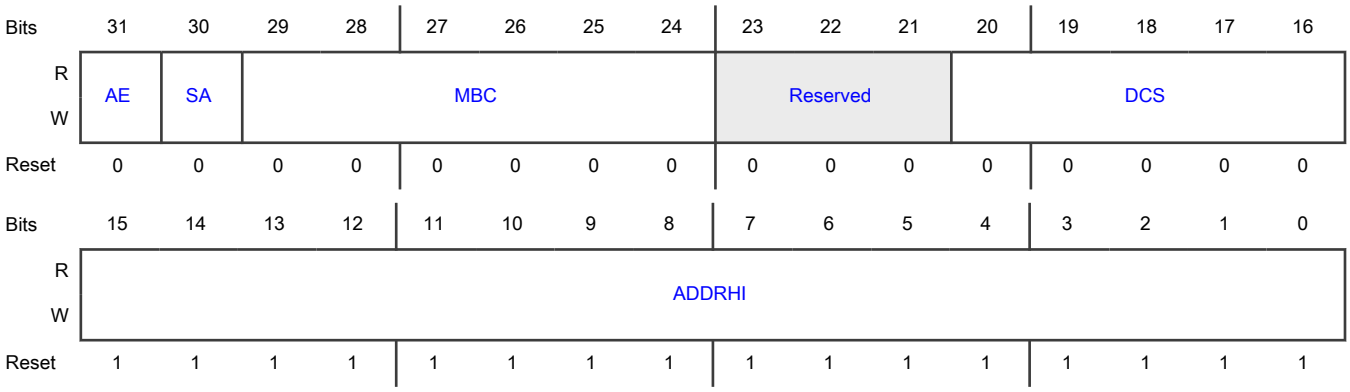
The MAC ADDRESS18 High register holds the upper 16 bits of the 19th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS18 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS18_HIGH	390h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 19th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 19th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS18[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS18[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS18[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS18 registers. Each bit controls the masking of the bytes as follows:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- Bit 29: ADDRESS18_High[15:8] - Bit 28: ADDRESS18_High[7:0] - Bit 27: ADDRESS18_Low[31:24] - .. - Bit 24: ADDRESS18_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS18 [47:32] This field contains the upper 16 bits[47:32] of the 19th 6-byte MAC address.

43.8.1.85 MAC Address18 Low (MAC_ADDRESS18_LOW)

The MAC ADDRESS18 Low register holds the lower 32 bits of the 19th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS18_LOW	394h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS18 [31:0] This field contains the lower 32 bits of 19th 6-byte MAC address. MAC ADDRESS18 [31:0] This field contains the lower 32 bits of 19th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.86 MAC Address19 High (MAC_ADDRESS19_HIGH)

The MAC ADDRESS19 High register holds the upper 16 bits of the 20th 6-byte MAC address of the station.

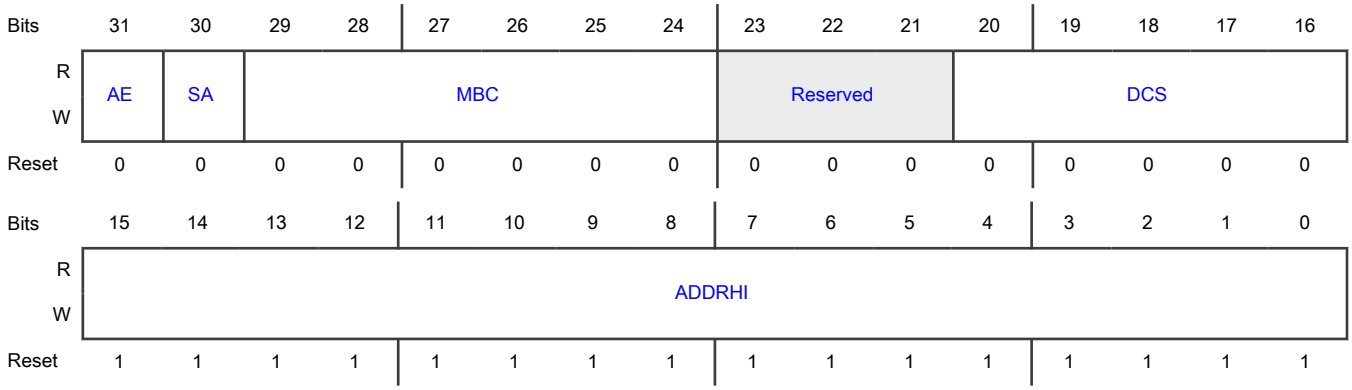
The MAC ADDRESS19 High register holds the upper 16 bits of the 20th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS19 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS19_HIGH	398h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 20th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 20th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS19[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS19[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS19[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS19 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS19_High[15:8] - Bit 28: ADDRESS19_High[7:0] - Bit 27: ADDRESS19_Low[31:24] - .. - Bit 24: ADDRESS19_Low[7:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS19 [47:32] This field contains the upper 16 bits[47:32] of the 20th 6-byte MAC address.

43.8.1.87 MAC Address19 Low (MAC_ADDRESS19_LOW)

The MAC ADDRESS19 Low register holds the lower 32 bits of the 20th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS19_LOW	39Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS19 [31:0] This field contains the lower 32 bits of 20th 6-byte MAC address. MAC ADDRESS19 [31:0] This field contains the lower 32 bits of 20th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.88 MAC Address20 High (MAC_ADDRESS20_HIGH)

The MAC ADDRESS20 High register holds the upper 16 bits of the 21st 6-byte MAC address of the station.

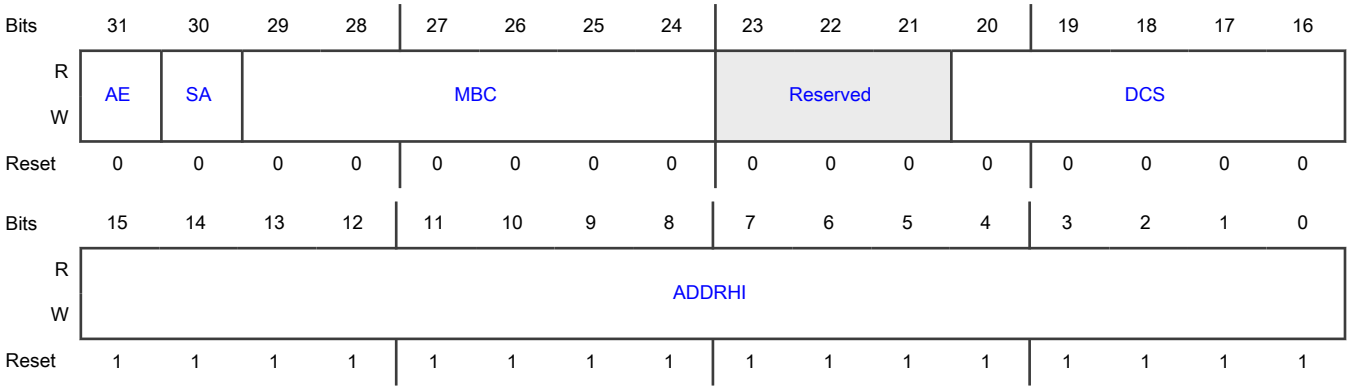
The MAC ADDRESS20 High register holds the upper 16 bits of the 21st 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS20 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS20_HIGH	3A0h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 21st MAC address for perfect filtering. Address Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the address filter module uses the 21st MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS20[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS20[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS20[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS20 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS20_High[15:8] - Bit 28: ADDRESS20_High[7:0] - Bit 27: ADDRESS20_Low[31:24] - .. - Bit 24: ADDRESS20_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-0 ADDRHI	MAC ADDRESS20 [47:32] This field contains the upper 16 bits[47:32] of the 21st 6-byte MAC address.

43.8.1.89 MAC Address20 Low (MAC_ADDRESS20_LOW)

The MAC ADDRESS20 Low register holds the lower 32 bits of the 21st 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS20_LOW	3A4h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS20 [31:0] This field contains the lower 32 bits of 21st 6-byte MAC address. MAC ADDRESS20 [31:0] This field contains the lower 32 bits of 21st 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.90 MAC Address21 High (MAC_ADDRESS21_HIGH)

The MAC ADDRESS21 High register holds the upper 16 bits of the 22nd 6-byte MAC address of the station.

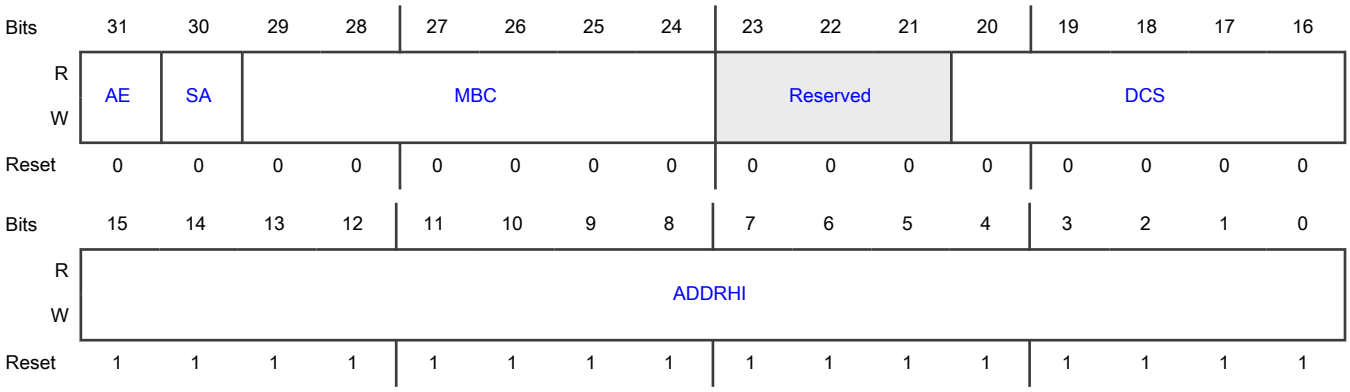
The MAC ADDRESS21 High register holds the upper 16 bits of the 22nd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS21 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS21_HIGH	3A8h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 22nd MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 22nd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS21[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS21[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS21[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS21 registers. Each bit controls the masking of the bytes as follows:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- Bit 29: ADDRESS21_High[15:8] - Bit 28: ADDRESS21_High[7:0] - Bit 27: ADDRESS21_Low[31:24] - .. - Bit 24: ADDRESS21_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS21 [47:32] This field contains the upper 16 bits[47:32] of the 22nd 6-byte MAC address.

43.8.1.91 MAC Address21 Low (MAC_ADDRESS21_LOW)

The MAC ADDRESS21 Low register holds the lower 32 bits of the 22nd 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS21_LOW	3ACh

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS21 [31:0] This field contains the lower 32 bits of 22nd 6-byte MAC address. MAC ADDRESS21 [31:0] This field contains the lower 32 bits of 22nd 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.92 MAC Address22 High (MAC_ADDRESS22_HIGH)

The MAC ADDRESS22 High register holds the upper 16 bits of the 23rd 6-byte MAC address of the station.

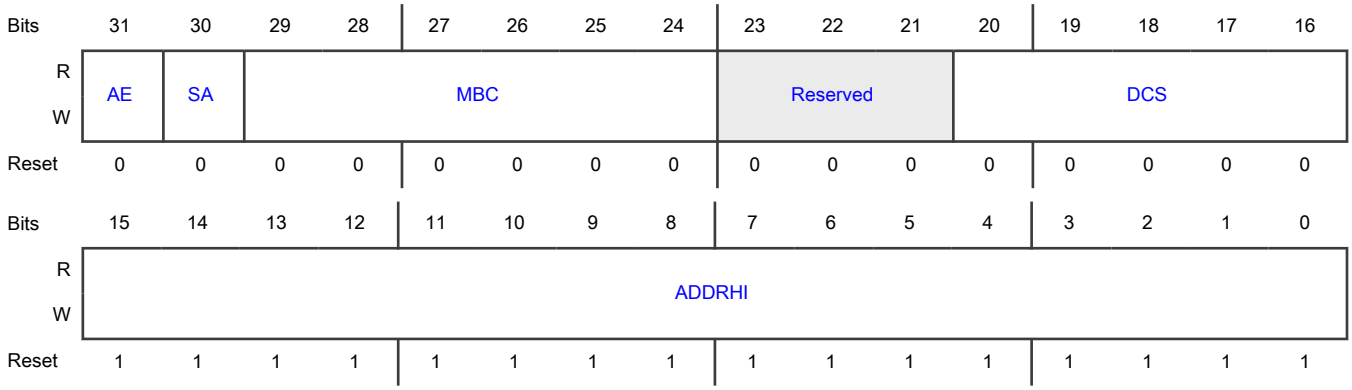
The MAC ADDRESS22 High register holds the upper 16 bits of the 23rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS22 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS22_HIGH	3B0h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 23rd MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 23rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS22[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS22[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS22[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS22 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS22_High[15:8] - Bit 28: ADDRESS22_High[7:0] - Bit 27: ADDRESS22_Low[31:24] - .. - Bit 24: ADDRESS22_Low[7:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS22 [47:32] This field contains the upper 16 bits[47:32] of the 23rd 6-byte MAC address.

43.8.1.93 MAC Address22 Low (MAC_ADDRESS22_LOW)

The MAC ADDRESS22 Low register holds the lower 32 bits of the 23rd 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS22_LOW	3B4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS22 [31:0] This field contains the lower 32 bits of 23rd 6-byte MAC address. MAC ADDRESS22 [31:0] This field contains the lower 32 bits of 23rd 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.94 MAC Address23 High (MAC_ADDRESS23_HIGH)

The MAC ADDRESS23 High register holds the upper 16 bits of the 24th 6-byte MAC address of the station.

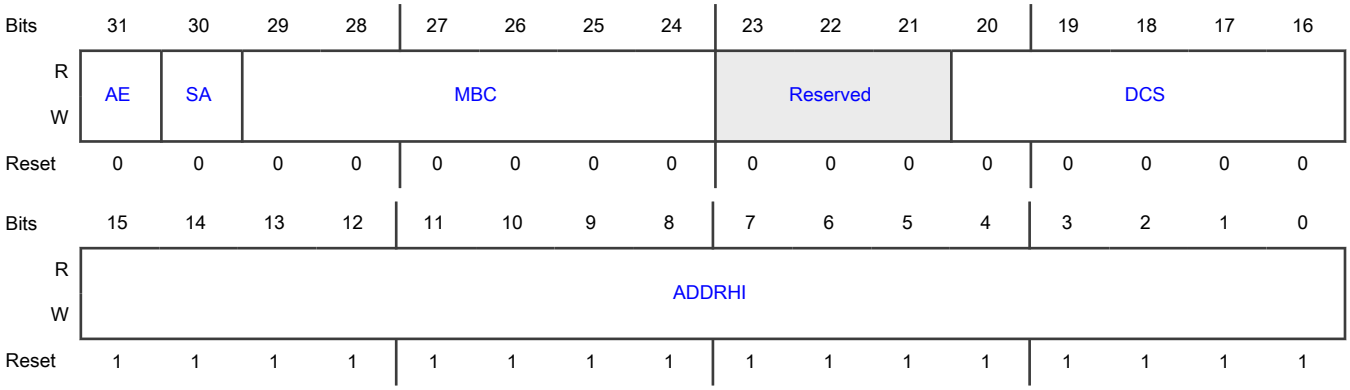
The MAC ADDRESS23 High register holds the upper 16 bits of the 24th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS23 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS23_HIGH	3B8h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 24th MAC address for perfect filtering. Address Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the address filter module uses the 24th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS23[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS23[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS23[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS23 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS23_High[15:8] - Bit 28: ADDRESS23_High[7:0] - Bit 27: ADDRESS23_Low[31:24] - .. - Bit 24: ADDRESS23_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-0 ADDRHI	MAC ADDRESS23 [47:32] This field contains the upper 16 bits[47:32] of the 24th 6-byte MAC address.

43.8.1.95 MAC Address23 Low (MAC_ADDRESS23_LOW)

The MAC ADDRESS23 Low register holds the lower 32 bits of the 24th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS23_LOW	3BCh

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS23 [31:0] This field contains the lower 32 bits of 24th 6-byte MAC address. MAC ADDRESS23 [31:0] This field contains the lower 32 bits of 24th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.96 MAC Address24 High (MAC_ADDRESS24_HIGH)

The MAC ADDRESS24 High register holds the upper 16 bits of the 25th 6-byte MAC address of the station.

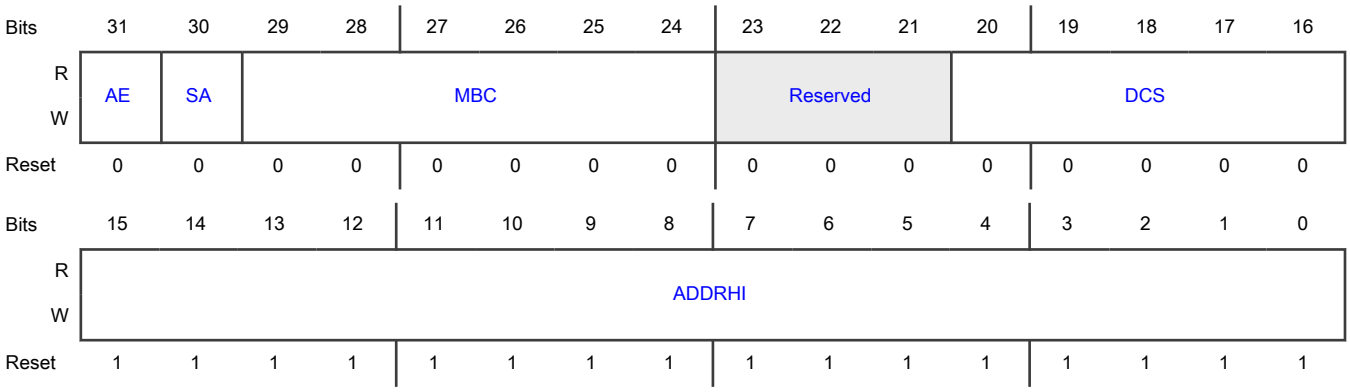
The MAC ADDRESS24 High register holds the upper 16 bits of the 25th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS24 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS24_HIGH	3C0h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 25th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 25th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS24[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS24[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS24[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS24 registers. Each bit controls the masking of the bytes as follows:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- Bit 29: ADDRESS24_High[15:8] - Bit 28: ADDRESS24_High[7:0] - Bit 27: ADDRESS24_Low[31:24] - .. - Bit 24: ADDRESS24_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS24 [47:32] This field contains the upper 16 bits[47:32] of the 25th 6-byte MAC address.

43.8.1.97 MAC Address24 Low (MAC_ADDRESS24_LOW)

The MAC ADDRESS24 Low register holds the lower 32 bits of the 25th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS24_LOW	3C4h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS24 [31:0] This field contains the lower 32 bits of 25th 6-byte MAC address. MAC ADDRESS24 [31:0] This field contains the lower 32 bits of 25th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.98 MAC Address25 High (MAC_ADDRESS25_HIGH)

The MAC ADDRESS25 High register holds the upper 16 bits of the 26th 6-byte MAC address of the station.

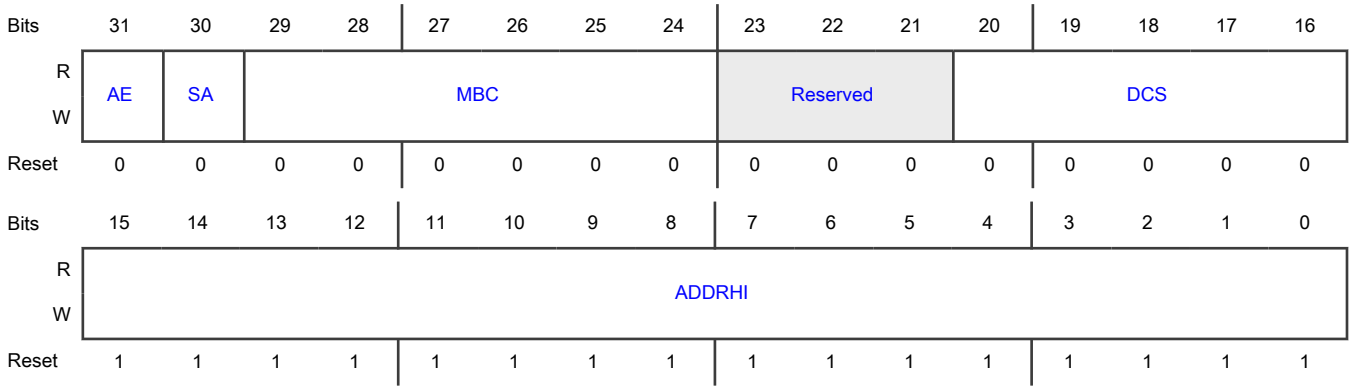
The MAC ADDRESS25 High register holds the upper 16 bits of the 26th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS25 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS25_HIGH	3C8h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 26th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 26th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS25[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS25[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS25[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS25 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS25_High[15:8] - Bit 28: ADDRESS25_High[7:0] - Bit 27: ADDRESS25_Low[31:24] - .. - Bit 24: ADDRESS25_Low[7:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS25 [47:32] This field contains the upper 16 bits[47:32] of the 26th 6-byte MAC address.

43.8.1.99 MAC Address25 Low (MAC_ADDRESS25_LOW)

The MAC ADDRESS25 Low register holds the lower 32 bits of the 26th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS25_LOW	3CCh

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS25 [31:0] This field contains the lower 32 bits of 26th 6-byte MAC address. MAC ADDRESS25 [31:0] This field contains the lower 32 bits of 26th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.100 MAC Address26 High (MAC_ADDRESS26_HIGH)

The MAC ADDRESS26 High register holds the upper 16 bits of the 27th 6-byte MAC address of the station.

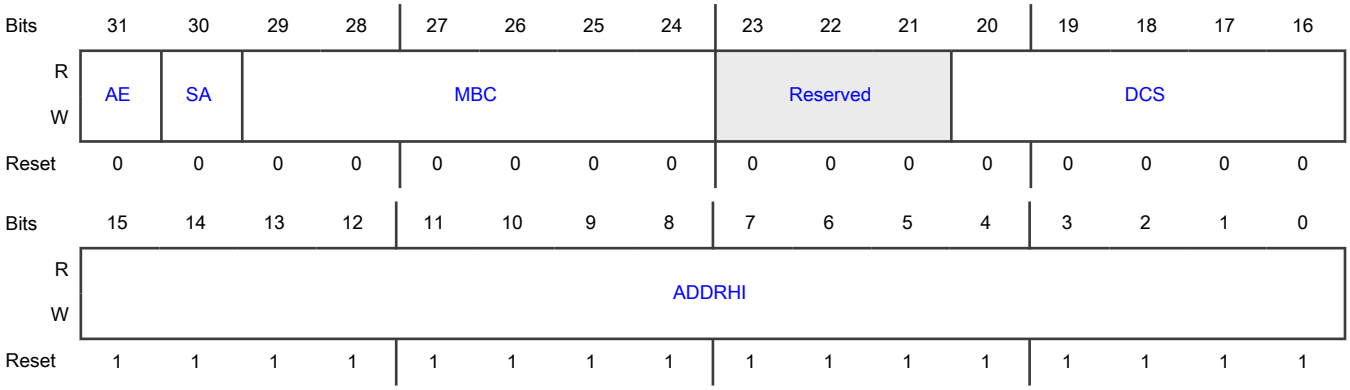
The MAC ADDRESS26 High register holds the upper 16 bits of the 27th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS26 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS26_HIGH	3D0h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 27th MAC address for perfect filtering. Address Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the address filter module uses the 27th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS26[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS26[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS26[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS26 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS26_High[15:8] - Bit 28: ADDRESS26_High[7:0] - Bit 27: ADDRESS26_Low[31:24] - .. - Bit 24: ADDRESS26_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-0 ADDRHI	MAC ADDRESS26 [47:32] This field contains the upper 16 bits[47:32] of the 27th 6-byte MAC address.

43.8.1.101 MAC Address26 Low (MAC_ADDRESS26_LOW)

The MAC ADDRESS26 Low register holds the lower 32 bits of the 27th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS26_LOW	3D4h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS26 [31:0] This field contains the lower 32 bits of 27th 6-byte MAC address. MAC ADDRESS26 [31:0] This field contains the lower 32 bits of 27th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.102 MAC Address27 High (MAC_ADDRESS27_HIGH)

The MAC ADDRESS27 High register holds the upper 16 bits of the 28th 6-byte MAC address of the station.

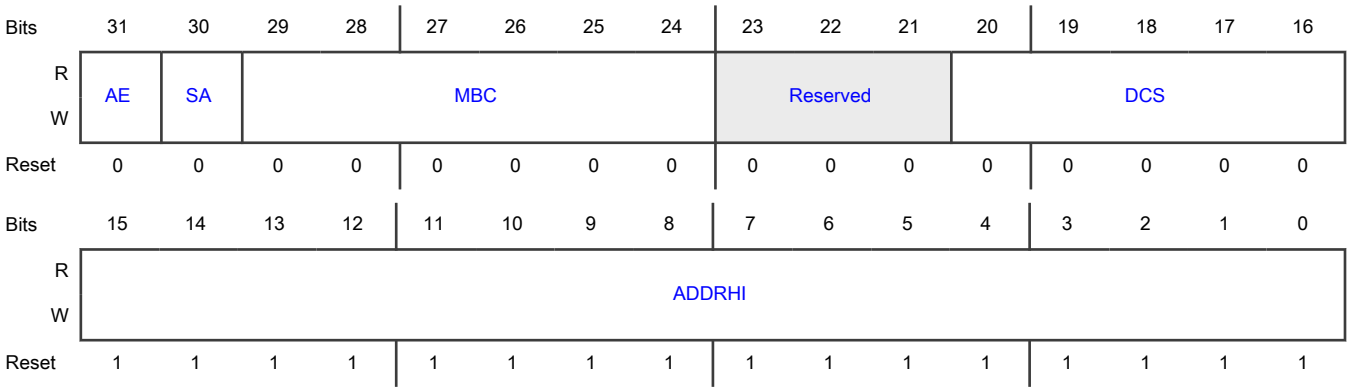
The MAC ADDRESS27 High register holds the upper 16 bits of the 28th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS27 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS27_HIGH	3D8h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 28th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 28th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS27[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS27[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS27[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS27 registers. Each bit controls the masking of the bytes as follows:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- Bit 29: ADDRESS27_High[15:8] - Bit 28: ADDRESS27_High[7:0] - Bit 27: ADDRESS27_Low[31:24] - .. - Bit 24: ADDRESS27_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS27 [47:32] This field contains the upper 16 bits[47:32] of the 28th 6-byte MAC address.

43.8.1.103 MAC Address27 Low (MAC_ADDRESS27_LOW)

The MAC ADDRESS27 Low register holds the lower 32 bits of the 28th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS27_LOW	3DCh

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS27 [31:0] This field contains the lower 32 bits of 28th 6-byte MAC address. MAC ADDRESS27 [31:0] This field contains the lower 32 bits of 28th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.104 MAC Address28 High (MAC_ADDRESS28_HIGH)

The MAC ADDRESS28 High register holds the upper 16 bits of the 29th 6-byte MAC address of the station.

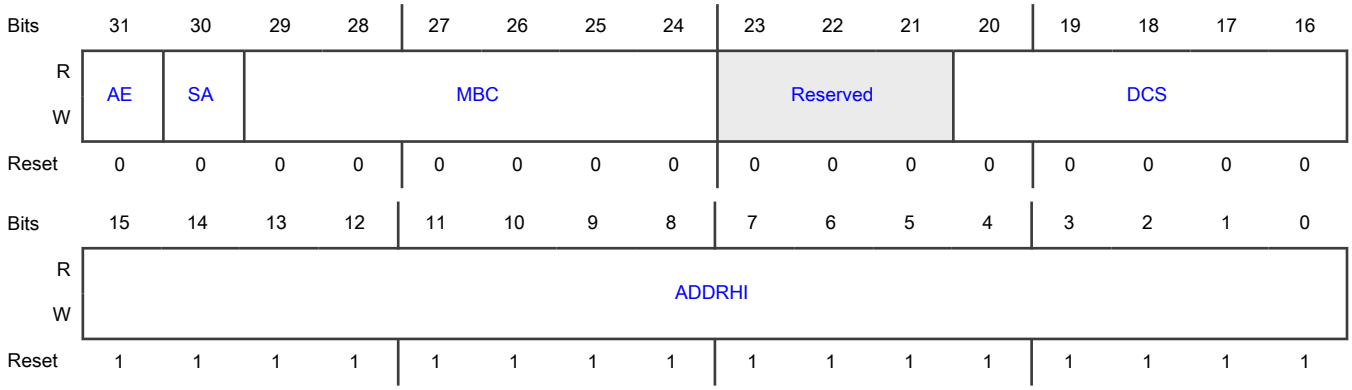
The MAC ADDRESS28 High register holds the upper 16 bits of the 29th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS28 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS28_HIGH	3E0h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 29th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 29th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS28[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS28[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS28[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS28 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS28_High[15:8] - Bit 28: ADDRESS28_High[7:0] - Bit 27: ADDRESS28_Low[31:24] - .. - Bit 24: ADDRESS28_Low[7:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS28 [47:32] This field contains the upper 16 bits[47:32] of the 29th 6-byte MAC address.

43.8.1.105 MAC Address28 Low (MAC_ADDRESS28_LOW)

The MAC ADDRESS28 Low register holds the lower 32 bits of the 29th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS28_LOW	3E4h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS28 [31:0] This field contains the lower 32 bits of 29th 6-byte MAC address. MAC ADDRESS28 [31:0] This field contains the lower 32 bits of 29th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.106 MAC Address29 High (MAC_ADDRESS29_HIGH)

The MAC ADDRESS29 High register holds the upper 16 bits of the 30th 6-byte MAC address of the station.

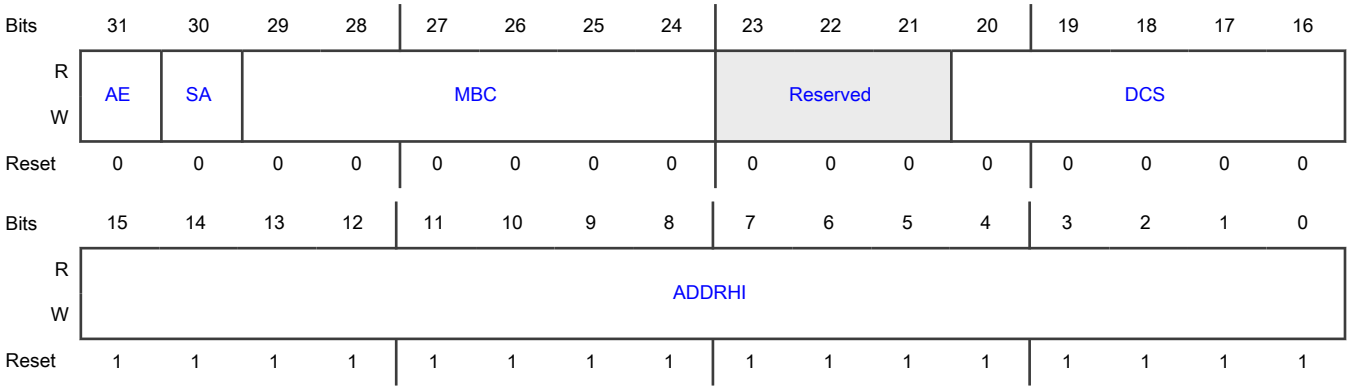
The MAC ADDRESS29 High register holds the upper 16 bits of the 30th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS29 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS29_HIGH	3E8h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 30th MAC address for perfect filtering. Address Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the address filter module uses the 30th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS29[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS29[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS29[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS29 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS29_High[15:8] - Bit 28: ADDRESS29_High[7:0] - Bit 27: ADDRESS29_Low[31:24] - .. - Bit 24: ADDRESS29_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-0 ADDRHI	MAC ADDRESS29 [47:32] This field contains the upper 16 bits[47:32] of the 30th 6-byte MAC address.

43.8.1.107 MAC Address29 Low (MAC_ADDRESS29_LOW)

The MAC ADDRESS29 Low register holds the lower 32 bits of the 30th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS29_LOW	3ECh

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS29 [31:0] This field contains the lower 32 bits of 30th 6-byte MAC address. MAC ADDRESS29 [31:0] This field contains the lower 32 bits of 30th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.108 MAC Address30 High (MAC_ADDRESS30_HIGH)

The MAC ADDRESS30 High register holds the upper 16 bits of the 31st 6-byte MAC address of the station.

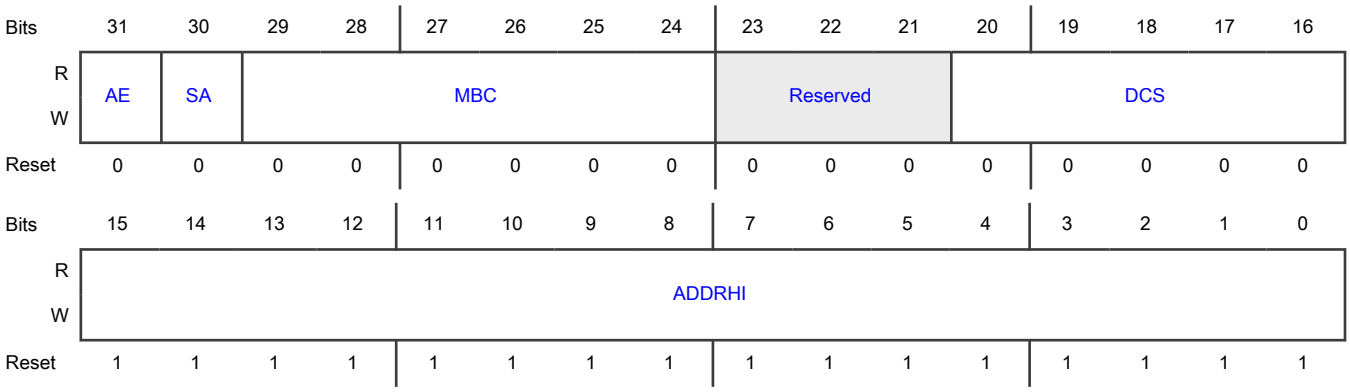
The MAC ADDRESS30 High register holds the upper 16 bits of the 31st 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS30 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS30_HIGH	3F0h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 31st MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 31st MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS30[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS30[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS30[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS30 registers. Each bit controls the masking of the bytes as follows:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- Bit 29: ADDRESS30_High[15:8] - Bit 28: ADDRESS30_High[7:0] - Bit 27: ADDRESS30_Low[31:24] - .. - Bit 24: ADDRESS30_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS30 [47:32] This field contains the upper 16 bits[47:32] of the 31st 6-byte MAC address.

43.8.1.109 MAC Address30 Low (MAC_ADDRESS30_LOW)

The MAC ADDRESS30 Low register holds the lower 32 bits of the 31st 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS30_LOW	3F4h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS30 [31:0] This field contains the lower 32 bits of 31st 6-byte MAC address. MAC ADDRESS30 [31:0] This field contains the lower 32 bits of 31st 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.110 MAC Address31 High (MAC_ADDRESS31_HIGH)

The MAC ADDRESS31 High register holds the upper 16 bits of the 32nd 6-byte MAC address of the station.

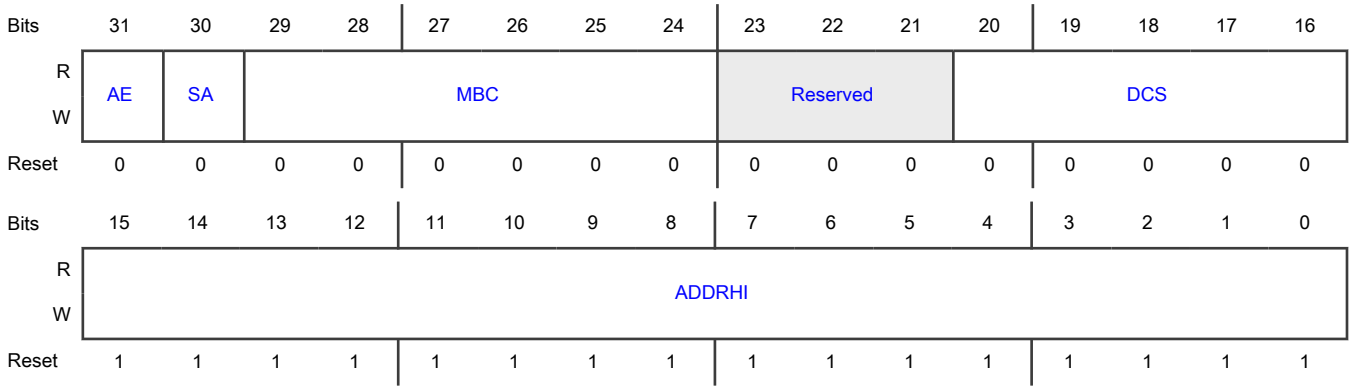
The MAC ADDRESS31 High register holds the upper 16 bits of the 32nd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS31 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Offset

Register	Offset
MAC_ADDRESS31_HIGH	3F8h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the 32nd MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 32nd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS31[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS31[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS31[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS31 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS31_High[15:8] - Bit 28: ADDRESS31_High[7:0] - Bit 27: ADDRESS31_Low[31:24] - .. - Bit 24: ADDRESS31_Low[7:0]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS31 [47:32] This field contains the upper 16 bits[47:32] of the 32nd 6-byte MAC address.

43.8.1.111 MAC Address31 Low (MAC_ADDRESS31_LOW)

The MAC ADDRESS31 Low register holds the lower 32 bits of the 32nd 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS31_LOW	3FCh

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS31 [31:0] This field contains the lower 32 bits of 32nd 6-byte MAC address. MAC ADDRESS31 [31:0] This field contains the lower 32 bits of 32nd 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

43.8.1.112 MAC Address32 High (MAC_ADDRESS32_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

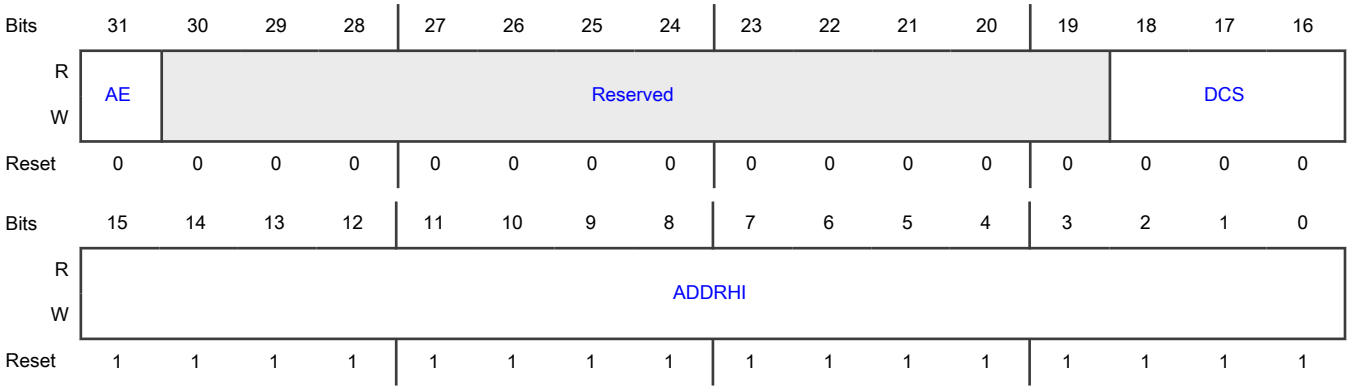
The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS32_HIGH	400h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

43.8.1.113 MAC Address32 Low (MAC_ADDRESS32_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS32_LOW	404h

Diagram

Fields

Field	Description
31-0	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address.

Table continues on the next page...

Field	Description
ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.114 MAC Address33 High (MAC_ADDRESS33_HIGH)

The MAC ADDRESS33 High register holds the upper 16 bits of the 34th 6-byte MAC address of the station.

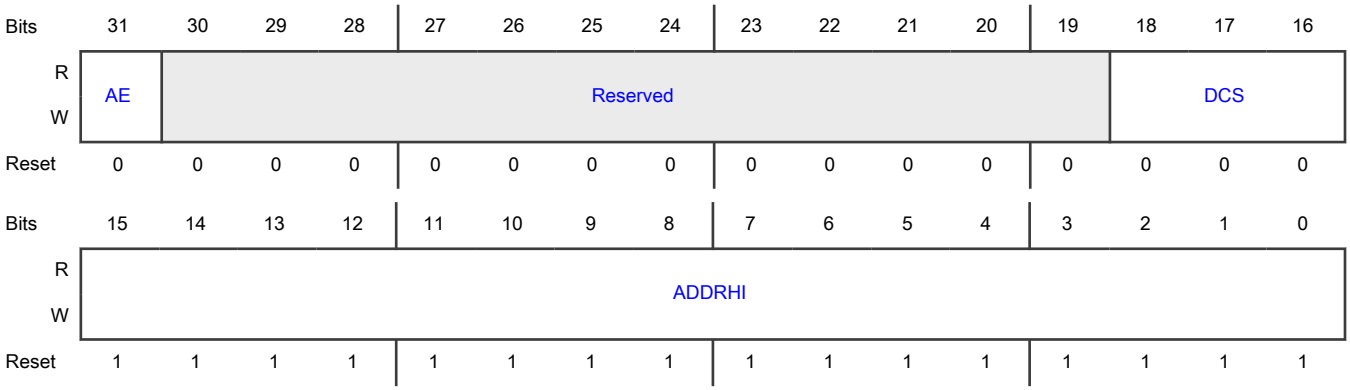
The MAC ADDRESS33 High register holds the upper 16 bits of the 34th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS33_HIGH	408h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 34th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 34th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS33 content is routed.
15-0 ADDRHI	MAC ADDRESS33 [47:32] This field contains the upper 16 bits (47:32) of the 34th 6-byte MAC address.

43.8.1.115 MAC Address33 Low (MAC_ADDRESS33_LOW)

The MAC ADDRESS33 Low register holds the lower 16 bits of the 34th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS33_LOW	40Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS33 [31:0] This field contains the lower 32 bits of the 34th 6-byte MAC address. MAC ADDRESS33 [31:0] This field contains the lower 32 bits of the 34th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.116 MAC Address34 High (MAC_ADDRESS34_HIGH)

The MAC ADDRESS34 High register holds the upper 16 bits of the 35th 6-byte MAC address of the station.

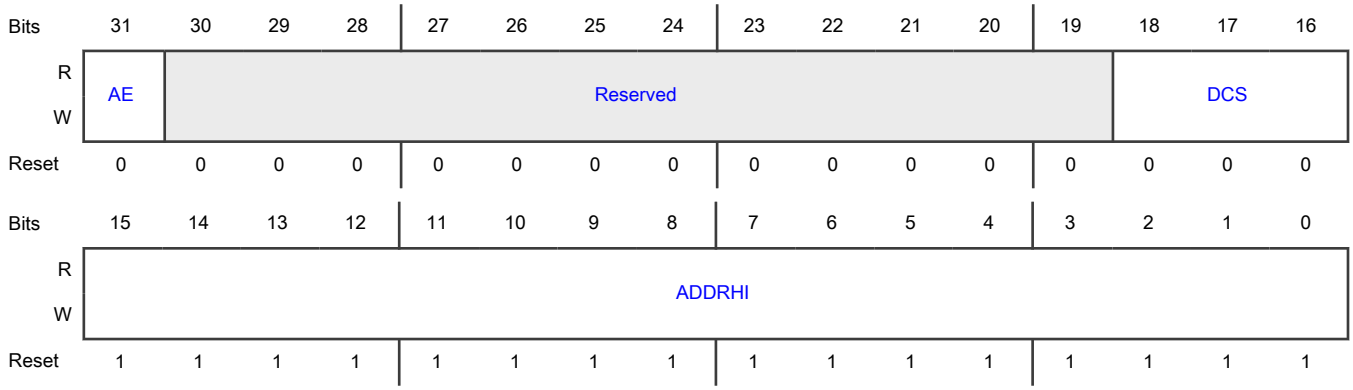
The MAC ADDRESS34 High register holds the upper 16 bits of the 35th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS34_HIGH	410h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 35th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 35th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS34 content is routed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-0 ADDRHI	MAC ADDRESS34 [47:32] This field contains the upper 16 bits (47:32) of the 35th 6-byte MAC address.

43.8.1.117 MAC Address34 Low (MAC_ADDRESS34_LOW)

The MAC ADDRESS34 Low register holds the lower 16 bits of the 35th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS34_LOW	414h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS34 [31:0] This field contains the lower 32 bits of the 35th 6-byte MAC address. MAC ADDRESS34 [31:0] This field contains the lower 32 bits of the 35th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.118 MAC Address35 High (MAC_ADDRESS35_HIGH)

The MAC ADDRESS35 High register holds the upper 16 bits of the 36th 6-byte MAC address of the station.

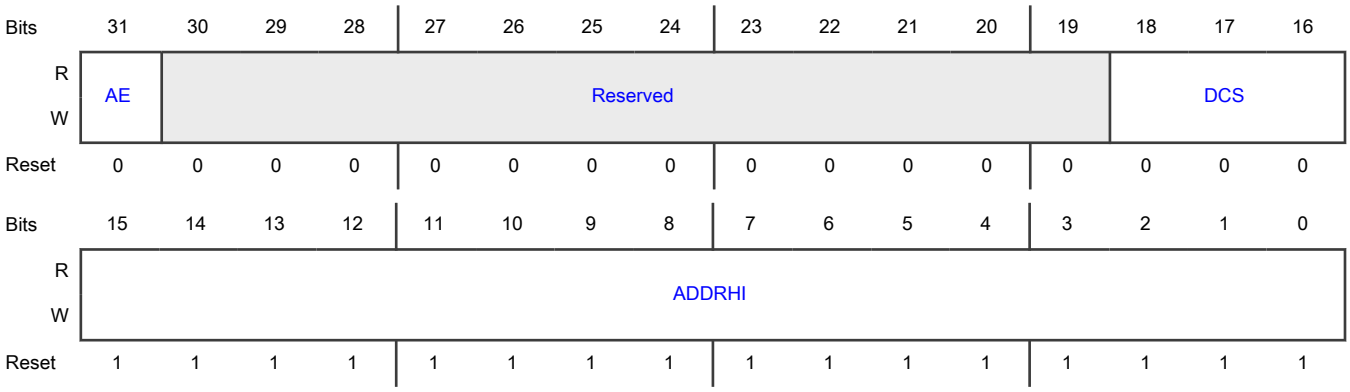
The MAC ADDRESS35 High register holds the upper 16 bits of the 36th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS35_HIGH	418h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 36th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 36th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS35 content is routed.
15-0 ADDRHI	MAC ADDRESS35 [47:32] This field contains the upper 16 bits (47:32) of the 36th 6-byte MAC address.

43.8.1.119 MAC Address35 Low (MAC_ADDRESS35_LOW)

The MAC ADDRESS35 Low register holds the lower 16 bits of the 36th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS35_LOW	41Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS35 [31:0] This field contains the lower 32 bits of the 36th 6-byte MAC address. MAC ADDRESS35 [31:0] This field contains the lower 32 bits of the 36th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.120 MAC Address36 High (MAC_ADDRESS36_HIGH)

The MAC ADDRESS36 High register holds the upper 16 bits of the 37th 6-byte MAC address of the station.

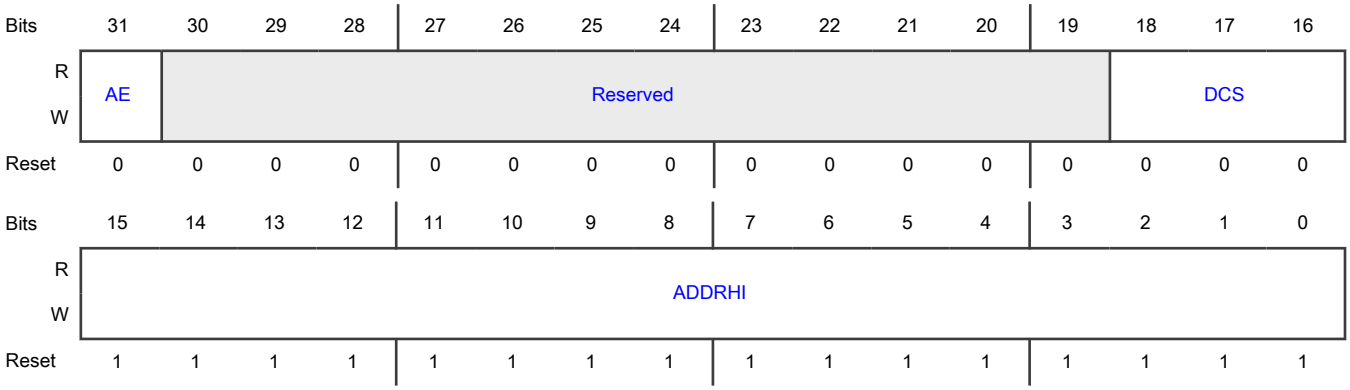
The MAC ADDRESS36 High register holds the upper 16 bits of the 37th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS36_HIGH	420h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 37th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 37th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS36 content is routed.
15-0 ADDRHI	MAC ADDRESS36 [47:32] This field contains the upper 16 bits (47:32) of the 37th 6-byte MAC address.

43.8.1.121 MAC Address36 Low (MAC_ADDRESS36_LOW)

The MAC ADDRESS36 Low register holds the lower 16 bits of the 37th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS36_LOW	424h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS36 [31:0] This field contains the lower 32 bits of the 37th 6-byte MAC address. MAC ADDRESS36 [31:0] This field contains the lower 32 bits of the 37th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.122 MAC Address37 High (MAC_ADDRESS37_HIGH)

The MAC ADDRESS37 High register holds the upper 16 bits of the 38th 6-byte MAC address of the station.

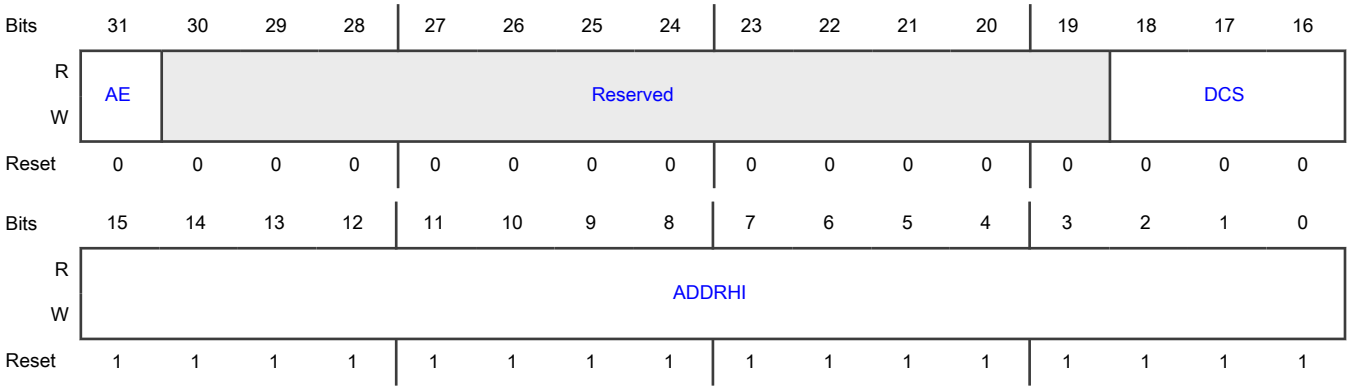
The MAC ADDRESS37 High register holds the upper 16 bits of the 38th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS37_HIGH	428h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 38th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 38th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS37 content is routed.
15-0 ADDRHI	MAC ADDRESS37 [47:32] This field contains the upper 16 bits (47:32) of the 38th 6-byte MAC address.

43.8.1.123 MAC Address37 Low (MAC_ADDRESS37_LOW)

The MAC ADDRESS37 Low register holds the lower 16 bits of the 38th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS37_LOW	42Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS37 [31:0] This field contains the lower 32 bits of the 38th 6-byte MAC address. MAC ADDRESS37 [31:0] This field contains the lower 32 bits of the 38th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.124 MAC Address38 High (MAC_ADDRESS38_HIGH)

The MAC ADDRESS38 High register holds the upper 16 bits of the 39th 6-byte MAC address of the station.

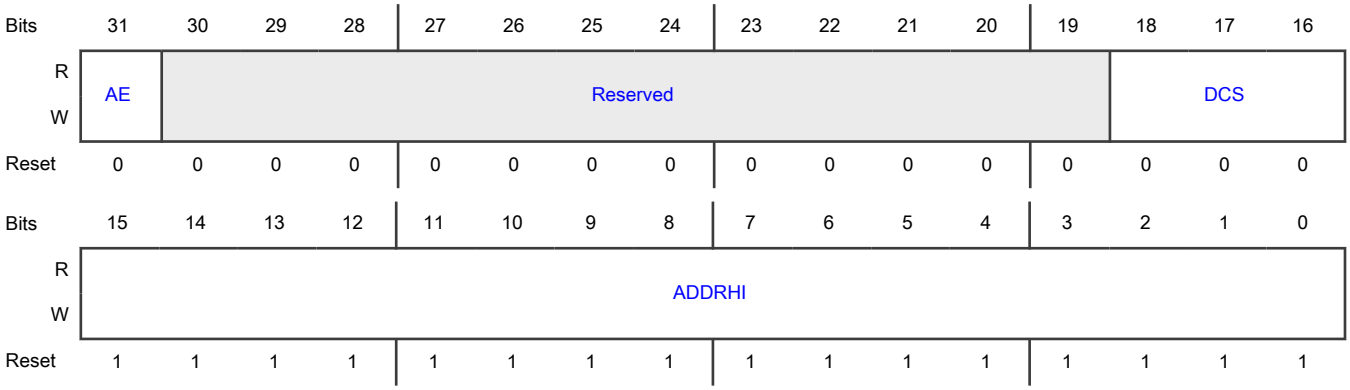
The MAC ADDRESS38 High register holds the upper 16 bits of the 39th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS38_HIGH	430h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 39th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 39th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS38 content is routed.
15-0 ADDRHI	MAC ADDRESS38 [47:32] This field contains the upper 16 bits (47:32) of the 39th 6-byte MAC address.

43.8.1.125 MAC Address38 Low (MAC_ADDRESS38_LOW)

The MAC ADDRESS38 Low register holds the lower 16 bits of the 39th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS38_LOW	434h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS38 [31:0] This field contains the lower 32 bits of the 39th 6-byte MAC address. MAC ADDRESS38 [31:0] This field contains the lower 32 bits of the 39th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.126 MAC Address39 High (MAC_ADDRESS39_HIGH)

The MAC ADDRESS39 High register holds the upper 16 bits of the 40th 6-byte MAC address of the station.

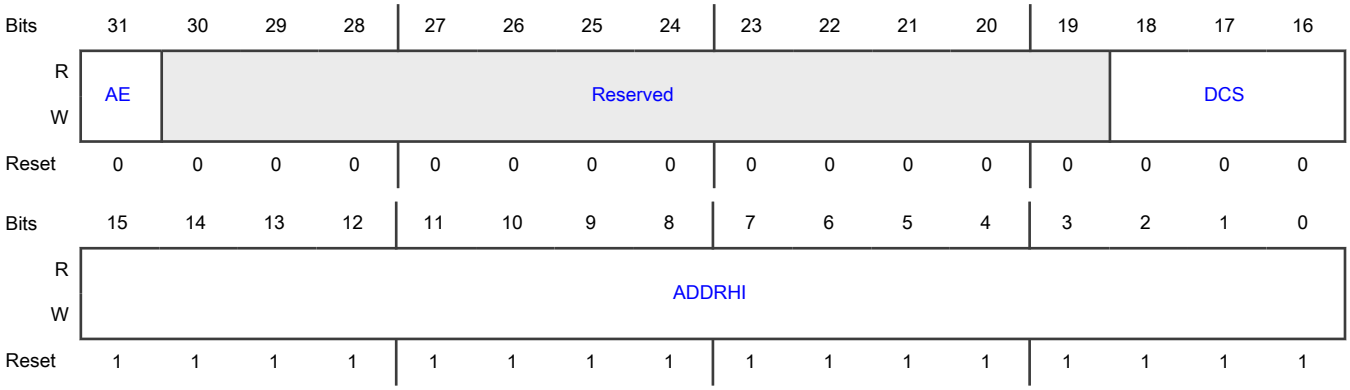
The MAC ADDRESS39 High register holds the upper 16 bits of the 40th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS39_HIGH	438h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 40th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 40th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS39 content is routed.
15-0 ADDRHI	MAC ADDRESS39 [47:32] This field contains the upper 16 bits (47:32) of the 40th 6-byte MAC address.

43.8.1.127 MAC Address39 Low (MAC_ADDRESS39_LOW)

The MAC ADDRESS39 Low register holds the lower 16 bits of the 40th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS39_LOW	43Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS39 [31:0] This field contains the lower 32 bits of the 40th 6-byte MAC address. MAC ADDRESS39 [31:0] This field contains the lower 32 bits of the 40th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.128 MAC Address40 High (MAC_ADDRESS40_HIGH)

The MAC ADDRESS40 High register holds the upper 16 bits of the 41st 6-byte MAC address of the station.

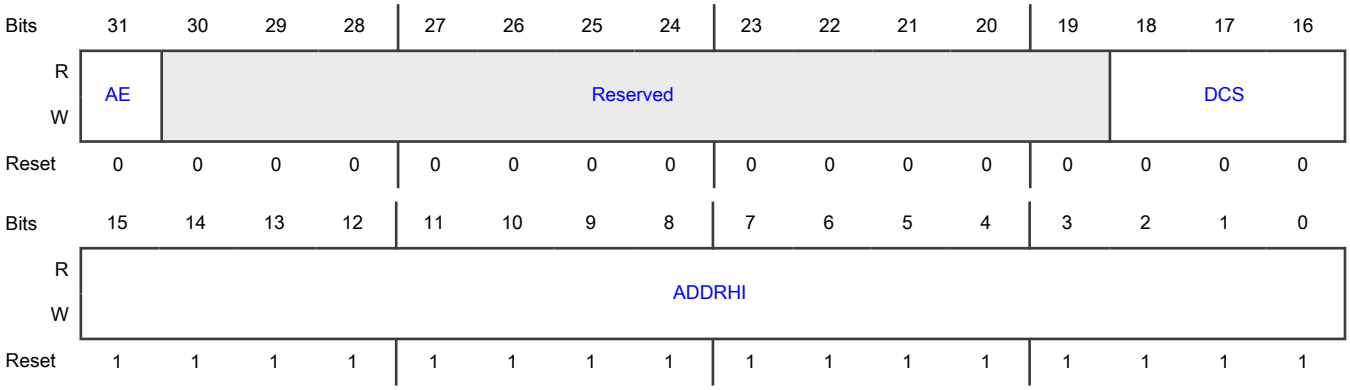
The MAC ADDRESS40 High register holds the upper 16 bits of the 41st 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS40_HIGH	440h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 41st MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 41st MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS40 content is routed.
15-0 ADDRHI	MAC ADDRESS40 [47:32] This field contains the upper 16 bits (47:32) of the 41st 6-byte MAC address.

43.8.1.129 MAC Address40 Low (MAC_ADDRESS40_LOW)

The MAC ADDRESS40 Low register holds the lower 16 bits of the 41st 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS40_LOW	444h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS40 [31:0] This field contains the lower 32 bits of the 41st 6-byte MAC address. MAC ADDRESS40 [31:0] This field contains the lower 32 bits of the 41st 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.130 MAC Address41 High (MAC_ADDRESS41_HIGH)

The MAC ADDRESS41 High register holds the upper 16 bits of the 42nd 6-byte MAC address of the station.

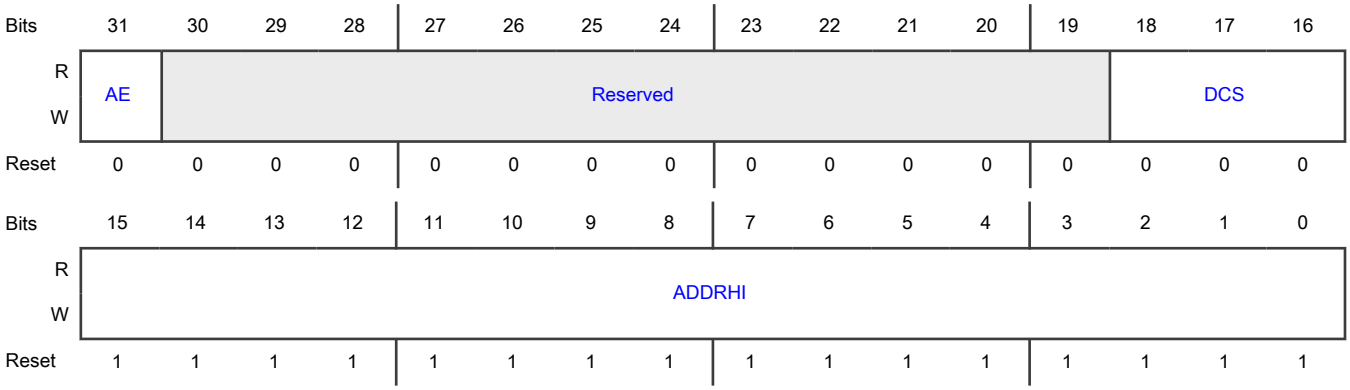
The MAC ADDRESS41 High register holds the upper 16 bits of the 42nd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS41_HIGH	448h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 42nd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 42nd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS41 content is routed.
15-0 ADDRHI	MAC ADDRESS41 [47:32] This field contains the upper 16 bits (47:32) of the 42nd 6-byte MAC address.

43.8.1.131 MAC Address41 Low (MAC_ADDRESS41_LOW)

The MAC ADDRESS41 Low register holds the lower 16 bits of the 42nd 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS41_LOW	44Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS41 [31:0] This field contains the lower 32 bits of the 42nd 6-byte MAC address. MAC ADDRESS41 [31:0] This field contains the lower 32 bits of the 42nd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.132 MAC Address42 High (MAC_ADDRESS42_HIGH)

The MAC ADDRESS42 High register holds the upper 16 bits of the 43rd 6-byte MAC address of the station.

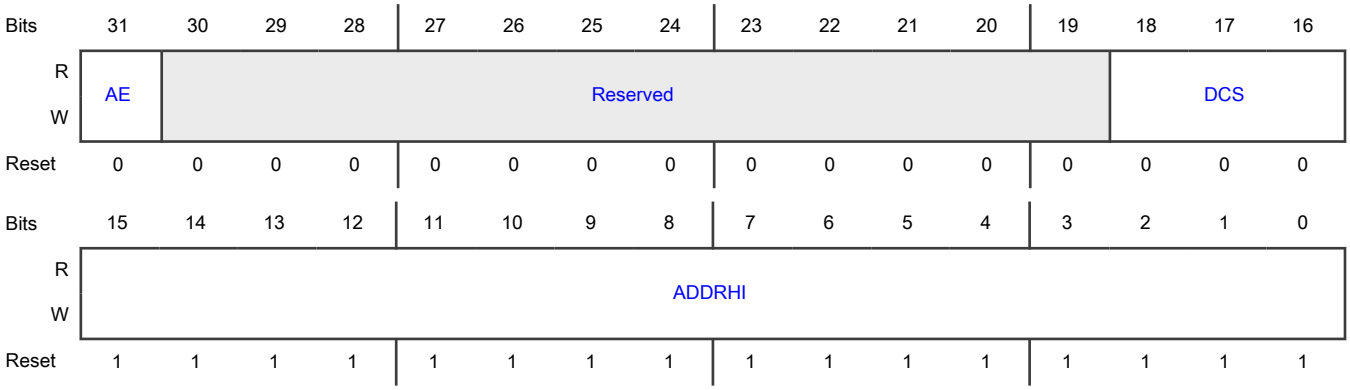
The MAC ADDRESS42 High register holds the upper 16 bits of the 43rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS42_HIGH	450h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 43rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 43rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS42 content is routed.
15-0 ADDRHI	MAC ADDRESS42 [47:32] This field contains the upper 16 bits (47:32) of the 43rd 6-byte MAC address.

43.8.1.133 MAC Address42 Low (MAC_ADDRESS42_LOW)

The MAC ADDRESS42 Low register holds the lower 16 bits of the 43rd 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS42_LOW	454h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS42 [31:0] This field contains the lower 32 bits of the 43rd 6-byte MAC address. MAC ADDRESS42 [31:0] This field contains the lower 32 bits of the 43rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.134 MAC Address43 High (MAC_ADDRESS43_HIGH)

The MAC ADDRESS43 High register holds the upper 16 bits of the 44th 6-byte MAC address of the station.

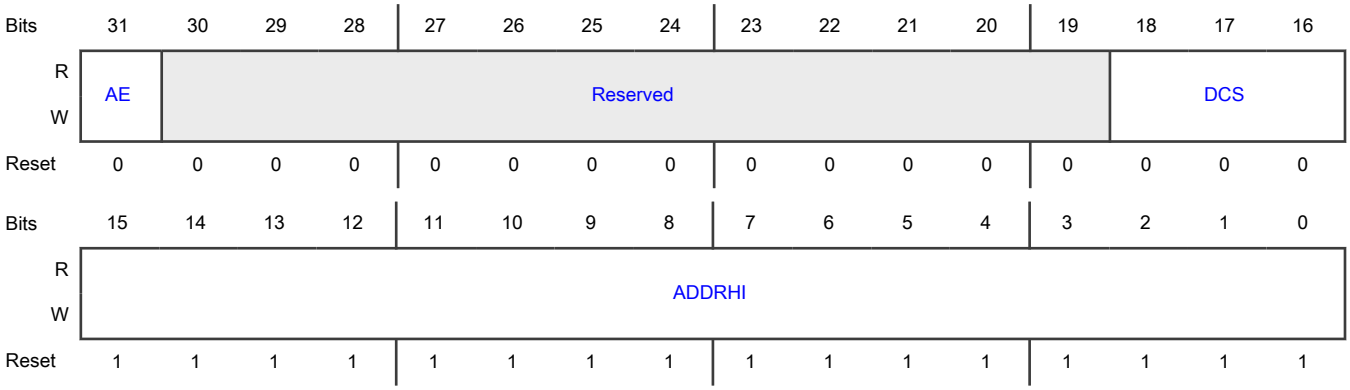
The MAC ADDRESS43 High register holds the upper 16 bits of the 44th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS43_HIGH	458h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 44th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 44th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS43 content is routed.
15-0 ADDRHI	MAC ADDRESS43 [47:32] This field contains the upper 16 bits (47:32) of the 44th 6-byte MAC address.

43.8.1.135 MAC Address43 Low (MAC_ADDRESS43_LOW)

The MAC ADDRESS43 Low register holds the lower 16 bits of the 44th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS43_LOW	45Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS43 [31:0] This field contains the lower 32 bits of the 44th 6-byte MAC address. MAC ADDRESS43 [31:0] This field contains the lower 32 bits of the 44th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.136 MAC Address44 High (MAC_ADDRESS44_HIGH)

The MAC ADDRESS44 High register holds the upper 16 bits of the 45th 6-byte MAC address of the station.

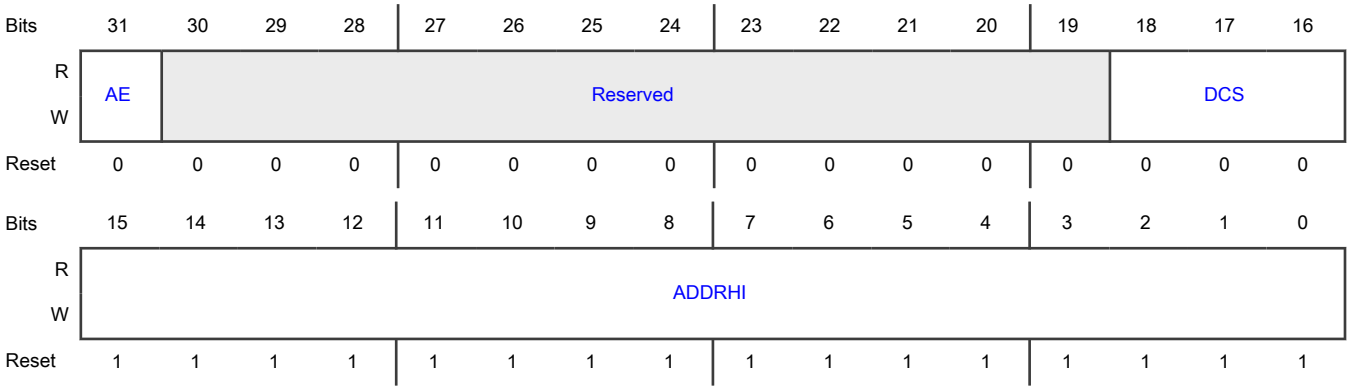
The MAC ADDRESS44 High register holds the upper 16 bits of the 45th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS44_HIGH	460h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 45th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 45th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS44 content is routed.
15-0 ADDRHI	MAC ADDRESS44 [47:32] This field contains the upper 16 bits (47:32) of the 45th 6-byte MAC address.

43.8.1.137 MAC Address44 Low (MAC_ADDRESS44_LOW)

The MAC ADDRESS44 Low register holds the lower 16 bits of the 45th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS44_LOW	464h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS44 [31:0] This field contains the lower 32 bits of the 45th 6-byte MAC address. MAC ADDRESS44 [31:0] This field contains the lower 32 bits of the 45th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.138 MAC Address45 High (MAC_ADDRESS45_HIGH)

The MAC ADDRESS45 High register holds the upper 16 bits of the 46th 6-byte MAC address of the station.

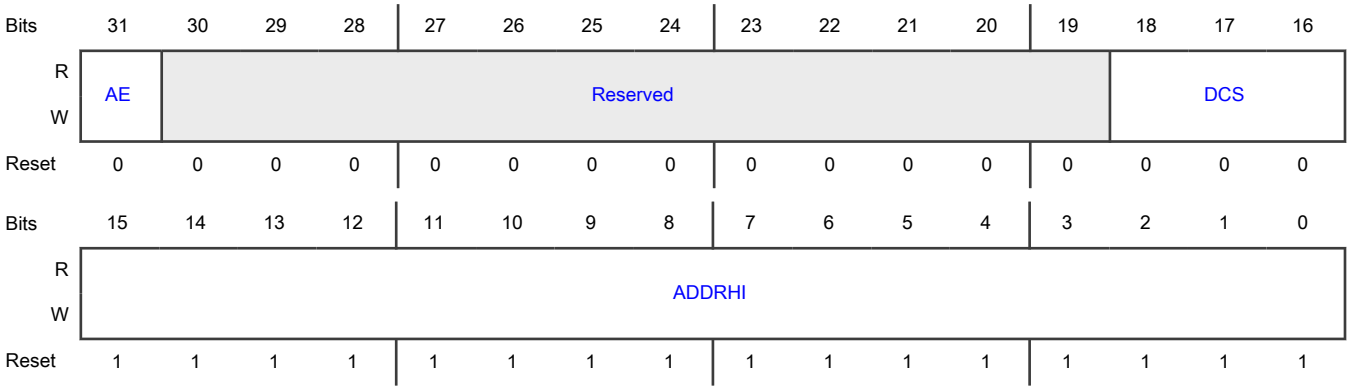
The MAC ADDRESS45 High register holds the upper 16 bits of the 46th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS45_HIGH	468h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 46th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 46th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS45 content is routed.
15-0 ADDRHI	MAC ADDRESS45 [47:32] This field contains the upper 16 bits (47:32) of the 46th 6-byte MAC address.

43.8.1.139 MAC Address45 Low (MAC_ADDRESS45_LOW)

The MAC ADDRESS45 Low register holds the lower 16 bits of the 46th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS45_LOW	46Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS45 [31:0] This field contains the lower 32 bits of the 46th 6-byte MAC address. MAC ADDRESS45 [31:0] This field contains the lower 32 bits of the 46th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.140 MAC Address46 High (MAC_ADDRESS46_HIGH)

The MAC ADDRESS46 High register holds the upper 16 bits of the 47th 6-byte MAC address of the station.

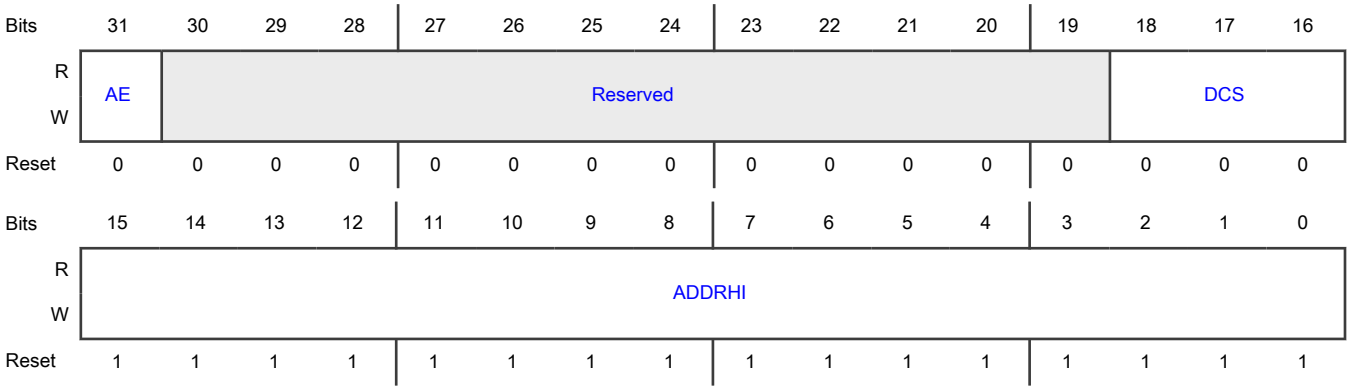
The MAC ADDRESS46 High register holds the upper 16 bits of the 47th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS46_HIGH	470h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 47th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 47th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS46 content is routed.
15-0 ADDRHI	MAC ADDRESS46 [47:32] This field contains the upper 16 bits (47:32) of the 47th 6-byte MAC address.

43.8.1.141 MAC Address46 Low (MAC_ADDRESS46_LOW)

The MAC ADDRESS46 Low register holds the lower 16 bits of the 47th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS46_LOW	474h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS46 [31:0] This field contains the lower 32 bits of the 47th 6-byte MAC address. MAC ADDRESS46 [31:0] This field contains the lower 32 bits of the 47th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.142 MAC Address47 High (MAC_ADDRESS47_HIGH)

The MAC ADDRESS47 High register holds the upper 16 bits of the 48th 6-byte MAC address of the station.

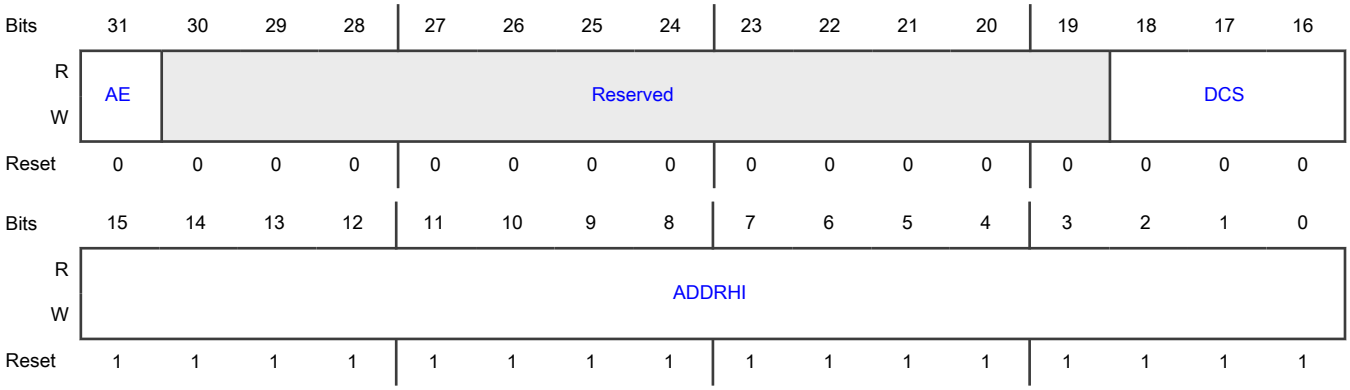
The MAC ADDRESS47 High register holds the upper 16 bits of the 48th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS47_HIGH	478h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 48th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 48th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS47 content is routed.
15-0 ADDRHI	MAC ADDRESS47 [47:32] This field contains the upper 16 bits (47:32) of the 48th 6-byte MAC address.

43.8.1.143 MAC Address47 Low (MAC_ADDRESS47_LOW)

The MAC ADDRESS47 Low register holds the lower 16 bits of the 48th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS47_LOW	47Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS47 [31:0] This field contains the lower 32 bits of the 48th 6-byte MAC address. MAC ADDRESS47 [31:0] This field contains the lower 32 bits of the 48th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.144 MAC Address48 High (MAC_ADDRESS48_HIGH)

The MAC ADDRESS48 High register holds the upper 16 bits of the 49th 6-byte MAC address of the station.

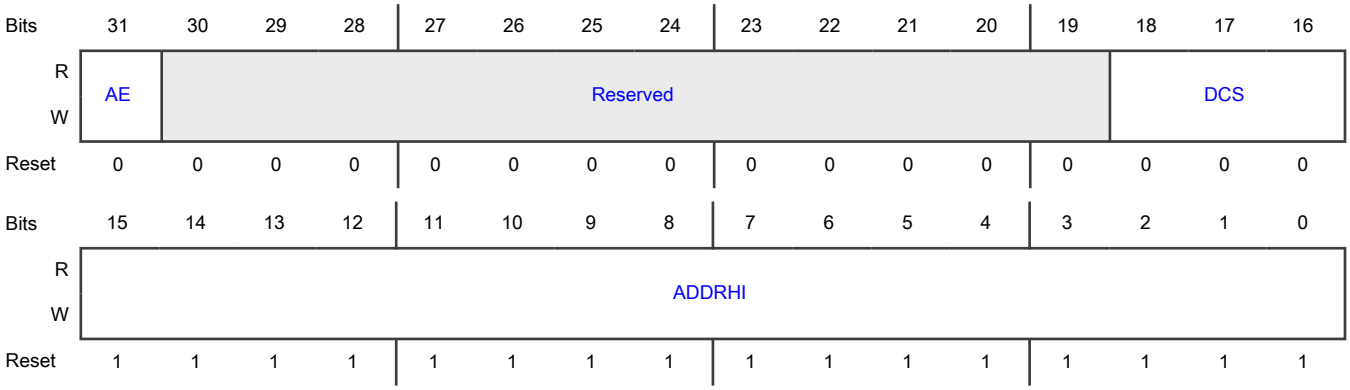
The MAC ADDRESS48 High register holds the upper 16 bits of the 49th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS48_HIGH	480h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 49th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 49th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS48 content is routed.
15-0 ADDRHI	MAC ADDRESS48 [47:32] This field contains the upper 16 bits (47:32) of the 49th 6-byte MAC address.

43.8.1.145 MAC Address48 Low (MAC_ADDRESS48_LOW)

The MAC ADDRESS48 Low register holds the lower 16 bits of the 49th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS48_LOW	484h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS48 [31:0] This field contains the lower 32 bits of the 49th 6-byte MAC address. MAC ADDRESS48 [31:0] This field contains the lower 32 bits of the 49th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.146 MAC Address49 High (MAC_ADDRESS49_HIGH)

The MAC ADDRESS49 High register holds the upper 16 bits of the 50th 6-byte MAC address of the station.

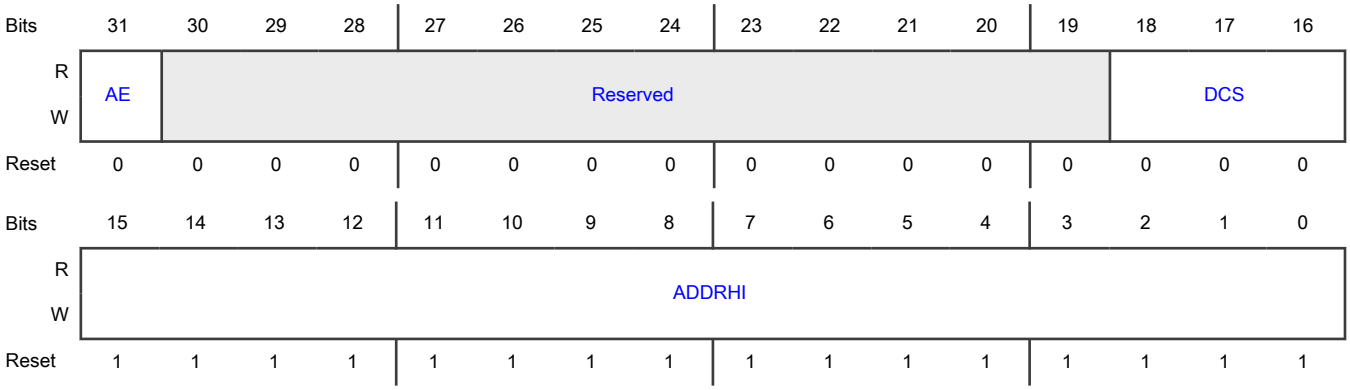
The MAC ADDRESS49 High register holds the upper 16 bits of the 50th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS49_HIGH	488h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 50th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 50th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS49 content is routed.
15-0 ADDRHI	MAC ADDRESS49 [47:32] This field contains the upper 16 bits (47:32) of the 50th 6-byte MAC address.

43.8.1.147 MAC Address49 Low (MAC_ADDRESS49_LOW)

The MAC ADDRESS49 Low register holds the lower 16 bits of the 50th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS49_LOW	48Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS49 [31:0] This field contains the lower 32 bits of the 50th 6-byte MAC address. MAC ADDRESS49 [31:0] This field contains the lower 32 bits of the 50th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.148 MAC Address50 High (MAC_ADDRESS50_HIGH)

The MAC ADDRESS50 High register holds the upper 16 bits of the 51st 6-byte MAC address of the station.

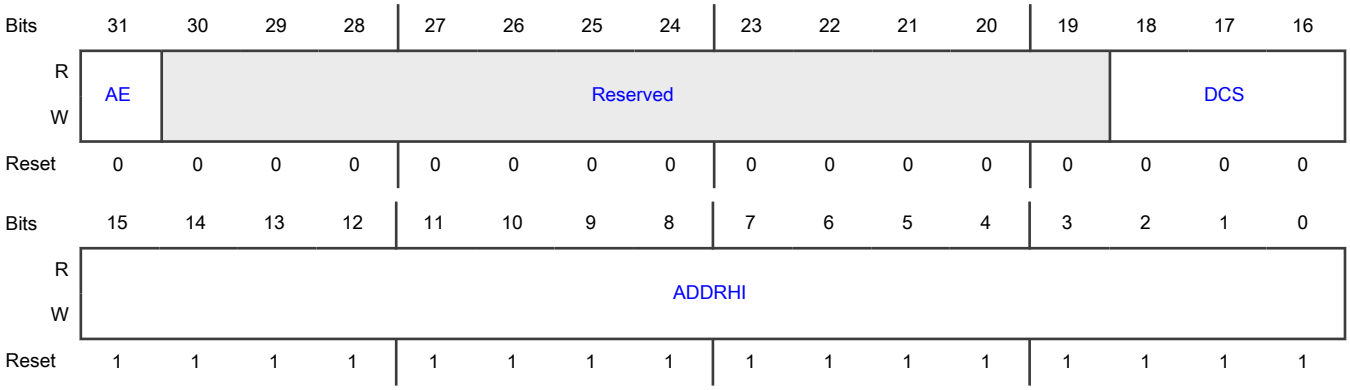
The MAC ADDRESS50 High register holds the upper 16 bits of the 51st 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS50_HIGH	490h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 51st MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 51st MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS50 content is routed.
15-0 ADDRHI	MAC ADDRESS50 [47:32] This field contains the upper 16 bits (47:32) of the 51st 6-byte MAC address.

43.8.1.149 MAC Address50 Low (MAC_ADDRESS50_LOW)

The MAC ADDRESS50 Low register holds the lower 16 bits of the 51st 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS50_LOW	494h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS50 [31:0] This field contains the lower 32 bits of the 51st 6-byte MAC address. MAC ADDRESS50 [31:0] This field contains the lower 32 bits of the 51st 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.150 MAC Address51 High (MAC_ADDRESS51_HIGH)

The MAC ADDRESS51 High register holds the upper 16 bits of the 52nd 6-byte MAC address of the station.

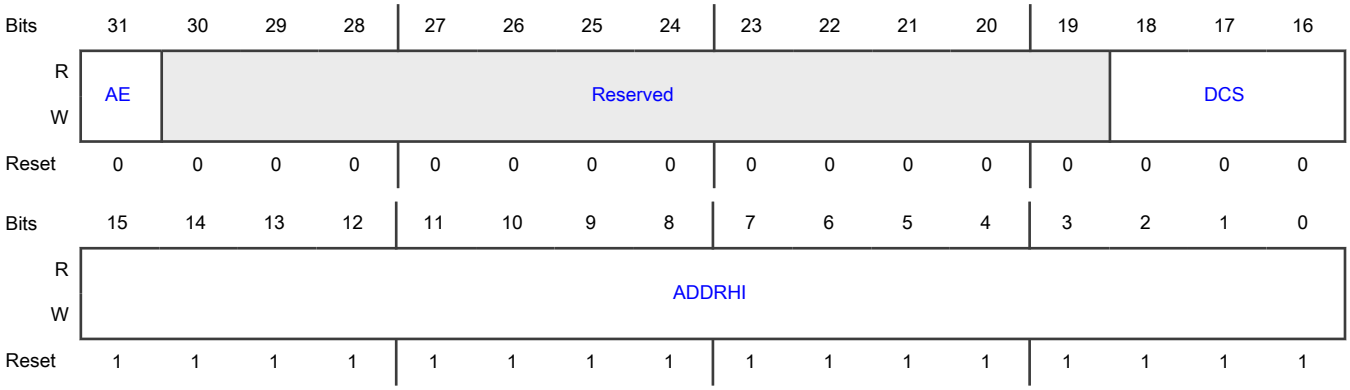
The MAC ADDRESS51 High register holds the upper 16 bits of the 52nd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS51_HIGH	498h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 52nd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 52nd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS51 content is routed.
15-0 ADDRHI	MAC ADDRESS51 [47:32] This field contains the upper 16 bits (47:32) of the 52nd 6-byte MAC address.

43.8.1.151 MAC Address51 Low (MAC_ADDRESS51_LOW)

The MAC ADDRESS51 Low register holds the lower 16 bits of the 52nd 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS51_LOW	49Ch

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS51 [31:0] This field contains the lower 32 bits of the 52nd 6-byte MAC address. MAC ADDRESS51 [31:0] This field contains the lower 32 bits of the 52nd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.152 MAC Address52 High (MAC_ADDRESS52_HIGH)

The MAC ADDRESS52 High register holds the upper 16 bits of the 53rd 6-byte MAC address of the station.

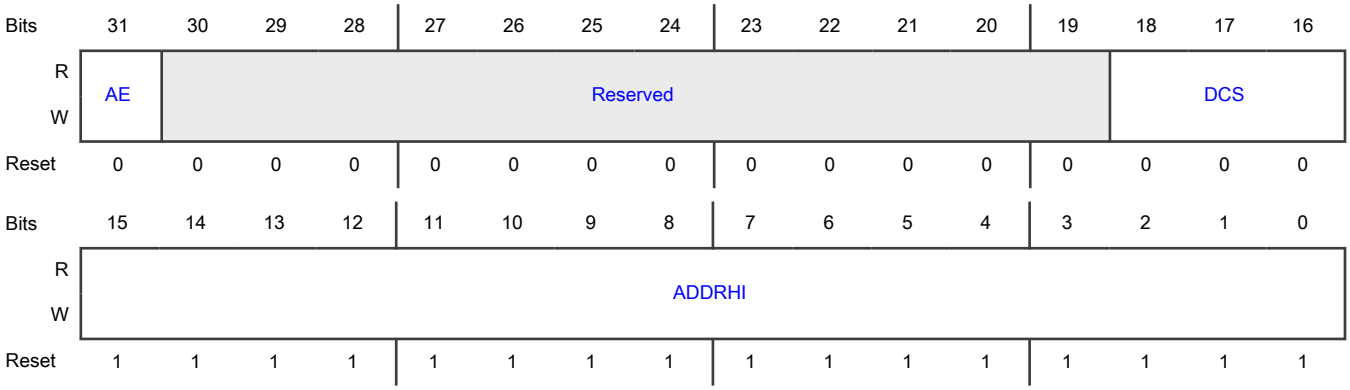
The MAC ADDRESS52 High register holds the upper 16 bits of the 53rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS52_HIGH	4A0h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 53rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 53rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS52 content is routed.
15-0 ADDRHI	MAC ADDRESS52 [47:32] This field contains the upper 16 bits (47:32) of the 53rd 6-byte MAC address.

43.8.1.153 MAC Address52 Low (MAC_ADDRESS52_LOW)

The MAC ADDRESS52 Low register holds the lower 16 bits of the 53rd 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS52_LOW	4A4h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS52 [31:0] This field contains the lower 32 bits of the 53rd 6-byte MAC address. MAC ADDRESS52 [31:0] This field contains the lower 32 bits of the 53rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.154 MAC Address53 High (MAC_ADDRESS53_HIGH)

The MAC ADDRESS53 High register holds the upper 16 bits of the 54th 6-byte MAC address of the station.

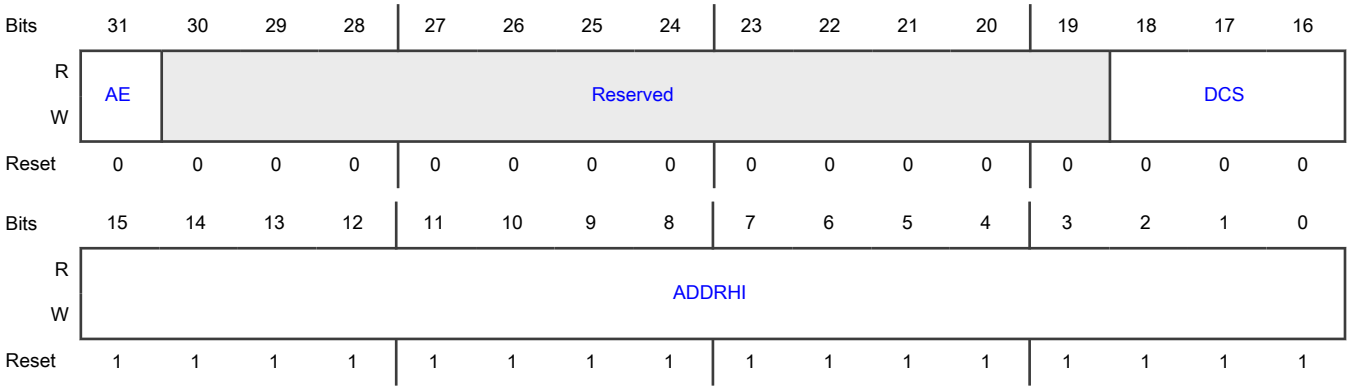
The MAC ADDRESS53 High register holds the upper 16 bits of the 54th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS53_HIGH	4A8h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 54th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 54th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS53 content is routed.
15-0 ADDRHI	MAC ADDRESS53 [47:32] This field contains the upper 16 bits (47:32) of the 54th 6-byte MAC address.

43.8.1.155 MAC Address53 Low (MAC_ADDRESS53_LOW)

The MAC ADDRESS53 Low register holds the lower 16 bits of the 54th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS53_LOW	4ACh

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS53 [31:0] This field contains the lower 32 bits of the 54th 6-byte MAC address. MAC ADDRESS53 [31:0] This field contains the lower 32 bits of the 54th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.156 MAC Address54 High (MAC_ADDRESS54_HIGH)

The MAC ADDRESS54 High register holds the upper 16 bits of the 55th 6-byte MAC address of the station.

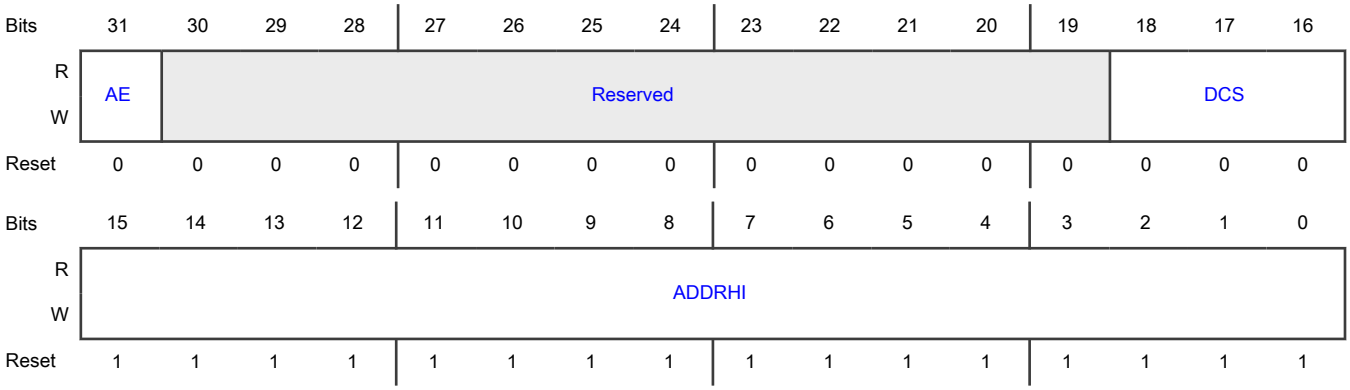
The MAC ADDRESS54 High register holds the upper 16 bits of the 55th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS54_HIGH	4B0h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 55th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 55th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS54 content is routed.
15-0 ADDRHI	MAC ADDRESS54 [47:32] This field contains the upper 16 bits (47:32) of the 55th 6-byte MAC address.

43.8.1.157 MAC Address54 Low (MAC_ADDRESS54_LOW)

The MAC ADDRESS54 Low register holds the lower 16 bits of the 55th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS54_LOW	4B4h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS54 [31:0] This field contains the lower 32 bits of the 55th 6-byte MAC address. MAC ADDRESS54 [31:0] This field contains the lower 32 bits of the 55th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.158 MAC Address55 High (MAC_ADDRESS55_HIGH)

The MAC ADDRESS55 High register holds the upper 16 bits of the 56th 6-byte MAC address of the station.

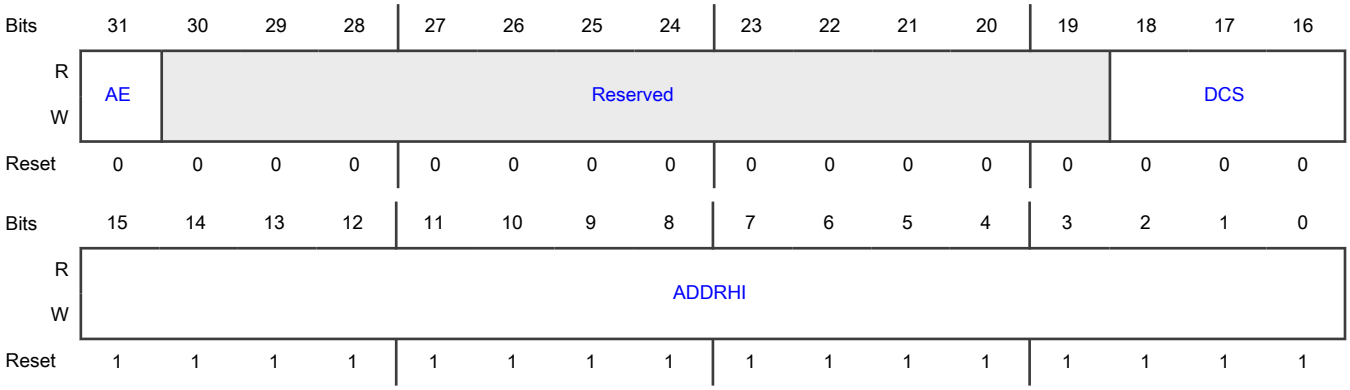
The MAC ADDRESS55 High register holds the upper 16 bits of the 56th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS55_HIGH	4B8h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 56th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 56th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS55 content is routed.
15-0 ADDRHI	MAC ADDRESS55 [47:32] This field contains the upper 16 bits (47:32) of the 56th 6-byte MAC address.

43.8.1.159 MAC Address55 Low (MAC_ADDRESS55_LOW)

The MAC ADDRESS55 Low register holds the lower 16 bits of the 56th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS55_LOW	4BCh

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS55 [31:0] This field contains the lower 32 bits of the 56th 6-byte MAC address. MAC ADDRESS55 [31:0] This field contains the lower 32 bits of the 56th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.160 MAC Address56 High (MAC_ADDRESS56_HIGH)

The MAC ADDRESS56 High register holds the upper 16 bits of the 57th 6-byte MAC address of the station.

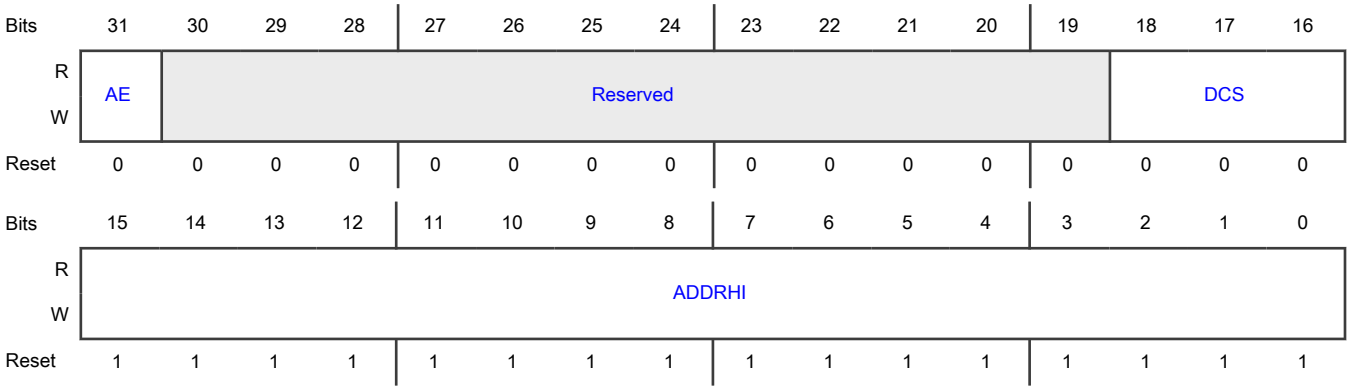
The MAC ADDRESS56 High register holds the upper 16 bits of the 57th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS56_HIGH	4C0h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 57th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 57th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS56 content is routed.
15-0 ADDRHI	MAC ADDRESS56 [47:32] This field contains the upper 16 bits (47:32) of the 57th 6-byte MAC address.

43.8.1.161 MAC Address56 Low (MAC_ADDRESS56_LOW)

The MAC ADDRESS56 Low register holds the lower 16 bits of the 57th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS56_LOW	4C4h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS56 [31:0] This field contains the lower 32 bits of the 57th 6-byte MAC address. MAC ADDRESS56 [31:0] This field contains the lower 32 bits of the 57th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.162 MAC Address57 High (MAC_ADDRESS57_HIGH)

The MAC ADDRESS57 High register holds the upper 16 bits of the 58th 6-byte MAC address of the station.

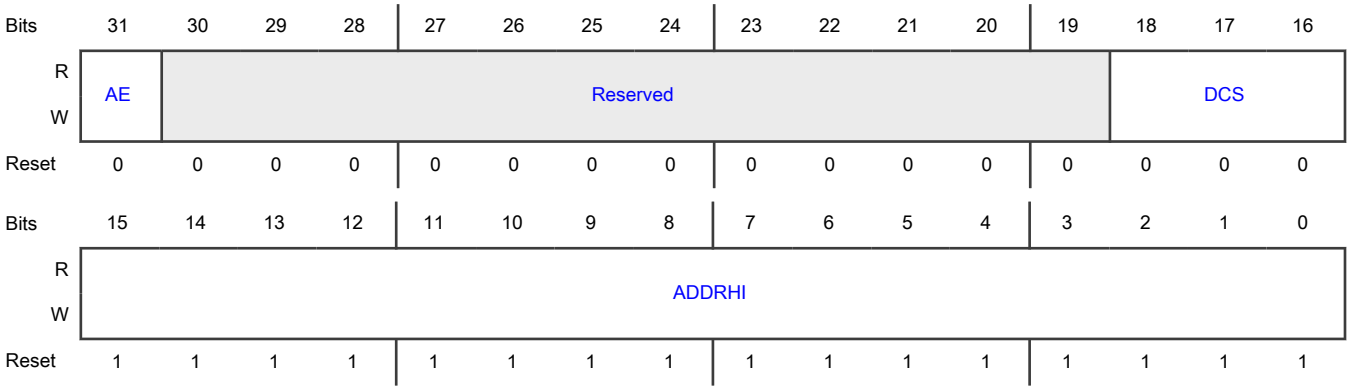
The MAC ADDRESS57 High register holds the upper 16 bits of the 58th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS57_HIGH	4C8h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 58th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 58th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS57 content is routed.
15-0 ADDRHI	MAC ADDRESS57 [47:32] This field contains the upper 16 bits (47:32) of the 58th 6-byte MAC address.

43.8.1.163 MAC Address57 Low (MAC_ADDRESS57_LOW)

The MAC ADDRESS57 Low register holds the lower 16 bits of the 58th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS57_LOW	4CCh

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS57 [31:0] This field contains the lower 32 bits of the 58th 6-byte MAC address. MAC ADDRESS57 [31:0] This field contains the lower 32 bits of the 58th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.164 MAC Address58 High (MAC_ADDRESS58_HIGH)

The MAC ADDRESS58 High register holds the upper 16 bits of the 59th 6-byte MAC address of the station.

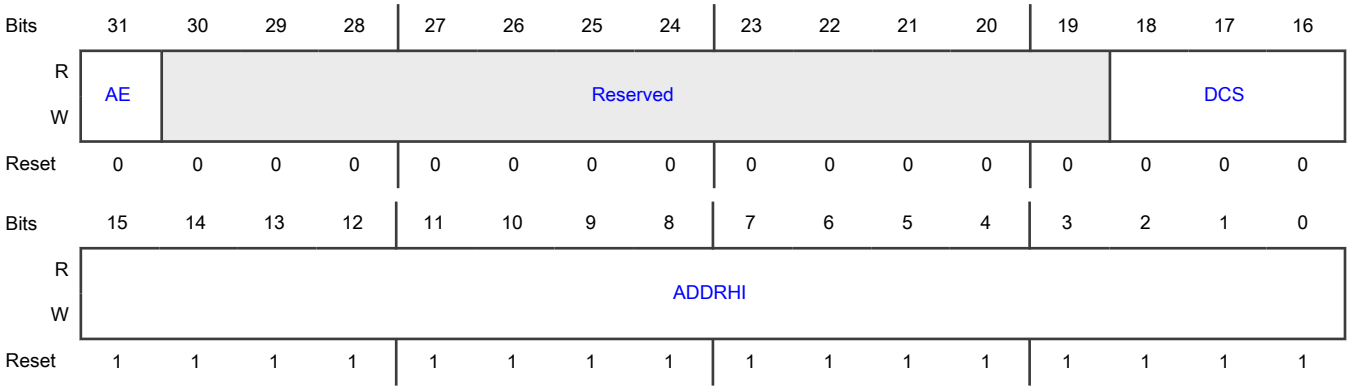
The MAC ADDRESS58 High register holds the upper 16 bits of the 59th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS58_HIGH	4D0h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 59th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 59th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS58 content is routed.
15-0 ADDRHI	MAC ADDRESS58 [47:32] This field contains the upper 16 bits (47:32) of the 59th 6-byte MAC address.

43.8.1.165 MAC Address58 Low (MAC_ADDRESS58_LOW)

The MAC ADDRESS58 Low register holds the lower 16 bits of the 59th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS58_LOW	4D4h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS58 [31:0] This field contains the lower 32 bits of the 59th 6-byte MAC address. MAC ADDRESS58 [31:0] This field contains the lower 32 bits of the 59th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.166 MAC Address59 High (MAC_ADDRESS59_HIGH)

The MAC ADDRESS59 High register holds the upper 16 bits of the 60th 6-byte MAC address of the station.

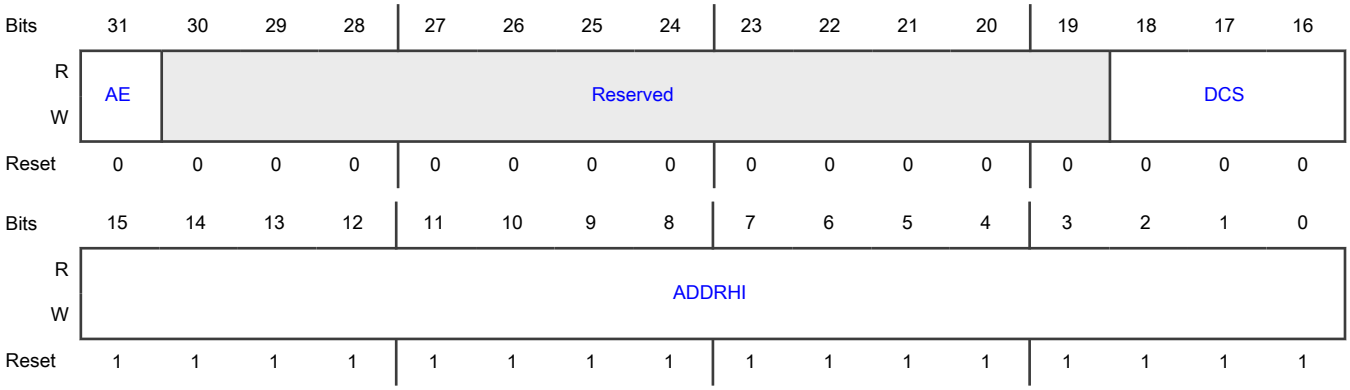
The MAC ADDRESS59 High register holds the upper 16 bits of the 60th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS59_HIGH	4D8h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 60th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 60th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS59 content is routed.
15-0 ADDRHI	MAC ADDRESS59 [47:32] This field contains the upper 16 bits (47:32) of the 60th 6-byte MAC address.

43.8.1.167 MAC Address59 Low (MAC_ADDRESS59_LOW)

The MAC ADDRESS59 Low register holds the lower 16 bits of the 60th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS59_LOW	4DCh

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS59 [31:0] This field contains the lower 32 bits of the 60th 6-byte MAC address. MAC ADDRESS59 [31:0] This field contains the lower 32 bits of the 60th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.168 MAC Address60 High (MAC_ADDRESS60_HIGH)

The MAC ADDRESS60 High register holds the upper 16 bits of the 61st 6-byte MAC address of the station.

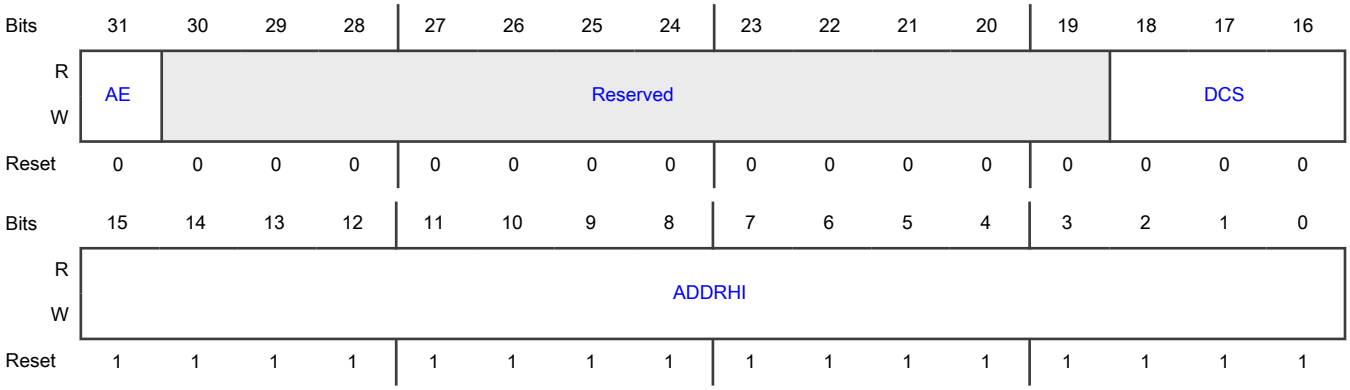
The MAC ADDRESS60 High register holds the upper 16 bits of the 61st 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS60_HIGH	4E0h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 61st MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 61st MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS60 content is routed.
15-0 ADDRHI	MAC ADDRESS60 [47:32] This field contains the upper 16 bits (47:32) of the 61st 6-byte MAC address.

43.8.1.169 MAC Address60 Low (MAC_ADDRESS60_LOW)

The MAC ADDRESS60 Low register holds the lower 16 bits of the 61st 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS60_LOW	4E4h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS60 [31:0] This field contains the lower 32 bits of the 61st 6-byte MAC address. MAC ADDRESS60 [31:0] This field contains the lower 32 bits of the 61st 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.170 MAC Address61 High (MAC_ADDRESS61_HIGH)

The MAC ADDRESS61 High register holds the upper 16 bits of the 62nd 6-byte MAC address of the station.

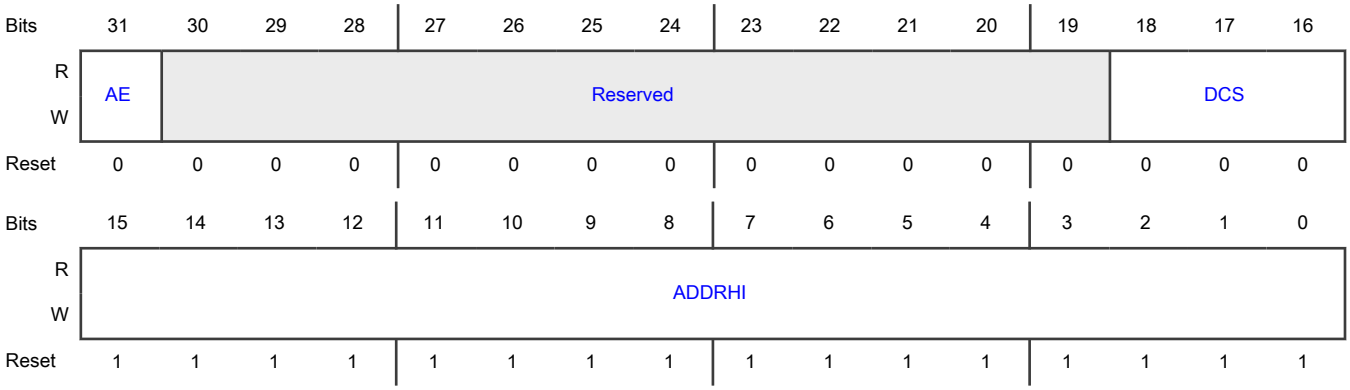
The MAC ADDRESS61 High register holds the upper 16 bits of the 62nd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS61_HIGH	4E8h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 62nd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 62nd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS61 content is routed.
15-0 ADDRHI	MAC ADDRESS61 [47:32] This field contains the upper 16 bits (47:32) of the 62nd 6-byte MAC address.

43.8.1.171 MAC Address61 Low (MAC_ADDRESS61_LOW)

The MAC ADDRESS61 Low register holds the lower 16 bits of the 62nd 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS61_LOW	4ECh

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS61 [31:0] This field contains the lower 32 bits of the 62nd 6-byte MAC address. MAC ADDRESS61 [31:0] This field contains the lower 32 bits of the 62nd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.172 MAC Address62 High (MAC_ADDRESS62_HIGH)

The MAC ADDRESS62 High register holds the upper 16 bits of the 63rd 6-byte MAC address of the station.

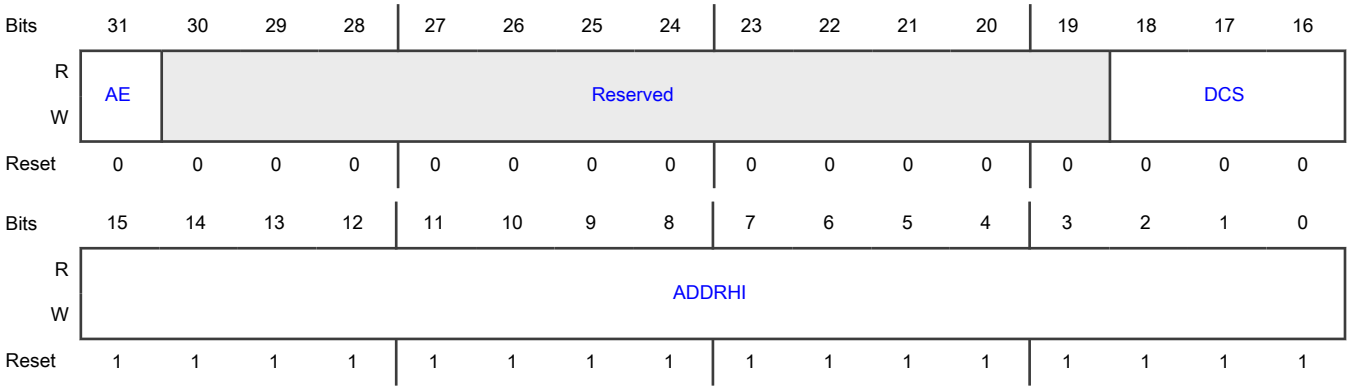
The MAC ADDRESS62 High register holds the upper 16 bits of the 63rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS62_HIGH	4F0h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 63rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 63rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS62 content is routed.
15-0 ADDRHI	MAC ADDRESS62 [47:32] This field contains the upper 16 bits (47:32) of the 63rd 6-byte MAC address.

43.8.1.173 MAC Address62 Low (MAC_ADDRESS62_LOW)

The MAC ADDRESS62 Low register holds the lower 16 bits of the 63rd 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS62_LOW	4F4h

Diagram

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS62 [31:0] This field contains the lower 32 bits of the 63rd 6-byte MAC address. MAC ADDRESS62 [31:0] This field contains the lower 32 bits of the 63rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.174 MAC Address63 High (MAC_ADDRESS63_HIGH)

The MAC ADDRESS63 High register holds the upper 16 bits of the 64th 6-byte MAC address of the station.

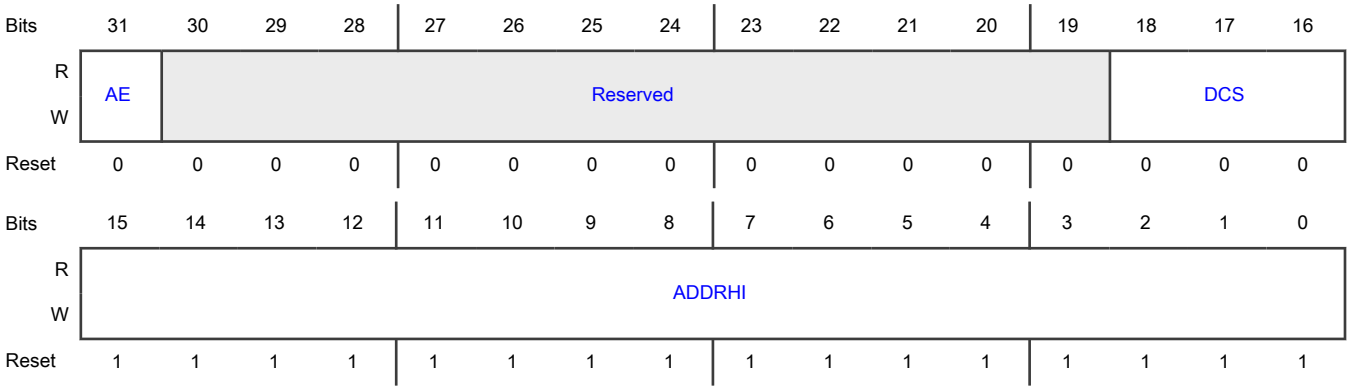
The MAC ADDRESS63 High register holds the upper 16 bits of the 64th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

Offset

Register	Offset
MAC_ADDRESS63_HIGH	4F8h

Diagram



Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 64th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 64th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS63 content is routed.
15-0 ADDRHI	MAC ADDRESS63 [47:32] This field contains the upper 16 bits (47:32) of the 64th 6-byte MAC address.

43.8.1.175 MAC Address63 Low (MAC_ADDRESS63_LOW)

The MAC ADDRESS63 Low register holds the lower 16 bits of the 64th 6-byte MAC address of the station.

Offset

Register	Offset
MAC_ADDRESS63_LOW	4FCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS63 [31:0] This field contains the lower 32 bits of the 64th 6-byte MAC address. MAC ADDRESS63 [31:0] This field contains the lower 32 bits of the 64th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

43.8.1.176 MMC Control (MAC_MMC_CONTROL)

This register establishes the operating mode of MMC.

Offset

Register	Offset
MAC_MMC_CONTROL	700h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							UCDBC	Reserved		CNTPRST...	CNTPRST	CNTFREEZ	RSTONRD	CNTSTOP...	CNTRST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-9 —	Reserved
8 UCDBC	Update MMC Counters for Dropped Broadcast Packets Note: The CNTRST bit has a higher priority than the CNTPRST bit. Update MMC Counters for Dropped Broadcast Packets Note: The CNTRST bit has a higher priority than the CNTPRST bit. Therefore, when the software tries to set both bits in the same write cycle, all counters are cleared and the CNTPRST bit is not set. When set, the MAC updates all related MMC Counters for Broadcast packets that are dropped because of the setting of the DBF bit of MAC_Packet_Filter register. When reset, the MMC Counters are not updated for dropped Broadcast packets. 0 - Update MMC Counters for Dropped Broadcast Packets is disabled 1 - Update MMC Counters for Dropped Broadcast Packets is enabled
7-6 —	Reserved
5 CNTPRSTLVL	Full-Half Preset When this bit is low and the CNTPRST bit is set, all MMC counters get preset to almost-half value. Full-Half Preset When this bit is low and the CNTPRST bit is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (Half 2KBytes) and all packet-counters gets preset to 0x7FFF_FFF0 (Half 16). When this bit is high and the CNTPRST bit is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (Full 2KBytes) and all packet-counters gets preset to 0xFFFF_FFF0 (Full 16). For 16-bit counters, the almost-half preset values are 0x7800 and 0x7FF0 for the respective octet and packet counters. Similarly, the almost-full preset values for the 16-bit counters are 0xF800 and 0xFFF0. 0 - Full-Half Preset is disabled 1 - Full-Half Preset is enabled
4 CNTPRST	Counters Preset When this bit is set, all counters are initialized or preset to almost full or almost half according to the CNTPRSTLVL bit. Counters Preset When this bit is set, all counters are initialized or preset to almost full or almost half according to the CNTPRSTLVL bit. This bit is cleared automatically after 1 clock cycle. This bit, along with the CNTPRSTLVL bit, is useful for debugging and testing the assertion of interrupts because of MMC counter becoming half-full or full. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0 - Counters Preset is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Counters Preset is enabled
3 CNTFREEZ	MMC Counter Freeze When this bit is set, it freezes all MMC counters to their current value. MMC Counter Freeze When this bit is set, it freezes all MMC counters to their current value. Until this bit is reset to 0, no MMC counter is updated because of any transmitted or received packet. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode. 0 - MMC Counter Freeze is disabled 1 - MMC Counter Freeze is enabled
2 RSTONRD	Reset on Read When this bit is set, the MMC counters are reset to zero after Read (self-clearing after reset). Reset on Read When this bit is set, the MMC counters are reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (Bits[7:0]) is read. 0 - Reset on Read is disabled 1 - Reset on Read is enabled
1 CNTSTOPRO	Counter Stop Rollover When this bit is set, the counter does not roll over to zero after reaching the maximum value. 0 - Counter Stop Rollover is disabled 1 - Counter Stop Rollover is enabled
0 CNTRST	Counters Reset When this bit is set, all counters are reset. Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0 - Counters are not reset 1 - All counters are reset

43.8.1.177 MMC Rx Interrupt (MAC_MMC_RX_INTERRUPT)

This register maintains the interrupts generated from all Receive statistics counters.

This register maintains the interrupts generated from all Receive statistics counters.

The MMC Receive Interrupt register maintains the interrupts that are generated when the following occur:

- Receive statistic counters reach half of their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter).

- Receive statistic counters cross their maximum values (0xFFFF_FFFF for 32 bit counter and 0xFFFF for 16 bit counter).

When the Counter Stop Rollover is set, interrupts are set but the counter remains at all-ones. The MMC Receive Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (Bits[7:0]) of the respective counter must be read to clear the interrupt bit.

Note: R_SS_RC means that this register bit is set internally, and it is cleared when the Counter register is read.

Offset

Register	Offset
MAC_MMC_RX_INTERRUPT	704h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				RXLPITR...	RXLPIUS...	RXCTRLP...	RXRCVER...	RXWDOGP...	RXVLANG...	RXFOVPIS	RXPAUSP...	RXORANG...	RXLENER...	RXUCGPIS	RX1024T...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX512T1...	RX256T5...	RX128T2...	RX65T12...	RX64OCT...	RXOSIZE...	RXUSIZE...	RXJABER...	RXRUNTP...	RXALGNE...	RXCRCER...	RXMCGPIS	RXBCGPIS	RXGOCTIS	RXGBOCT...	RXGBPKT...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27 RXLPITRCIS	MMC Receive LPI transition counter interrupt status This bit is set when the Rx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. MMC Receive LPI transition counter interrupt status This bit is set when the Rx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive LPI transition Counter Interrupt Status not detected 1 - MMC Receive LPI transition Counter Interrupt Status detected
26 RXLPIUSCIS	MMC Receive LPI microsecond counter interrupt status This bit is set when the Rx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. MMC Receive LPI microsecond counter interrupt status This bit is set when the Rx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive LPI microsecond Counter Interrupt Status not detected 1 - MMC Receive LPI microsecond Counter Interrupt Status detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
25 RXCTRLPIS	MMC Receive Control Packet Counter Interrupt Status This bit is set when the rxctrlpackets_g counter reaches half of the maximum value or the maximum value. MMC Receive Control Packet Counter Interrupt Status This bit is set when the rxctrlpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Control Packet Counter Interrupt Status not detected 1 - MMC Receive Control Packet Counter Interrupt Status detected
24 RXRCVERRPIS	MMC Receive Error Packet Counter Interrupt Status This bit is set when the rxrcverror counter reaches half of the maximum value or the maximum value. MMC Receive Error Packet Counter Interrupt Status This bit is set when the rxrcverror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Error Packet Counter Interrupt Status not detected 1 - MMC Receive Error Packet Counter Interrupt Status detected
23 RXWDOGPIS	MMC Receive Watchdog Error Packet Counter Interrupt Status This bit is set when the rxwatchdog error counter reaches half of the maximum value or the maximum value. MMC Receive Watchdog Error Packet Counter Interrupt Status This bit is set when the rxwatchdog error counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Watchdog Error Packet Counter Interrupt Status not detected 1 - MMC Receive Watchdog Error Packet Counter Interrupt Status detected
22 RXVLANGBPIS	MMC Receive VLAN Good Bad Packet Counter Interrupt Status This bit is set when the rxvlanpackets_gb counter reaches half of the maximum value or the maximum value. MMC Receive VLAN Good Bad Packet Counter Interrupt Status This bit is set when the rxvlanpackets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive VLAN Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive VLAN Good Bad Packet Counter Interrupt Status detected
21 RXFOVPIS	MMC Receive FIFO Overflow Packet Counter Interrupt Status This bit is set when the rxfifooverflow counter reaches half of the maximum value or the maximum value. MMC Receive FIFO Overflow Packet Counter Interrupt Status This bit is set when the rxfifooverflow counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive FIFO Overflow Packet Counter Interrupt Status not detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - MMC Receive FIFO Overflow Packet Counter Interrupt Status detected
20 RXPAUSPIS	MMC Receive Pause Packet Counter Interrupt Status This bit is set when the rxpausepackets counter reaches half of the maximum value or the maximum value. MMC Receive Pause Packet Counter Interrupt Status This bit is set when the rxpausepackets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Pause Packet Counter Interrupt Status not detected 1 - MMC Receive Pause Packet Counter Interrupt Status detected
19 RXORANGEPIS	MMC Receive Out Of Range Error Packet Counter Interrupt Status. MMC Receive Out Of Range Error Packet Counter Interrupt Status. This bit is set when the rxoutofrangetype counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Out Of Range Error Packet Counter Interrupt Status not detected 1 - MMC Receive Out Of Range Error Packet Counter Interrupt Status detected
18 RXLENERPIS	MMC Receive Length Error Packet Counter Interrupt Status This bit is set when the rxlengtherror counter reaches half of the maximum value or the maximum value. MMC Receive Length Error Packet Counter Interrupt Status This bit is set when the rxlengtherror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Length Error Packet Counter Interrupt Status not detected 1 - MMC Receive Length Error Packet Counter Interrupt Status detected
17 RXUCGPIS	MMC Receive Unicast Good Packet Counter Interrupt Status This bit is set when the rxunicastpackets_g counter reaches half of the maximum value or the maximum value. MMC Receive Unicast Good Packet Counter Interrupt Status This bit is set when the rxunicastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Unicast Good Packet Counter Interrupt Status not detected 1 - MMC Receive Unicast Good Packet Counter Interrupt Status detected
16 RX1024TMAXOCTGBPIS	MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status detected
15 RX512T1023OCTGBPIS	MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status detected
14 RX256T511OCTGBPIS	MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status detected
13 RX128T255OCTGBPIS	MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status detected
12 RX65T127OCTGBPIS	MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
11 RX64OCTGBPIS	MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx64octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx64octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status detected
10 RXOSIZEGPIS	MMC Receive Oversize Good Packet Counter Interrupt Status This bit is set when the rxoversize_g counter reaches half of the maximum value or the maximum value. MMC Receive Oversize Good Packet Counter Interrupt Status This bit is set when the rxoversize_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Oversize Good Packet Counter Interrupt Status not detected 1 - MMC Receive Oversize Good Packet Counter Interrupt Status detected
9 RXUSIZEGPIS	MMC Receive Undersize Good Packet Counter Interrupt Status This bit is set when the rxundersize_g counter reaches half of the maximum value or the maximum value. MMC Receive Undersize Good Packet Counter Interrupt Status This bit is set when the rxundersize_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Undersize Good Packet Counter Interrupt Status not detected 1 - MMC Receive Undersize Good Packet Counter Interrupt Status detected
8 RXJABERPIS	MMC Receive Jabber Error Packet Counter Interrupt Status This bit is set when the rxjabbererror counter reaches half of the maximum value or the maximum value. MMC Receive Jabber Error Packet Counter Interrupt Status This bit is set when the rxjabbererror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Jabber Error Packet Counter Interrupt Status not detected 1 - MMC Receive Jabber Error Packet Counter Interrupt Status detected
7 RXRUNTPIS	MMC Receive Runt Packet Counter Interrupt Status This bit is set when the rxrunterror counter reaches half of the maximum value or the maximum value. MMC Receive Runt Packet Counter Interrupt Status This bit is set when the rxrunterror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Runt Packet Counter Interrupt Status not detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - MMC Receive Runt Packet Counter Interrupt Status detected
6 RXALGNERPIS	MMC Receive Alignment Error Packet Counter Interrupt Status This bit is set when the rxalignmenterror counter reaches half of the maximum value or the maximum value. MMC Receive Alignment Error Packet Counter Interrupt Status This bit is set when the rxalignmenterror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Alignment Error Packet Counter Interrupt Status not detected 1 - MMC Receive Alignment Error Packet Counter Interrupt Status detected
5 RXCRCERPIS	MMC Receive CRC Error Packet Counter Interrupt Status This bit is set when the rxcrcerror counter reaches half of the maximum value or the maximum value. MMC Receive CRC Error Packet Counter Interrupt Status This bit is set when the rxcrcerror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive CRC Error Packet Counter Interrupt Status not detected 1 - MMC Receive CRC Error Packet Counter Interrupt Status detected
4 RXMCGPIS	MMC Receive Multicast Good Packet Counter Interrupt Status This bit is set when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. MMC Receive Multicast Good Packet Counter Interrupt Status This bit is set when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Multicast Good Packet Counter Interrupt Status not detected 1 - MMC Receive Multicast Good Packet Counter Interrupt Status detected
3 RXBCGPIS	MMC Receive Broadcast Good Packet Counter Interrupt Status This bit is set when the rxbroadcastpackets_g counter reaches half of the maximum value or the maximum value. MMC Receive Broadcast Good Packet Counter Interrupt Status This bit is set when the rxbroadcastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Broadcast Good Packet Counter Interrupt Status not detected 1 - MMC Receive Broadcast Good Packet Counter Interrupt Status detected
2 RXGOCTIS	MMC Receive Good Octet Counter Interrupt Status This bit is set when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. MMC Receive Good Octet Counter Interrupt Status

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This bit is set when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Good Octet Counter Interrupt Status not detected 1 - MMC Receive Good Octet Counter Interrupt Status detected
1 RXGBOCTIS	MMC Receive Good Bad Octet Counter Interrupt Status This bit is set when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. MMC Receive Good Bad Octet Counter Interrupt Status This bit is set when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Good Bad Octet Counter Interrupt Status not detected 1 - MMC Receive Good Bad Octet Counter Interrupt Status detected
0 RXGBPKTIS	MMC Receive Good Bad Packet Counter Interrupt Status This bit is set when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. MMC Receive Good Bad Packet Counter Interrupt Status This bit is set when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive Good Bad Packet Counter Interrupt Status detected

43.8.1.178 MMC Tx Interrupt (MAC_MMC_TX_INTERRUPT)

This register maintains the interrupts generated from all Transmit statistics counters.

This register maintains the interrupts generated from all Transmit statistics counters.

The MMC Transmit Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values

(0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter).

When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones.

The MMC Transmit Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read.

The least significant byte lane (Bits[7:0]) of the respective counter must be read to clear the interrupt bit.

Offset

Register	Offset
MAC_MMC_TX_INTERRUPT	708h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				TXLPITR...	TXLPIUS...	TXOSIZE...	TXVLANG...	TXPAUSP...	TXEXDEF...	TXGPKTIS	TXGOCTIS	TXCARER...	TXEXCOL...	TXLATCO...	TXDEFPIS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXMCOLG...	TXSCOLG...	TXUFLOW...	TXBCGBP...	TXMCGBP...	TXUCGBP...	TX1024T...	TX512T1...	TX256T5...	TX128T2...	TX65T12...	TX64OCT...	TXMCGPIS	TXBCGPIS	TXGBPKT...	TXGBOCT...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27 TXLPITRCIS	MMC Transmit LPI transition counter interrupt status This bit is set when the Tx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. MMC Transmit LPI transition counter interrupt status This bit is set when the Tx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit LPI transition Counter Interrupt Status not detected 1 - MMC Transmit LPI transition Counter Interrupt Status detected
26 TXLPIUSCIS	MMC Transmit LPI microsecond counter interrupt status This bit is set when the Tx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. MMC Transmit LPI microsecond counter interrupt status This bit is set when the Tx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit LPI microsecond Counter Interrupt Status not detected 1 - MMC Transmit LPI microsecond Counter Interrupt Status detected
25 TXOSIZEGPIS	MMC Transmit Oversize Good Packet Counter Interrupt Status This bit is set when the txoversize_g counter reaches half of the maximum value or the maximum value. MMC Transmit Oversize Good Packet Counter Interrupt Status This bit is set when the txoversize_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - MMC Transmit Oversize Good Packet Counter Interrupt Status not detected 1 - MMC Transmit Oversize Good Packet Counter Interrupt Status detected
24 TXVLANGPIS	MMC Transmit VLAN Good Packet Counter Interrupt Status This bit is set when the txvlanpackets_g counter reaches half of the maximum value or the maximum value. MMC Transmit VLAN Good Packet Counter Interrupt Status This bit is set when the txvlanpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit VLAN Good Packet Counter Interrupt Status not detected 1 - MMC Transmit VLAN Good Packet Counter Interrupt Status detected
23 TXPAUSPIS	MMC Transmit Pause Packet Counter Interrupt Status This bit is set when the txpausepacketserror counter reaches half of the maximum value or the maximum value. MMC Transmit Pause Packet Counter Interrupt Status This bit is set when the txpausepacketserror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Pause Packet Counter Interrupt Status not detected 1 - MMC Transmit Pause Packet Counter Interrupt Status detected
22 TXEXDEFPIS	MMC Transmit Excessive Deferral Packet Counter Interrupt Status This bit is set when the txexcessdef counter reaches half of the maximum value or the maximum value. MMC Transmit Excessive Deferral Packet Counter Interrupt Status This bit is set when the txexcessdef counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Excessive Deferral Packet Counter Interrupt Status not detected 1 - MMC Transmit Excessive Deferral Packet Counter Interrupt Status detected
21 TXGPKTIS	MMC Transmit Good Packet Counter Interrupt Status This bit is set when the txpacketcount_g counter reaches half of the maximum value or the maximum value. MMC Transmit Good Packet Counter Interrupt Status This bit is set when the txpacketcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Good Packet Counter Interrupt Status not detected 1 - MMC Transmit Good Packet Counter Interrupt Status detected
20 TXGOCTIS	MMC Transmit Good Octet Counter Interrupt Status This bit is set when the txoctetcount_g counter reaches half of the maximum value or the maximum value. MMC Transmit Good Octet Counter Interrupt Status

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This bit is set when the txoctetcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Good Octet Counter Interrupt Status not detected 1 - MMC Transmit Good Octet Counter Interrupt Status detected
19 TXCARERPIS	MMC Transmit Carrier Error Packet Counter Interrupt Status This bit is set when the txcarriererror counter reaches half of the maximum value or the maximum value. MMC Transmit Carrier Error Packet Counter Interrupt Status This bit is set when the txcarriererror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Carrier Error Packet Counter Interrupt Status not detected 1 - MMC Transmit Carrier Error Packet Counter Interrupt Status detected
18 TXEXCOLPIS	MMC Transmit Excessive Collision Packet Counter Interrupt Status This bit is set when the txexesscol counter reaches half of the maximum value or the maximum value. MMC Transmit Excessive Collision Packet Counter Interrupt Status This bit is set when the txexesscol counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Excessive Collision Packet Counter Interrupt Status not detected 1 - MMC Transmit Excessive Collision Packet Counter Interrupt Status detected
17 TXLATCOLPIS	MMC Transmit Late Collision Packet Counter Interrupt Status This bit is set when the txlatecol counter reaches half of the maximum value or the maximum value. MMC Transmit Late Collision Packet Counter Interrupt Status This bit is set when the txlatecol counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Late Collision Packet Counter Interrupt Status not detected 1 - MMC Transmit Late Collision Packet Counter Interrupt Status detected
16 TXDEFPIS	MMC Transmit Deferred Packet Counter Interrupt Status This bit is set when the txdeferred counter reaches half of the maximum value or the maximum value. MMC Transmit Deferred Packet Counter Interrupt Status This bit is set when the txdeferred counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Deferred Packet Counter Interrupt Status not detected 1 - MMC Transmit Deferred Packet Counter Interrupt Status detected
15	MMC Transmit Multiple Collision Good Packet Counter Interrupt Status This bit is set when the txmulticol_g counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Table continued from the previous page...

Field	Description
TXMCOLGPIS	MMC Transmit Multiple Collision Good Packet Counter Interrupt Status This bit is set when the txmulticol_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Multiple Collision Good Packet Counter Interrupt Status not detected 1 - MMC Transmit Multiple Collision Good Packet Counter Interrupt Status detected
14 TXSCOLGPIS	MMC Transmit Single Collision Good Packet Counter Interrupt Status This bit is set when the txsinglecol_g counter reaches half of the maximum value or the maximum value. MMC Transmit Single Collision Good Packet Counter Interrupt Status This bit is set when the txsinglecol_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Single Collision Good Packet Counter Interrupt Status not detected 1 - MMC Transmit Single Collision Good Packet Counter Interrupt Status detected
13 TXUFLOWERPIS	MMC Transmit Underflow Error Packet Counter Interrupt Status This bit is set when the txunderflowerror counter reaches half of the maximum value or the maximum value. MMC Transmit Underflow Error Packet Counter Interrupt Status This bit is set when the txunderflowerror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Underflow Error Packet Counter Interrupt Status not detected 1 - MMC Transmit Underflow Error Packet Counter Interrupt Status detected
12 TXBCGBPIS	MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status This bit is set when the txbroadcastpackets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status This bit is set when the txbroadcastpackets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status detected
11 TXMCGBPIS	MMC Transmit Multicast Good Bad Packet Counter Interrupt Status The bit is set when the txmulticastpackets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Multicast Good Bad Packet Counter Interrupt Status The bit is set when the txmulticastpackets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Multicast Good Bad Packet Counter Interrupt Status not detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - MMC Transmit Multicast Good Bad Packet Counter Interrupt Status detected
10 TXUCGBPIS	MMC Transmit Unicast Good Bad Packet Counter Interrupt Status This bit is set when the txunicastpackets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Unicast Good Bad Packet Counter Interrupt Status This bit is set when the txunicastpackets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Unicast Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit Unicast Good Bad Packet Counter Interrupt Status detected
9 TX1024TMAXOCTGBPIS	MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status detected
8 TX512T1023OCTGBPIS	MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status detected
7 TX256T511OCTGBPIS	MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status detected
6	MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Table continued from the previous page...

Field	Description
TX128T255OCTGBPIS	MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status detected
5 TX65T127OCTGBPIS	MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx65to127octets_gb counter reaches half the maximum value, and also when it reaches the maximum value. MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx65to127octets_gb counter reaches half the maximum value, and also when it reaches the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status detected
4 TX64OCTGBPIS	MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx64octets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx64octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status detected
3 TXMCGPIS	MMC Transmit Multicast Good Packet Counter Interrupt Status This bit is set when the txmulticastpackets_g counter reaches half of the maximum value or the maximum value. MMC Transmit Multicast Good Packet Counter Interrupt Status This bit is set when the txmulticastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Multicast Good Packet Counter Interrupt Status not detected 1 - MMC Transmit Multicast Good Packet Counter Interrupt Status detected
2 TXBCGPIS	MMC Transmit Broadcast Good Packet Counter Interrupt Status This bit is set when the txbroadcastpackets_g counter reaches half of the maximum value or the maximum value. MMC Transmit Broadcast Good Packet Counter Interrupt Status This bit is set when the txbroadcastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - MMC Transmit Broadcast Good Packet Counter Interrupt Status not detected 1 - MMC Transmit Broadcast Good Packet Counter Interrupt Status detected
1 TXGBPKTIS	MMC Transmit Good Bad Packet Counter Interrupt Status This bit is set when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Good Bad Packet Counter Interrupt Status This bit is set when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit Good Bad Packet Counter Interrupt Status detected
0 TXGBOCTIS	MMC Transmit Good Bad Octet Counter Interrupt Status This bit is set when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Good Bad Octet Counter Interrupt Status This bit is set when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Good Bad Octet Counter Interrupt Status not detected 1 - MMC Transmit Good Bad Octet Counter Interrupt Status detected

43.8.1.179 MMC Rx Interrupt Mask (MAC_MMC_RX_INTERRUPT_MASK)

This register maintains the masks for interrupts generated from all Receive statistics counters.

This register maintains the masks for interrupts generated from all Receive statistics counters.

The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when receive statistic counters reach half of their maximum value or the maximum values.

This register is 32 bit wide.

Offset

Register	Offset
MAC_MMC_RX_INTERRUPT_MASK	70Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				RXLPITR...	RXLPIUS...	RXCTRLP...	RXRCVER...	RXWDOGP...	RXVLANG...	RXFOVPIM	RXPAUSP...	RXORANG...	RXLENER...	RXUCGPIM	RX1024T...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX512T1...	RX256T5...	RX128T2...	RX65T12...	RX64OCT...	RXOSIZE...	RXUSIZE...	RXJABER...	RXRUNTP...	RXALGNE...	RXCRCER...	RXMCGPIM	RXBCGPIM	RXGOCTIM	RXGBOCT...	RXGBPKT...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27 RXLPITRCIM	MMC Receive LPI transition counter interrupt Mask Setting this bit masks the interrupt when the Rx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. 0 - MMC Receive LPI transition counter interrupt Mask is disabled 1 - MMC Receive LPI transition counter interrupt Mask is enabled
26 RXLPIUSCIM	MMC Receive LPI microsecond counter interrupt Mask Setting this bit masks the interrupt when the Rx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. 0 - MMC Receive LPI microsecond counter interrupt Mask is disabled 1 - MMC Receive LPI microsecond counter interrupt Mask is enabled
25 RXCTRLPIM	MMC Receive Control Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxctrlpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Control Packet Counter Interrupt Mask is disabled 1 - MMC Receive Control Packet Counter Interrupt Mask is enabled
24 RXRCVERRPIM	MMC Receive Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxrcverror counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive Error Packet Counter Interrupt Mask is enabled
23 RXWDOGPIM	MMC Receive Watchdog Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxwatchdog counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Watchdog Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive Watchdog Error Packet Counter Interrupt Mask is enabled
22 RXVLANGBPIM	MMC Receive VLAN Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxvlanpackets_gb counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - MMC Receive VLAN Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive VLAN Good Bad Packet Counter Interrupt Mask is enabled
21 RXFOVPIM	MMC Receive FIFO Overflow Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxfifooverflow counter reaches half of the maximum value or the maximum value. 0 - MMC Receive FIFO Overflow Packet Counter Interrupt Mask is disabled 1 - MMC Receive FIFO Overflow Packet Counter Interrupt Mask is enabled
20 RXPAUSPIM	MMC Receive Pause Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxpausepackets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Pause Packet Counter Interrupt Mask is disabled 1 - MMC Receive Pause Packet Counter Interrupt Mask is enabled
19 RXORANGEPIM	MMC Receive Out Of Range Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoutofrangetype counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Out Of Range Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive Out Of Range Error Packet Counter Interrupt Mask is enabled
18 RXLENERPIM	MMC Receive Length Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxlengtherror counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Length Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive Length Error Packet Counter Interrupt Mask is enabled
17 RXUCGPIM	MMC Receive Unicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxunicastpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Unicast Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive Unicast Good Packet Counter Interrupt Mask is enabled
16 RX1024TMAXOCTGBPIM	MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask. MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask. Setting this bit masks the interrupt when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is enabled
15 RX512T1023OCTGBPIM	MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
14 RX256T511OCTGBPIM	MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is enabled
13 RX128T255OCTGBPIM	MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is enabled
12 RX65T127OCTGBPIM	MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is enabled
11 RX64OCTGBPIM	MMC Receive 64 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx64octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive 64 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive 64 Octet Good Bad Packet Counter Interrupt Mask is enabled
10 RXOSIZEGPIM	MMC Receive Oversize Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoversize_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Oversize Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive Oversize Good Packet Counter Interrupt Mask is enabled
9 RXUSIZEGPIM	MMC Receive Undersize Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxundersize_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Undersize Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive Undersize Good Packet Counter Interrupt Mask is enabled
8 RXJABERPIM	MMC Receive Jabber Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxjabbererror counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Jabber Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive Jabber Error Packet Counter Interrupt Mask is enabled
7 RXRUNTPIM	MMC Receive Runt Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxrunterror counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Runt Packet Counter Interrupt Mask is disabled 1 - MMC Receive Runt Packet Counter Interrupt Mask is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
6 RXALGNERPIM	MMC Receive Alignment Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxalignmenterror counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Alignment Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive Alignment Error Packet Counter Interrupt Mask is enabled
5 RXCRCERPIM	MMC Receive CRC Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxcrcerror counter reaches half of the maximum value or the maximum value. 0 - MMC Receive CRC Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive CRC Error Packet Counter Interrupt Mask is enabled
4 RXMCGPIM	MMC Receive Multicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Multicast Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive Multicast Good Packet Counter Interrupt Mask is enabled
3 RXBCGPIM	MMC Receive Broadcast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxbroadcastpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Broadcast Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive Broadcast Good Packet Counter Interrupt Mask is enabled
2 RXGOCTIM	MMC Receive Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive Good Octet Counter Interrupt Mask is enabled
1 RXGBOCTIM	MMC Receive Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Good Bad Octet Counter Interrupt Mask is disabled 1 - MMC Receive Good Bad Octet Counter Interrupt Mask is enabled
0 RXGBPKTIM	MMC Receive Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive Good Bad Packet Counter Interrupt Mask is enabled

43.8.1.180 MMC Tx Interrupt Mask (MAC_MMC_TX_INTERRUPT_MASK)

This register maintains the masks for interrupts generated from all Transmit statistics counters.

This register maintains the masks for interrupts generated from all Transmit statistics counters.

The MMC Transmit Interrupt Mask register maintains the masks for the interrupts generated when the transmit statistic counters reach half of their maximum value or the maximum values. This register is 32 bit wide.

Offset

Register	Offset
MAC_MMC_TX_INTERRUPT_MASK	710h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				TXLPITR...	TXLPIUS...	TXOSIZE...	TXVLANG...	TXPAUSP...	TXEXDEF...	TXGPKTIM	TXGOCTIM	TXCARER...	TXEXCOL...	TXLATCO...	TXDEFPIM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXMCOLG...	TXSCOLG...	TXUFLOW...	TXBCGBP...	TXMCGBP...	TXUCGBP...	TX1024T...	TX512T1...	TX256T5...	TX128T2...	TX65T12...	TX64OCT...	TXMCGPIM	TXBCGPIM	TXGBPKT...	TXGBOCT...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27 TXLPITRCIM	MMC Transmit LPI transition counter interrupt Mask Setting this bit masks the interrupt when the Tx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit LPI transition counter interrupt Mask is disabled 1 - MMC Transmit LPI transition counter interrupt Mask is enabled
26 TXLPIUSCIM	MMC Transmit LPI microsecond counter interrupt Mask Setting this bit masks the interrupt when the Tx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit LPI microsecond counter interrupt Mask is disabled 1 - MMC Transmit LPI microsecond counter interrupt Mask is enabled
25 TXOSIZEGPIM	MMC Transmit Oversize Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txoversize_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Oversize Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Oversize Good Packet Counter Interrupt Mask is enabled
24 TXVLANGPIM	MMC Transmit VLAN Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txvlanpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit VLAN Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit VLAN Good Packet Counter Interrupt Mask is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
23 TXPAUSPIM	MMC Transmit Pause Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpausepackets counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Pause Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Pause Packet Counter Interrupt Mask is enabled
22 TXEXDEFPIM	MMC Transmit Excessive Deferral Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txexcessdef counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Excessive Deferral Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Excessive Deferral Packet Counter Interrupt Mask is enabled
21 TXGPKTIM	MMC Transmit Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpacketcount_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Good Packet Counter Interrupt Mask is enabled
20 TXGOCTIM	MMC Transmit Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Good Octet Counter Interrupt Mask is disabled 1 - MMC Transmit Good Octet Counter Interrupt Mask is enabled
19 TXCARERPIM	MMC Transmit Carrier Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txcarriererror counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Carrier Error Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Carrier Error Packet Counter Interrupt Mask is enabled
18 TXEXCOLPIM	MMC Transmit Excessive Collision Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txexcesscol counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Excessive Collision Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Excessive Collision Packet Counter Interrupt Mask is enabled
17 TXLATCOLPIM	MMC Transmit Late Collision Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txlatecol counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Late Collision Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Late Collision Packet Counter Interrupt Mask is enabled
16 TXDEFPIM	MMC Transmit Deferred Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txdeferred counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Deferred Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Deferred Packet Counter Interrupt Mask is enabled
15 TXMCOLGPIM	MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticol_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask is enabled
14 TXSCOLGPIM	MMC Transmit Single Collision Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txsinglecol_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Single Collision Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Single Collision Good Packet Counter Interrupt Mask is enabled
13 TXUFLOWERPIM	MMC Transmit Underflow Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txunderflowerror counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Underflow Error Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Underflow Error Packet Counter Interrupt Mask is enabled
12 TXBCGBPIM	MMC Transmit Broadcast Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txbroadcastpackets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Broadcast Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Broadcast Good Bad Packet Counter Interrupt Mask is enabled
11 TXMCGBPIM	MMC Transmit Multicast Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticastpackets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Multicast Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Multicast Good Bad Packet Counter Interrupt Mask is enabled
10 TXUCGBPIM	MMC Transmit Unicast Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txunicastpackets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Unicast Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Unicast Good Bad Packet Counter Interrupt Mask is enabled
9 TX1024TMAXOCTGBPIM	MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is enabled
8 TX512T1023OCTGBPIM	MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is enabled
7 TX256T511OCTGBPIM	MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is enabled
6 TX128T255OCTGBPIM	MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is enabled
5 TX65T127OCTGBPIM	MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx65to127octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is enabled
4 TX64OCTGBPIM	MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx64octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Mask is enabled
3 TXMCGPIM	MMC Transmit Multicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticastpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Multicast Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Multicast Good Packet Counter Interrupt Mask is enabled
2 TXBCGPIM	MMC Transmit Broadcast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txbroadcastpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Broadcast Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Broadcast Good Packet Counter Interrupt Mask is enabled
1 TXGBPKTIM	MMC Transmit Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Good Bad Packet Counter Interrupt Mask is enabled
0 TXGBOCTIM	MMC Transmit Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Good Bad Octet Counter Interrupt Mask is disabled 1 - MMC Transmit Good Bad Octet Counter Interrupt Mask is enabled

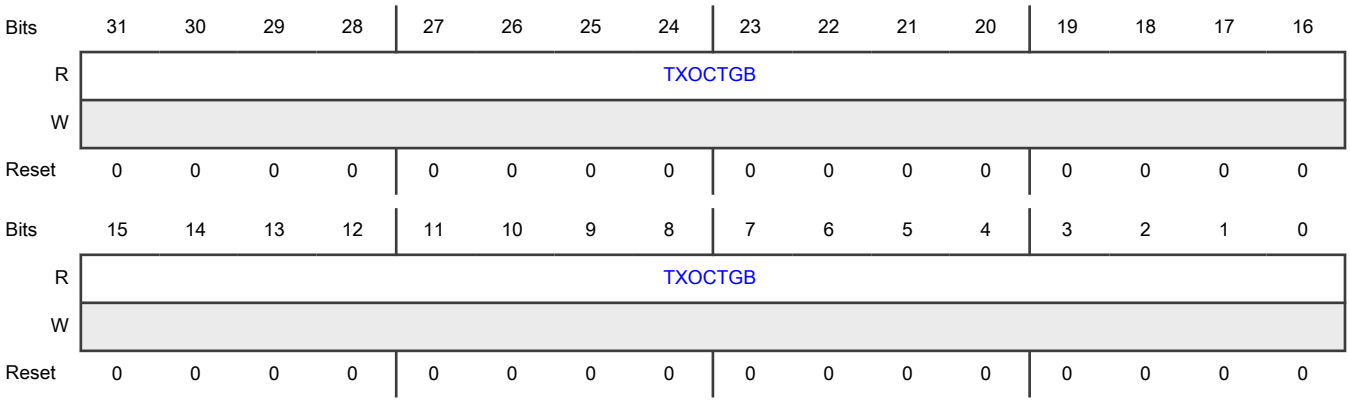
43.8.1.181 Tx Octet Count Good and Bad (MAC_TX_OCTET_COUNT_GOOD_BAD)

This register provides the number of bytes transmitted by the DWC_ether_qos, exclusive of preamble and retried bytes, in good and bad packets.

Offset

Register	Offset
MAC_TX_OCTET_COUNT_GOOD_BAD	714h

Diagram



Fields

Field	Description
31-0 TXOCTGB	Tx Octet Count Good Bad This field indicates the number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad packets.

43.8.1.182 Tx Packet Count Good and Bad (MAC_TX_PACKET_COUNT_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos, exclusive of retried packets.

Offset

Register	Offset
MAC_TX_PACKET_COUNT_GOOD_BAD	718h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TXPKTGB	Tx Packet Count Good Bad This field indicates the number of good and bad packets transmitted, exclusive of retried packets.

43.8.1.183 Tx Broadcast Packets Good (MAC_TX_BROADCAST_PACKETS_GOOD)

This register provides the number of good broadcast packets transmitted by DWC_ether_qos.

Offset

Register	Offset
MAC_TX_BROADCAST_PACKETS_GOOD	71Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXBCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXBCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TXBCASTG	Tx Broadcast Packets Good This field indicates the number of good broadcast packets transmitted.

43.8.1.184 Tx Multicast Packets Good (MAC_TX_MULTICAST_PACKETS_GOOD)

This register provides the number of good multicast packets transmitted by DWC_ether_qos.

Offset

Register	Offset
MAC_TX_MULTICAST_ PACKETS_GOOD	720h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXMCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXMCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TXMCASTG	Tx Multicast Packets Good This field indicates the number of good multicast packets transmitted.

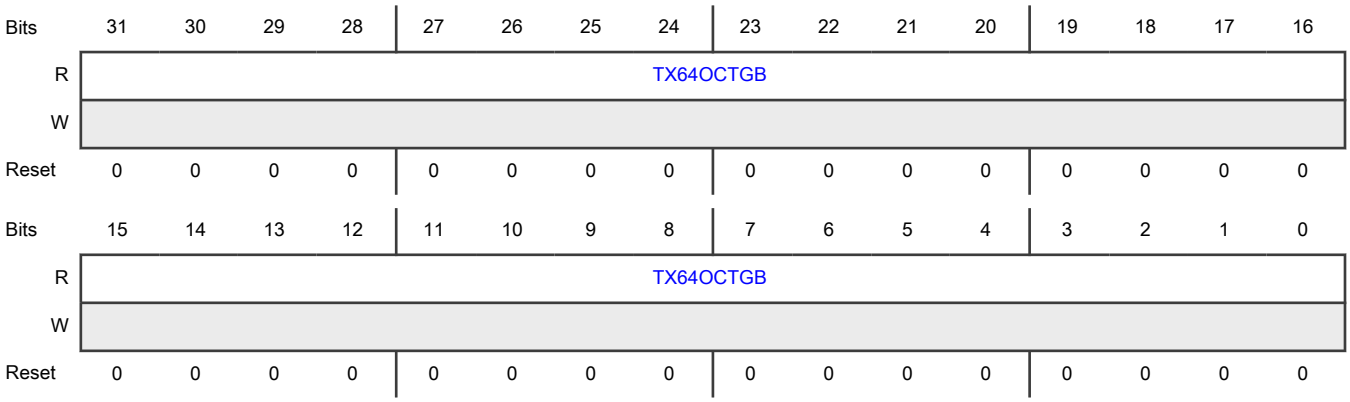
43.8.1.185 Tx Good and Bad 64-Byte Packets (MAC_TX_64OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length 64 bytes, exclusive of preamble and retried packets.

Offset

Register	Offset
MAC_TX_64OCTETS_PACKETS_GOOD_BAD	724h

Diagram



Fields

Field	Description
31-0 TX64OCTGB	Tx 64Octets Packets Good_Bad This field indicates the number of good and bad packets transmitted with length 64 bytes, exclusive of preamble and retried packets.

43.8.1.186 Tx Good and Bad 65 to 127-Byte Packets (MAC_TX_65TO127OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried packets.

Offset

Register	Offset
MAC_TX_65TO127OCTETS_PACKETS_GOOD_BAD	728h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX65_127OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX65_127OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TX65_127OCTGB	Tx 65To127Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried packets.

43.8.1.187 Tx Good and Bad 128 to 255-Byte Packets (MAC_TX_128TO255OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length between 128 to 255 (inclusive) bytes, exclusive of preamble and retried packets.

Offset

Register	Offset
MAC_TX_128TO255OCTETS_PACKETS_GOOD_BAD	72Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX128_255OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX128_255OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TX128_255OCTGB	Tx 128To255Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 128 and 255 (inclusive) bytes, exclusive of preamble and retried packets.

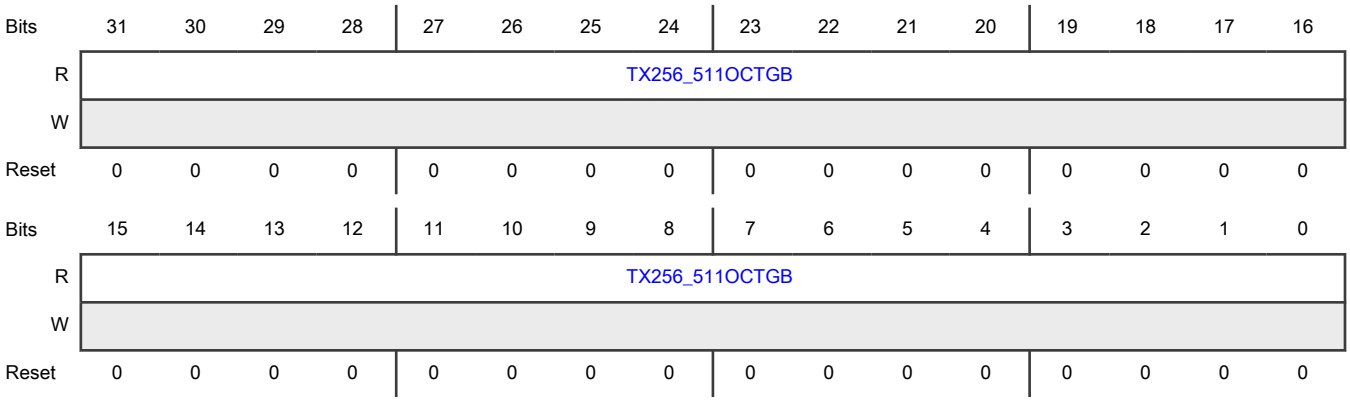
43.8.1.188 Tx Good and Bad 256 to 511-Byte Packets (MAC_TX_256TO511OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length between 256 to 511 (inclusive) bytes, exclusive of preamble and retried packets.

Offset

Register	Offset
MAC_TX_256TO511OCTETS_PACKETS_GOOD_BAD	730h

Diagram



Fields

Field	Description
31-0 TX256_511OCTGB	Tx 256To511Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 256 and 511 (inclusive) bytes, exclusive of preamble and retried packets.

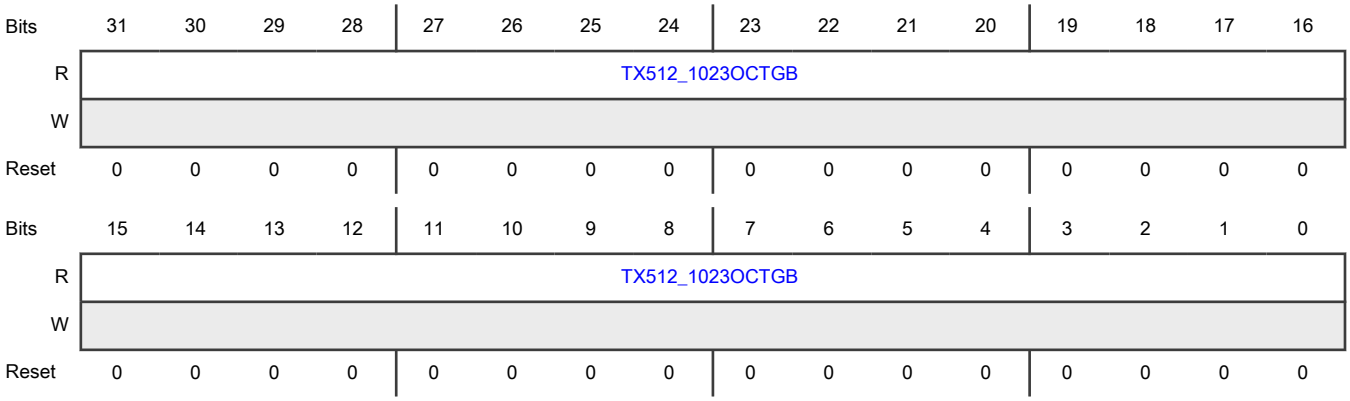
43.8.1.189 Tx Good and Bad 512 to 1023-Byte Packets (MAC_TX_512TO1023OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length 512 to 1023 (inclusive) bytes, exclusive of preamble and retried packets.

Offset

Register	Offset
MAC_TX_512TO1023OCTETS_PACKETS_GOOD_BAD	734h

Diagram



Fields

Field	Description
31-0 TX512_1023OCTGB	Tx 512To1023Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 512 and 1023 (inclusive) bytes, exclusive of preamble and retried packets.

43.8.1.190 Tx Good and Bad 1024 to Max-Byte Packets (MAC_TX_1024TOMAXOCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length 1024 to maxsize (inclusive) bytes, exclusive of preamble and retried packets.

Offset

Register	Offset
MAC_TX_1024TOMAXOCTETS_PACKETS_GOOD_BAD	738h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX1024_MAXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX1024_MAXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TX1024_MAXOCTGB	Tx 1024ToMaxOctets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 1024 and maxsize (inclusive) bytes, exclusive of preamble and retried packets.

43.8.1.191 Good and Bad Unicast Packets Transmitted (MAC_TX_UNICAST_PACKETS_GOOD_BAD)

This register provides the number of good and bad unicast packets transmitted by DWC_ether_qos.

Offset

Register	Offset
MAC_TX_UNICAST_PACKETS_GOOD_BAD	73Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXUCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXUCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TXUCASTGB	Tx Unicast Packets Good Bad This field indicates the number of good and bad unicast packets transmitted.

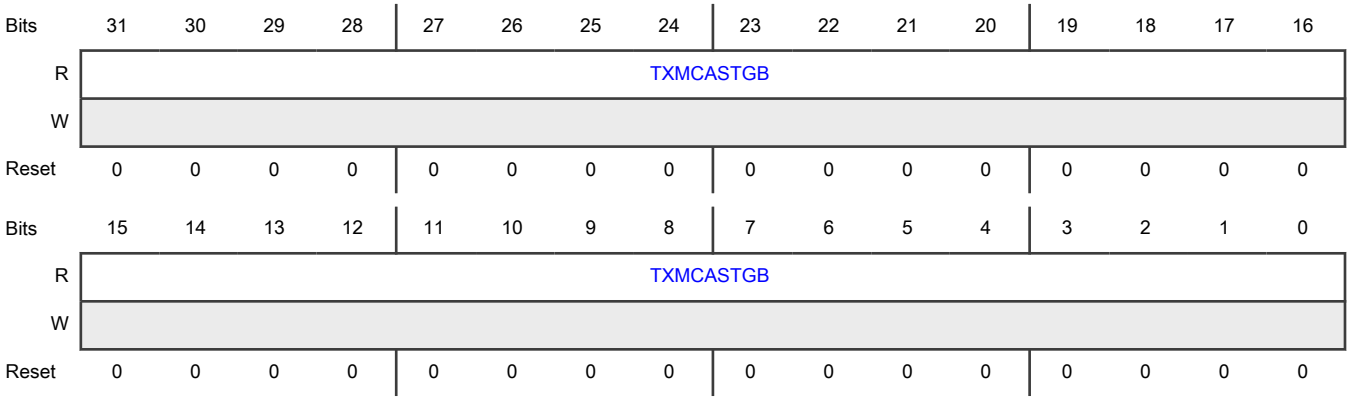
43.8.1.192 Good and Bad Multicast Packets Transmitted (MAC_TX_MULTICAST_PACKETS_GOOD_BAD)

This register provides the number of good and bad multicast packets transmitted by DWC_ether_qos.

Offset

Register	Offset
MAC_TX_MULTICAST_ PACKETS_GOOD_BAD	740h

Diagram



Fields

Field	Description
31-0 TXMCASTGB	Tx Multicast Packets Good Bad This field indicates the number of good and bad multicast packets transmitted.

43.8.1.193 Good and Bad Broadcast Packets Transmitted (MAC_TX_BROADCAST_PACKETS_GOOD_BAD)

This register provides the number of good and bad broadcast packets transmitted by DWC_ether_qos.

Offset

Register	Offset
MAC_TX_BROADCAST_ PACKETS_GOOD_BAD	744h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXBCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXBCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TXBCASTGB	Tx Broadcast Packets Good Bad This field indicates the number of good and bad broadcast packets transmitted.

43.8.1.194 Tx Packets Aborted By Underflow Error (MAC_TX_UNDERFLOW_ERROR_PACKETS)

This register provides the number of packets aborted by DWC_ether_qos because of packets underflow error.

Offset

Register	Offset
MAC_TX_UNDERFLOW_ERROR_PACKETS	748h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXUNDRFLW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXUNDRFLW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TXUNDRFLW	Tx Underflow Error Packets This field indicates the number of packets aborted because of packets underflow error.

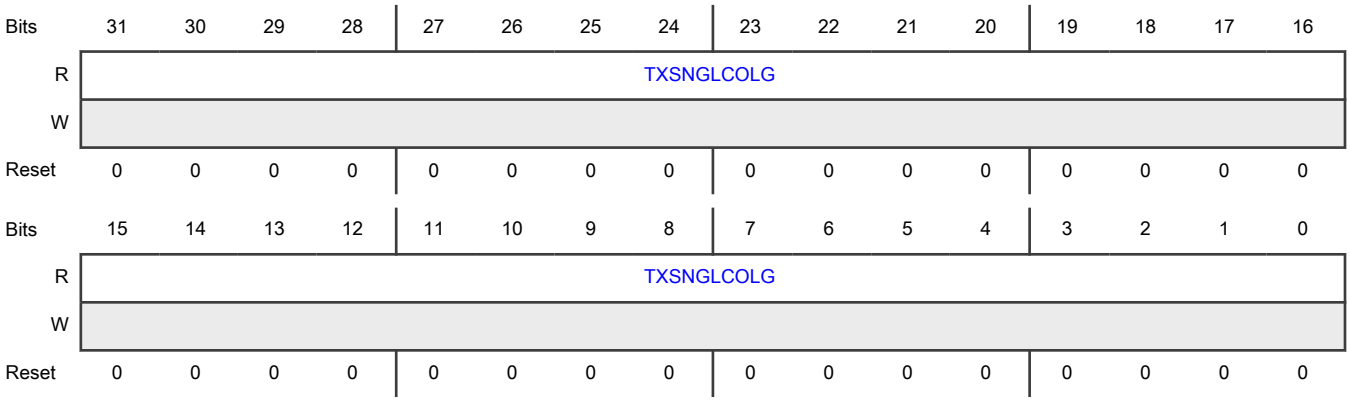
43.8.1.195 Single Collision Good Packets Transmitted (MAC_TX_SINGLE_COLLISION_GOOD_PACKETS)

This register provides the number of successfully transmitted packets by DWC_ether_qos after a single collision in the half-duplex mode.

Offset

Register	Offset
MAC_TX_SINGLE_COLLISION_GOOD_PACKETS	74Ch

Diagram



Fields

Field	Description
31-0 TXSNGLCOLG	Tx Single Collision Good Packets This field indicates the number of successfully transmitted packets after a single collision in the half-duplex mode.

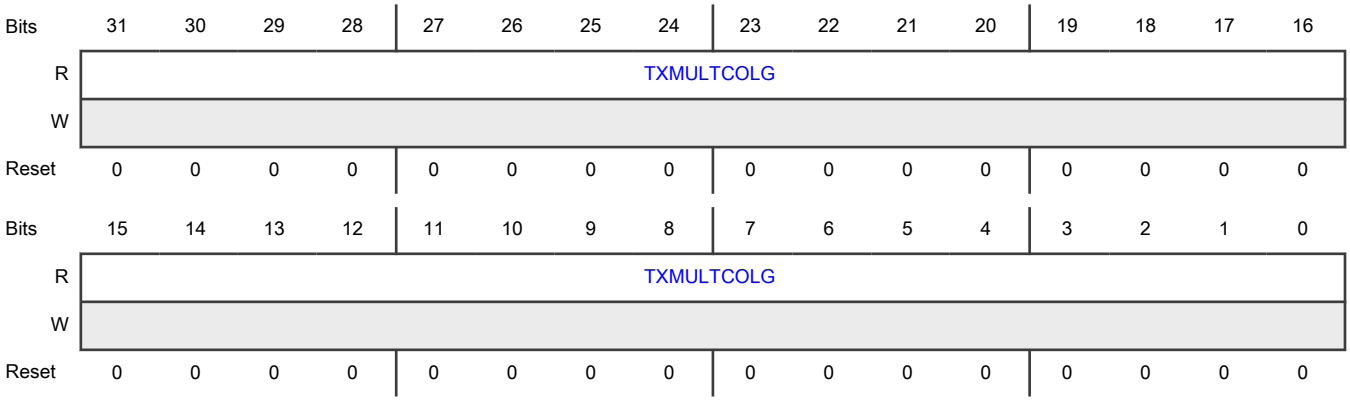
43.8.1.196 Multiple Collision Good Packets Transmitted (MAC_TX_MULTIPLE_COLLISION_GOOD_PACKETS)

This register provides the number of successfully transmitted packets by DWC_ether_qos after multiple collisions in the half-duplex mode.

Offset

Register	Offset
MAC_TX_MULTIPLE_COLLISION_GOOD_PACKETS	750h

Diagram



Fields

Field	Description
31-0 TXMULTCOLG	Tx Multiple Collision Good Packets This field indicates the number of successfully transmitted packets after multiple collisions in the half-duplex mode.

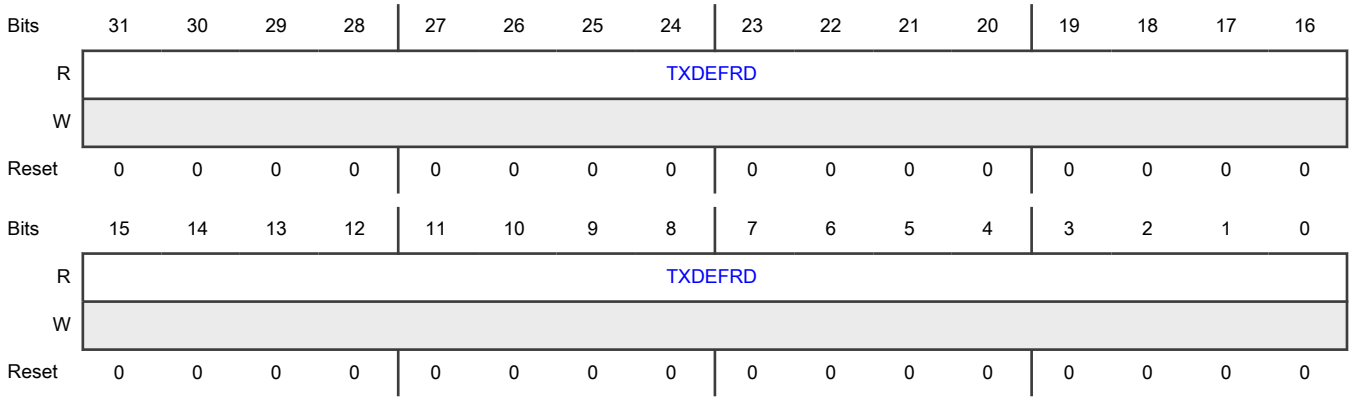
43.8.1.197 Deferred Packets Transmitted (MAC_TX_DEFERRED_PACKETS)

This register provides the number of successfully transmitted by DWC_ether_qos after a deferral in the half-duplex mode.

Offset

Register	Offset
MAC_TX_DEFERRED_PACKETS	754h

Diagram



Fields

Field	Description
31-0 TXDEFRD	Tx Deferred Packets This field indicates the number of successfully transmitted after a deferral in the half-duplex mode.

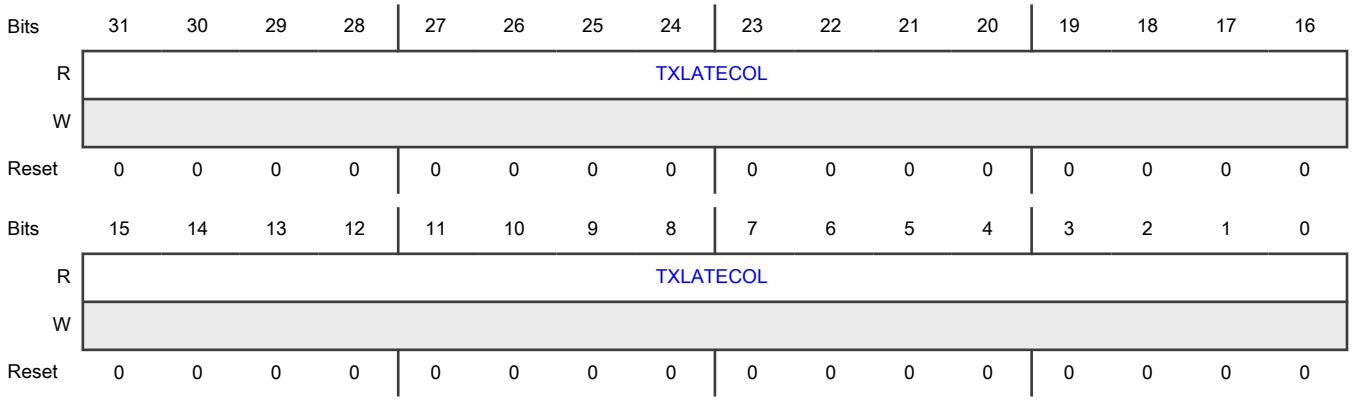
43.8.1.198 Late Collision Packets Transmitted (MAC_TX_LATE_COLLISION_PACKETS)

This register provides the number of packets aborted by DWC_ether_qos because of late collision error.

Offset

Register	Offset
MAC_TX_LATE_COLLISION_PACKETS	758h

Diagram



Fields

Field	Description
31-0 TXLATECOL	Tx Late Collision Packets This field indicates the number of packets aborted because of late collision error.

43.8.1.199 Excessive Collision Packets Transmitted (MAC_TX_EXCESSIVE_COLLISION_PACKETS)

This register provides the number of packets aborted by DWC_ether_qos because of excessive (16) collision errors.

Offset

Register	Offset
MAC_TX_EXCESSIVE_ COLLISION_PACKETS	75Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXEXSCOL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXEXSCOL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TXEXSCOL	Tx Excessive Collision Packets This field indicates the number of packets aborted because of excessive (16) collision errors.

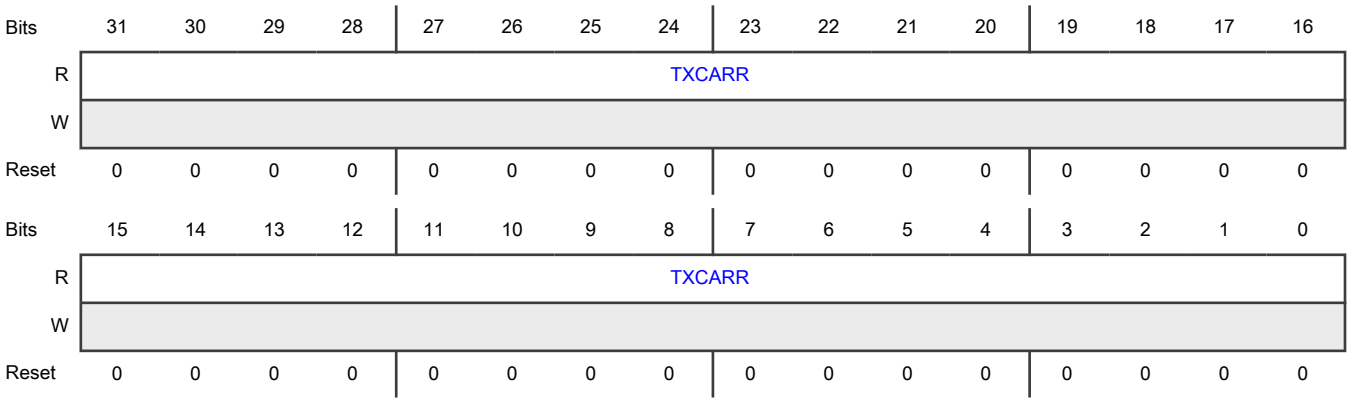
43.8.1.200 Carrier Error Packets Transmitted (MAC_TX_CARRIER_ERROR_PACKETS)

This register provides the number of packets aborted by DWC_ether_qos because of carrier sense error (no carrier or loss of carrier).

Offset

Register	Offset
MAC_TX_CARRIER_ERROR_PACKETS	760h

Diagram



Fields

Field	Description
31-0 TXCARR	Tx Carrier Error Packets This field indicates the number of packets aborted because of carrier sense error (no carrier or loss of carrier).

43.8.1.201 Bytes Transmitted in Good Packets (MAC_TX_OCTET_COUNT_GOOD)

This register provides the number of bytes transmitted by DWC_ether_qos, exclusive of preamble, only in good packets.

Offset

Register	Offset
MAC_TX_OCTET_COUNT_GOOD	764h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXOCTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXOCTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TXOCTG	Tx Octet Count Good This field indicates the number of bytes transmitted, exclusive of preamble, only in good packets.

43.8.1.202 Good Packets Transmitted (MAC_TX_PACKET_COUNT_GOOD)

This register provides the number of good packets transmitted by DWC_ether_qos.

Offset

Register	Offset
MAC_TX_PACKET_COUNT_GOOD	768h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXPKTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TXPKTG	Tx Packet Count Good This field indicates the number of good packets transmitted.

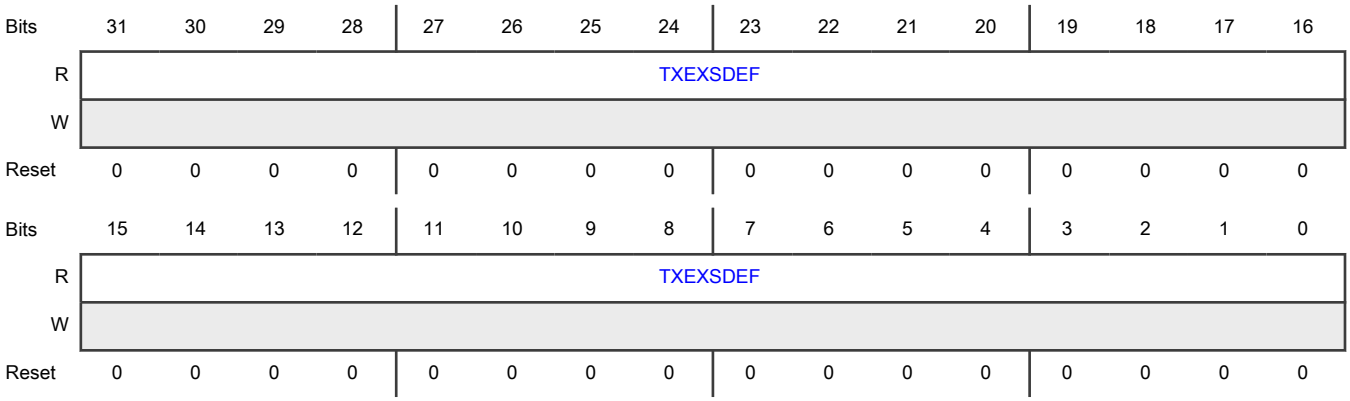
43.8.1.203 Packets Aborted By Excessive Deferral Error (MAC_TX_EXCESSIVE_DEFERRAL_ERROR)

This register provides the number of packets aborted by DWC_ether_qos because of excessive deferral error (deferred for more than two max-sized packet times).

Offset

Register	Offset
MAC_TX_EXCESSIVE_ DEFERRAL_ERROR	76Ch

Diagram



Fields

Field	Description
31-0 TXEXSDEF	Tx Excessive Deferral Error This field indicates the number of packets aborted because of excessive deferral error (deferred for more than two max-sized packet times).

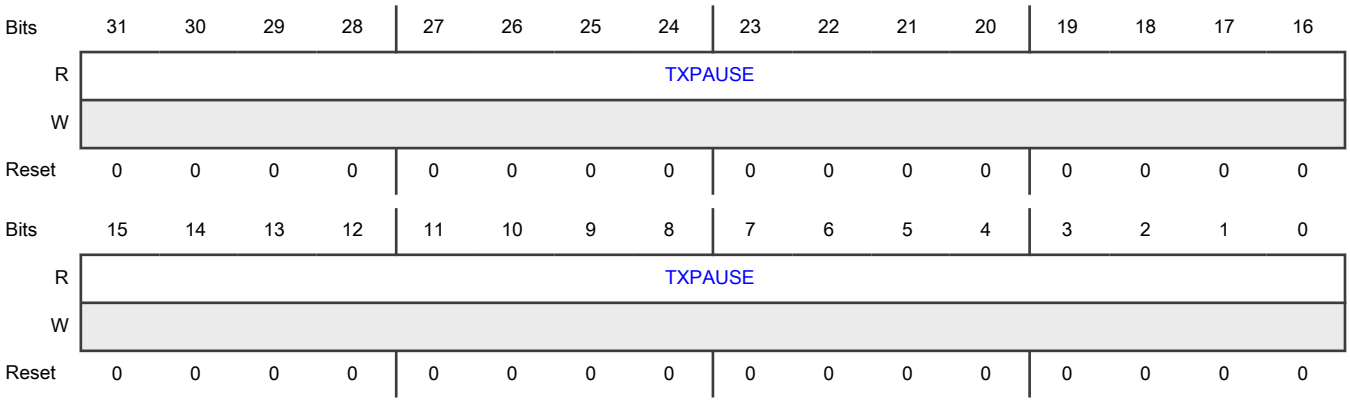
43.8.1.204 Pause Packets Transmitted (MAC_TX_PAUSE_PACKETS)

This register provides the number of good Pause packets transmitted by DWC_ether_qos.

Offset

Register	Offset
MAC_TX_PAUSE_PACKETS	770h

Diagram



Fields

Field	Description
31-0 TXPAUSE	Tx Pause Packets This field indicates the number of good Pause packets transmitted.

43.8.1.205 Good VLAN Packets Transmitted (MAC_TX_VLAN_PACKETS_GOOD)

This register provides the number of good VLAN packets transmitted by DWC_ether_qos.

Offset

Register	Offset
MAC_TX_VLAN_PACKETS_GOOD	774h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXVLANG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXVLANG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TXVLANG	Tx VLAN Packets Good This field provides the number of good VLAN packets transmitted.

43.8.1.206 Good Oversize Packets Transmitted (MAC_TX_OSIZE_PACKETS_GOOD)

This register provides the number of packets transmitted by DWC_ether_qos without errors and with length greater than the maxsize (1,518 or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in S2KP bit of the CONFIGURATION register).

Offset

Register	Offset
MAC_TX_OSIZE_PACKETS_GOOD	778h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXOSIZG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXOSIZG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TXOSIZG	Tx OSize Packets Good This field indicates the number of packets transmitted without errors and with length greater than the maxsize (1,518 or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in S2KP bit of the CONFIGURATION register).

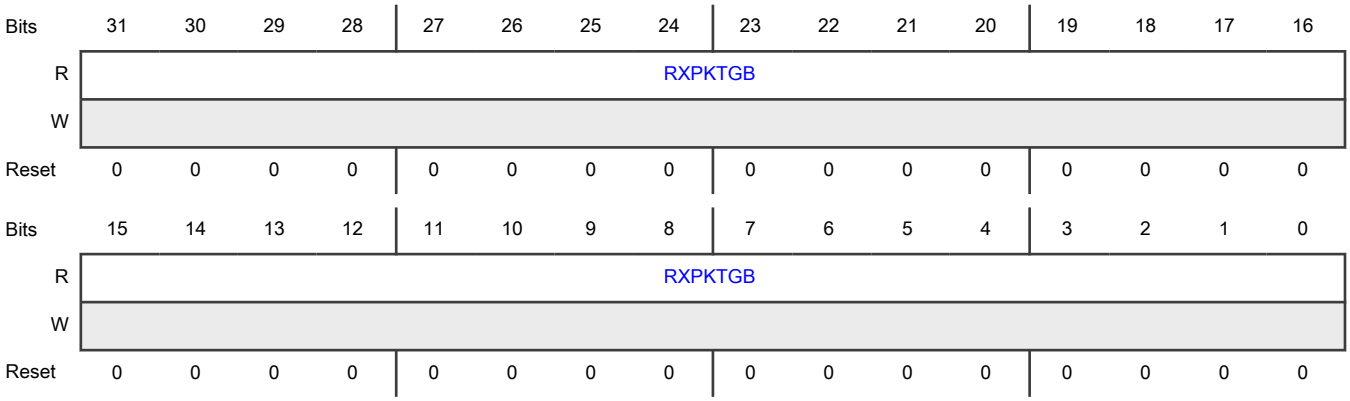
43.8.1.207 Good and Bad Packets Received (MAC_RX_PACKETS_COUNT_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos.

Offset

Register	Offset
MAC_RX_PACKETS_COUNT_GOOD_BAD	780h

Diagram



Fields

Field	Description
31-0 RXPKTGB	Rx Packets Count Good Bad This field indicates the number of good and bad packets received.

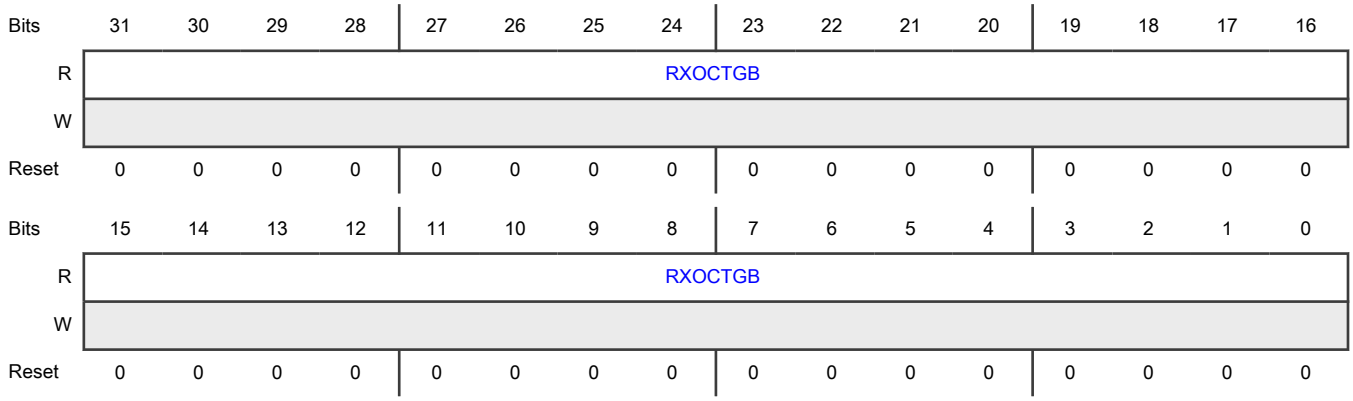
43.8.1.208 Bytes in Good and Bad Packets Received (MAC_RX_OCTET_COUNT_GOOD_BAD)

This register provides the number of bytes received by DWC_ther_qos, exclusive of preamble, in good and bad packets.

Offset

Register	Offset
MAC_RX_OCTET_COUNT_GOOD_BAD	784h

Diagram



Fields

Field	Description
31-0 RXOCTGB	Rx Octet Count Good Bad This field indicates the number of bytes received, exclusive of preamble, in good and bad packets.

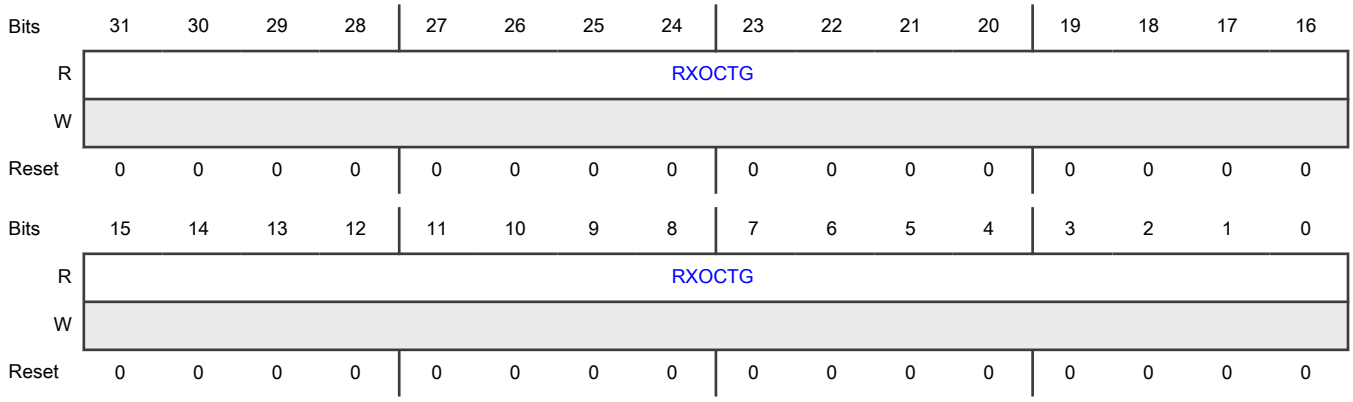
43.8.1.209 Bytes in Good Packets Received (MAC_RX_OCTET_COUNT_GOOD)

This register provides the number of bytes received by DWC_ether_qos, exclusive of preamble, only in good packets.

Offset

Register	Offset
MAC_RX_OCTET_COUNT_GOOD	788h

Diagram



Fields

Field	Description
31-0 RXOCTG	Rx Octet Count Good This field indicates the number of bytes received, exclusive of preamble, only in good packets.

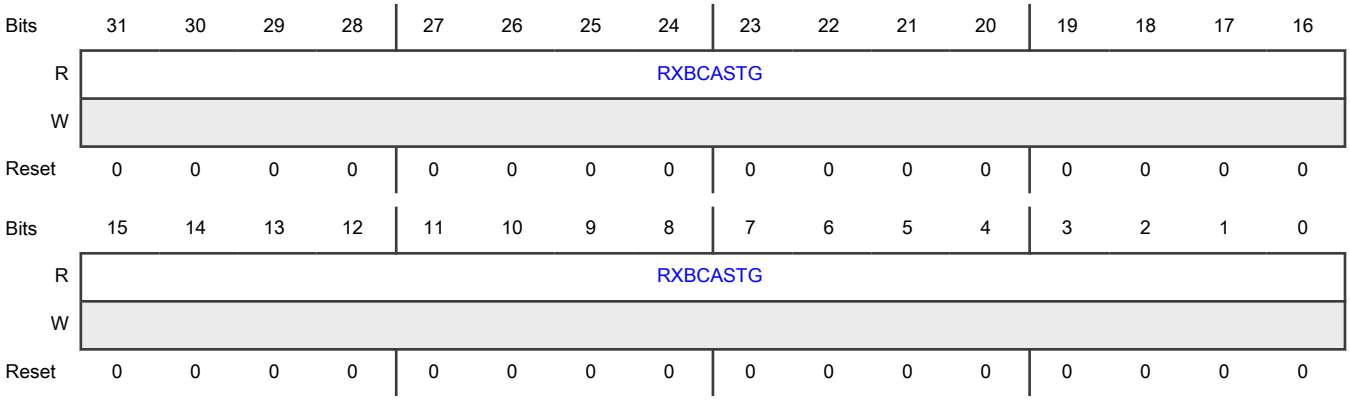
43.8.1.210 Good Broadcast Packets Received (MAC_RX_BROADCAST_PACKETS_GOOD)

This register provides the number of good broadcast packets received by DWC_ether_qos.

Offset

Register	Offset
MAC_RX_BROADCAST_PACKETS_GOOD	78Ch

Diagram



Fields

Field	Description
31-0 RXBCASTG	Rx Broadcast Packets Good This field indicates the number of good broadcast packets received.

43.8.1.211 Good Multicast Packets Received (MAC_RX_MULTICAST_PACKETS_GOOD)

This register provides the number of good multicast packets received by DWC_ether_qos.

Offset

Register	Offset
MAC_RX_MULTICAST_PACKETS_GOOD	790h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXMCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXMCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXMCASTG	Rx Multicast Packets Good This field indicates the number of good multicast packets received.

43.8.1.212 CRC Error Packets Received (MAC_RX_CRC_ERROR_PACKETS)

This register provides the number of packets received by DWC_ether_qos with CRC error.

Offset

Register	Offset
MAC_RX_CRC_ERROR_PACKETS	794h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXCRCERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXCRCERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXCRCERR	Rx CRC Error Packets This field indicates the number of packets received with CRC error.

43.8.1.213 Alignment Error Packets Received (MAC_RX_ALIGNMENT_ERROR_PACKETS)

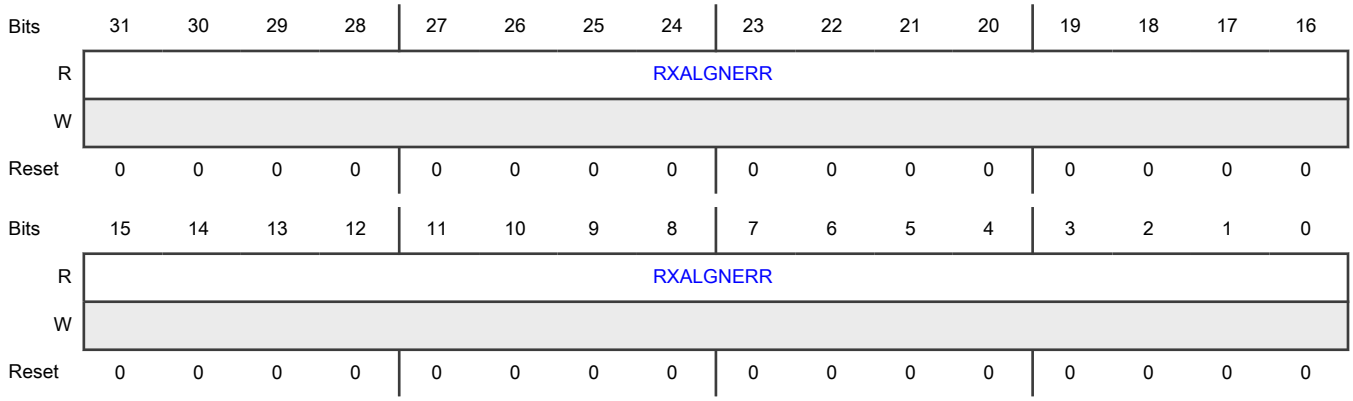
This register provides the number of packets received by DWC_ether_qos with alignment (dribble) error.

This register provides the number of packets received by DWC_ether_qos with alignment (dribble) error. It is valid only in 10/100 mode.

Offset

Register	Offset
MAC_RX_ALIGNMENT_ERROR_PACKETS	798h

Diagram



Fields

Field	Description
31-0 RXALGNERR	Rx Alignment Error Packets This field indicates the number of packets received with alignment (dribble) error. Rx Alignment Error Packets This field indicates the number of packets received with alignment (dribble) error. It is valid only in 10/100 mode.

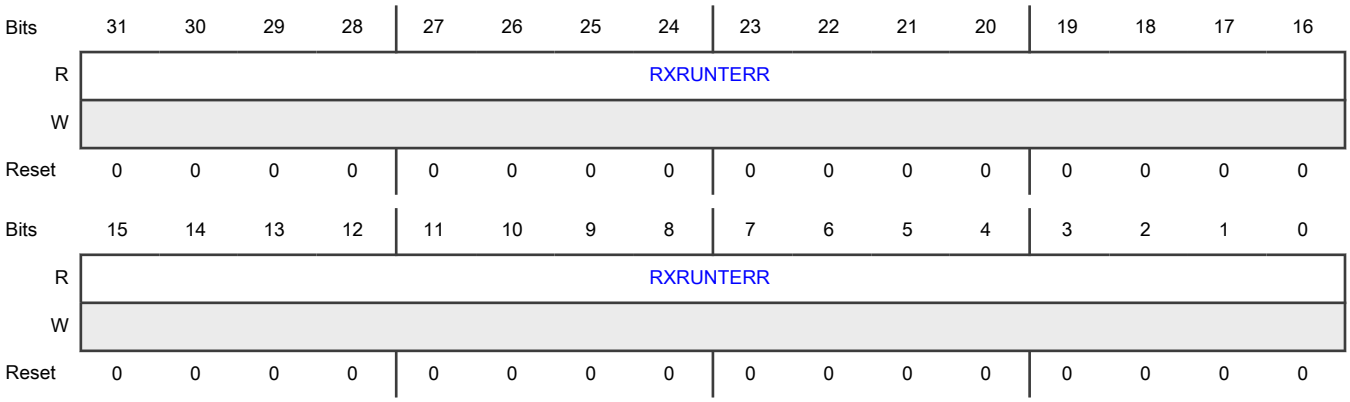
43.8.1.214 Runt Error Packets Received (MAC_RX_RUNT_ERROR_PACKETS)

This register provides the number of packets received by DWC_ether_qos with runt (length less than 64 bytes and CRC error) error.

Offset

Register	Offset
MAC_RX_RUNT_ERROR_PACKETS	79Ch

Diagram



Fields

Field	Description
31-0 RXRUNTERR	Rx Runt Error Packets This field indicates the number of packets received with runt (length less than 64 bytes and CRC error) error.

43.8.1.215 Jabber Error Packets Received (MAC_RX_JABBER_ERROR_PACKETS)

This register provides the number of giant packets received by DWC_ether_qos with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error.

This register provides the number of giant packets received by DWC_ether_qos with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Packet mode is enabled, packets of length greater than 9,018 bytes (9,022 bytes for VLAN tagged) are considered as giant packets.

Offset

Register	Offset
MAC_RX_JABBER_ERROR_PACKETS	7A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXJABERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXJABERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXJABERR	Rx Jabber Error Packets This field indicates the number of giant packets received with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. Rx Jabber Error Packets This field indicates the number of giant packets received with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Packet mode is enabled, packets of length greater than 9,018 bytes (9,022 bytes for VLAN tagged) are considered as giant packets.

43.8.1.216 Good Undersize Packets Received (MAC_RX_UNDERSIZE_PACKETS_GOOD)

This register provides the number of packets received by DWC_ether_qos with length less than 64 bytes, without any errors.

Offset

Register	Offset
MAC_RX_UNDERSIZE_PACKETS_GOOD	7A4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXUNDERSZG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXUNDERSZG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXUNDERSZG	Rx Undersize Packets Good This field indicates the number of packets received with length less than 64 bytes, without any errors.

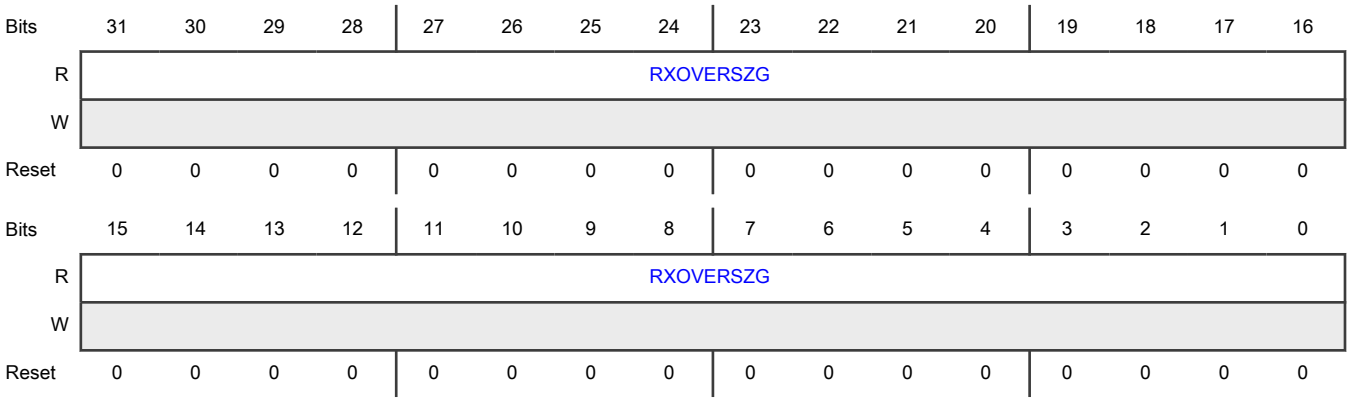
43.8.1.217 Good Oversize Packets Received (MAC_RX_OVERSIZE_PACKETS_GOOD)

This register provides the number of packets received by DWC_ether_qos without errors, with length greater than the maxsize (1,518 bytes or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in the S2KP bit of the MAC_CONFIGURATION register).

Offset

Register	Offset
MAC_RX_OVERSIZE_PACKETS_GOOD	7A8h

Diagram



Fields

Field	Description
31-0 RXOVERSZG	Rx Oversize Packets Good This field indicates the number of packets received without errors, with length greater than the maxsize (1,518 bytes or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in the S2KP bit of the MAC_CONFIGURATION register).

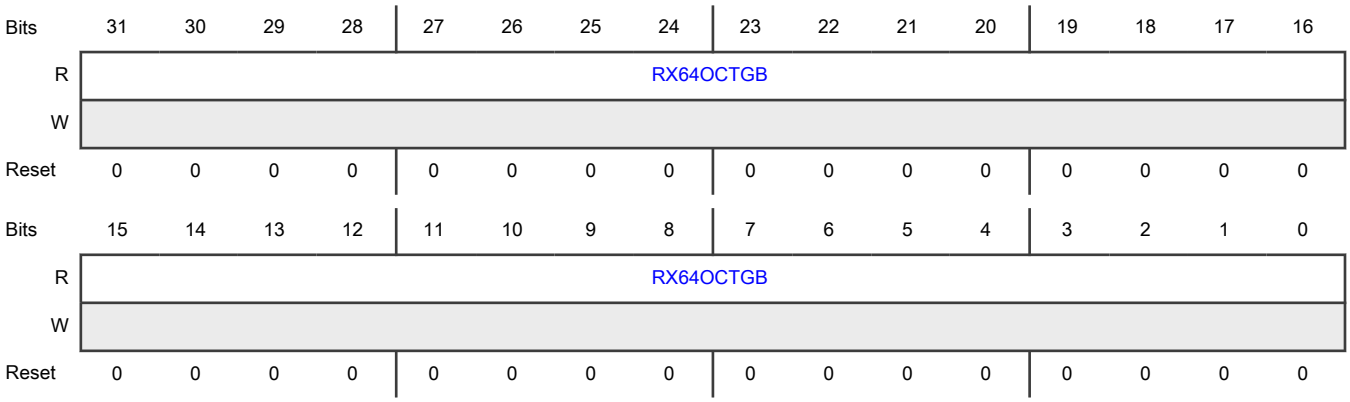
43.8.1.218 Good and Bad 64-Byte Packets Received (MAC_RX_64OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos with length 64 bytes, exclusive of the preamble.

Offset

Register	Offset
MAC_RX_64OCTETS_PACKETS_GOOD_BAD	7ACh

Diagram



Fields

Field	Description
31-0 RX64OCTGB	Rx 64 Octets Packets Good Bad This field indicates the number of good and bad packets received with length 64 bytes, exclusive of the preamble.

43.8.1.219 Good and Bad 64-to-127 Byte Packets Received (MAC_RX_65TO127OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos with length between 65 and 127 (inclusive) bytes, exclusive of the preamble.

Offset

Register	Offset
MAC_RX_65TO127OCTETS_PACKETS_GOOD_BAD	7B0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX65_127OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX65_127OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RX65_127OCTGB	Rx 65-127 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 65 and 127 (inclusive) bytes, exclusive of the preamble.

43.8.1.220 Good and Bad 128-to-255 Byte Packets Received (MAC_RX_128TO255OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos with length between 128 and 255 (inclusive) bytes, exclusive of the preamble.

Offset

Register	Offset
MAC_RX_128TO255OCTETS_PACKETS_GOOD_BAD	7B4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX128_255OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX128_255OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RX128_255OCTGB	Rx 128-255 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 128 and 255 (inclusive) bytes, exclusive of the preamble.

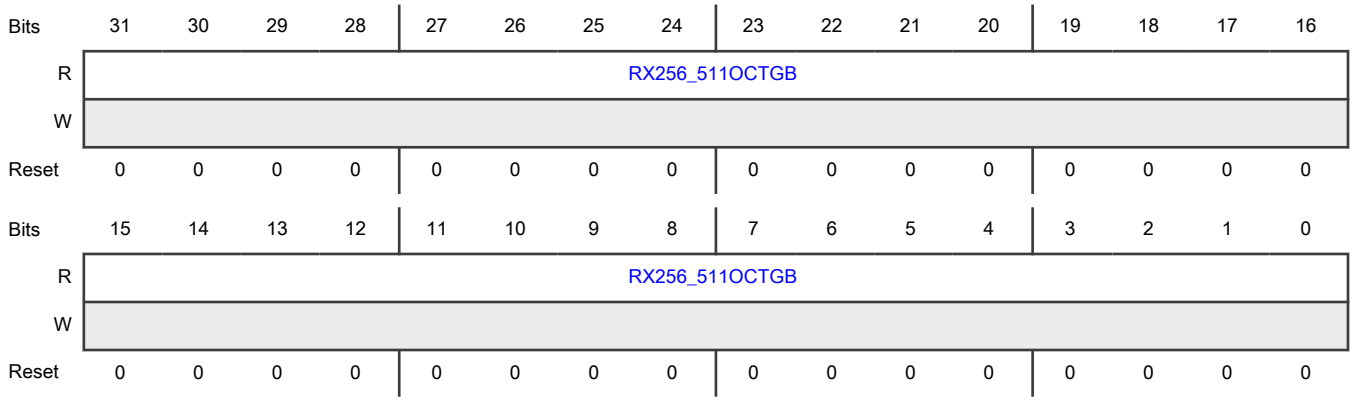
43.8.1.221 Good and Bad 256-to-511 Byte Packets Received (MAC_RX_256TO511OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos with length between 256 and 511 (inclusive) bytes, exclusive of the preamble.

Offset

Register	Offset
MAC_RX_256TO511OCTETS_PACKETS_GOOD_BAD	7B8h

Diagram



Fields

Field	Description
31-0 RX256_511OCTGB	Rx 256-511 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 256 and 511 (inclusive) bytes, exclusive of the preamble.

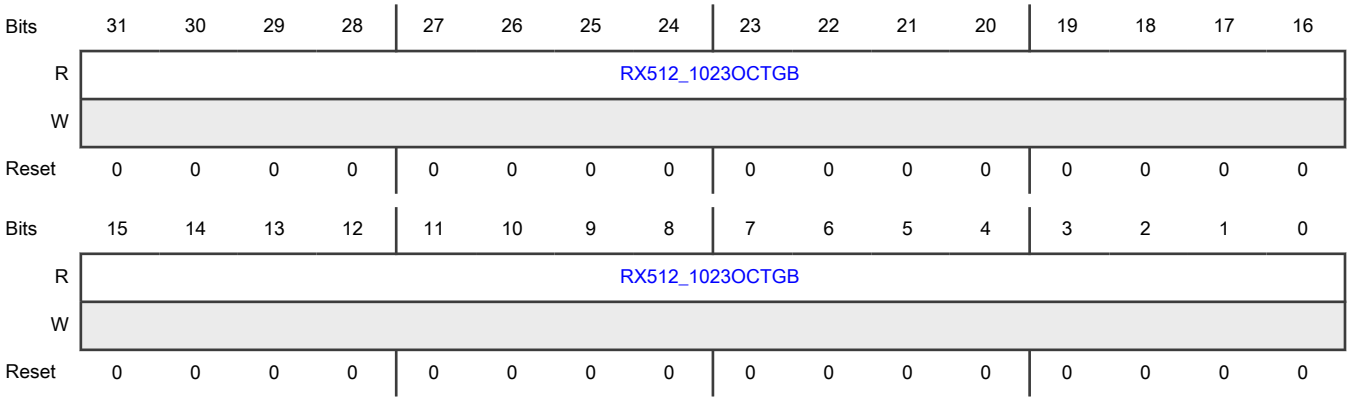
43.8.1.222 Good and Bad 512-to-1023 Byte Packets Received (MAC_RX_512TO1023OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos with length between 512 and 1023 (inclusive) bytes, exclusive of the preamble.

Offset

Register	Offset
MAC_RX_512TO1023OCTETS_PACKETS_GOOD_BAD	7BCh

Diagram



Fields

Field	Description
31-0 RX512_1023OCTGB	RX 512-1023 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 512 and 1023 (inclusive) bytes, exclusive of the preamble.

43.8.1.223 Good and Bad 1024-to-Max Byte Packets Received (MAC_RX_1024TOMAXOCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos with length between 1024 and maxsize (inclusive) bytes, exclusive of the preamble.

Offset

Register	Offset
MAC_RX_1024TOMAXOCTETS_PACKETS_GOOD_BAD	7C0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX1024_MAXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX1024_MAXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RX1024_MAXOCTGB	Rx 1024-Max Octets Good Bad This field indicates the number of good and bad packets received with length between 1024 and maxsize (inclusive) bytes, exclusive of the preamble.

43.8.1.224 Good Unicast Packets Received (MAC_RX_UNICAST_PACKETS_GOOD)

This register provides the number of good unicast packets received by DWC_ether_qos.

Offset

Register	Offset
MAC_RX_UNICAST_PACKETS_GOOD	7C4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXUCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXUCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXUCASTG	Rx Unicast Packets Good This field indicates the number of good unicast packets received.

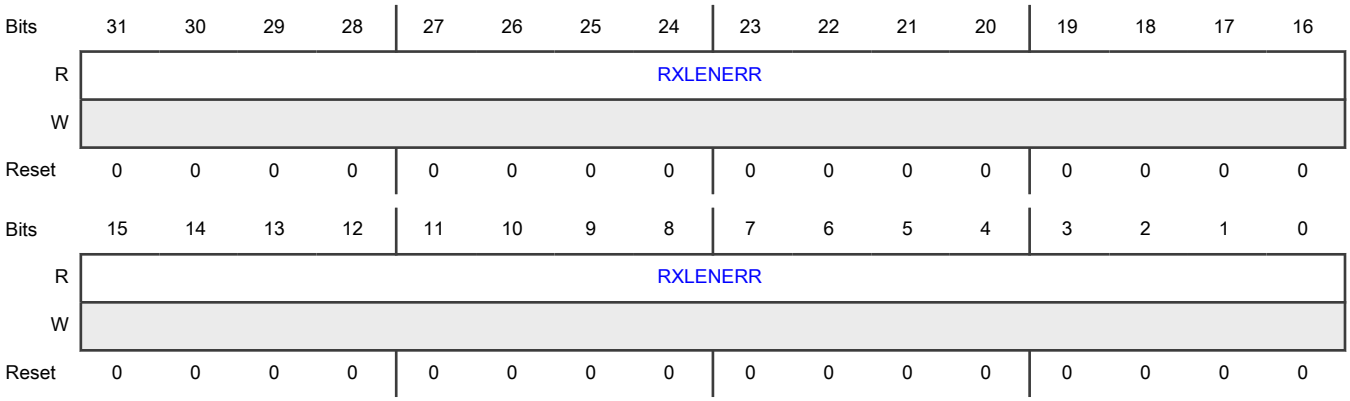
43.8.1.225 Length Error Packets Received (MAC_RX_LENGTH_ERROR_PACKETS)

This register provides the number of packets received by DWC_ether_qos with length error (Length Type field not equal to packet size), for all packets with valid length field.

Offset

Register	Offset
MAC_RX_LENGTH_ERROR_PACKETS	7C8h

Diagram



Fields

Field	Description
31-0 RXLENERR	Rx Length Error Packets This field indicates the number of packets received with length error (Length Type field not equal to packet size), for all packets with valid length field.

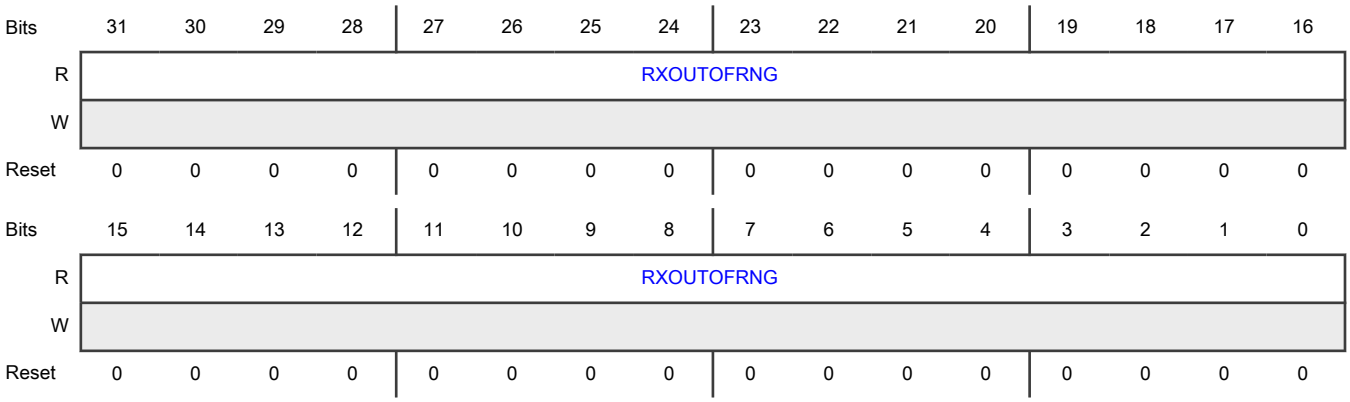
43.8.1.226 Out-of-range Type Packets Received (MAC_RX_OUT_OF_RANGE_TYPE_PACKETS)

This register provides the number of packets received by DWC_ether_qos with length field not equal to the valid packet size (greater than 1,500 but less than 1,536).

Offset

Register	Offset
MAC_RX_OUT_OF_RANGE_TYPE_PACKETS	7CCh

Diagram



Fields

Field	Description
31-0 RXOUTOFRNG	Rx Out of Range Type Packet This field indicates the number of packets received with length field not equal to the valid packet size (greater than 1,500 but less than 1,536).

43.8.1.227 Pause Packets Received (MAC_RX_PAUSE_PACKETS)

This register provides the number of good and valid Pause packets received by DWC_ether_qos.

Offset

Register	Offset
MAC_RX_PAUSE_PACKETS	7D0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXPAUSEPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXPAUSEPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXPAUSEPKT	Rx Pause Packets This field indicates the number of good and valid Pause packets received.

43.8.1.228 Missed Packets Due to FIFO Overflow (MAC_RX_FIFO_OVERFLOW_PACKETS)

This register provides the number of missed received packets because of FIFO overflow in DWC_ether_qos.

Offset

Register	Offset
MAC_RX_FIFO_OVERFLOW_PACKETS	7D4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXFIFOOVFL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXFIFOOVFL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXFIFOOVFL	Rx FIFO Overflow Packets This field indicates the number of missed received packets because of FIFO overflow.

43.8.1.229 Good and Bad VLAN Packets Received (MAC_RX_VLAN_PACKETS_GOOD_BAD)

This register provides the number of good and bad VLAN packets received by DWC_ether_qos.

Offset

Register	Offset
MAC_RX_VLAN_PACKETS_GOOD_BAD	7D8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXVLANPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXVLANPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXVLANPKTGB	Rx VLAN Packets Good Bad This field indicates the number of good and bad VLAN packets received.

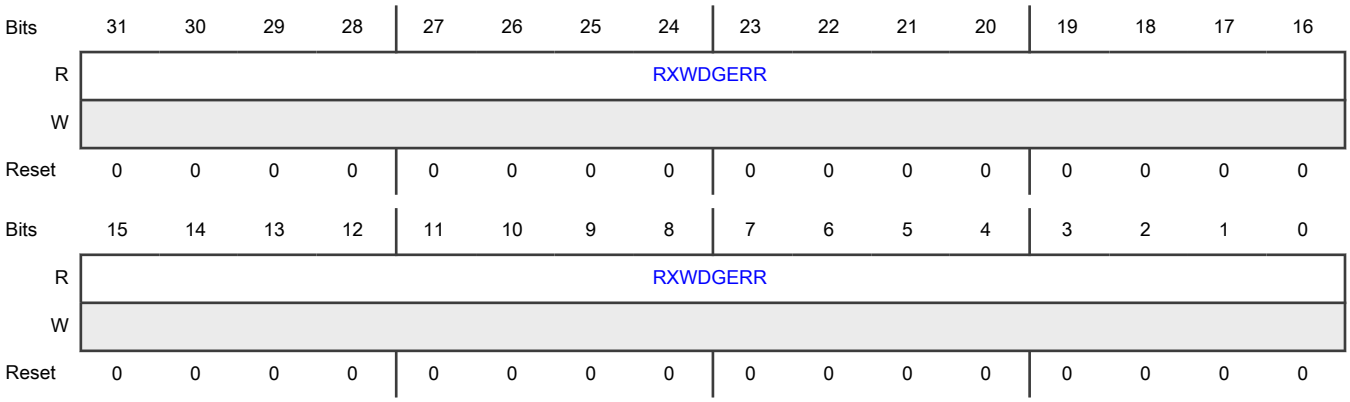
43.8.1.230 Watchdog Error Packets Received (MAC_RX_WATCHDOG_ERROR_PACKETS)

This register provides the number of packets received by DWC_ether_qos with error because of watchdog timeout error (packets with a data load larger than 2,048 bytes (when JE and WD bits are reset in MAC_CONFIGURATION register), 10,240 bytes (when JE bit is set and WD bit is reset in MAC_CONFIGURATION register), 16,384 bytes (when WD bit is set in MAC_CONFIGURATION register) or the value programmed in the MAC_WATCHDOG_TIMEOUT register).

Offset

Register	Offset
MAC_RX_WATCHDOG_ERROR_PACKETS	7DCh

Diagram



Fields

Field	Description
31-0 RXWDGERR	Rx Watchdog Error Packets This field indicates the number of packets received with error because of watchdog timeout error (packets with a data load larger than 2,048 bytes (when JE and WD bits are reset in MAC_CONFIGURATION register), 10,240 bytes (when JE bit is set and WD bit is reset in MAC_CONFIGURATION register), 16,384 bytes (when WD bit is set in MAC_CONFIGURATION register) or the value programmed in the MAC_WATCHDOG_TIMEOUT register).

43.8.1.231 Receive Error Packets Received (MAC_RX_RECEIVE_ERROR_PACKETS)

This register provides the number of packets received by DWC_ether_qos with Receive error or Packet Extension error on the GMII interface.

Offset

Register	Offset
MAC_RX_RECEIVE_ERROR_PACKETS	7E0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXRCVERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXRCVERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXRCVERR	Rx Receive Error Packets This field indicates the number of packets received with Receive error or Packet Extension error on the GMII interface.

43.8.1.232 Good Control Packets Received (MAC_RX_CONTROL_PACKETS_GOOD)

This register provides the number of good control packets received by DWC_ether_qos.

Offset

Register	Offset
MAC_RX_CONTROL_PACKETS_GOOD	7E4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXCTRLG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXCTRLG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXCTRLG	Rx Control Packets Good This field indicates the number of good control packets received.

43.8.1.233 Microseconds Tx LPI Asserted (MAC_TX_LPI_USEC_CNTR)

This register provides the number of microseconds Tx LPI is asserted by DWC_ether_qos.

Offset

Register	Offset
MAC_TX_LPI_USEC_CNTR	7ECh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXLPIUSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXLPIUSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TXLPIUSC	Tx LPI Microseconds Counter This field indicates the number of microseconds Tx LPI is asserted. Tx LPI Microseconds Counter This field indicates the number of microseconds Tx LPI is asserted. For every Tx LPI Entry and Exit, the Timer value can have an error of +/- 1 microsecond.

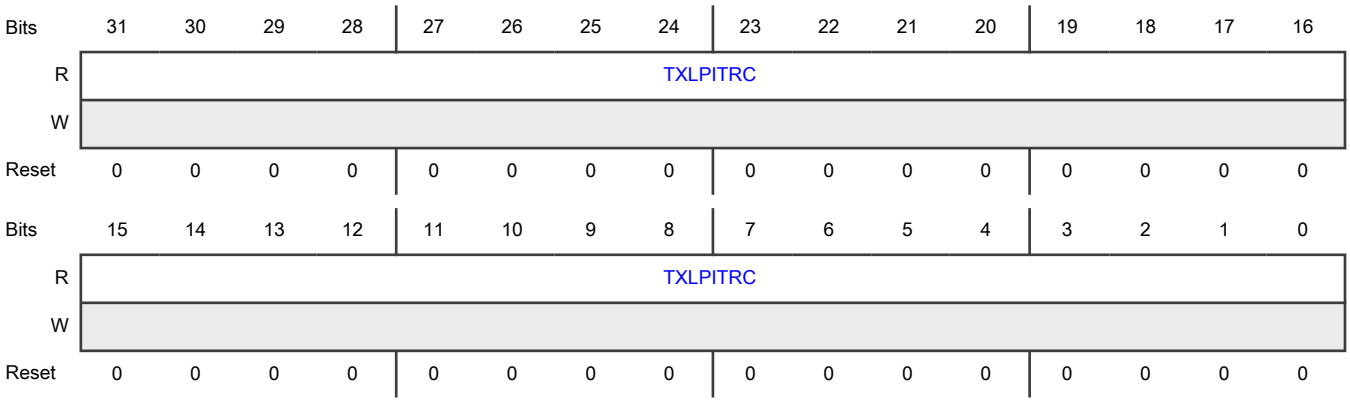
43.8.1.234 Number of Times Tx LPI Asserted (MAC_TX_LPI_TRAN_CNTR)

This register provides the number of times DWC_ether_qos has entered Tx LPI.

Offset

Register	Offset
MAC_TX_LPI_TRAN_CNTR	7F0h

Diagram



Fields

Field	Description
31-0 TXLPITRC	Tx LPI Transition counter This field indicates the number of times Tx LPI Entry has occurred. Tx LPI Transition counter This field indicates the number of times Tx LPI Entry has occurred. Even if Tx LPI Entry occurs in Automate Mode (because of LPITXA bit set in the LPI Control and Status register), the counter will increment.

43.8.1.235 Microseconds Rx LPI Sampled (MAC_RX_LPI_USEC_CNTR)

This register provides the number of microseconds Rx LPI is sampled by DWC_ether_qos.

Offset

Register	Offset
MAC_RX_LPI_USEC_CNTR	7F4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXLPIUSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXLPIUSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXLPIUSC	Rx LPI Microseconds Counter This field indicates the number of microseconds Rx LPI is asserted. Rx LPI Microseconds Counter This field indicates the number of microseconds Rx LPI is asserted. For every Rx LPI Entry and Exit, the Timer value can have an error of +/- 1 microsecond.

43.8.1.236 Number of Times Rx LPI Entered (MAC_RX_LPI_TRAN_CNTR)

This register provides the number of times DWC_ether_qos has entered Rx LPI.

Offset

Register	Offset
MAC_RX_LPI_TRAN_CNTR	7F8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXLPITRC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXLPITRC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXLPITRC	Rx LPI Transition counter This field indicates the number of times Rx LPI Entry has occurred.

43.8.1.237 MMC IPC Receive Interrupt Mask (MAC_MMC_IPC_RX_INTERRUPT_MASK)

This register maintains the mask for the interrupt generated from the receive IPC statistic counters.

The MMC Receive Checksum Off load Interrupt Mask register maintains the masks for the interrupts generated when the receive IPC (Checksum Off load) statistic counters reach half their maximum value, and when they reach their maximum values. This register is 32 bits wide.

Offset

Register	Offset
MAC_MMC_IPC_RX_INTERRUPT_MASK	800h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		RXICMPE...	RXICMPG...	RXTCPER...	RXTCPGO...	RXUDPER...	RXUDPGO...	RXIPV6N...	RXIPV6H...	RXIPV6G...	RXIPV4U...	RXIPV4F...	RXIPV4N...	RXIPV4H...	RXIPV4G...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		RXICMPE...	RXICMPG...	RXTCPER...	RXTCPGP...	RXUDPER...	RXUDPGP...	RXIPV6N...	RXIPV6H...	RXIPV6G...	RXIPV4U...	RXIPV4F...	RXIPV4N...	RXIPV4H...	RXIPV4G...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-30 —	Reserved
29 RXICMPEROIM	MMC Receive ICMP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive ICMP Error Octet Counter Interrupt Mask is disabled 1 - MMC Receive ICMP Error Octet Counter Interrupt Mask is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
28 RXICMPGOIM	MMC Receive ICMP Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_gd_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive ICMP Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive ICMP Good Octet Counter Interrupt Mask is enabled
27 RXTCPEROIM	MMC Receive TCP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive TCP Error Octet Counter Interrupt Mask is disabled 1 - MMC Receive TCP Error Octet Counter Interrupt Mask is enabled
26 RXTCPGOIM	MMC Receive TCP Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_gd_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive TCP Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive TCP Good Octet Counter Interrupt Mask is enabled
25 RXUDPEROIM	MMC Receive UDP Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive UDP Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive UDP Good Octet Counter Interrupt Mask is enabled
24 RXUDPGOIM	MMC Receive IPV6 No Payload Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_gd_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 No Payload Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 No Payload Octet Counter Interrupt Mask is enabled
23 RXIPV6NOPAYOIM	MMC Receive IPV6 Header Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 Header Error Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 Header Error Octet Counter Interrupt Mask is enabled
22 RXIPV6HEROIM	MMC Receive IPV6 Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 Good Octet Counter Interrupt Mask is enabled
21 RXIPV6GOIM	MMC Receive IPV6 Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_gd_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 Good Octet Counter Interrupt Mask is enabled
20 RXIPV4UDSBLOIM	MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_udsbl_octets counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Mask is enabled
19 RXIPV4FRAGOIM	MMC Receive IPV4 Fragmented Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_frag_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 Fragmented Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 Fragmented Octet Counter Interrupt Mask is enabled
18 RXIPV4NOPAYOIM	MMC Receive IPV4 No Payload Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_nopay_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 No Payload Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 No Payload Octet Counter Interrupt Mask is enabled
17 RXIPV4HEROIM	MMC Receive IPV4 Header Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 Header Error Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 Header Error Octet Counter Interrupt Mask is enabled
16 RXIPV4GOIM	MMC Receive IPV4 Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_gd_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 Good Octet Counter Interrupt Mask is enabled
15-14 —	Reserved
13 RXICMPERPIM	MMC Receive ICMP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive ICMP Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive ICMP Error Packet Counter Interrupt Mask is enabled
12 RXICMPGPIM	MMC Receive ICMP Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_gd_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive ICMP Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive ICMP Good Packet Counter Interrupt Mask is enabled
11 RXTCPERPIM	MMC Receive TCP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive TCP Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive TCP Error Packet Counter Interrupt Mask is enabled
10 RXTCPGPIM	MMC Receive TCP Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_gd_pkts counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - MMC Receive TCP Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive TCP Good Packet Counter Interrupt Mask is enabled
9 RXUDPERPIM	MMC Receive UDP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive UDP Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive UDP Error Packet Counter Interrupt Mask is enabled
8 RXUDPGPIM	MMC Receive UDP Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_gd_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive UDP Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive UDP Good Packet Counter Interrupt Mask is enabled
7 RXIPV6NOPAYPIM	MMC Receive IPV6 No Payload Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_nopay_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 No Payload Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 No Payload Packet Counter Interrupt Mask is enabled
6 RXIPV6HERPIM	MMC Receive IPV6 Header Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 Header Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 Header Error Packet Counter Interrupt Mask is enabled
5 RXIPV6GPIM	MMC Receive IPV6 Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 Good Packet Counter Interrupt Mask is enabled
4 RXIPV4UDSBLPIM	MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_udsbl_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Mask is enabled
3 RXIPV4FRAGPIM	MMC Receive IPV4 Fragmented Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_frag_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 Fragmented Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 Fragmented Packet Counter Interrupt Mask is enabled
2 RXIPV4NOPAYPIM	MMC Receive IPV4 No Payload Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_nopay_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 No Payload Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 No Payload Packet Counter Interrupt Mask is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 RXIPV4HERPIM	MMC Receive IPV4 Header Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 Header Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 Header Error Packet Counter Interrupt Mask is enabled
0 RXIPV4GPIM	MMC Receive IPV4 Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 Good Packet Counter Interrupt Mask is enabled

43.8.1.238 MMC IPC Receive Interrupt (MAC_MMC_IPC_RX_INTERRUPT)

This register maintains the interrupt that the receive IPC statistic counters generate.

The MMC Receive Checksum Offload Interrupt register maintains the interrupts generated when receive IPC statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32 bit counter and 0xFFFF for 16 bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones.

The MMC Receive Checksum Offload Interrupt register is 32 bit wide. When the MMC IPC counter that caused the interrupt is read, its corresponding interrupt bit is cleared. The counter's least-significant byte lane (Bits[7:0]) must be read to clear the interrupt bit.

Offset

Register	Offset
MAC_MMC_IPC_RX_INTERRUPT	808h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		RXICMPE...	RXICMPG...	RXTCPER...	RXTCPGO...	RXUDPER...	RXUDPGO...	RXIPV6N...	RXIPV6H...	RXIPV6G...	RXIPV4U...	RXIPV4F...	RXIPV4N...	RXIPV4H...	RXIPV4G...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		RXICMPE...	RXICMPG...	RXTCPER...	RXTCPGP...	RXUDPER...	RXUDPGP...	RXIPV6N...	RXIPV6H...	RXIPV6G...	RXIPV4U...	RXIPV4F...	RXIPV4N...	RXIPV4H...	RXIPV4G...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-30 —	Reserved
29 RXICMPEROIS	MMC Receive ICMP Error Octet Counter Interrupt Status This bit is set when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. MMC Receive ICMP Error Octet Counter Interrupt Status This bit is set when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive ICMP Error Octet Counter Interrupt Status not detected 1 - MMC Receive ICMP Error Octet Counter Interrupt Status detected
28 RXICMPGOIS	MMC Receive ICMP Good Octet Counter Interrupt Status This bit is set when the rxicmp_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive ICMP Good Octet Counter Interrupt Status This bit is set when the rxicmp_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive ICMP Good Octet Counter Interrupt Status not detected 1 - MMC Receive ICMP Good Octet Counter Interrupt Status detected
27 RXTCPEROIS	MMC Receive TCP Error Octet Counter Interrupt Status This bit is set when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. MMC Receive TCP Error Octet Counter Interrupt Status This bit is set when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive TCP Error Octet Counter Interrupt Status not detected 1 - MMC Receive TCP Error Octet Counter Interrupt Status detected
26 RXTCPGOIS	MMC Receive TCP Good Octet Counter Interrupt Status This bit is set when the rxtcp_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive TCP Good Octet Counter Interrupt Status This bit is set when the rxtcp_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive TCP Good Octet Counter Interrupt Status not detected 1 - MMC Receive TCP Good Octet Counter Interrupt Status detected
25 RXUDPEROIS	MMC Receive UDP Error Octet Counter Interrupt Status This bit is set when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. MMC Receive UDP Error Octet Counter Interrupt Status This bit is set when the rxudp_err_octets counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive UDP Error Octet Counter Interrupt Status not detected 1 - MMC Receive UDP Error Octet Counter Interrupt Status detected
24 RXUDPGOIS	MMC Receive UDP Good Octet Counter Interrupt Status This bit is set when the rxudp_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive UDP Good Octet Counter Interrupt Status This bit is set when the rxudp_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive UDP Good Octet Counter Interrupt Status not detected 1 - MMC Receive UDP Good Octet Counter Interrupt Status detected
23 RXIPV6NOPAYOIS	MMC Receive IPV6 No Payload Octet Counter Interrupt Status This bit is set when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 No Payload Octet Counter Interrupt Status This bit is set when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV6 No Payload Octet Counter Interrupt Status not detected 1 - MMC Receive IPV6 No Payload Octet Counter Interrupt Status detected
22 RXIPV6HEROIS	MMC Receive IPV6 Header Error Octet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 Header Error Octet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV6 Header Error Octet Counter Interrupt Status not detected 1 - MMC Receive IPV6 Header Error Octet Counter Interrupt Status detected
21 RXIPV6GOIS	MMC Receive IPV6 Good Octet Counter Interrupt Status This bit is set when the rxipv6_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 Good Octet Counter Interrupt Status This bit is set when the rxipv6_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV6 Good Octet Counter Interrupt Status not detected 1 - MMC Receive IPV6 Good Octet Counter Interrupt Status detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
20 RXIPV4UDSBLOIS	MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status This bit is set when the rxipv4_udsbl_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status This bit is set when the rxipv4_udsbl_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status not detected 1 - MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status detected
19 RXIPV4FRAGOIS	MMC Receive IPV4 Fragmented Octet Counter Interrupt Status This bit is set when the rxipv4_frag_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Fragmented Octet Counter Interrupt Status This bit is set when the rxipv4_frag_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 Fragmented Octet Counter Interrupt Status not detected 1 - MMC Receive IPV4 Fragmented Octet Counter Interrupt Status detected
18 RXIPV4NOPAYOIS	MMC Receive IPV4 No Payload Octet Counter Interrupt Status This bit is set when the rxipv4_nopay_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 No Payload Octet Counter Interrupt Status This bit is set when the rxipv4_nopay_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 No Payload Octet Counter Interrupt Status not detected 1 - MMC Receive IPV4 No Payload Octet Counter Interrupt Status detected
17 RXIPV4HEROIS	MMC Receive IPV4 Header Error Octet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Header Error Octet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 Header Error Octet Counter Interrupt Status not detected 1 - MMC Receive IPV4 Header Error Octet Counter Interrupt Status detected
16 RXIPV4GOIS	MMC Receive IPV4 Good Octet Counter Interrupt Status This bit is set when the rxipv4_gd_octets counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	MMC Receive IPV4 Good Octet Counter Interrupt Status This bit is set when the rxipv4_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 Good Octet Counter Interrupt Status not detected 1 - MMC Receive IPV4 Good Octet Counter Interrupt Status detected
15-14 —	Reserved
13 RXICMPERPIS	MMC Receive ICMP Error Packet Counter Interrupt Status This bit is set when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. MMC Receive ICMP Error Packet Counter Interrupt Status This bit is set when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive ICMP Error Packet Counter Interrupt Status not detected 1 - MMC Receive ICMP Error Packet Counter Interrupt Status detected
12 RXICMPGPIS	MMC Receive ICMP Good Packet Counter Interrupt Status This bit is set when the rxicmp_gd_pkts counter reaches half of the maximum value or the maximum value. MMC Receive ICMP Good Packet Counter Interrupt Status This bit is set when the rxicmp_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive ICMP Good Packet Counter Interrupt Status not detected 1 - MMC Receive ICMP Good Packet Counter Interrupt Status detected
11 RXTCPERPIS	MMC Receive TCP Error Packet Counter Interrupt Status This bit is set when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. MMC Receive TCP Error Packet Counter Interrupt Status This bit is set when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive TCP Error Packet Counter Interrupt Status not detected 1 - MMC Receive TCP Error Packet Counter Interrupt Status detected
10 RXTCPGPIS	MMC Receive TCP Good Packet Counter Interrupt Status This bit is set when the rxtcp_gd_pkts counter reaches half of the maximum value or the maximum value. MMC Receive TCP Good Packet Counter Interrupt Status This bit is set when the rxtcp_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - MMC Receive TCP Good Packet Counter Interrupt Status not detected 1 - MMC Receive TCP Good Packet Counter Interrupt Status detected
9 RXUDPERPIS	MMC Receive UDP Error Packet Counter Interrupt Status This bit is set when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. MMC Receive UDP Error Packet Counter Interrupt Status This bit is set when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive UDP Error Packet Counter Interrupt Status not detected 1 - MMC Receive UDP Error Packet Counter Interrupt Status detected
8 RXUDPGPIS	MC Receive UDP Good Packet Counter Interrupt Status This bit is set when the rxudp_gd_pkts counter reaches half of the maximum value or the maximum value. MC Receive UDP Good Packet Counter Interrupt Status This bit is set when the rxudp_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive UDP Good Packet Counter Interrupt Status not detected 1 - MMC Receive UDP Good Packet Counter Interrupt Status detected
7 RXIPV6NOPAYPIS	MMC Receive IPV6 No Payload Packet Counter Interrupt Status This bit is set when the rxipv6_nopay_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 No Payload Packet Counter Interrupt Status This bit is set when the rxipv6_nopay_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV6 No Payload Packet Counter Interrupt Status not detected 1 - MMC Receive IPV6 No Payload Packet Counter Interrupt Status detected
6 RXIPV6HERPIS	MMC Receive IPV6 Header Error Packet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 Header Error Packet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV6 Header Error Packet Counter Interrupt Status not detected 1 - MMC Receive IPV6 Header Error Packet Counter Interrupt Status detected
5 RXIPV6GPIS	MMC Receive IPV6 Good Packet Counter Interrupt Status This bit is set when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	MMC Receive IPV6 Good Packet Counter Interrupt Status This bit is set when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV6 Good Packet Counter Interrupt Status not detected 1 - MMC Receive IPV6 Good Packet Counter Interrupt Status detected
4 RXIPV4UDSBLPIS	MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Status This bit is set when the rxipv4_udsbl_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Status not detected 1 - MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Status detected
3 RXIPV4FRAGPIS	MMC Receive IPV4 Fragmented Packet Counter Interrupt Status This bit is set when the rxipv4_frag_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Fragmented Packet Counter Interrupt Status This bit is set when the rxipv4_frag_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 Fragmented Packet Counter Interrupt Status not detected 1 - MMC Receive IPV4 Fragmented Packet Counter Interrupt Status detected
2 RXIPV4NOPAYPIS	MMC Receive IPV4 No Payload Packet Counter Interrupt Status This bit is set when the rxipv4_nopay_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 No Payload Packet Counter Interrupt Status This bit is set when the rxipv4_nopay_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 No Payload Packet Counter Interrupt Status not detected 1 - MMC Receive IPV4 No Payload Packet Counter Interrupt Status detected
1 RXIPV4HERPIS	MMC Receive IPV4 Header Error Packet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Header Error Packet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 Header Error Packet Counter Interrupt Status not detected 1 - MMC Receive IPV4 Header Error Packet Counter Interrupt Status detected
0 RXIPV4GPIS	MMC Receive IPV4 Good Packet Counter Interrupt Status This bit is set when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	MMC Receive IPV4 Good Packet Counter Interrupt Status This bit is set when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 Good Packet Counter Interrupt Status not detected 1 - MMC Receive IPV4 Good Packet Counter Interrupt Status detected

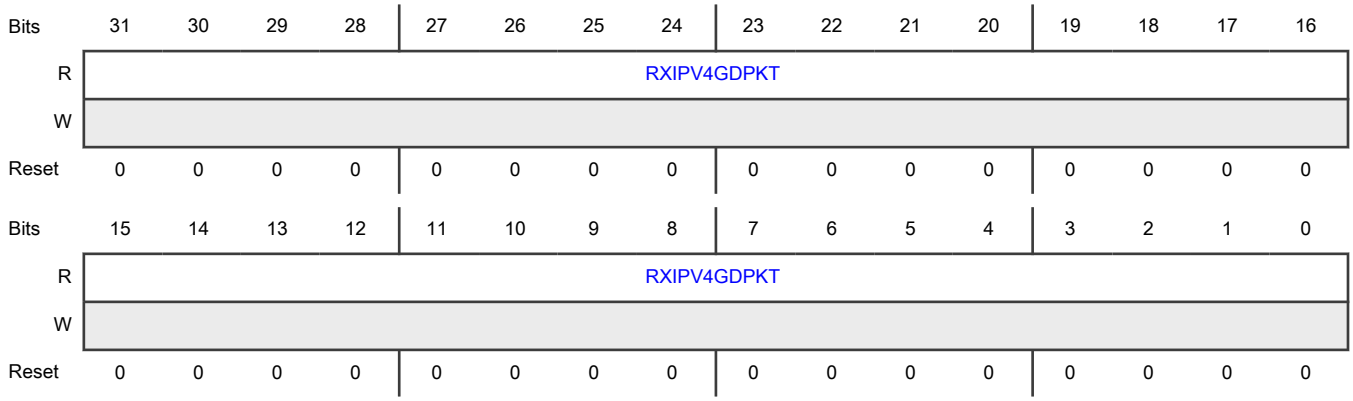
43.8.1.239 Good IPv4 Datagrams Received (MAC_RXIPV4_GOOD_PACKETS)

This register provides the number of good IPv4 datagrams received by DWC_ether_qos with the TCP, UDP, or ICMP payload.

Offset

Register	Offset
MAC_RXIPV4_GOOD_PACKETS	810h

Diagram



Fields

Field	Description
31-0 RXIPV4GDPKT	RxIPv4 Good Packets This field indicates the number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload.

43.8.1.240 IPv4 Datagrams Received with Header Errors (MAC_RXIPV4_HEADER_ERROR_PACKETS)

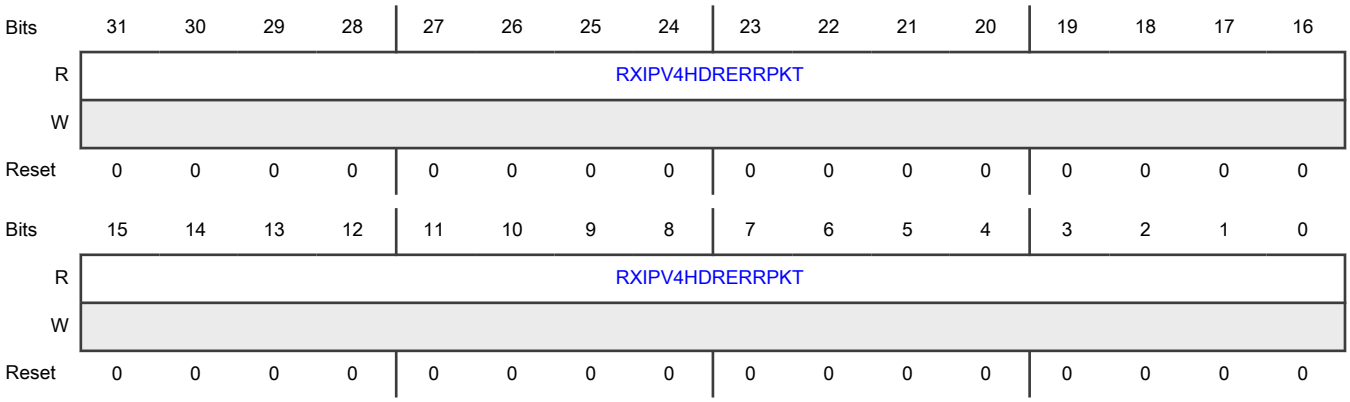
RxIPv4 Header Error Packets

This register provides the number of IPv4 datagrams received by DWC_ether_qos with header (checksum, length, or version mismatch) errors.

Offset

Register	Offset
MAC_RXIPV4_HEADER_ERROR_PACKETS	814h

Diagram



Fields

Field	Description
31-0 RXIPV4HDRERRPKT	RxIPv4 Header Error Packets This field indicates the number of IPv4 datagrams received with header (checksum, length, or version mismatch) errors.

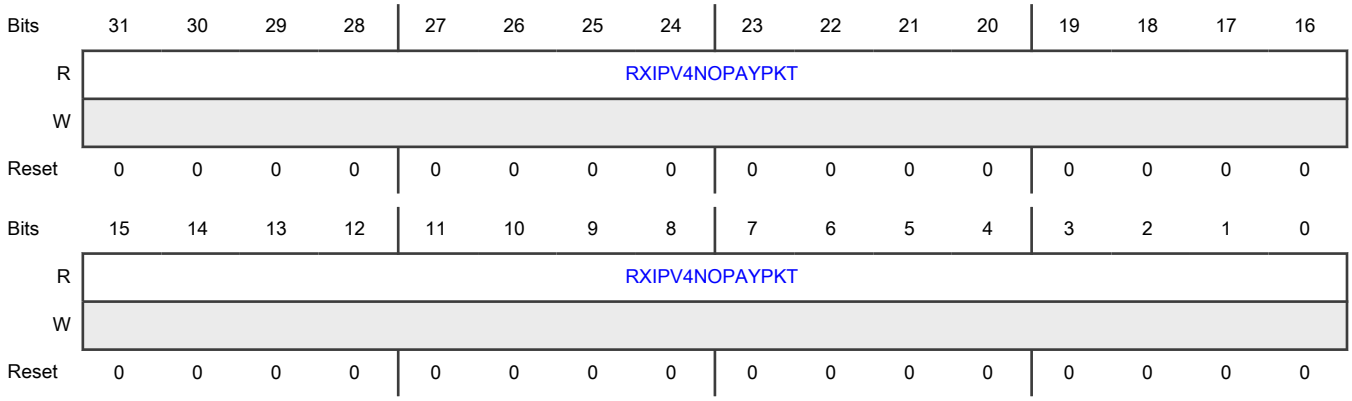
43.8.1.241 IPv4 Datagrams Received with No Payload (MAC_RXIPV4_NO_PAYLOAD_PACKETS)

This register provides the number of IPv4 datagram packets received by DWC_ether_qos that did not have a TCP, UDP, or ICMP payload.

Offset

Register	Offset
MAC_RXIPV4_NO_PAYLOAD_PACKETS	818h

Diagram



Fields

Field	Description
31-0 RXIPV4NOPAYPKT	RxIPv4 Payload Packets This field indicates the number of IPv4 datagram packets received that did not have a TCP, UDP, or ICMP payload.

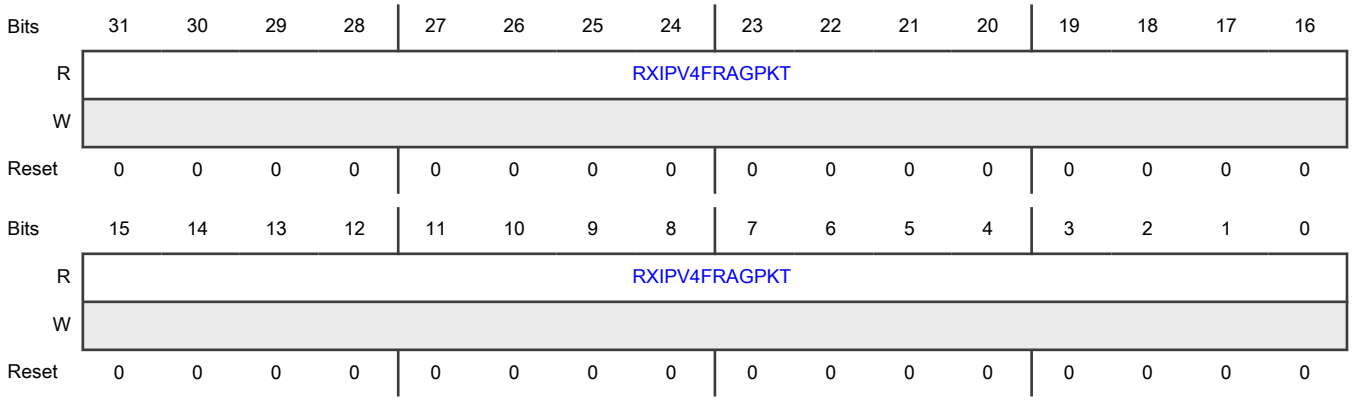
43.8.1.242 IPv4 Datagrams Received with Fragmentation (MAC_RXIPV4_FRAGMENTED_PACKETS)

This register provides the number of good IPv4 datagrams received by DWC_ether_qos with fragmentation.

Offset

Register	Offset
MAC_RXIPV4_FRAGMENTED_PACKETS	81Ch

Diagram



Fields

Field	Description
31-0 RXIPV4FRAGPKT	RxIPv4 Fragmented Packets This field indicates the number of good IPv4 datagrams received with fragmentation.

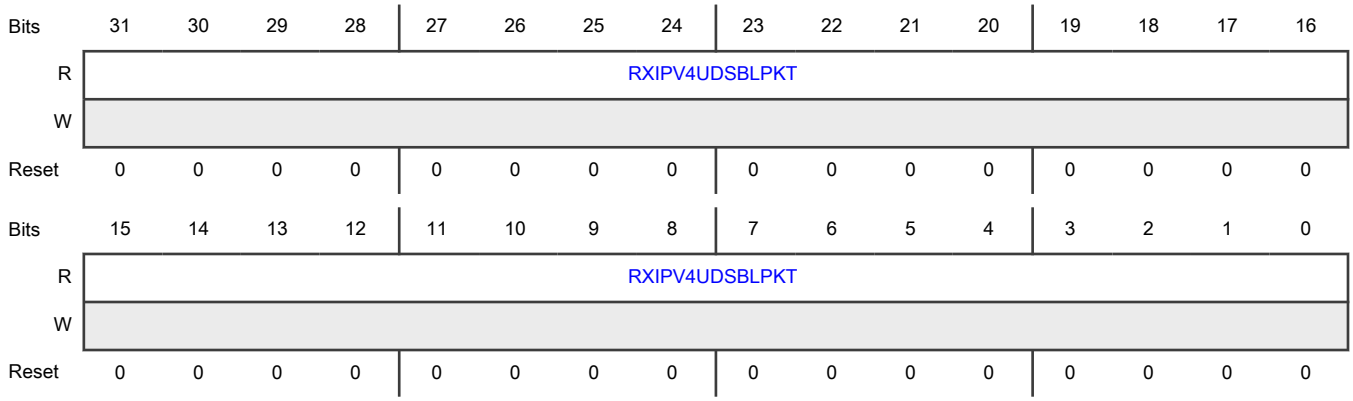
43.8.1.243 IPv4 Datagrams Received with UDP Checksum Disabled (MAC_RXIPV4_UDP_CHECKSUM_DISABLED_PACKETS)

This register provides the number of good IPv4 datagrams received by DWC_ether_qos that had a UDP payload with checksum disabled.

Offset

Register	Offset
MAC_RXIPV4_UDP_CHECKSUM_DISABLED_PACKETS	820h

Diagram



Fields

Field	Description
31-0 RXIPV4UDSBLPKT	RxIPv4 UDP Checksum Disabled Packets This field indicates the number of good IPv4 datagrams received that had a UDP payload with checksum disabled.

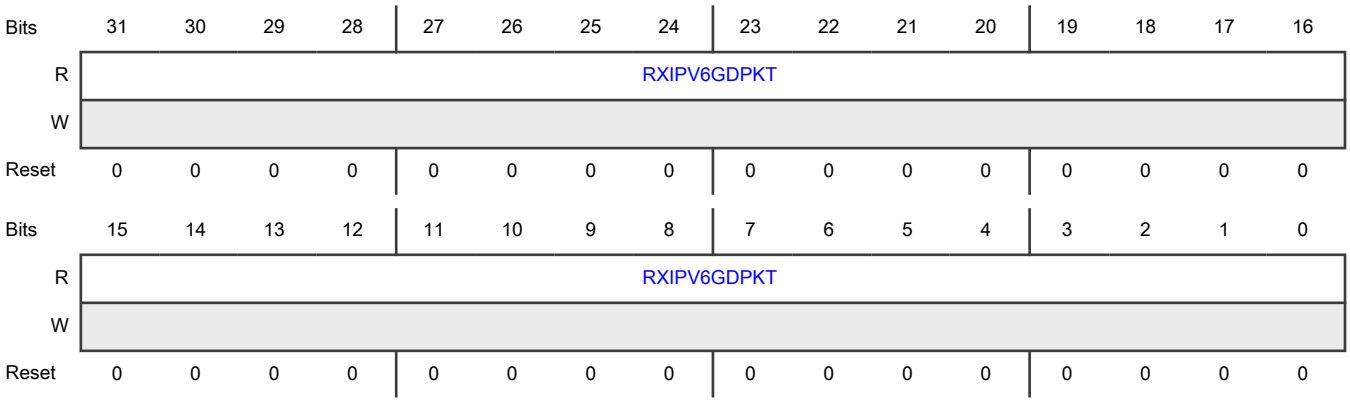
43.8.1.244 Good IPv6 Datagrams Received (MAC_RXIPV6_GOOD_PACKETS)

This register provides the number of good IPv6 datagrams received by DWC_ether_qos with the TCP, UDP, or ICMP payload.

Offset

Register	Offset
MAC_RXIPV6_GOOD_PACKETS	824h

Diagram



Fields

Field	Description
31-0 RXIPV6GDPKT	RxIPv6 Good Packets This field indicates the number of good IPv6 datagrams received with the TCP, UDP, or ICMP payload.

43.8.1.245 IPv6 Datagrams Received with Header Errors (MAC_RXIPV6_HEADER_ERROR_PACKETS)

This register provides the number of IPv6 datagrams received by DWC_ether_qos with header (length or version mismatch) errors.

Offset

Register	Offset
MAC_RXIPV6_HEADER_ERROR_PACKETS	828h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV6HDRERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV6HDRERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXIPV6HDRERRPKT	RxIPv6 Header Error Packets This field indicates the number of IPv6 datagrams received with header (length or version mismatch) errors.

43.8.1.246 IPv6 Datagrams Received with No Payload (MAC_RXIPV6_NO_PAYLOAD_PACKETS)

This register provides the number of IPv6 datagram packets received by DWC_ether_qos that did not have a TCP, UDP, or ICMP payload.

This register provides the number of IPv6 datagram packets received by DWC_ether_qos that did not have a TCP, UDP, or ICMP payload. This includes all IPv6 datagrams with fragmentation or security extension headers.

Offset

Register	Offset
MAC_RXIPV6_NO_PAYLOAD_PACKETS	82Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV6NOPAYPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV6NOPAYPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXIPV6NOPAYPKT	RxIPv6 Payload Packets This field indicates the number of IPv6 datagram packets received that did not have a TCP, UDP, or ICMP payload. RxIPv6 Payload Packets This field indicates the number of IPv6 datagram packets received that did not have a TCP, UDP, or ICMP payload. This includes all IPv6 datagrams with fragmentation or security extension headers.

43.8.1.247 IPv6 Datagrams Received with Good UDP (MAC_RXUDP_GOOD_PACKETS)

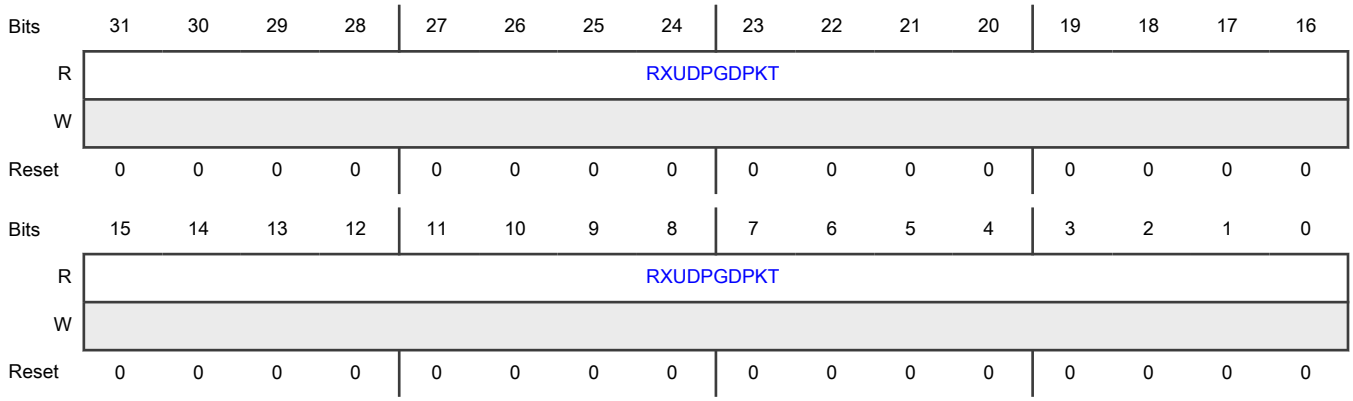
This register provides the number of good IP datagrams received by DWC_ether_qos with a good UDP payload.

This register provides the number of good IP datagrams received by DWC_ether_qos with a good UDP payload. This counter is not updated when the RxIPv4_UDP_Checksum_Disabled_Packets counter is incremented.

Offset

Register	Offset
MAC_RXUDP_GOOD_PACKETS	830h

Diagram



Fields

Field	Description
31-0 RXUDPGDPKT	RxUDP Good Packets This field indicates the number of good IP datagrams received with a good UDP payload. RxUDP Good Packets This field indicates the number of good IP datagrams received with a good UDP payload. This counter is not updated when the RxIPv4_UDP_Checksum_Disabled_Packets counter is incremented.

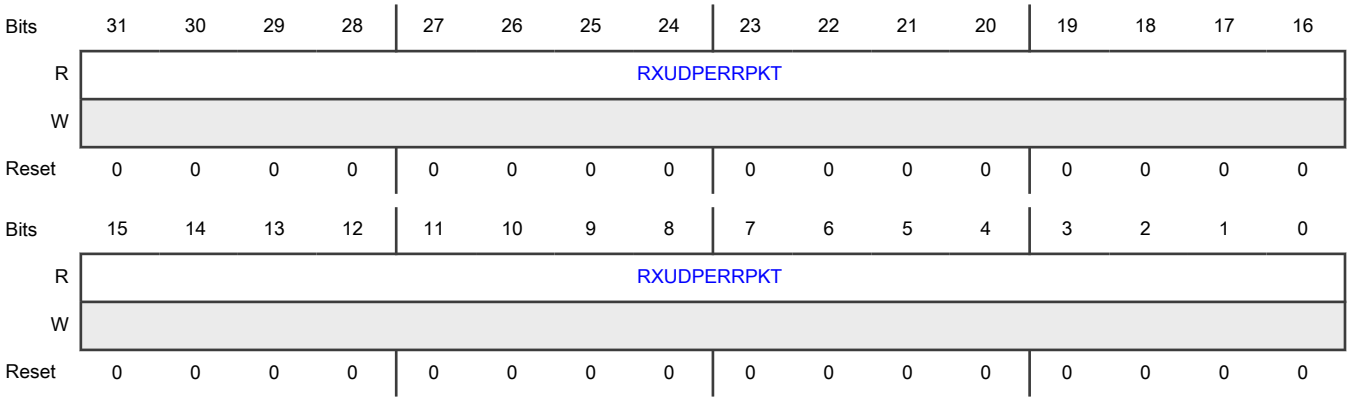
43.8.1.248 IPv6 Datagrams Received with UDP Checksum Error (MAC_RXUDP_ERROR_PACKETS)

This register provides the number of good IP datagrams received by DWC_ether_qos whose UDP payload has a checksum error.

Offset

Register	Offset
MAC_RXUDP_ERROR_PACKETS	834h

Diagram



Fields

Field	Description
31-0 RXUDPERRPKT	RxUDP Error Packets This field indicates the number of good IP datagrams received whose UDP payload has a checksum error.

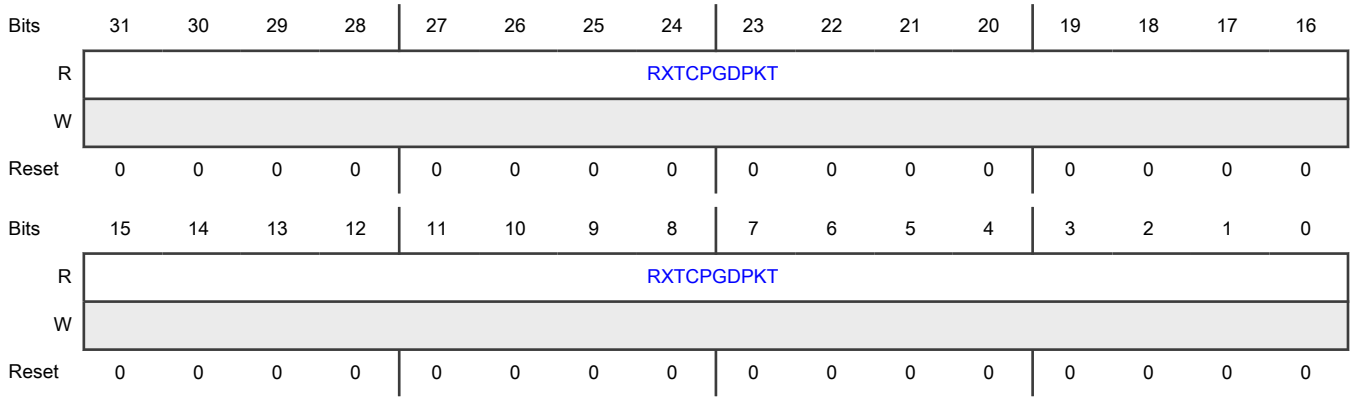
43.8.1.249 IPv6 Datagrams Received with Good TCP Payload (MAC_RXTCP_GOOD_PACKETS)

This register provides the number of good IP datagrams received by DWC_ether_qos with a good TCP payload.

Offset

Register	Offset
MAC_RXTCP_GOOD_PACKETS	838h

Diagram



Fields

Field	Description
31-0 RXTCPGDPKT	RxTCP Good Packets This field indicates the number of good IP datagrams received with a good TCP payload.

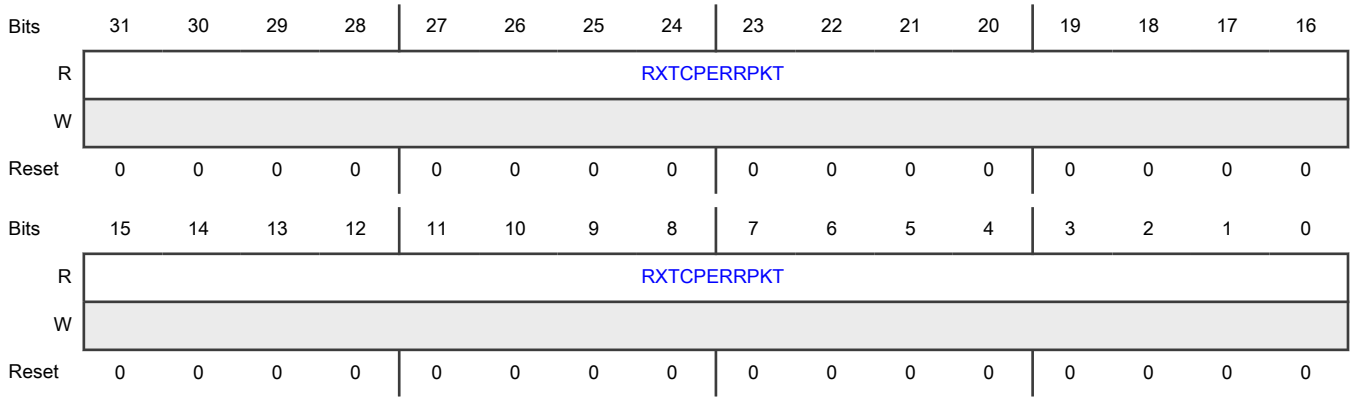
43.8.1.250 IPv6 Datagrams Received with TCP Checksum Error (MAC_RXTCP_ERROR_PACKETS)

This register provides the number of good IP datagrams received by DWC_ether_qos whose TCP payload has a checksum error.

Offset

Register	Offset
MAC_RXTCP_ERROR_PACKETS	83Ch

Diagram



Fields

Field	Description
31-0 RXTCPERRPKT	RxTCP Error Packets This field indicates the number of good IP datagrams received whose TCP payload has a checksum error.

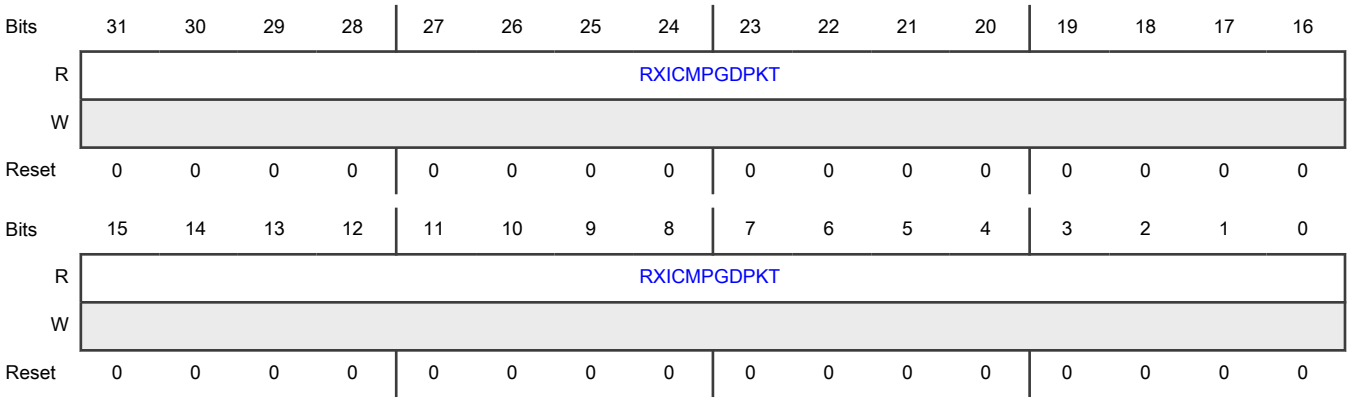
43.8.1.251 IPv6 Datagrams Received with Good ICMP Payload (MAC_RXICMP_GOOD_PACKETS)

This register provides the number of good IP datagrams received by DWC_ether_qos with a good ICMP payload.

Offset

Register	Offset
MAC_RXICMP_GOOD_PACKETS	840h

Diagram



Fields

Field	Description
31-0 RXICMPGDPKT	RxICMP Good Packets This field indicates the number of good IP datagrams received with a good ICMP payload.

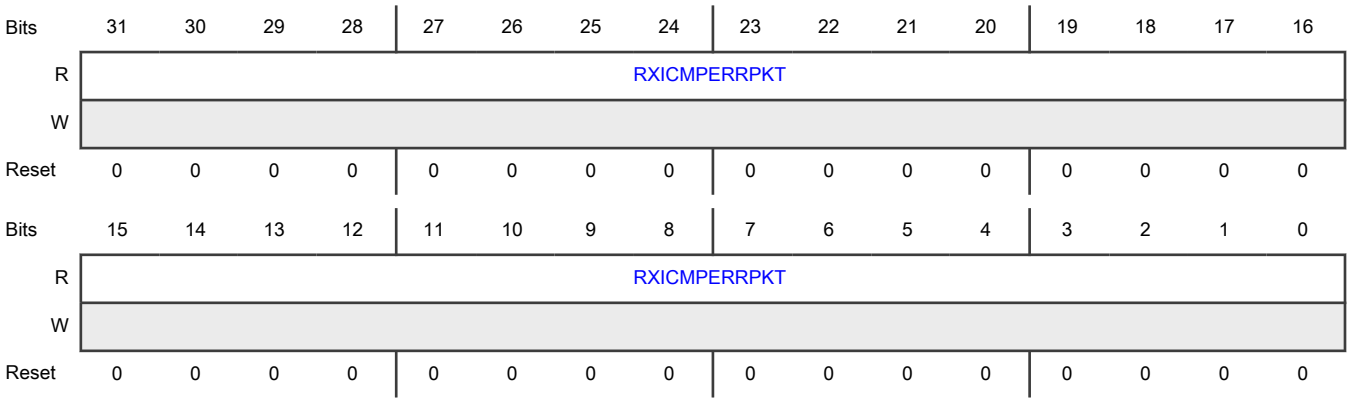
43.8.1.252 IPv6 Datagrams Received with ICMP Checksum Error (MAC_RXICMP_ERROR_PACKETS)

This register provides the number of good IP datagrams received by DWC_ether_qos whose ICMP payload has a checksum error.

Offset

Register	Offset
MAC_RXICMP_ERROR_PACKETS	844h

Diagram



Fields

Field	Description
31-0 RXICMPERRPKT	RxICMP Error Packets This field indicates the number of good IP datagrams received whose ICMP payload has a checksum error.

43.8.1.253 Good Bytes Received in IPv4 Datagrams (MAC_RXIPV4_GOOD_OCTETS)

This register provides the number of bytes received by DWC_ether_qos in good IPv4 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

Offset

Register	Offset
MAC_RXIPV4_GOOD_OCTETS	850h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4GDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4GDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXIPV4GDOCT	RxIPv4 Good Octets This field indicates the number of bytes received in good IPv4 datagrams encapsulating TCP, UDP, or ICMP data. RxIPv4 Good Octets This field indicates the number of bytes received in good IPv4 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

43.8.1.254 Bytes Received in IPv4 Datagrams with Header Errors (MAC_RXIPV4_HEADER_ERROR_OCTETS)

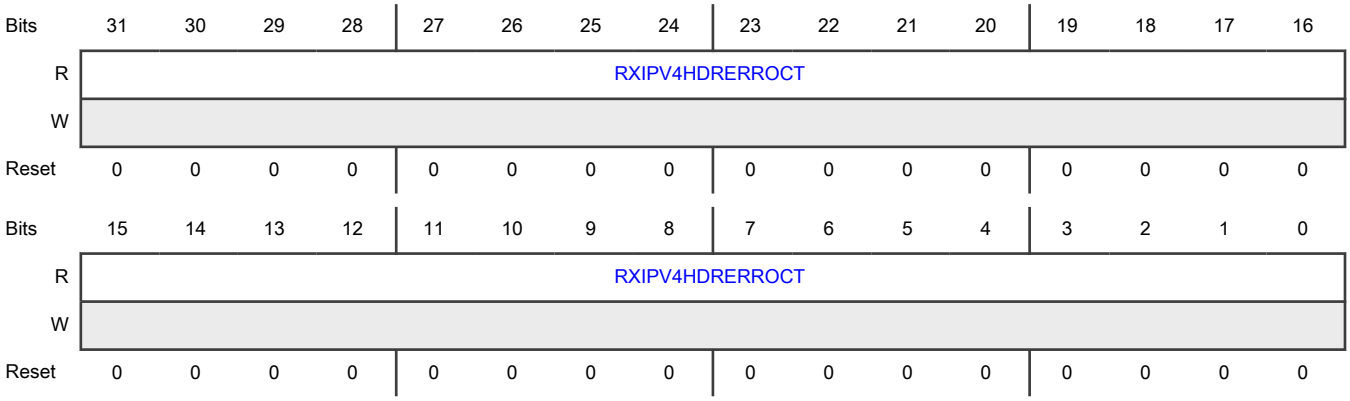
This register provides the number of bytes received by DWC_ether_qos in IPv4 datagrams with header errors (checksum, length, version mismatch).

This register provides the number of bytes received by DWC_ether_qos in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

Offset

Register	Offset
MAC_RXIPV4_HEADER_ERROR_OCTETS	854h

Diagram



Fields

Field	Description
31-0 RXIPV4HDRERROCT	RxIPv4 Header Error Octets This field indicates the number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). RxIPv4 Header Error Octets This field indicates the number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

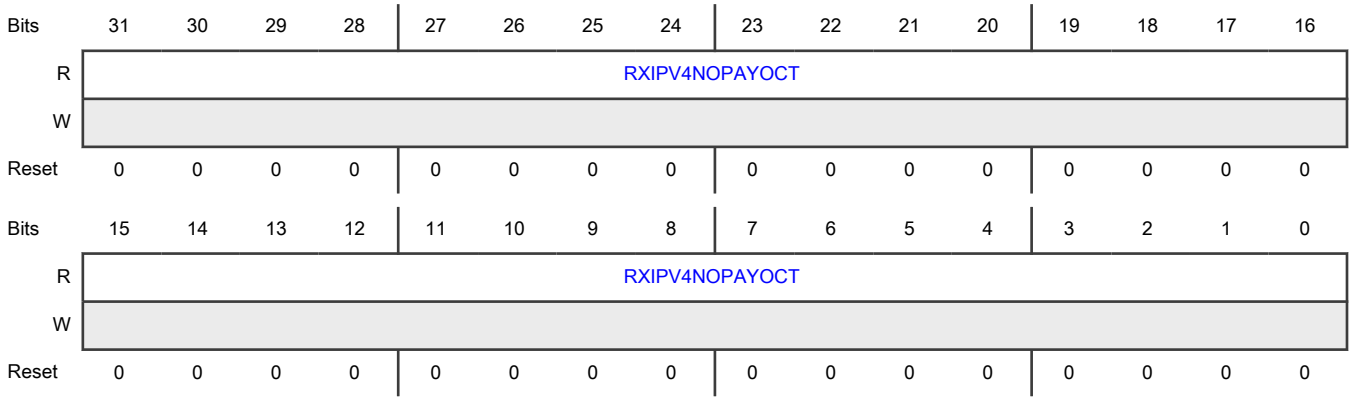
43.8.1.255 Bytes Received in IPv4 Datagrams with No Payload (MAC_RXIPV4_NO_PAYLOAD_OCTETS)

This register provides the number of bytes received by DWC_ether_qos in IPv4 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

Offset

Register	Offset
MAC_RXIPV4_NO_PAYLOAD_OCTETS	858h

Diagram



Fields

Field	Description
31-0 RXIPV4NOPAYOCT	RxIPv4 Payload Octets This field indicates the number of bytes received in IPv4 datagrams that did not have a TCP, UDP, or ICMP payload. RxIPv4 Payload Octets This field indicates the number of bytes received in IPv4 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

43.8.1.256 Bytes Received in Fragmented IPv4 Datagrams (MAC_RXIPV4_FRAGMENTED_OCTETS)

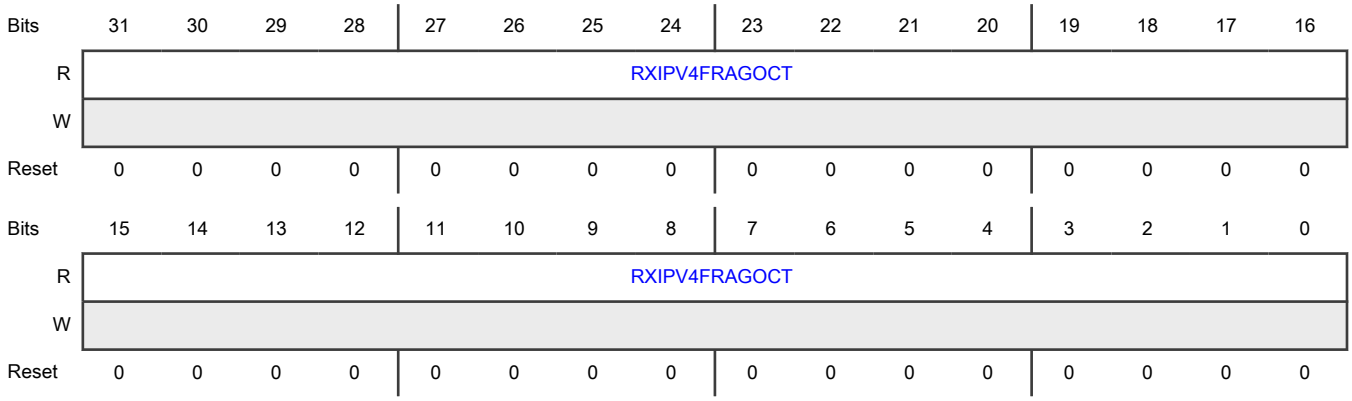
This register provides the number of bytes received by DWC_ether_qos in fragmented IPv4 datagrams.

This register provides the number of bytes received by DWC_ether_qos in fragmented IPv4 datagrams. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

Offset

Register	Offset
MAC_RXIPV4_FRAGMENTED_OCTETS	85Ch

Diagram



Fields

Field	Description
31-0 RXIPV4FRAGOCT	RxIPv4 Fragmented Octets This field indicates the number of bytes received in fragmented IPv4 datagrams. RxIPv4 Fragmented Octets This field indicates the number of bytes received in fragmented IPv4 datagrams. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

43.8.1.257 Bytes Received with UDP Checksum Disabled (MAC_RXIPV4_UDP_CHECKSUM_DISABLE_OCTETS)

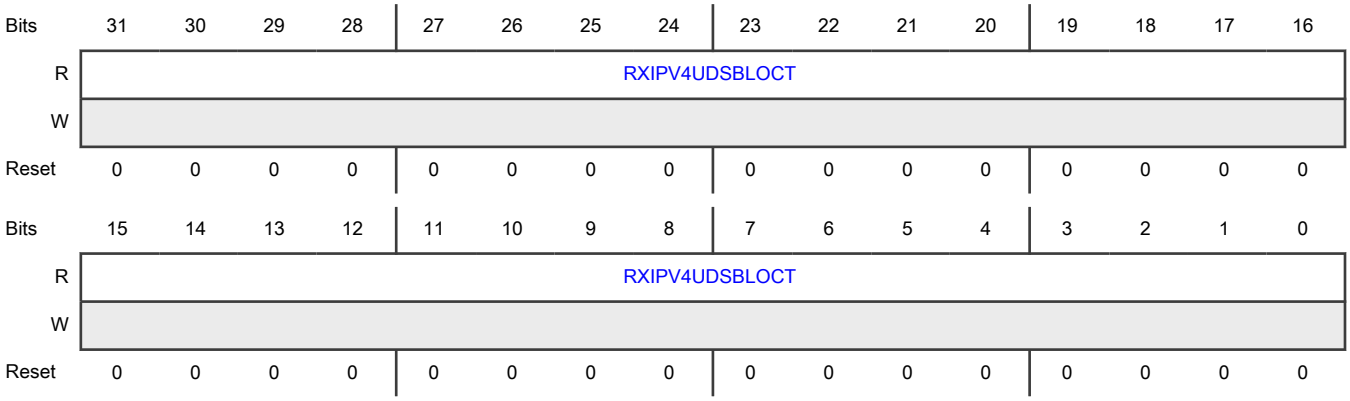
This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had the UDP checksum disabled.

This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had the UDP checksum disabled. This counter does not count IP Header bytes. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

Offset

Register	Offset
MAC_RXIPV4_UDP_CHECKSUM_DISABLE_OCTETS	860h

Diagram



Fields

Field	Description
31-0 RXIPV4UDSBLOCT	RxIPv4 UDP Checksum Disable Octets This field indicates the number of bytes received in a UDP segment that had the UDP checksum disabled. RxIPv4 UDP Checksum Disable Octets This field indicates the number of bytes received in a UDP segment that had the UDP checksum disabled. This counter does not count IP Header bytes. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

43.8.1.258 Bytes Received in Good IPv6 Datagrams (MAC_RXIPV6_GOOD_OCTETS)

This register provides the number of bytes received by DWC_ether_qos in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data.

This register provides the number of bytes received by DWC_ether_qos in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

Offset

Register	Offset
MAC_RXIPV6_GOOD_OCTETS	864h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV6GDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV6GDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXIPV6GDOCT	RxIPv6 Good Octets This field indicates the number of bytes received in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data. RxIPv6 Good Octets This field indicates the number of bytes received in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

43.8.1.259 Bytes Received in IPv6 Datagrams with Data Errors (MAC_RXIPV6_HEADER_ERROR_OCTETS)

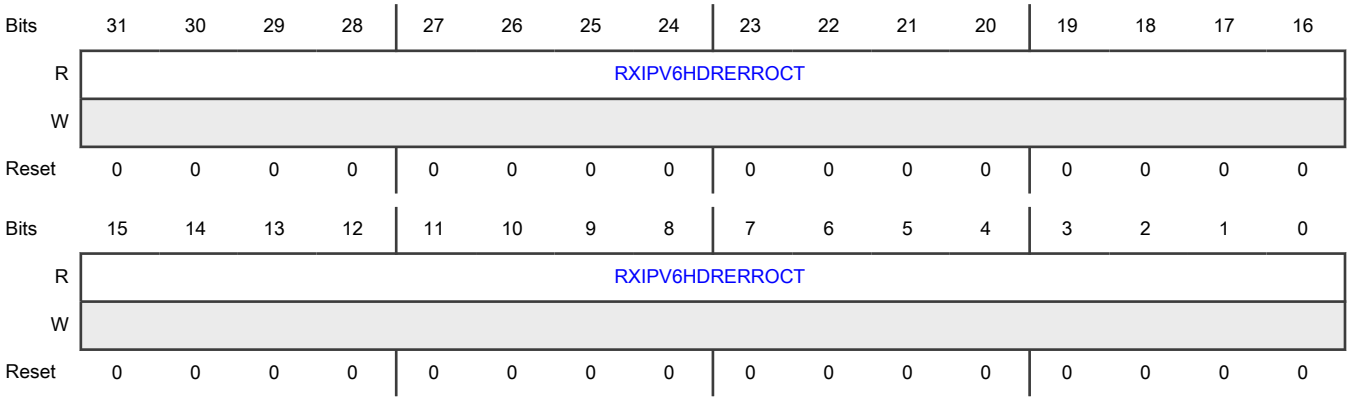
This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams with header errors (length, version mismatch).

This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams with header errors (length, version mismatch). The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

Offset

Register	Offset
MAC_RXIPV6_HEADER_ERROR_OCTETS	868h

Diagram



Fields

Field	Description
31-0 RXIPV6HDRERROCT	RxIPv6 Header Error Octets This field indicates the number of bytes received in IPv6 datagrams with header errors (length, version mismatch). RxIPv6 Header Error Octets This field indicates the number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

43.8.1.260 Bytes Received in IPv6 Datagrams with No Payload (MAC_RXIPV6_NO_PAYLOAD_OCTETS)

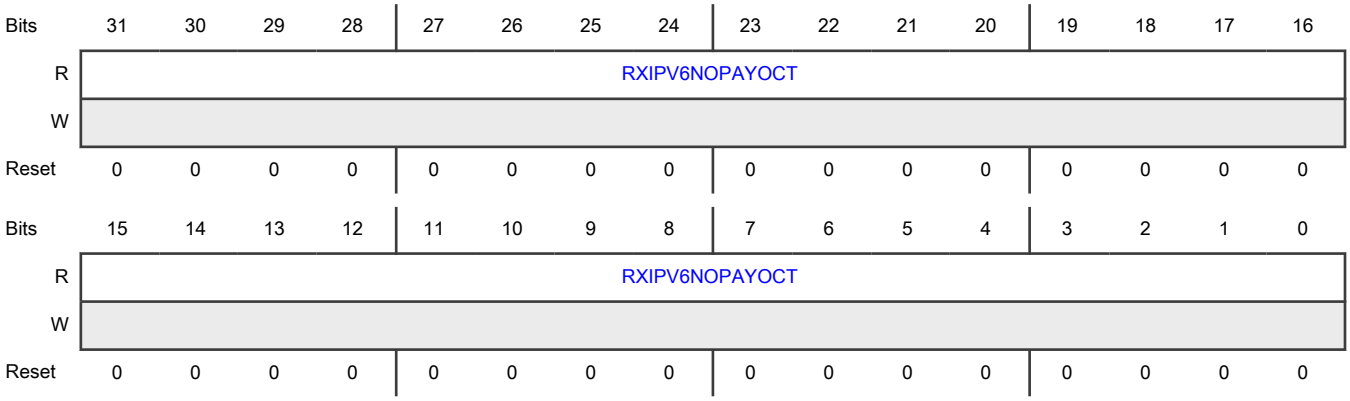
This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload.

This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

Offset

Register	Offset
MAC_RXIPV6_NO_PAYLOAD_OCTETS	86Ch

Diagram



Fields

Field	Description
31-0 RXIPV6NOPAYOCT	RxIPv6 Payload Octets This field indicates the number of bytes received in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload. RxIPv6 Payload Octets This field indicates the number of bytes received in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

43.8.1.261 Bytes Received in Good UDP Segment (MAC_RXUDP_GOOD_OCTETS)

This register provides the number of bytes received by DWC_ether_qos in a good UDP segment.

This register provides the number of bytes received by DWC_ether_qos in a good UDP segment. This counter does not count IP header bytes.

Offset

Register	Offset
MAC_RXUDP_GOOD_OCTETS	870h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXUDPGDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXUDPGDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXUDPGDOCT	RxUDP Good Octets This field indicates the number of bytes received in a good UDP segment. RxUDP Good Octets This field indicates the number of bytes received in a good UDP segment. This counter does not count IP header bytes.

43.8.1.262 Bytes Received in UDP Segment with Checksum Errors (MAC_RXUDP_ERROR_OCTETS)

This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had checksum errors.

This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had checksum errors. This counter does not count IP header bytes.

Offset

Register	Offset
MAC_RXUDP_ERROR_OCTETS	874h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXUDPERROCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXUDPERROCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 RXUDPERROCT	RxUDP Error Octets This field indicates the number of bytes received in a UDP segment that had checksum errors. RxUDP Error Octets This field indicates the number of bytes received in a UDP segment that had checksum errors. This counter does not count IP header bytes.

43.8.1.263 Bytes Received in Good TCP Segment (MAC_RXTCP_GOOD_OCTETS)

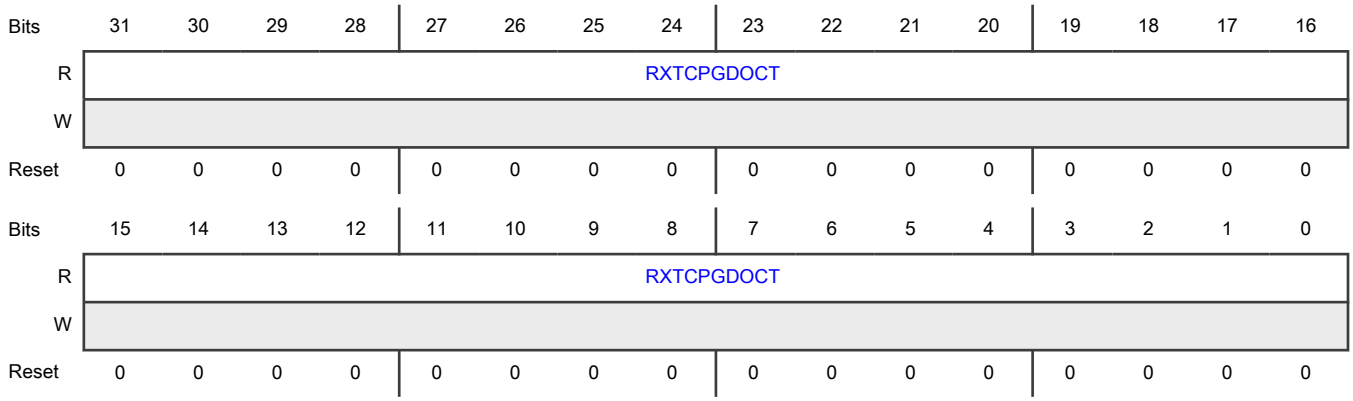
This register provides the number of bytes received by DWC_ether_qos in a good TCP segment.

This register provides the number of bytes received by DWC_ether_qos in a good TCP segment. This counter does not count IP header bytes.

Offset

Register	Offset
MAC_RXTCP_GOOD_OCTETS	878h

Diagram



Fields

Field	Description
31-0 RXTCPGDOCT	RxTCP Good Octets This field indicates the number of bytes received in a good TCP segment. RxTCP Good Octets This field indicates the number of bytes received in a good TCP segment. This counter does not count IP header bytes.

43.8.1.264 Bytes Received in TCP Segment with Checksum Errors (MAC_RXTCP_ERROR_OCTETS)

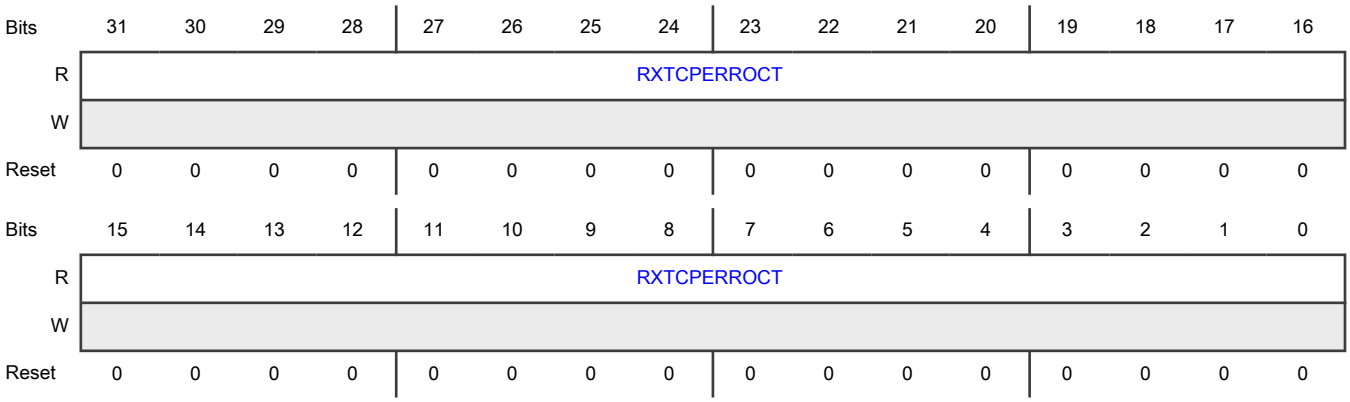
This register provides the number of bytes received by DWC_ether_qos in a TCP segment that had checksum errors.

This register provides the number of bytes received by DWC_ether_qos in a TCP segment that had checksum errors. This counter does not count IP header bytes.

Offset

Register	Offset
MAC_RXTCP_ERROR_OCTETS	87Ch

Diagram



Fields

Field	Description
31-0 RXTCPERROCT	RxTCP Error Octets This field indicates the number of bytes received in a TCP segment that had checksum errors. RxTCP Error Octets This field indicates the number of bytes received in a TCP segment that had checksum errors. This counter does not count IP header bytes.

43.8.1.265 Bytes Received in Good ICMP Segment (MAC_RXICMP_GOOD_OCTETS)

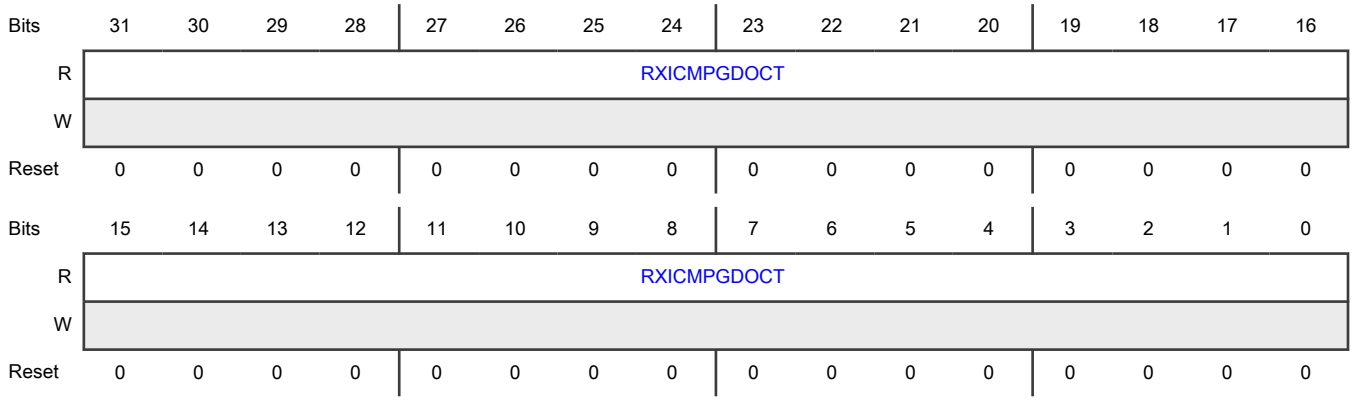
This register provides the number of bytes received by DWC_ether_qos in a good ICMP segment.

This register provides the number of bytes received by DWC_ether_qos in a good ICMP segment. This counter does not count IP header bytes.

Offset

Register	Offset
MAC_RXICMP_GOOD_OCTETS	880h

Diagram



Fields

Field	Description
31-0 RXICMPGDOCT	RxICMP Good Octets This field indicates the number of bytes received in a good ICMP segment. RxICMP Good Octets This field indicates the number of bytes received in a good ICMP segment. This counter does not count IP header bytes.

43.8.1.266 Bytes Received in ICMP Segment with Checksum Errors (MAC_RXICMP_ERROR_OCTETS)

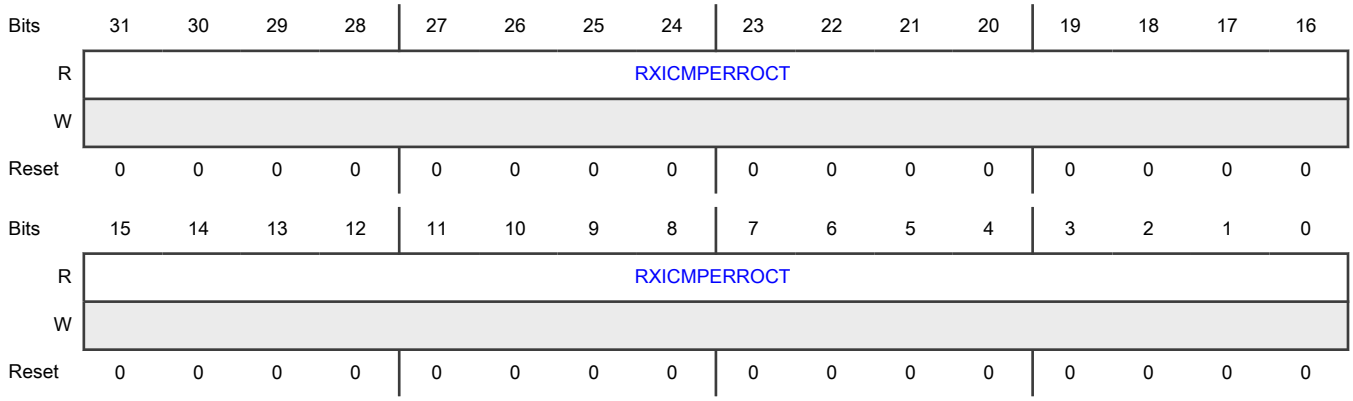
This register provides the number of bytes received by DWC_ether_qos in a ICMP segment that had checksum errors.

This register provides the number of bytes received by DWC_ether_qos in a ICMP segment that had checksum errors. This counter does not count IP header bytes.

Offset

Register	Offset
MAC_RXICMP_ERROR_OCTETS	884h

Diagram



Fields

Field	Description
31-0 RXICMPERROCT	RxICMP Error Octets This field indicates the number of bytes received in a ICMP segment that had checksum errors. RxICMP Error Octets This field indicates the number of bytes received in a ICMP segment that had checksum errors. This counter does not count IP header bytes.

43.8.1.267 MMC FPE Transmit Interrupt (MAC_MMC_FPE_TX_INTERRUPT)

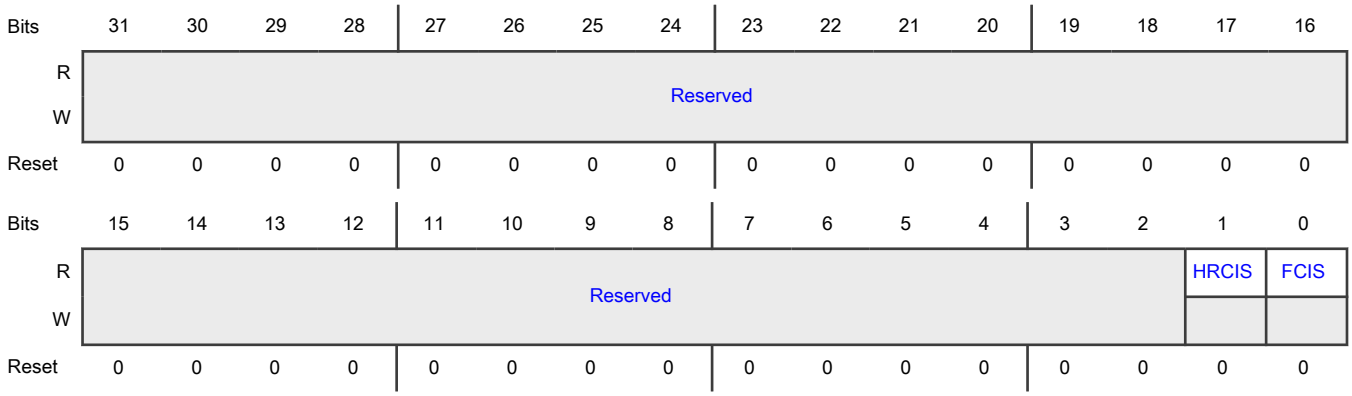
This register maintains the interrupts generated from all FPE related Transmit statistics counters.

This register maintains the interrupts generated from all FPE related Transmit statistics counters. The MMC FPE Transmit Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones. The MMC FPE Transmit Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (Bits[7:0]) of the respective counter must be read to clear the interrupt bit.

Offset

Register	Offset
MAC_MMC_FPE_TX_INTERRUPT	8A0h

Diagram



Fields

Field	Description
31-2 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 HRCIS	MMC Tx Hold Request Counter Interrupt Status This bit is set when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. MMC Tx Hold Request Counter Interrupt Status This bit is set when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE with AV_EST Enabled configuration. 0 - MMC Tx Hold Request Counter Interrupt Status not detected 1 - MMC Tx Hold Request Counter Interrupt Status detected
0 FCIS	MMC Tx FPE Fragment Counter Interrupt status This bit is set when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. MMC Tx FPE Fragment Counter Interrupt status This bit is set when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Tx FPE Fragment Counter Interrupt status not detected 1 - MMC Tx FPE Fragment Counter Interrupt status detected

43.8.1.268 MMC FPE Transmit Mask Interrupt (MAC_MMC_FPE_TX_INTERRUPT_MASK)

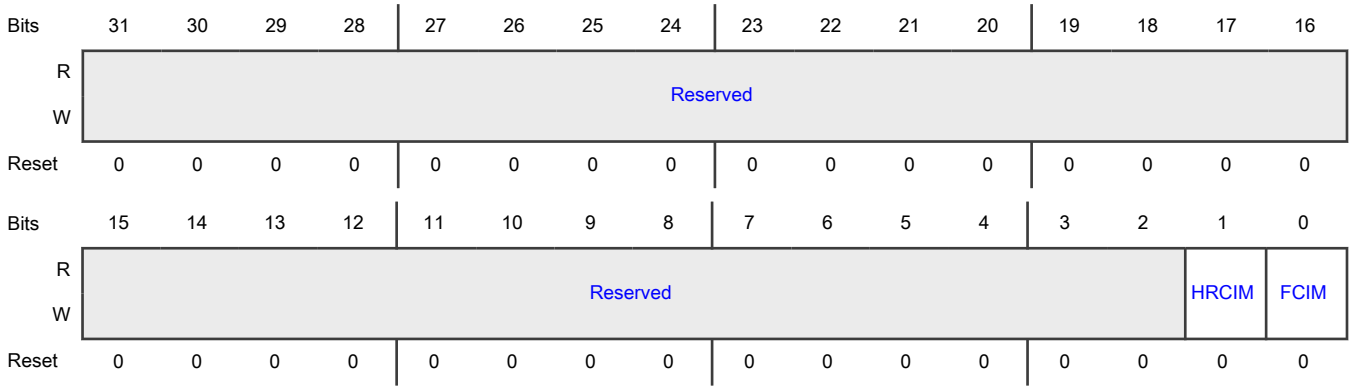
This register maintains the masks for interrupts generated from all FPE related Transmit statistics counters.

This register maintains the masks for interrupts generated from all FPE related Transmit statistics counters. The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when FPE related receive statistic counters reach half of their maximum value or the maximum values. This register is 32 bit wide.

Offset

Register	Offset
MAC_MMC_FPE_TX_INTERRUPT_MASK	8A4h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 HRCIM	MMC Transmit Hold Request Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. MMC Transmit Hold Request Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE with AV_EST Enabled configuration. 0 - MMC Transmit Hold Request Counter Interrupt Mask is disabled 1 - MMC Transmit Hold Request Counter Interrupt Mask is enabled
0 FCIM	MMC Transmit Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. MMC Transmit Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Transmit Fragment Counter Interrupt Mask is disabled 1 - MMC Transmit Fragment Counter Interrupt Mask is enabled

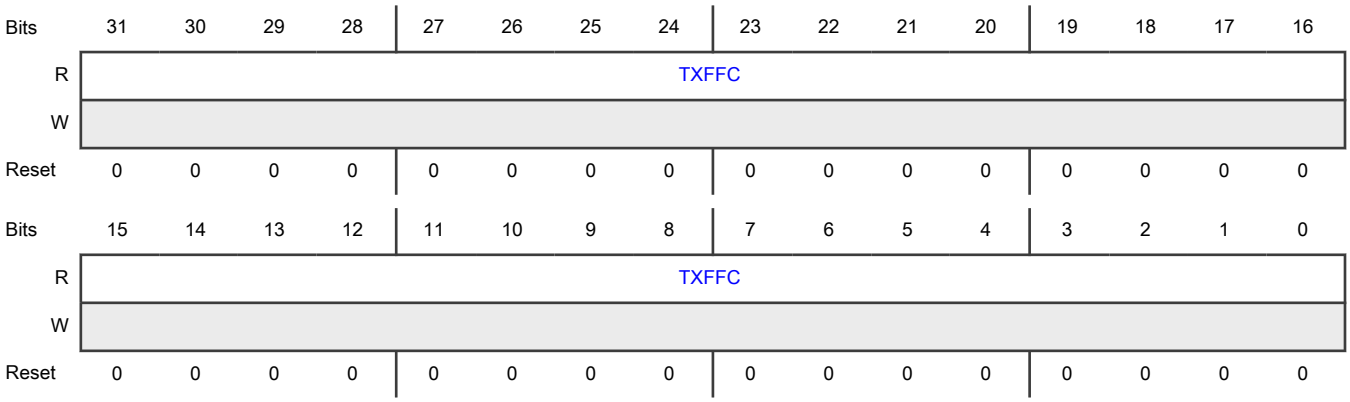
43.8.1.269 MMC FPE Transmitted Fragment Counter (MAC_MMC_TX_FPE_FRAGMENT_CNTR)

This register provides the number of additional mPackets transmitted due to preemption.

Offset

Register	Offset
MAC_MMC_TX_FPE_FRAGMENT_CNTR	8A8h

Diagram



Fields

Field	Description
31-0 TXFFC	Tx FPE Fragment counter This field indicates the number of additional mPackets that has been transmitted due to preemption Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

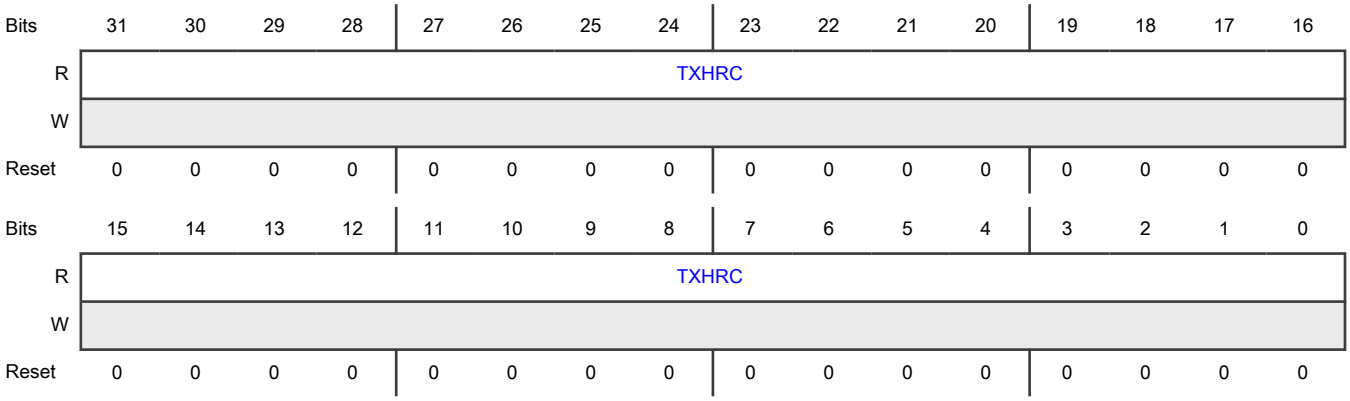
43.8.1.270 MMC FPE Transmitted Hold Request Counter (MAC_MMC_TX_HOLD_REQ_CNTR)

This register provides the count of number of times a hold request is given to MAC

Offset

Register	Offset
MAC_MMC_TX_HOLD_REQ_CNTR	8ACh

Diagram



Fields

Field	Description
31-0 TXHRC	Tx Hold Request Counter This field indicates count of number of a hold request is given to MAC. Tx Hold Request Counter This field indicates count of number of a hold request is given to MAC. Exists when any one of the RX/TX MMC counters are enabled during FPE with AV_EST Enabled configuration.

43.8.1.271 MMC FPE Receive Interrupt (MAC_MMC_FPE_RX_INTERRUPT)

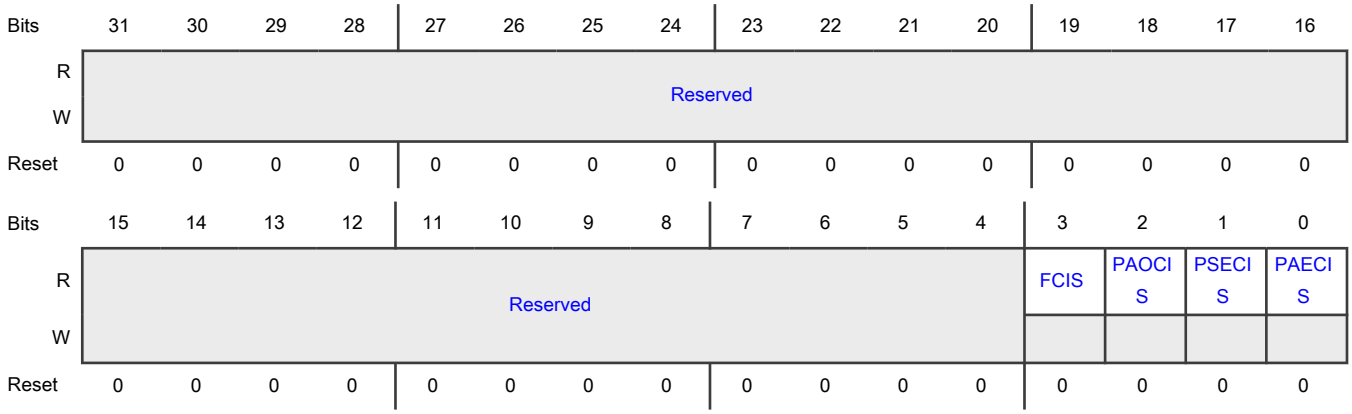
This register maintains the interrupts generated from all FPE related Receive statistics counters.

This register maintains the interrupts generated from all FPE related Receive statistics counters. The MMC FPE Receive Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones. The MMC FPE Receive Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (Bits[7:0]) of the respective counter must be read to clear the interrupt bit.

Offset

Register	Offset
MAC_MMC_FPE_RX_INTERRUPT	8C0h

Diagram



Fields

Field	Description
31-4 —	Reserved
3 FCIS	MMC Rx FPE Fragment Counter Interrupt Status This bit is set when the Rx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx FPE Fragment Counter Interrupt Status This bit is set when the Rx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx FPE Fragment Counter Interrupt Status not detected 1 - MMC Rx FPE Fragment Counter Interrupt Status detected
2 PAOCIS	MMC Rx Packet Assembly OK Counter Interrupt Status This bit is set when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet Assembly OK Counter Interrupt Status This bit is set when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx Packet Assembly OK Counter Interrupt Status not detected 1 - MMC Rx Packet Assembly OK Counter Interrupt Status detected
1 PSECIS	MMC Rx Packet SMD Error Counter Interrupt Status This bit is set when the Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet SMD Error Counter Interrupt Status

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This bit is set when the Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx Packet SMD Error Counter Interrupt Status not detected 1 - MMC Rx Packet SMD Error Counter Interrupt Status detected
0 PAECIS	MMC Rx Packet Assembly Error Counter Interrupt Status This bit is set when the Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet Assembly Error Counter Interrupt Status This bit is set when the Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx Packet Assembly Error Counter Interrupt Status not detected 1 - MMC Rx Packet Assembly Error Counter Interrupt Status detected

43.8.1.272 MMC FPE Receive Interrupt Mask (MAC_MMC_FPE_RX_INTERRUPT_MASK)

This register maintains the masks for interrupts generated from all FPE related Receive statistics counters.

This register maintains the masks for interrupts generated from all FPE related Receive statistics counters. The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when FPE related receive statistic counters reach half of their maximum value or the maximum values. This register is 32 bit wide.

Offset

Register	Offset
MAC_MMC_FPE_RX_INTERRUPT_MASK	8C4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												FCIM	PAOCIM	PSECIM	PAECIM
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved
3 FCIM	MMC Rx FPE Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx FPE Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx FPE Fragment Counter Interrupt Mask is disabled 1 - MMC Rx FPE Fragment Counter Interrupt Mask is enabled
2 PAOCIM	MMC Rx Packet Assembly OK Counter Interrupt Mask Setting this bit masks the interrupt when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet Assembly OK Counter Interrupt Mask Setting this bit masks the interrupt when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx Packet Assembly OK Counter Interrupt Mask is disabled 1 - MMC Rx Packet Assembly OK Counter Interrupt Mask is enabled
1 PSECIM	MMC Rx Packet SMD Error Counter Interrupt Mask Setting this bit masks the interrupt when the R Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet SMD Error Counter Interrupt Mask Setting this bit masks the interrupt when the R Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx Packet SMD Error Counter Interrupt Mask is disabled 1 - MMC Rx Packet SMD Error Counter Interrupt Mask is enabled
0 PAECIM	MMC Rx Packet Assembly Error Counter Interrupt Mask Setting this bit masks the interrupt when the R Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet Assembly Error Counter Interrupt Mask Setting this bit masks the interrupt when the R Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx Packet Assembly Error Counter Interrupt Mask is disabled 1 - MMC Rx Packet Assembly Error Counter Interrupt Mask is enabled

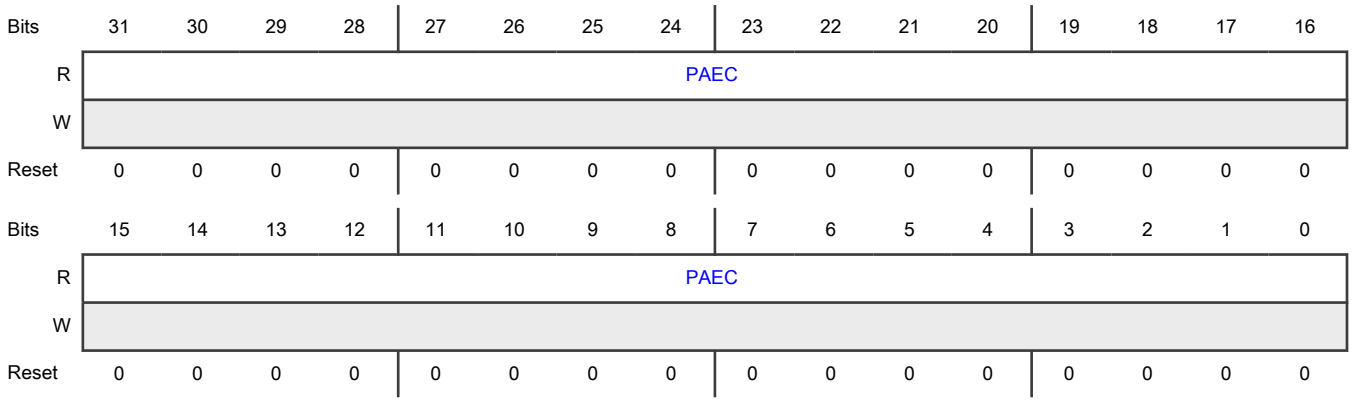
43.8.1.273 MMC Receive Packet Reassembly Error Counter (MAC_MMC_RX_PACKET_ASSEMBLY_ERR_CNTR)

This register provides the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value.

Offset

Register	Offset
MAC_MMC_RX_PACKET_ASSEMBLY_ERR_CNTR	8C8h

Diagram



Fields

Field	Description
31-0 PAEC	Rx Packet Assembly Error Counter This field indicates the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value. Rx Packet Assembly Error Counter This field indicates the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

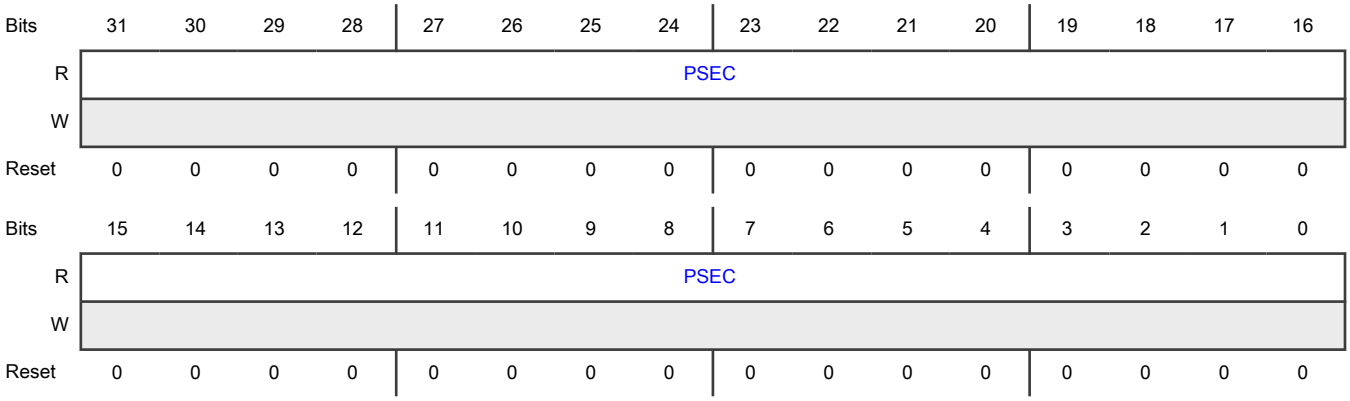
43.8.1.274 MMC Receive Packet SMD Error Counter (MAC_MMC_RX_PACKET_SMD_ERR_CNTR)

This register provides the number of received MAC frames rejected due to unknown SMD value and MAC frame fragments rejected due to arriving with an SMD-C when there was no preceding preempted frame.

Offset

Register	Offset
MAC_MMC_RX_PACKET_SMD_ERR_CNTR	8CCh

Diagram



Fields

Field	Description
31-0 PSEC	Rx Packet SMD Error Counter This field indicates the number of MAC frames rejected due to unknown SMD value and MAC frame fragments rejected due to arriving with an SMD-C when there was no preceding preempted frame. Rx Packet SMD Error Counter This field indicates the number of MAC frames rejected due to unknown SMD value and MAC frame fragments rejected due to arriving with an SMD-C when there was no preceding preempted frame. Exists when at least one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

43.8.1.275 MMC Receive Packet Successful Reassembly Counter (MAC_MMC_RX_PACKET_ASSEMBLY_OK_CNTR)

This register provides the number of MAC frames that were successfully reassembled and delivered to MAC.

Offset

Register	Offset
MAC_MMC_RX_PACKET_ASSEMBLY_OK_CNTR	8D0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PAOC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PAOC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PAOC	Rx Packet Assembly OK Counter This field indicates the number of MAC frames that were successfully reassembled and delivered to MAC. Rx Packet Assembly OK Counter This field indicates the number of MAC frames that were successfully reassembled and delivered to MAC. Exists when at least one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

43.8.1.276 MMC FPE Received Fragment Counter (MAC_MMC_RX_FPE_FRAGMENT_CNTR)

This register provides the number of additional mPackets received due to preemption.

Offset

Register	Offset
MAC_MMC_RX_FPE_FRAGMENT_CNTR	8D4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FFC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FFC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 FFC	Rx FPE Fragment Counter This field indicates the number of additional mPackets received due to preemption Exists when at least one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

43.8.1.277 Layer 3 and Layer 4 Control of Filter 0 (MAC_L3_L4_CONTROL0)

The Layer 3 and Layer 4 Control register controls the operations of filter 0 of Layer 3 and Layer 4.

Offset

Register	Offset
MAC_L3_L4_CONTROL0	900h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN0	Reserved	DMCHN0			Reserved		L4DPIM0	L4DPM0	L4SPIM0	L4SPM0	Reserved	L4PEN0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM0				L3HSBM0						L3DAIM0	L3DAM0	L3SAIM0	L3SAM0	Reserved	L3PEN0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 DMCHEN0	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
27 —	Reserved
26-24 DMCHN0	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM0	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM0 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM0	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM0	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching.

Table continues on the next page...

Table continued from the previous page...

Field	Description
L4SPM0	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN0	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM0 bit is set. 0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM0	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM0	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM0	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM0	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM0	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM0	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN0	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

43.8.1.278 Layer 4 Address 0 (MAC_LAYER4_ADDRESS0)

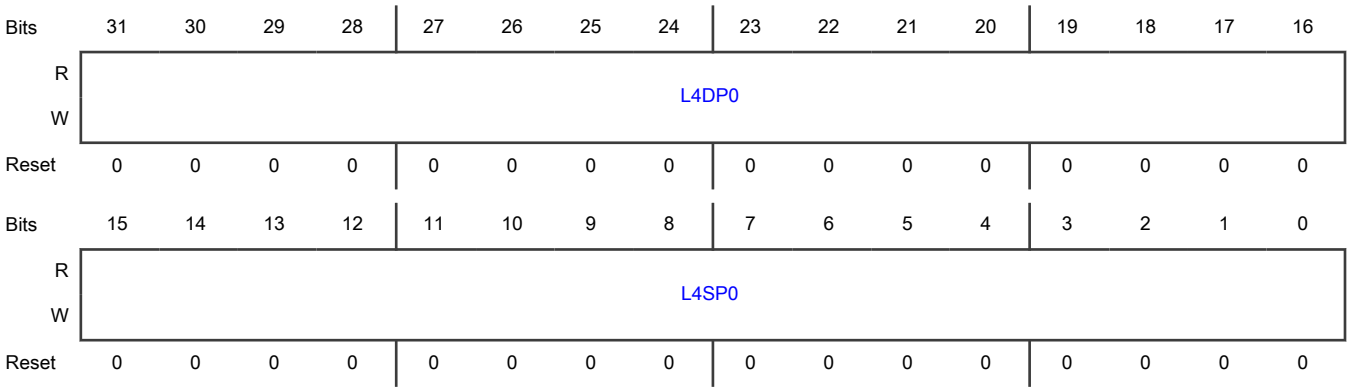
The MAC_LAYER4_ADDRESS0, MAC_L3_L4_CONTROL0, MAC_LAYER3_ADDR0_Reg0, MAC_LAYER3_ADDR1_Reg0, MAC_LAYER3_ADDR2_Reg0 and MAC_LAYER3_ADDR3_Reg0 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

Offset

Register	Offset
MAC_LAYER4_ADDRESS0	904h

Diagram



Fields

Field	Description
31-16 L4DP0	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4DPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP0	Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

43.8.1.279 Layer 3 Address 0 Register 0 (MAC_LAYER3_ADDR0_REG0)

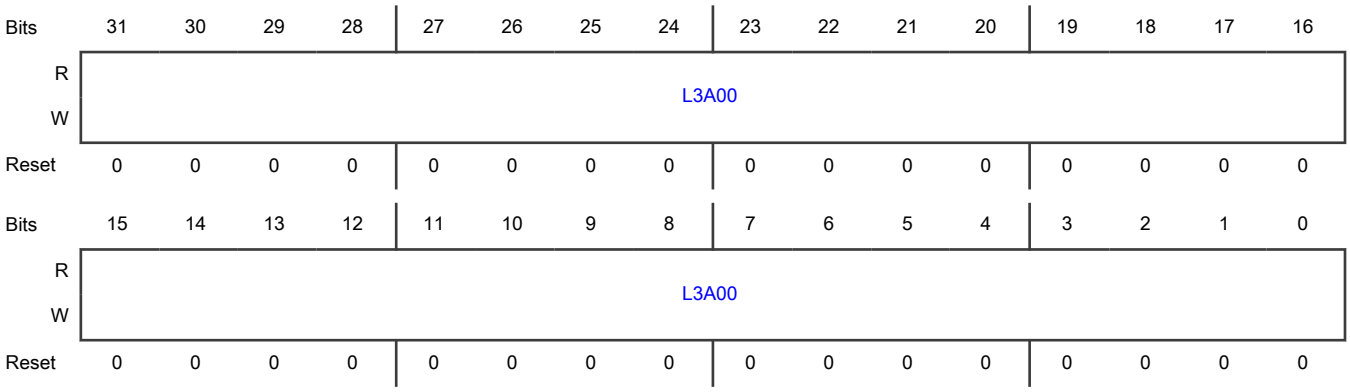
For IPv4 packets, the Layer 3 Address 0 Register 0 register contains the 32-bit IP Source Address field.

For IPv4 packets, the Layer 3 Address 0 Register 0 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR0_REG0	910h

Diagram



Fields

Field	Description
31-0 L3A00	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

43.8.1.280 Layer 3 Address 1 Register 0 (MAC_LAYER3_ADDR1_REG0)

For IPv4 packets, the Layer 3 Address 1 Register 0 register contains the 32-bit IP Destination Address field.

For IPv4 packets, the Layer 3 Address 1 Register 0 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR1_REG0	914h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 L3A10	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

43.8.1.281 Layer 3 Address 2 Register 0 (MAC_LAYER3_ADDR2_REG0)

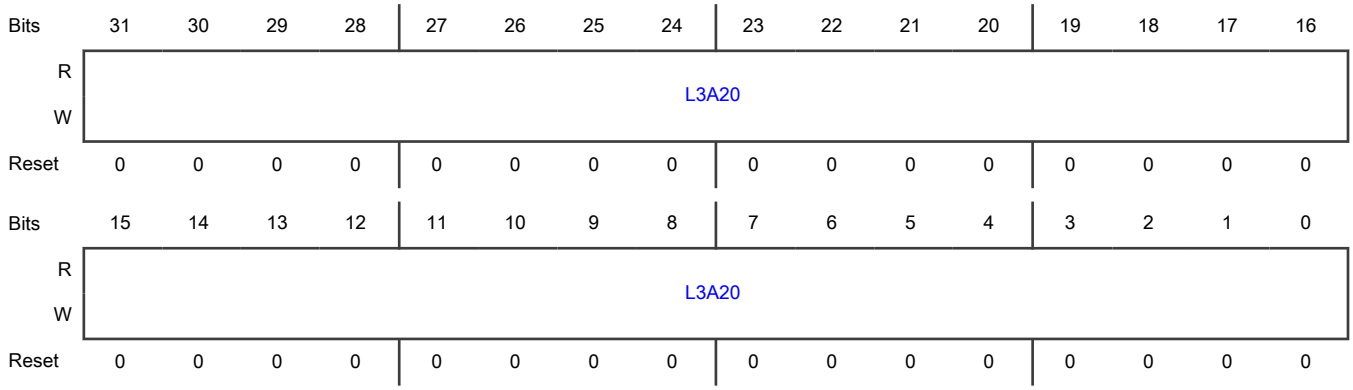
The Layer 3 Address 2 Register 0 register is reserved for IPv4 packets.

The Layer 3 Address 2 Register 0 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR2_REG0	918h

Diagram



Fields

Field	Description
31-0 L3A20	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

43.8.1.282 Layer 3 Address 3 Register 0 (MAC_LAYER3_ADDR3_REG0)

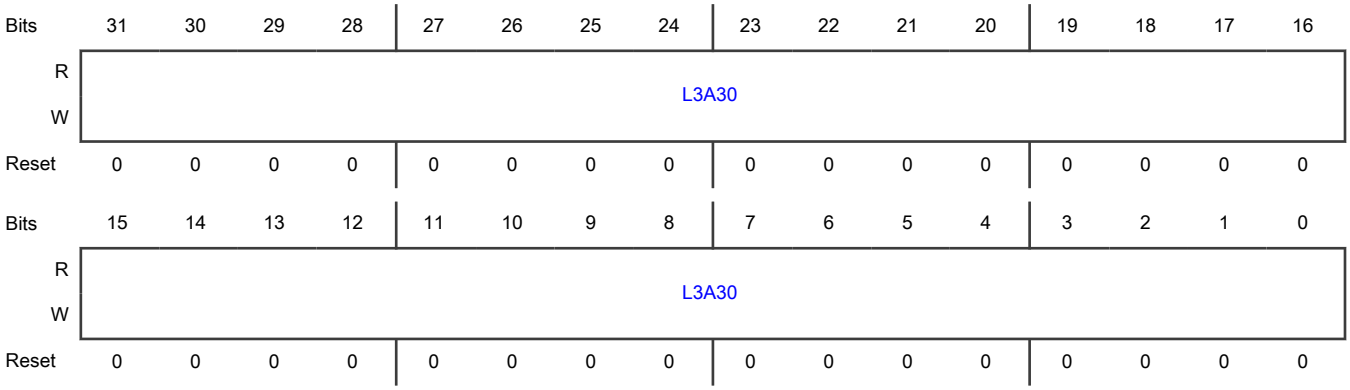
The Layer 3 Address 3 Register 0 register is reserved for IPv4 packets.

The Layer 3 Address 3 Register 0 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR3_REG0	91Ch

Diagram



Fields

Field	Description
31-0 L3A30	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

43.8.1.283 Layer 3 and Layer 4 Control of Filter 1 (MAC_L3_L4_CONTROL1)

The Layer 3 and Layer 4 Control register controls the operations of filter 1 of Layer 3 and Layer 4.

Offset

Register	Offset
MAC_L3_L4_CONTROL1	930h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN1	Reserved	DMCHN1			Reserved		L4DPIM1	L4DPM1	L4SPIM1	L4SPM1	Reserved	L4PEN1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM1				L3HSBM1						L3DAIM1	L3DAM1	L3SAIM1	L3SAM1	Reserved	L3PEN1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 DMCHEN1	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN1	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM1	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM1 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM1	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM1	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM1	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN1	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM1 bit is set.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM1	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM1	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM1	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM1	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM1	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM1	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 —	Reserved
0 L3PEN1	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

43.8.1.284 Layer 4 Address 0 (MAC_LAYER4_ADDRESS1)

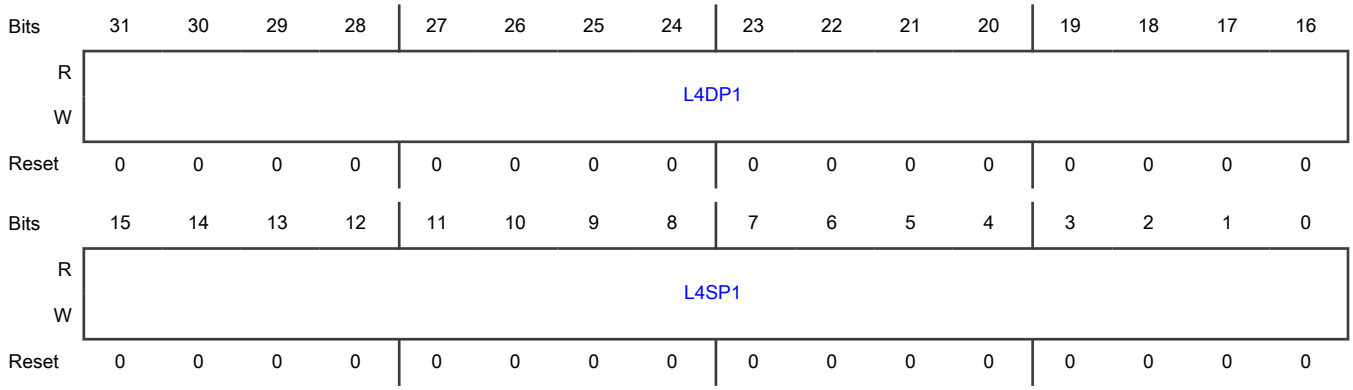
The MAC_LAYER4_ADDRESS1, MAC_L3_L4_CONTROL1, MAC_LAYER3_ADDR0_Reg1, MAC_LAYER3_ADDR1_Reg1, MAC_LAYER3_ADDR2_Reg1 and MAC_LAYER3_ADDR3_Reg1 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

Offset

Register	Offset
MAC_LAYER4_ADDRESS1	934h

Diagram



Fields

Field	Description
31-16 L4DP1	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM1 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN1 bit is reset and the L4DPM1 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN1 and L4DPM1 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP1	Layer 4 Source Port Number Field When the L4PEN1 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN1 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN1 and L4SPM1 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

43.8.1.285 Layer 3 Address 0 Register 1 (MAC_LAYER3_ADDR0_REG1)

For IPv4 packets, the Layer 3 Address 0 Register 1 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR0_REG1	940h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 L3A01	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

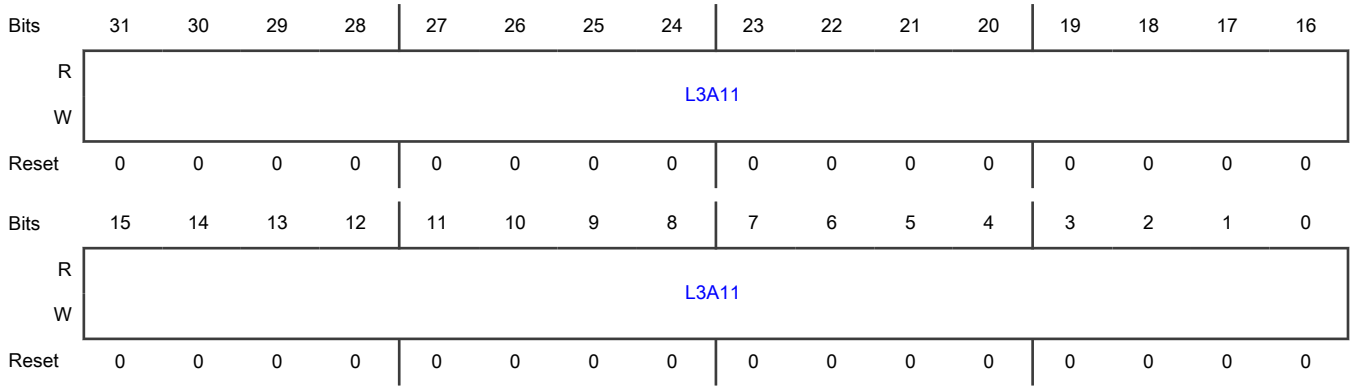
43.8.1.286 Layer 3 Address 1 Register 1 (MAC_LAYER3_ADDR1_REG1)

For IPv4 packets, the Layer 3 Address 1 Register 0 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR1_REG1	944h

Diagram



Fields

Field	Description
31-0 L3A11	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets.

Table continues on the next page...

Field	Description
	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

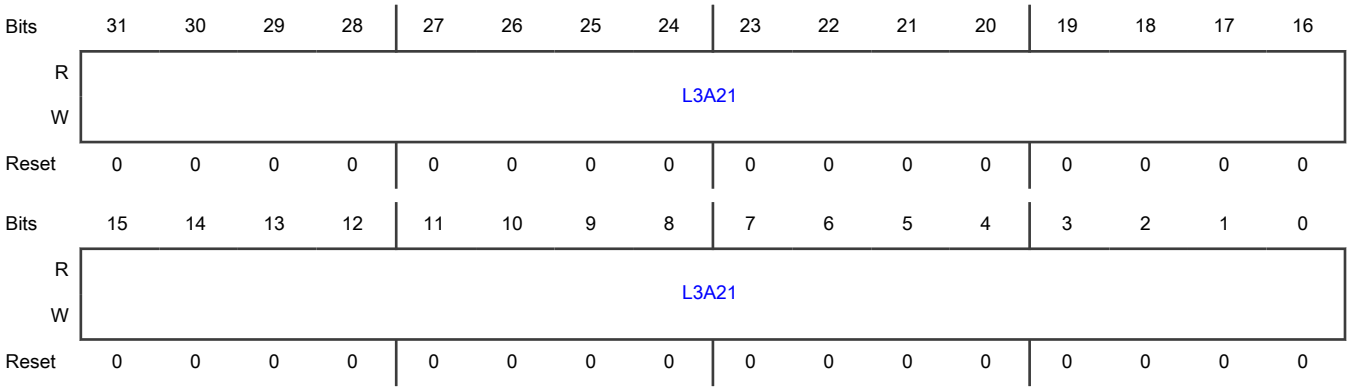
43.8.1.287 Layer 3 Address 2 Register 1 (MAC_LAYER3_ADDR2_REG1)

The Layer 3 Address 2 Register 1 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR2_REG1	948h

Diagram



Fields

Field	Description
31-0 L3A21	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets.

Table continues on the next page...

Field	Description
	When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

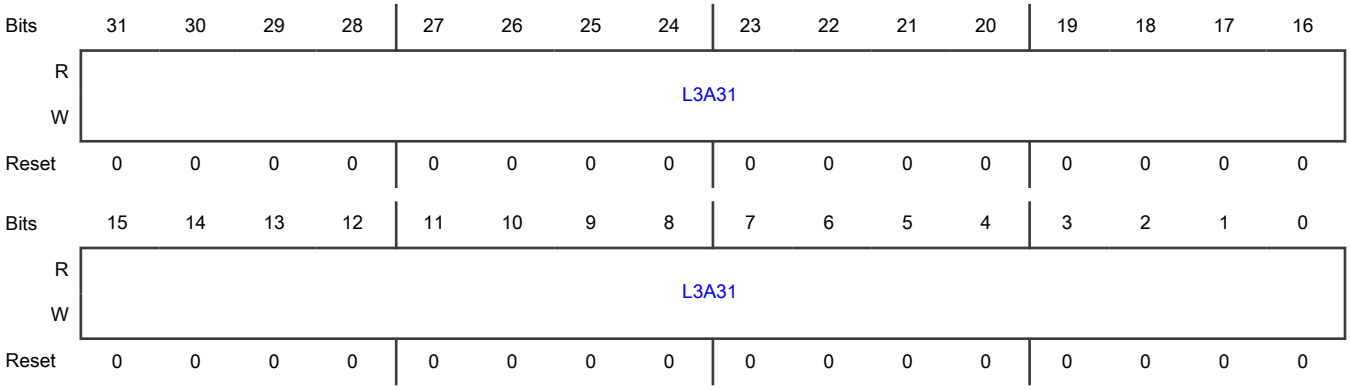
43.8.1.288 Layer 3 Address 3 Register 1 (MAC_LAYER3_ADDR3_REG1)

The Layer 3 Address 3 Register 1 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR3_REG1	94Ch

Diagram



Fields

Field	Description
31-0 L3A31	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

43.8.1.289 Layer 3 and Layer 4 Control of Filter 2 (MAC_L3_L4_CONTROL2)

The Layer 3 and Layer 4 Control register controls the operations of filter 2 of Layer 3 and Layer 4.

Offset

Register	Offset
MAC_L3_L4_CONTROL2	960h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN2	Reserved	DMCHN2			Reserved		L4DPIM2	L4DPM2	L4SPIM2	L4SPM2	Reserved	L4PEN2
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM2				L3HSBM2						L3DAIM2	L3DAM2	L3SAIM2	L3SAM2	Reserved	L3PEN2
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 DMCHEN2	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN2	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
21 L4DPIM2	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM2 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM2	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM2	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM2	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
16 L4PEN2	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM2 bit is set. 0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM2	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM2	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM2	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM2	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM2	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching.

Table continues on the next page...

Table continued from the previous page...

Field	Description
L3SAM2	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN2	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

43.8.1.290 Layer 4 Address 2 (MAC_LAYER4_ADDRESS2)

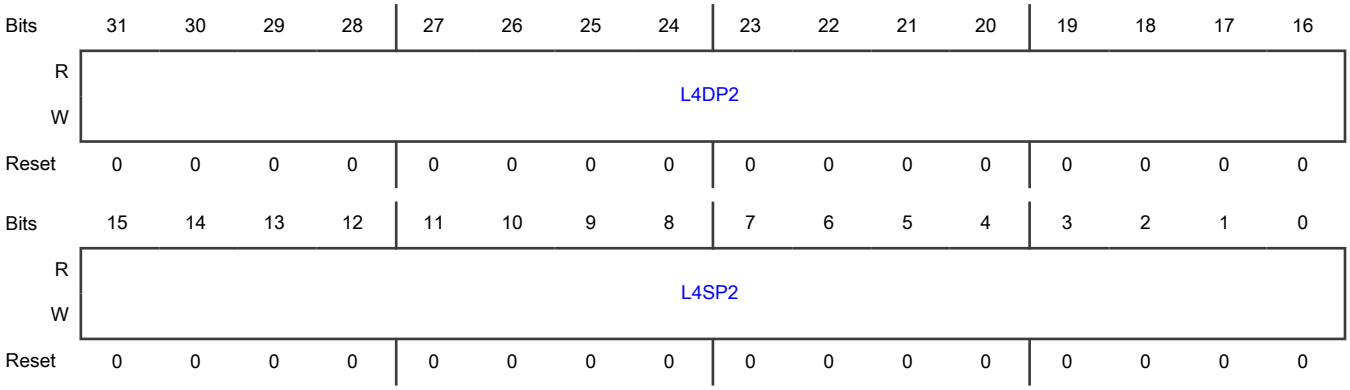
The MAC_LAYER4_ADDRESS2, MAC_L3_L4_CONTROL2), MAC_LAYER3_ADDR0_Reg2), MAC_LAYER3_ADDR1_Reg2, MAC_LAYER3_ADDR2_Reg2 and MAC_LAYER3_ADDR3_Reg2 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

Offset

Register	Offset
MAC_LAYER4_ADDRESS2	964h

Diagram



Fields

Field	Description
31-16 L4DP2	Layer 4 Destination Port Number Field When the L4PEN2 bit is reset and the L4DPM2 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN2 bit is reset and the L4DPM2 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN2 and L4DPM2 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP2	Layer 4 Source Port Number Field When the L4PEN2 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN2 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN2 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

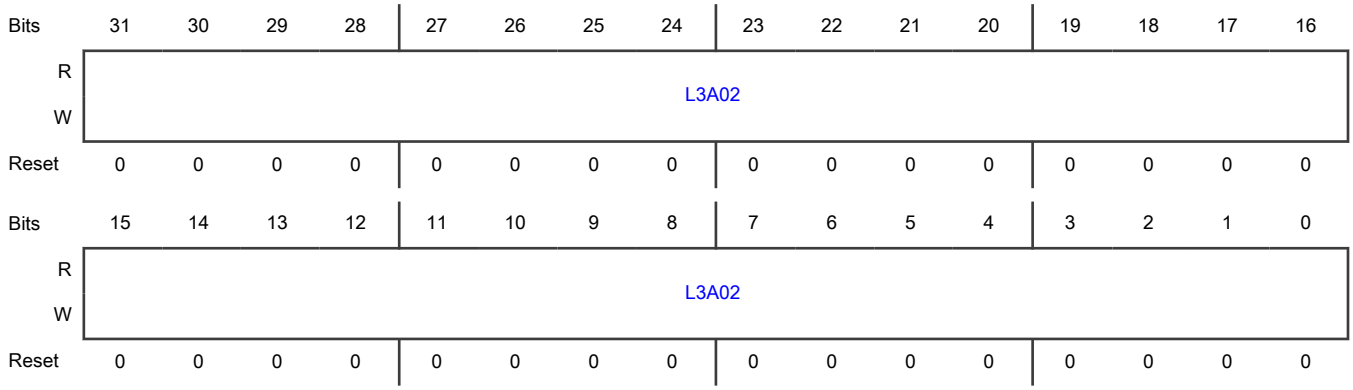
43.8.1.291 Layer 3 Address 0 Register 2 (MAC_LAYER3_ADDR0_REG2)

For IPv4 packets, the Layer 3 Address 0 Register 2 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR0_REG2	970h

Diagram



Fields

Field	Description
31-0 L3A02	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

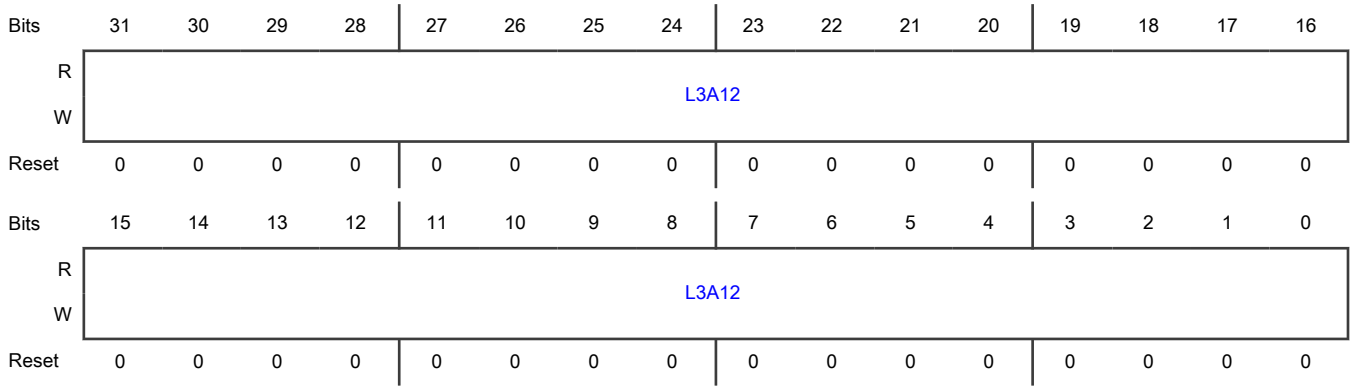
43.8.1.292 Layer 3 Address 0 Register 2 (MAC_LAYER3_ADDR1_REG2)

For IPv4 packets, the Layer 3 Address 1 Register 2 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR1_REG2	974h

Diagram



Fields

Field	Description
31-0 L3A12	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

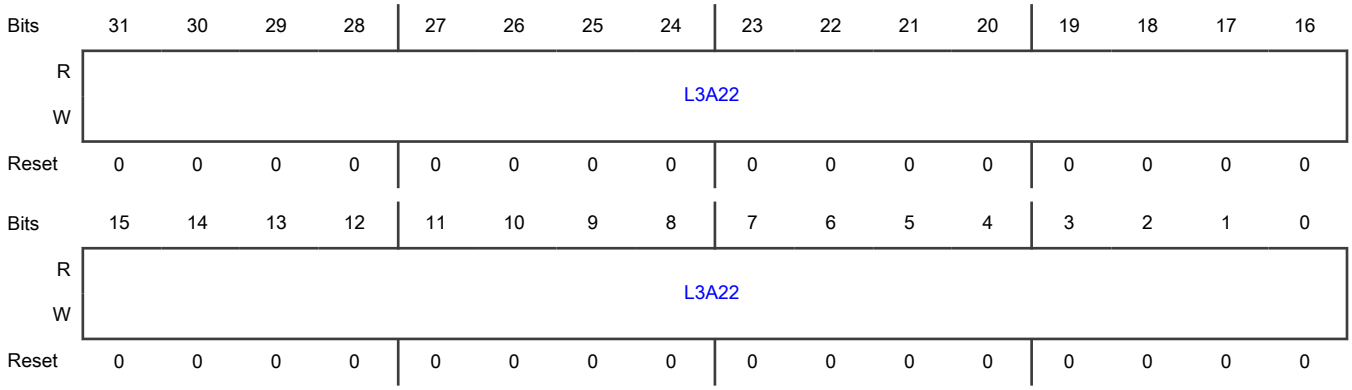
43.8.1.293 Layer 3 Address 2 Register 2 (MAC_LAYER3_ADDR2_REG2)

The Layer 3 Address 2 Register 2 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR2_REG2	978h

Diagram



Fields

Field	Description
31-0 L3A22	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

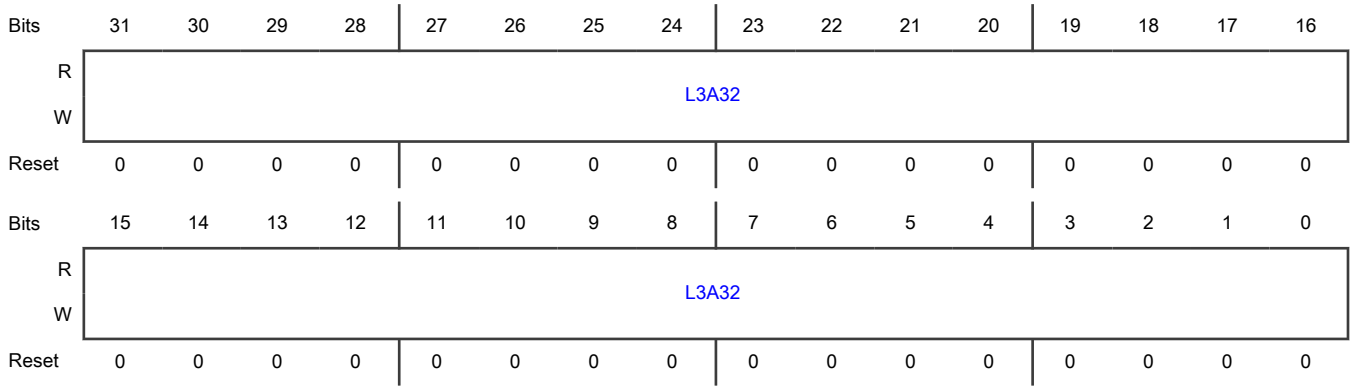
43.8.1.294 Layer 3 Address 3 Register 2 (MAC_LAYER3_ADDR3_REG2)

The Layer 3 Address 3 Register 2 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR3_REG2	97Ch

Diagram



Fields

Field	Description
31-0 L3A32	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

43.8.1.295 Layer 3 and Layer 4 Control of Filter 3 (MAC_L3_L4_CONTROL3)

The Layer 3 and Layer 4 Control register controls the operations of filter 3 of Layer 3 and Layer 4.

Offset

Register	Offset
MAC_L3_L4_CONTROL3	990h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN3	Reserved	DMCHN3			Reserved		L4DPIM3	L4DPM3	L4SPIM3	L4SPM3	Reserved	L4PEN3
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM3				L3HSBM3						L3DAIM3	L3DAM3	L3SAIM3	L3SAM3	Reserved	L3PEN3
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 DMCHEN3	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN3	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM3	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM3 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM3	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM3	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM3	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN3	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM3 bit is set.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM3	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM3	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM3	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM3	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM3	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM3	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 —	Reserved
0 L3PEN3	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

43.8.1.296 Layer 4 Address 3 (MAC_LAYER4_ADDRESS3)

The MAC_LAYER4_ADDRESS4, MAC_L3_L4_CONTROL4, MAC_LAYER3_ADDR0_Reg4, MAC_LAYER3_ADDR1_Reg4, MAC_LAYER3_ADDR2_Reg4 and MAC_LAYER3_ADDR3_Reg4 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

Offset

Register	Offset
MAC_LAYER4_ADDRESS3	994h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 L4DP3	Layer 4 Destination Port Number Field When the L4PEN3 bit is reset and the L4DPM3 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN3 bit is reset and the L4DPM3 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN3 and L4DPM3 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP3	Layer 4 Source Port Number Field When the L4PEN3 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN3 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN3 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

43.8.1.297 Layer 3 Address 0 Register 3 (MAC_LAYER3_ADDR0_REG3)

For IPv4 packets, the Layer 3 Address 0 Register 3 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR0_REG3	9A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A03															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A03															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 L3A03	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

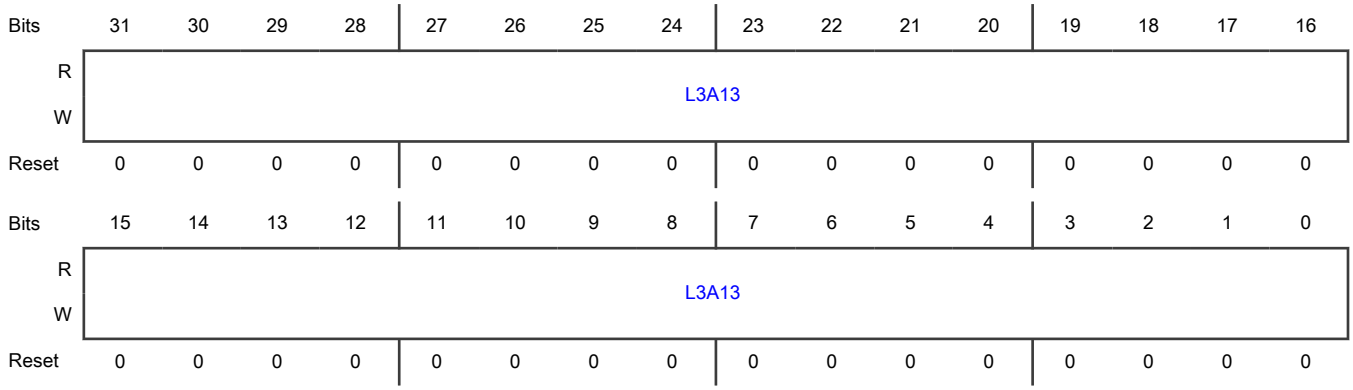
43.8.1.298 Layer 3 Address 1 Register 3 (MAC_LAYER3_ADDR1_REG3)

For IPv4 packets, the Layer 3 Address 1 Register 3 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR1_REG3	9A4h

Diagram



Fields

Field	Description
31-0 L3A13	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets.

Table continues on the next page...

Field	Description
	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

43.8.1.299 Layer 3 Address 2 Register 3 (MAC_LAYER3_ADDR2_REG3)

The Layer 3 Address 2 Register 3 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR2_REG3	9A8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A23															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A23															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 L3A23	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets.

Table continues on the next page...

Field	Description
	When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

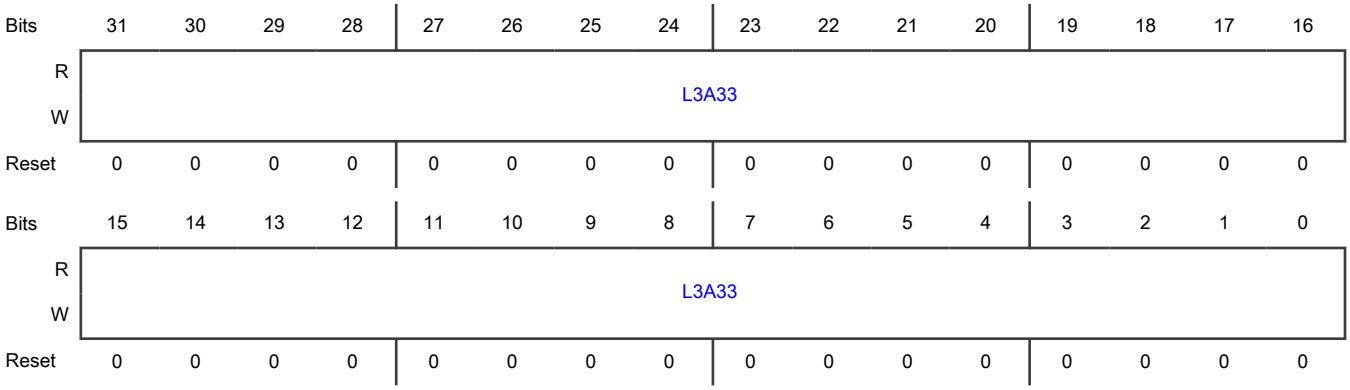
43.8.1.300 Layer 3 Address 3 Register 3 (MAC_LAYER3_ADDR3_REG3)

The Layer 3 Address 3 Register 3 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR3_REG3	9ACh

Diagram



Fields

Field	Description
31-0 L3A33	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

43.8.1.301 Layer 3 and Layer 4 Control of Filter 4 (MAC_L3_L4_CONTROL4)

The Layer 3 and Layer 4 Control register controls the operations of filter 3 of Layer 3 and Layer 4.

Offset

Register	Offset
MAC_L3_L4_CONTROL4	9C0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN4	Reserved	DMCHN4			Reserved		L4DPIM4	L4DPM4	L4SPIM4	L4SPM4	Reserved	L4PEN4
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM4				L3HSBM4						L3DAIM4	L3DAM4	L3SAIM4	L3SAM4	Reserved	L3PEN4
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 DMCHEN4	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN4	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
21 L4DPIM4	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM4 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM4	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM4	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM4	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
16 L4PEN4	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM4bit is set. 0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM4	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM4	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM4	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM4	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM4	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching.

Table continues on the next page...

Table continued from the previous page...

Field	Description
L3SAM4	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN4	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

43.8.1.302 Layer 4 Address 4 (MAC_LAYER4_ADDRESS4)

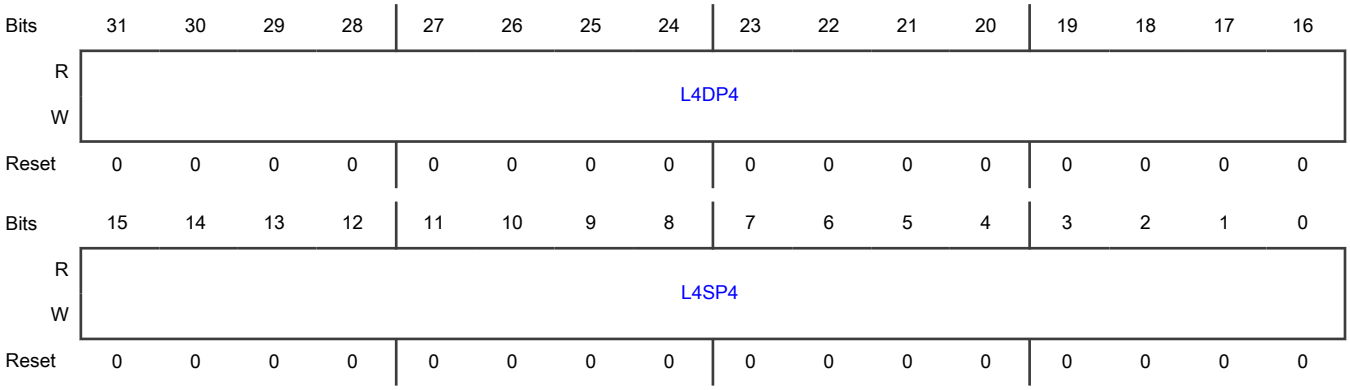
The MAC_LAYER4_ADDRESS4, MAC_L3_L4_CONTROL4, MAC_LAYER3_ADDR0_Reg4, MAC_LAYER3_ADDR1_Reg4, MAC_LAYER3_ADDR2_Reg4 and MAC_LAYER3_ADDR3_Reg4) registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

Offset

Register	Offset
MAC_LAYER4_ADDRESS4	9C4h

Diagram



Fields

Field	Description
31-16 L4DP4	Layer 4 Destination Port Number Field When the L4PEN4 bit is reset and the L4DPM4 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN4 bit is reset and the L4DPM4 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN4 and L4DPM4 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP4	Layer 4 Source Port Number Field When the L4PEN4 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN4 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN4 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

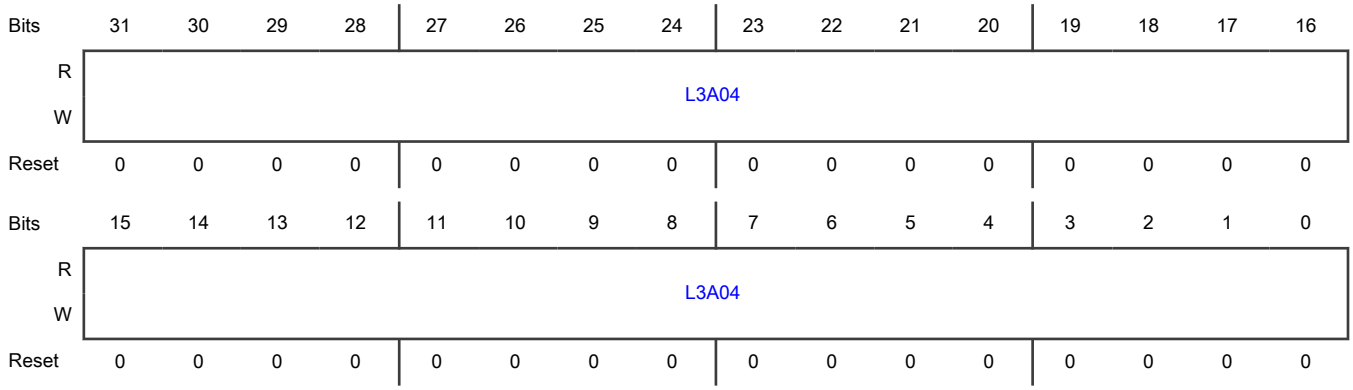
43.8.1.303 Layer 3 Address 0 Register 4 (MAC_LAYER3_ADDR0_REG4)

For IPv4 packets, the Layer 3 Address 0 Register 4 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR0_REG4	9D0h

Diagram



Fields

Field	Description
31-0 L3A04	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

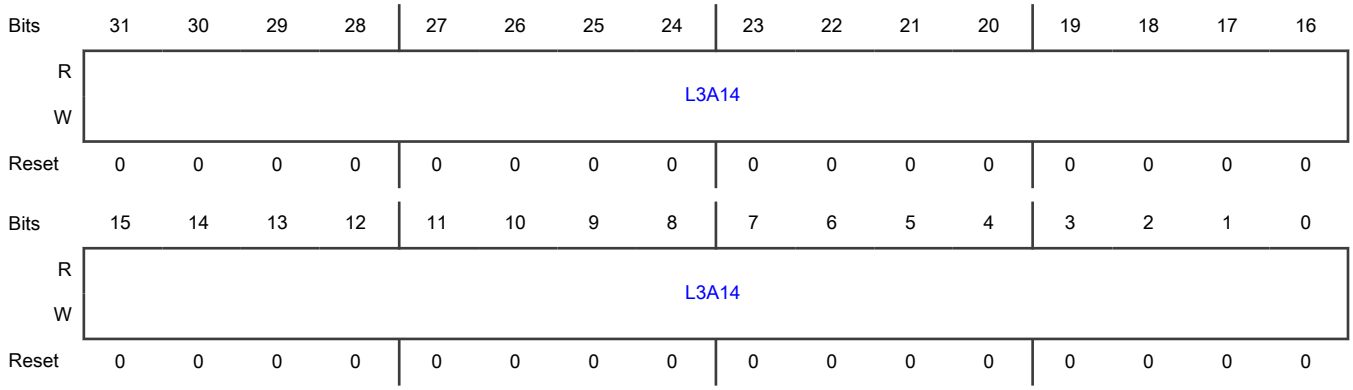
43.8.1.304 Layer 3 Address 1 Register 4 (MAC_LAYER3_ADDR1_REG4)

For IPv4 packets, the Layer 3 Address 1 Register 4 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR1_REG4	9D4h

Diagram



Fields

Field	Description
31-0 L3A14	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

43.8.1.305 Layer 3 Address 2 Register 4 (MAC_LAYER3_ADDR2_REG4)

The Layer 3 Address 2 Register 4 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR2_REG4	9D8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A24															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A24															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 L3A24	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

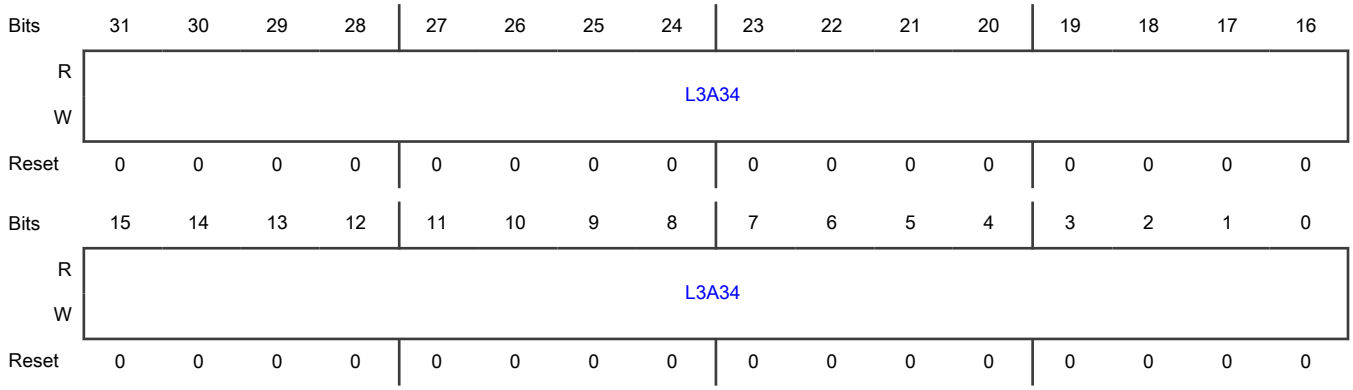
43.8.1.306 Layer 3 Address 3 Register 4 (MAC_LAYER3_ADDR3_REG4)

The Layer 3 Address 3 Register 4 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR3_REG4	9DCh

Diagram



Fields

Field	Description
31-0 L3A34	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

43.8.1.307 Layer 3 and Layer 4 Control of Filter 5 (MAC_L3_L4_CONTROL5)

The Layer 3 and Layer 4 Control register controls the operations of filter 5 of Layer 3 and Layer 4.

Offset

Register	Offset
MAC_L3_L4_CONTROL5	9F0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN5	Reserved	DMCHN5			Reserved		L4DPIM5	L4DPM5	L4SPIM5	L4SPM5	Reserved	L4PEN5
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM5				L3HSBM5						L3DAIM5	L3DAM5	L3SAIM5	L3SAM5	Reserved	L3PEN5
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 DMCHEN5	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN5	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM5	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM5 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM5	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM5	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM5	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN5	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM5 bit is set.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM5	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM5	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM5	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM5	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM5	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM5	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 —	Reserved
0 L3PEN5	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

43.8.1.308 Layer 4 Address 5 (MAC_LAYER4_ADDRESS5)

The MAC_LAYER4_ADDRESS5, MAC_L3_L4_CONTROL5, MAC_LAYER3_ADDR0_Reg5, MAC_LAYER3_ADDR1_Reg5, MAC_LAYER3_ADDR2_Reg5 and MAC_LAYER3_ADDR3_Reg5 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

Offset

Register	Offset
MAC_LAYER4_ADDRESS5	9F4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 L4DP5	Layer 4 Destination Port Number Field When the L4PEN5 bit is reset and the L4DPM5 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN5 bit is reset and the L4DPM5 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN5 and L4DPM5 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP5	Layer 4 Source Port Number Field When the L4PEN5 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN5 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN5 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

43.8.1.309 Layer 3 Address 0 Register 5 (MAC_LAYER3_ADDR0_REG5)

For IPv4 packets, the Layer 3 Address 0 Register 5 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR0_REG5	A00h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A05															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A05															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 L3A05	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

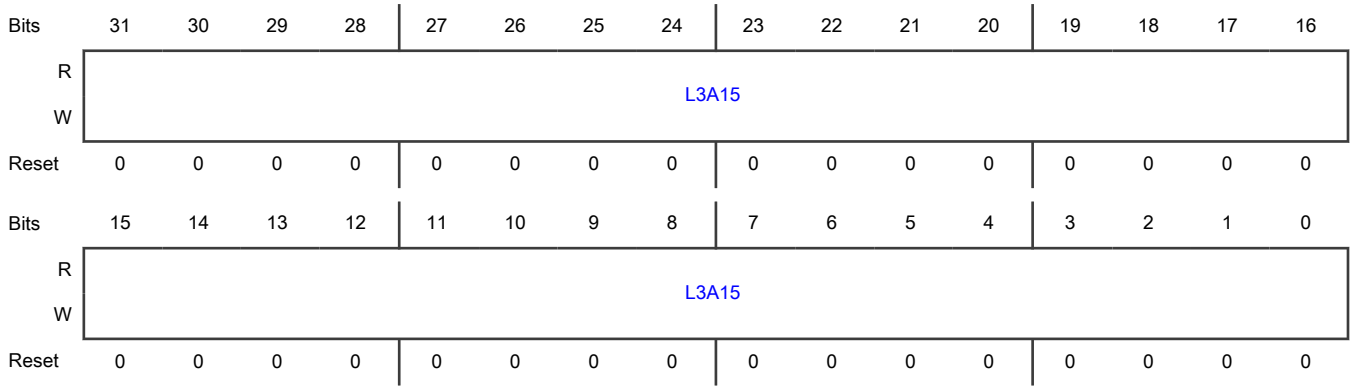
43.8.1.310 Layer 3 Address 1 Register 5 (MAC_LAYER3_ADDR1_REG5)

For IPv4 packets, the Layer 3 Address 1 Register 5 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR1_REG5	A04h

Diagram



Fields

Field	Description
31-0 L3A15	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets.

Table continues on the next page...

Field	Description
	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

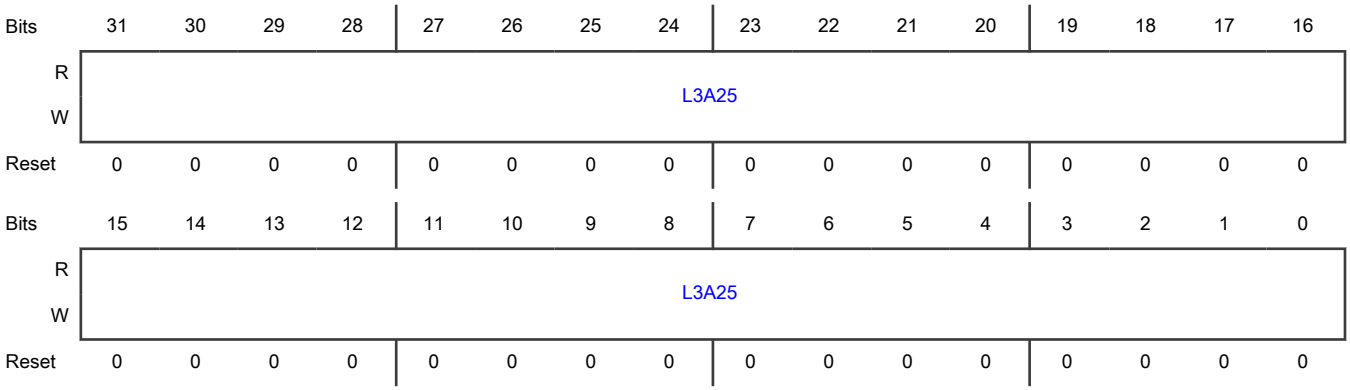
43.8.1.311 Layer 3 Address 2 Register 5 (MAC_LAYER3_ADDR2_REG5)

The Layer 3 Address 2 Register 5 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR2_REG5	A08h

Diagram



Fields

Field	Description
31-0 L3A25	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets.

Table continues on the next page...

Field	Description
	When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

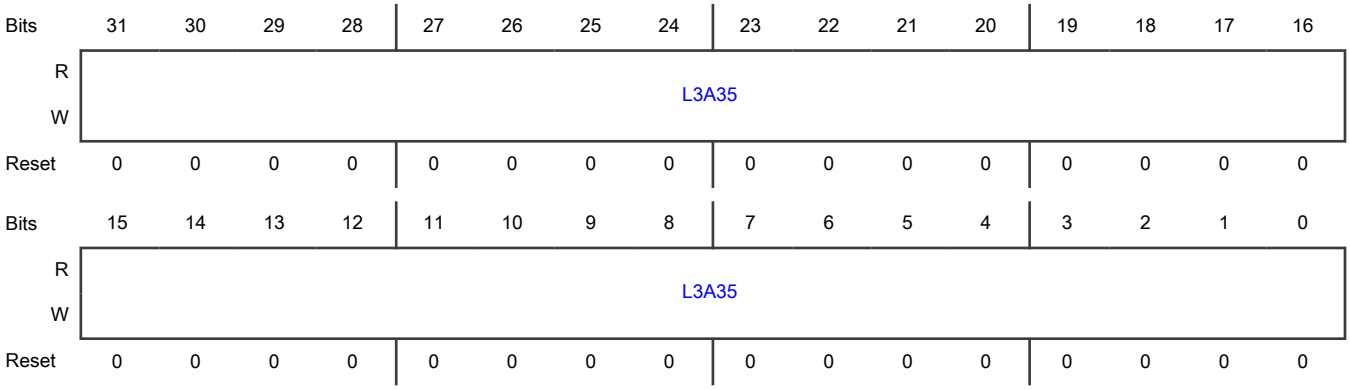
43.8.1.312 Layer 3 Address 3 Register 5 (MAC_LAYER3_ADDR3_REG5)

The Layer 3 Address 3 Register 5 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR3_REG5	A0Ch

Diagram



Fields

Field	Description
31-0 L3A35	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

43.8.1.313 Layer 3 and Layer 4 Control of Filter 6 (MAC_L3_L4_CONTROL6)

The Layer 3 and Layer 4 Control register controls the operations of filter 6 of Layer 3 and Layer 4.

Offset

Register	Offset
MAC_L3_L4_CONTROL6	A20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN6	Reserved	DMCHN6			Reserved		L4DPIM6	L4DPM6	L4SPIM6	L4SPM6	Reserved	L4PEN6
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM6				L3HSBM6						L3DAIM6	L3DAM6	L3SAIM6	L3SAM6	Reserved	L3PEN6
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 DMCHEN6	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN6	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
21 L4DPIM6	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM6 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM6	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM6	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM6	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
16 L4PEN6	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM6 bit is set. 0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM6	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM6	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM6	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM6	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM6	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching.

Table continues on the next page...

Table continued from the previous page...

Field	Description
L3SAM6	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN6	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

43.8.1.314 Layer 4 Address 6 (MAC_LAYER4_ADDRESS6)

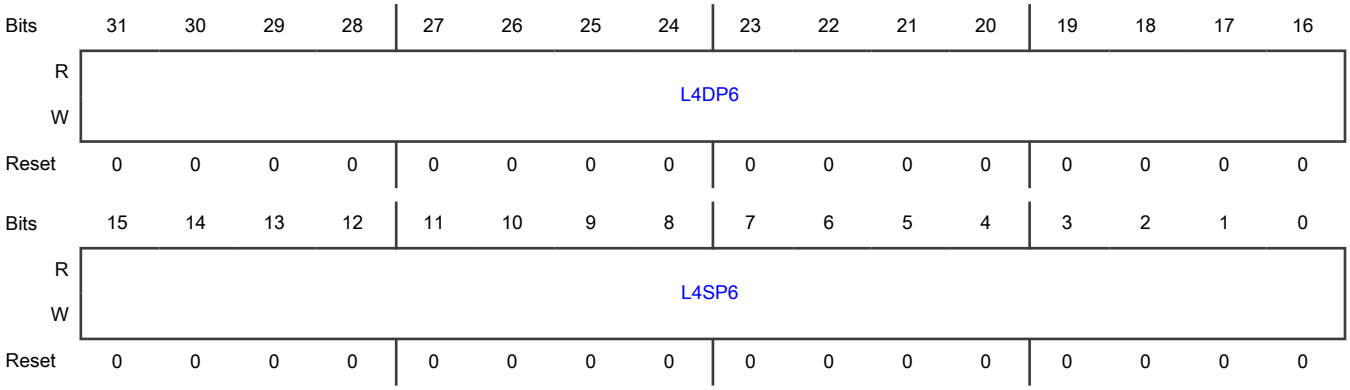
The MAC_LAYER4_ADDRESS6), MAC_L3_L4_CONTROL6, MAC_LAYER3_ADDR0_Reg6, MAC_LAYER3_ADDR1_Reg6, MAC_LAYER3_ADDR2_Reg5 and MAC_LAYER3_ADDR3_Reg6 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

Offset

Register	Offset
MAC_LAYER4_ADDRESS6	A24h

Diagram



Fields

Field	Description
31-16 L4DP6	Layer 4 Destination Port Number Field When the L4PEN6 bit is reset and the L4DPM6 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN6 bit is reset and the L4DPM6 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN6 and L4DPM6 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP6	Layer 4 Source Port Number Field When the L4PEN6 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN6 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN6 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

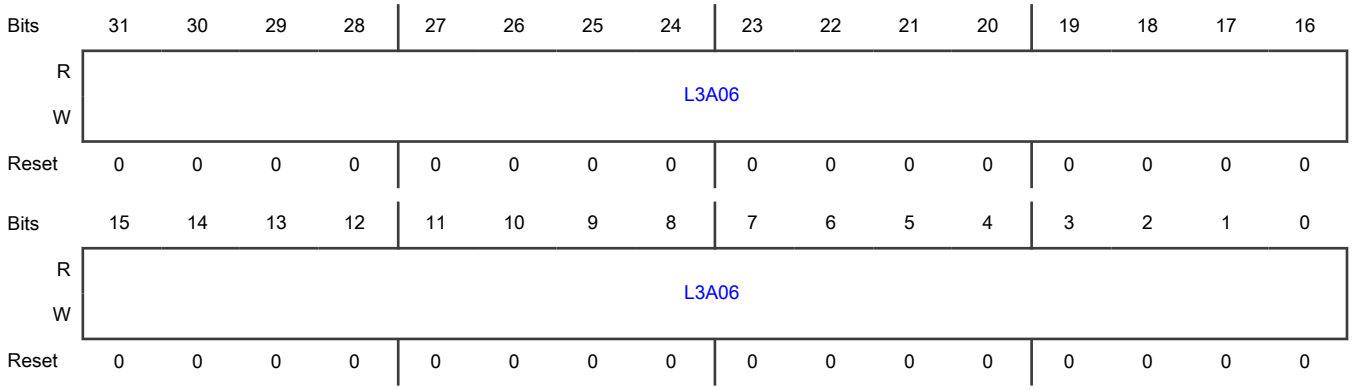
43.8.1.315 Layer 3 Address 0 Register 6 (MAC_LAYER3_ADDR0_REG6)

For IPv4 packets, the Layer 3 Address 0 Register 6 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR0_REG6	A30h

Diagram



Fields

Field	Description
31-0 L3A06	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

43.8.1.316 Layer 3 Address 1 Register 6 (MAC_LAYER3_ADDR1_REG6)

For IPv4 packets, the Layer 3 Address 1 Register 6 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR1_REG6	A34h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A16															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A16															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 L3A16	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

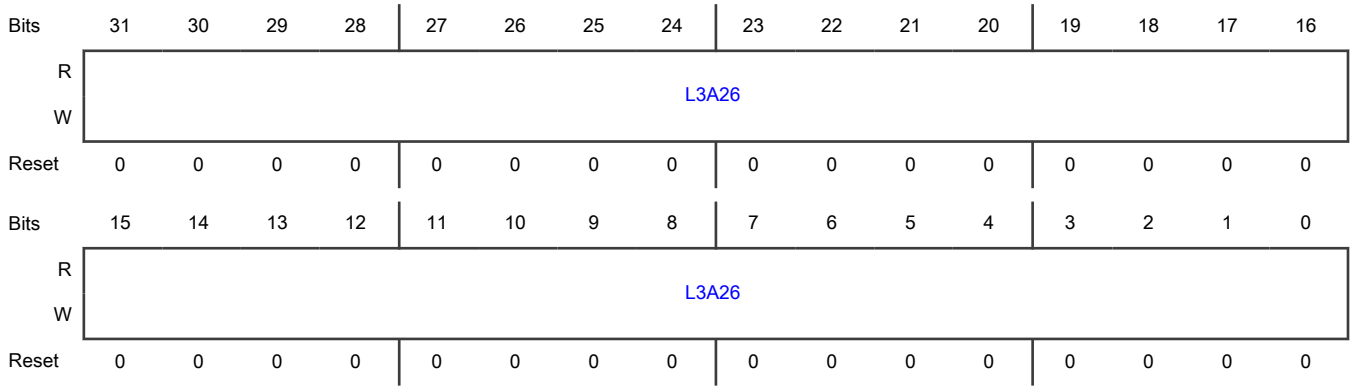
43.8.1.317 Layer 3 Address 2 Register 6 (MAC_LAYER3_ADDR2_REG6)

The Layer 3 Address 2 Register 6 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR2_REG6	A38h

Diagram



Fields

Field	Description
31-0 L3A26	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

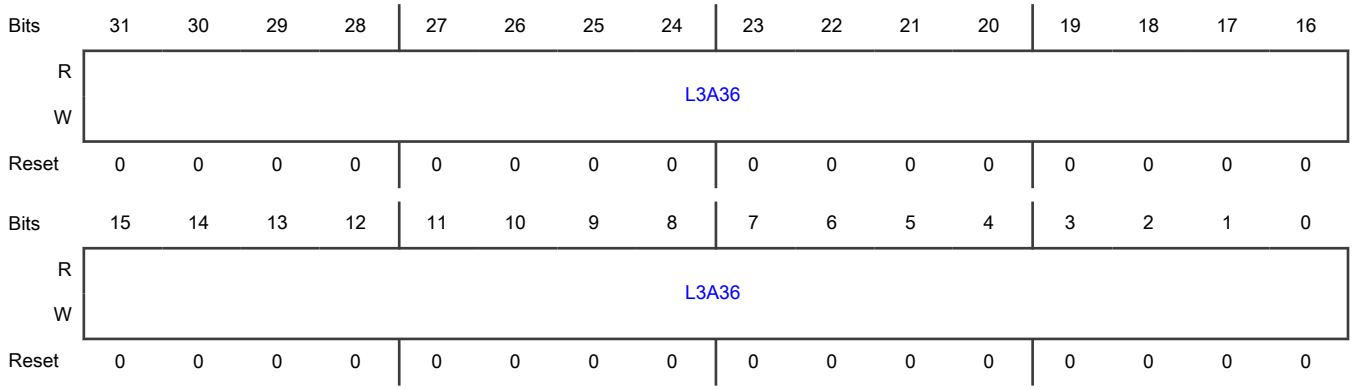
43.8.1.318 Layer 3 Address 3 Register 6 (MAC_LAYER3_ADDR3_REG6)

The Layer 3 Address 3 Register 6 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR3_REG6	A3Ch

Diagram



Fields

Field	Description
31-0 L3A36	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

43.8.1.319 Layer 3 and Layer 4 Control of Filter 0 (MAC_L3_L4_CONTROL7)

The Layer 3 and Layer 4 Control register controls the operations of filter 7 of Layer 3 and Layer 4.

Offset

Register	Offset
MAC_L3_L4_CONTROL7	A50h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN7	Reserved	DMCHN7			Reserved		L4DPIM7	L4DPM7	L4SPIM7	L4SPM7	Reserved	L4PEN7
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM7				L3HSBM7						L3DAIM7	L3DAM7	L3SAIM7	L3SAM7	Reserved	L3PEN7
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 DMCHEN7	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN7	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM7	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM7 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM7	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM7	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM7	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN7	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM7 bit is set.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM7	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM7	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM7	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM7	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM7	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM7	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 —	Reserved
0 L3PEN7	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

43.8.1.320 Layer 4 Address 7 (MAC_LAYER4_ADDRESS7)

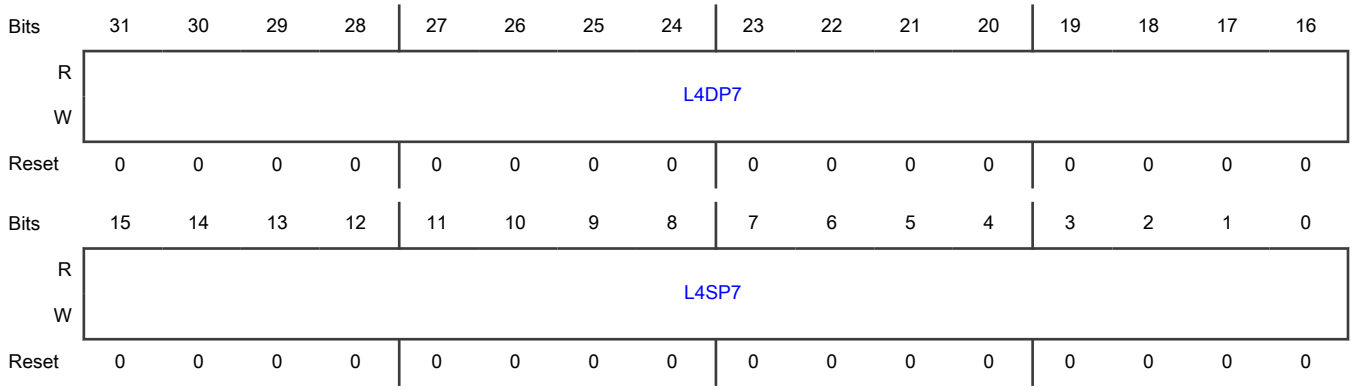
The MAC_LAYER4_ADDRESS7, MAC_L3_L4_CONTROL7, MAC_LAYER3_ADDR0_Reg7, MAC_LAYER3_ADDR1_Reg7, MAC_LAYER3_ADDR2_Reg7 and MAC_LAYER3_ADDR3_Reg7 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

Offset

Register	Offset
MAC_LAYER4_ADDRESS7	A54h

Diagram



Fields

Field	Description
31-16 L4DP7	Layer 4 Destination Port Number Field When the L4PEN7 bit is reset and the L4DPM7 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN7 bit is reset and the L4DPM7 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN7 and L4DPM7 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP7	Layer 4 Source Port Number Field When the L4PEN7 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN7 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN7 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

43.8.1.321 Layer 3 Address 0 Register 7 (MAC_LAYER3_ADDR0_REG7)

For IPv4 packets, the Layer 3 Address 0 Register 7 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR0_REG7	A60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A07															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A07															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 L3A07	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

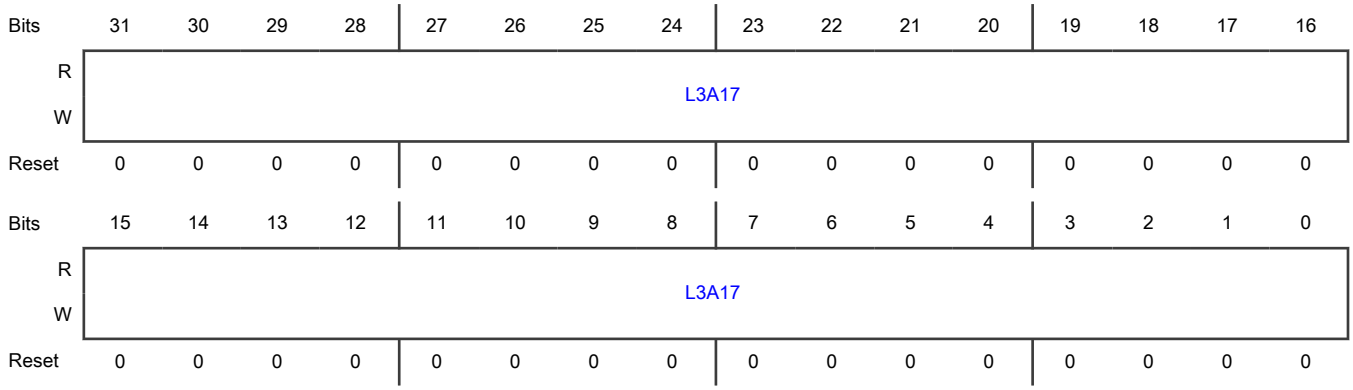
43.8.1.322 Layer 3 Address 1 Register 7 (MAC_LAYER3_ADDR1_REG7)

For IPv4 packets, the Layer 3 Address 1 Register 7 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR1_REG7	A64h

Diagram



Fields

Field	Description
31-0 L3A17	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets.

Table continues on the next page...

Field	Description
	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

43.8.1.323 Layer 3 Address 2 Register 7 (MAC_LAYER3_ADDR2_REG7)

The Layer 3 Address 2 Register 7 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR2_REG7	A68h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A27															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A27															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 L3A27	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets.

Table continues on the next page...

Field	Description
	When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

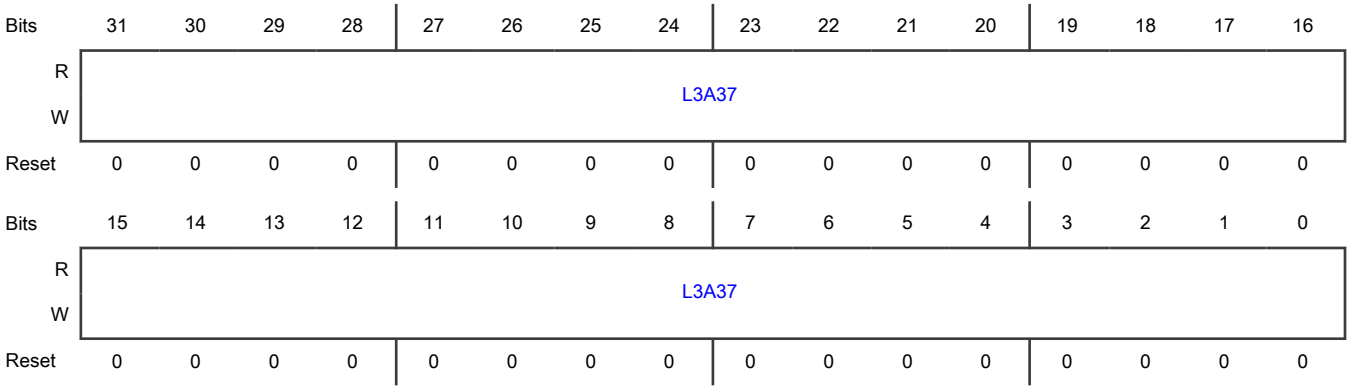
43.8.1.324 Layer 3 Address 3 Register 7 (MAC_LAYER3_ADDR3_REG7)

The Layer 3 Address 3 Register 7 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

Offset

Register	Offset
MAC_LAYER3_ADDR3_REG7	A6Ch

Diagram



Fields

Field	Description
31-0 L3A37	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

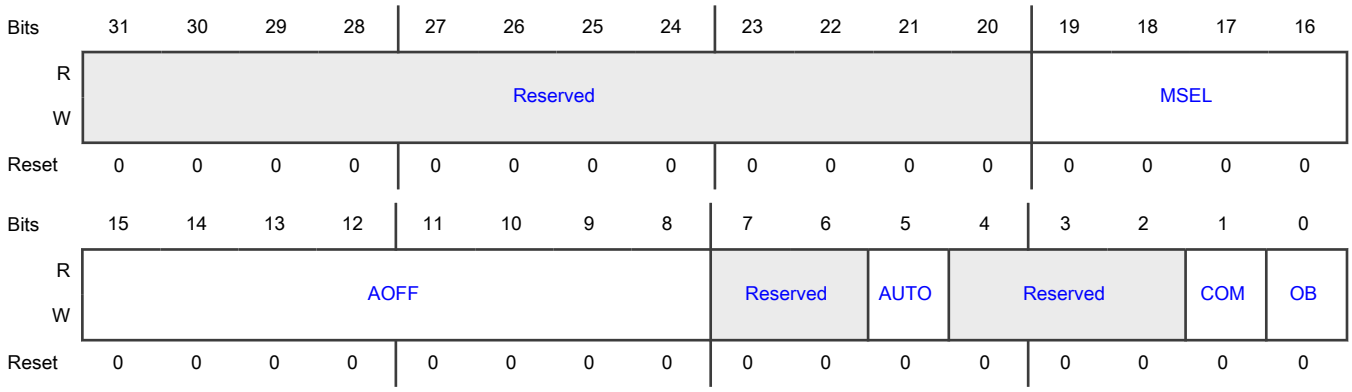
43.8.1.325 MAC_INDIR_ACCESS_CTRL (MAC_INDIR_ACCESS_CTRL)

This register provides the Indirect Access control and status for MAC_<MSEl>_<AOFF> registers.

Offset

Register	Offset
MAC_INDIR_ACCESS_CTRL	A70h

Diagram



Fields

Field	Description
31-20 —	Reserved
19-16 MSEL	Mode Select This field is used in indirect access of MAC_<MSEL>_<AOFF>. This field must be set along with initiation of read/write to MAC_<MSEL>_<AOFF> and should not be changed until the OB is reset. Values: 0000: ESTTyp_RXQ_ (Type based RXQ mappping) 0001-1111: Reserved for future use.
15-8 AOFF	Address Offset This field is used in indirect access of MAC_<MSEL>_<AOFF>. This field must be set along with initiation of read/write to MAC_<MSEL>_<AOFF> and should not be changed until the OB is reset. Values: 00000000: IndReg0(Indirect register 0) 00000001: IndReg1(Indirect register 1) ... 00000111: IndReg7(Indirect register 7) 00001000-11111111: Reserved for future use.
7-6 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
5 AUTO	Auto increment enable 0 - AOFF is not incremented automatically. Software should program the correct Address Offset for each access. 1 - AOFF is incremented by 1. Software should ensure not to cause a wrap condition. Byte wise read/write is not supported when auto increment is enabled.
4-2 —	Reserved
1 COM	Command type. Indicates the register access type. 0 - Indicates a write operation 1 - Indicates a read operation
0 OB	Operation Busy This bit is set along with a read or write command for initiating the indirect access to MAC_<MSEL>_<AOFF> register. This bit is reset when the read or write command to MAC_<MSEL>_<AOFF> register is complete. The next indirect register access can be initiated only after this bit is reset. During a write operation, the bit is reset only after the data has been written into MAC_<MSEL>_<AOFF> register. During a read operation, the data should be read from MAC_INDIR_ACCESS_DATA register only after this bit is reset.

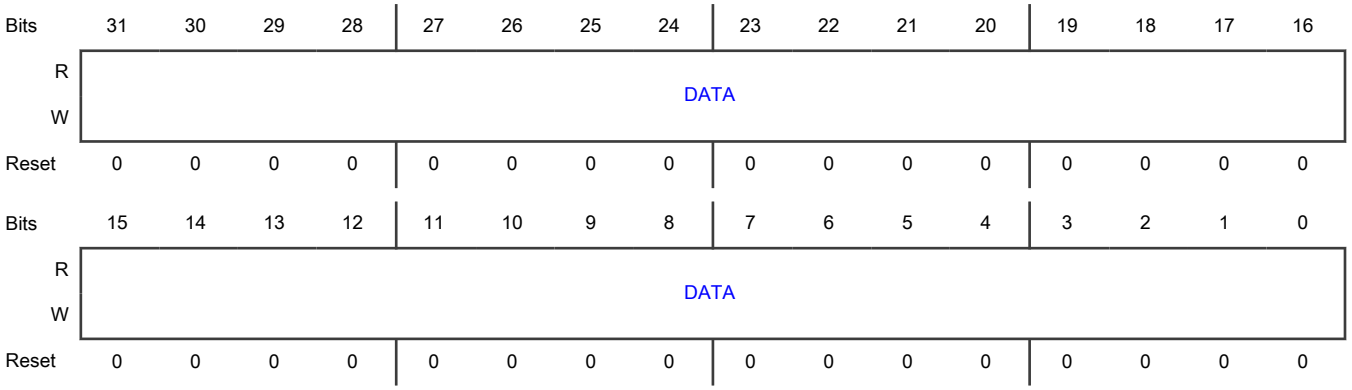
43.8.1.326 MAC_INDIR_ACCESS_DATA (MAC_INDIR_ACCESS_DATA)

This register holds the read/write data for Indirect Access of MAC_<MSEL_<AOFF> registers.

Offset

Register	Offset
MAC_INDIR_ACCESS_DATA	A74h

Diagram



Fields

Field	Description
31-0 DATA	This field contains data to read/write for Indirect address access associated with MAC_INDIR_ACCESS_CTRL register.

43.8.1.327 Timestamp Control (MAC_TIMESTAMP_CONTROL)

This register controls the operation of the System Time generator and processing of PTP packets for timestamping in the Receiver.

Offset

Register	Offset
MAC_TIMESTAMP_CONTROL	B00h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			AV8021A...	Reserved			TXTSSTSM	Reserved			ESTI	CSC	TSENMAC...	SNAPTYPSEL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSMSTRE...	TSEVNTE...	TSIPV4E...	TSIPV6E...	TSIPENA	TSVER2E...	TSCTRLS...	TSENALL	Reserved	PTGE	TSADDREG	Reserved	TSUPDT	TSINIT	TSCFUPDT	TSENA
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 AV8021ASMEN	AV 802. AV 802.1AS Mode Enable When this bit is set, the MAC processes only untagged PTP over Ethernet packets for providing PTP status and capturing timestamp snapshots, that is, IEEE 802.1AS mode of operation. When PTP offload feature is enabled, for the purpose of PTP offload, the transport specific field in the PTP header is generated and checked based on the value of this bit. 0 - AV 802.1AS Mode is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - AV 802.1AS Mode is enabled
27-25 —	Reserved
24 TXTSSTSM	Transmit Timestamp Status Mode When this bit is set, the MAC overwrites the earlier transmit timestamp status even if it is not read by the software. Transmit Timestamp Status Mode When this bit is set, the MAC overwrites the earlier transmit timestamp status even if it is not read by the software. The MAC indicates this by setting the TXTSSMIS bit of the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS register. When this bit is reset, the MAC ignores the timestamp status of current packet if the timestamp status of previous packet is not read by the software. The MAC indicates this by setting the TXTSSMIS bit of the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS register. 0 - Transmit Timestamp Status Mode is disabled 1 - Transmit Timestamp Status Mode is enabled
23-21 —	Reserved
20 ESTI	External System Time Input When this bit is set, the MAC uses the external 64-bit reference System Time input for the following: - To take the timestamp provided as status - To insert the timestamp in transmit PTP packets when One-step Timestamp or Timestamp Offload feature is enabled. External System Time Input When this bit is set, the MAC uses the external 64-bit reference System Time input for the following: - To take the timestamp provided as status - To insert the timestamp in transmit PTP packets when One-step Timestamp or Timestamp Offload feature is enabled. When this bit is reset, the MAC uses the internal reference System Time. 0 - External System Time Input is disabled 1 - External System Time Input is enabled
19 CSC	Enable checksum correction during OST for PTP over UDP/IPv4 packets When this bit is set, the last two bytes of PTP message sent over UDP/IPv4 is updated to keep the UDP checksum correct, for changes made to origin timestamp and/or correction field as part of one step timestamp operation. Enable checksum correction during OST for PTP over UDP/IPv4 packets When this bit is set, the last two bytes of PTP message sent over UDP/IPv4 is updated to keep the UDP checksum correct, for changes made to origin timestamp and/or correction field as part of one step timestamp operation. The application shall form the packet with these two dummy bytes. When reset, no updates are done to keep the UDP checksum correct. The application shall form the packet with UDP checksum set to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - checksum correction during OST for PTP over UDP/IPv4 packets is disabled 1 - checksum correction during OST for PTP over UDP/IPv4 packets is enabled
18 TSENMACADDR	Enable MAC Address for PTP Packet Filtering When this bit is set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP packets when PTP is directly sent over Ethernet. Enable MAC Address for PTP Packet Filtering When this bit is set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP packets when PTP is directly sent over Ethernet. When this bit is set, received PTP packets with DA containing a special multicast or unicast address that matches the one programmed in MAC address registers are considered for processing as indicated below, when PTP is directly sent over Ethernet. For normal time stamping operation, MAC address registers 0 to 31 is considered for unicast destination address matching. For PTP offload, only MAC address register 0 is considered for unicast destination address matching. 0 - MAC Address for PTP Packet Filtering is disabled 1 - MAC Address for PTP Packet Filtering is enabled
17-16 SNAPTYPSEL	Select PTP packets for Taking Snapshots These bits, along with Bits 15 and 14, decide the set of PTP packet types for which snapshot needs to be taken. Select PTP packets for Taking Snapshots These bits, along with Bits 15 and 14, decide the set of PTP packet types for which snapshot needs to be taken. The encoding is given in Timestamp Snapshot Dependency on Register Bits Table.
15 TSMSTRENA	Enable Snapshot for Messages Relevant to Master When this bit is set, the snapshot is taken only for the messages that are relevant to the master node. Enable Snapshot for Messages Relevant to Master When this bit is set, the snapshot is taken only for the messages that are relevant to the master node. Otherwise, the snapshot is taken for the messages relevant to the slave node. 0 - Snapshot for Messages Relevant to Master is disabled 1 - Snapshot for Messages Relevant to Master is enabled
14 TSEVNTENA	Enable Timestamp Snapshot for Event Messages When this bit is set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). Enable Timestamp Snapshot for Event Messages When this bit is set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). When this bit is reset, the snapshot is taken for all messages except Announce, Management, and Signaling. For more information about the timestamp snapshots, see Timestamp Snapshot Dependency on Register Bits Table. 0 - Timestamp Snapshot for Event Messages is disabled 1 - Timestamp Snapshot for Event Messages is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
13 TSIPV4ENA	Enable Processing of PTP Packets Sent over IPv4-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv4-UDP packets. Enable Processing of PTP Packets Sent over IPv4-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv4-UDP packets. When this bit is reset, the MAC ignores the PTP transported over IPv4-UDP packets. This bit is set by default. 0 - Processing of PTP Packets Sent over IPv4-UDP is disabled 1 - Processing of PTP Packets Sent over IPv4-UDP is enabled
12 TSIPV6ENA	Enable Processing of PTP Packets Sent over IPv6-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv6-UDP packets. Enable Processing of PTP Packets Sent over IPv6-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv6-UDP packets. When this bit is clear, the MAC ignores the PTP transported over IPv6-UDP packets. 0 - Processing of PTP Packets Sent over IPv6-UDP is disabled 1 - Processing of PTP Packets Sent over IPv6-UDP is enabled
11 TSIPENA	Enable Processing of PTP over Ethernet Packets When this bit is set, the MAC receiver processes the PTP packets encapsulated directly in the Ethernet packets. Enable Processing of PTP over Ethernet Packets When this bit is set, the MAC receiver processes the PTP packets encapsulated directly in the Ethernet packets. When this bit is reset, the MAC ignores the PTP over Ethernet packets. 0 - Processing of PTP over Ethernet Packets is disabled 1 - Processing of PTP over Ethernet Packets is enabled
10 TSVER2ENA	Enable PTP Packet Processing for Version 2 Format When this bit is set, the IEEE 1588 version 2 format is used to process the PTP packets. Enable PTP Packet Processing for Version 2 Format When this bit is set, the IEEE 1588 version 2 format is used to process the PTP packets. When this bit is reset, the IEEE 1588 version 1 format is used to process the PTP packets. The IEEE 1588 formats are described in 'PTP Processing and Control'. 0 - PTP Packet Processing for Version 2 Format is disabled 1 - PTP Packet Processing for Version 2 Format is enabled
9 TSCTRLSSR	Timestamp Digital or Binary Rollover Control When this bit is set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (High) seconds. Timestamp Digital or Binary Rollover Control When this bit is set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (High) seconds. When this bit is reset, the rollover value of sub-second register is 0x7FFF_FFFF. The sub-second increment must be programmed correctly depending on the PTP reference clock frequency and the value of this bit.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Timestamp Digital or Binary Rollover Control is disabled 1 - Timestamp Digital or Binary Rollover Control is enabled
8 TSENALL	Enable Timestamp for All Packets When this bit is set, the timestamp snapshot is enabled for all packets received by the MAC. 0 - Timestamp for All Packets disabled 1 - Timestamp for All Packets enabled
7 —	Reserved
6 PTGE	Presentation Time Generation Enable When this bit is set the Presentation Time generation will be enabled. 0 - Presentation Time Generation is disabled 1 - Presentation Time Generation is enabled
5 TSADDREG	Update Addend Register When this bit is set, the content of the Timestamp Addend register is updated in the PTP block for fine correction. Update Addend Register When this bit is set, the content of the Timestamp Addend register is updated in the PTP block for fine correction. This bit is cleared when the update is complete. This bit should be zero before it is set. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Addend Register is not updated 1 - Addend Register is updated
4 —	Reserved
3 TSUPDT	Update Timestamp When this bit is set, the system time is updated (added or subtracted) with the value specified in MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. Update Timestamp When this bit is set, the system time is updated (added or subtracted) with the value specified in MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. This bit should be zero before updating it. This bit is reset when the update is complete in hardware. The Timestamp Higher Word register (if enabled during core configuration) is not updated. When Media Clock Generation and Recovery is configured (DWC_EQOS_FLEXI_PPS_OUT_EN) and enabled MAC_Presn_Time_Updt should also be updated before setting this field. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Timestamp is not updated 1 - Timestamp is updated

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 TSINIT	Initialize Timestamp When this bit is set, the system time is initialized (overwritten) with the value specified in the MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. Initialize Timestamp When this bit is set, the system time is initialized (overwritten) with the value specified in the MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. This bit should be zero before it is updated. This bit is reset when the initialization is complete. The Timestamp Higher Word register (if enabled during core configuration) can only be initialized. When Media Clock Generation and Recovery is configured (DWC_EQOS_FLEXI_PPS_OUT_EN) and enabled MAC_Presn_Time_Updt should also be updated before setting this field. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Timestamp is not initialized 1 - Timestamp is initialized
1 TSCFUPDT	Fine or Coarse Timestamp Update When this bit is set, the Fine method is used to update system timestamp. Fine or Coarse Timestamp Update When this bit is set, the Fine method is used to update system timestamp. When this bit is reset, Coarse method is used to update the system timestamp. 0 - Coarse method is used to update system timestamp 1 - Fine method is used to update system timestamp
0 TSENA	Enable Timestamp When this bit is set, the timestamp is added for Transmit and Receive packets. Enable Timestamp When this bit is set, the timestamp is added for Transmit and Receive packets. When disabled, timestamp is not added for transmit and receive packets and the Timestamp Generator is also suspended. You need to initialize the Timestamp (system time) after enabling this mode. On the Receive side, the MAC processes the 1588 packets only if this bit is set. 0 - Timestamp is disabled 1 - Timestamp is enabled

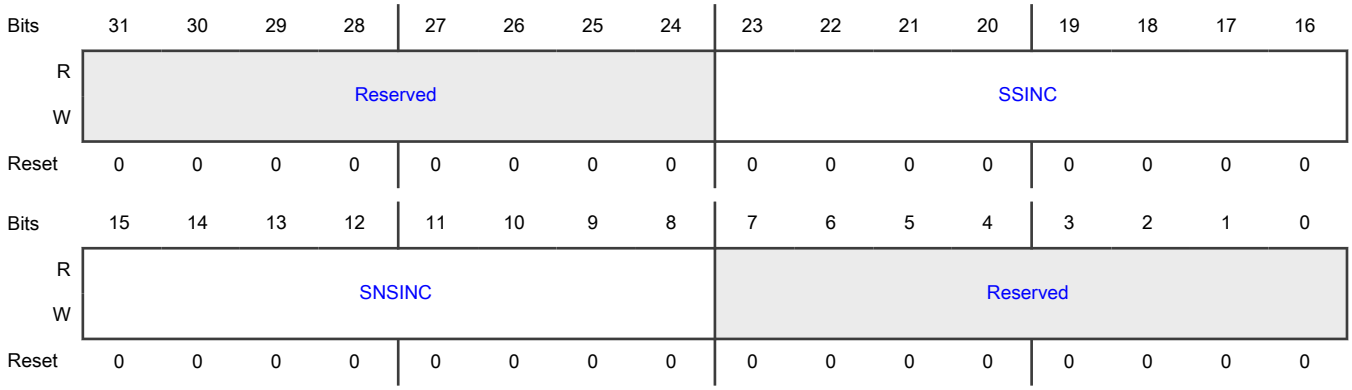
43.8.1.328 Subsecond Increment (MAC_SUB_SECOND_INCREMENT)

This register specifies the value to be added to the internal system time register every cycle of clk_ptp_ref_i clock.

Offset

Register	Offset
MAC_SUB_SECOND_INCREMENT	B04h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 SSINC	Sub-second Increment Value The value programmed in this field is accumulated every clock cycle (of clk_ptp_i) with the contents of the sub-second register. Sub-second Increment Value The value programmed in this field is accumulated every clock cycle (of clk_ptp_i) with the contents of the sub-second register. For example, when the PTP clock is 50 MHz (period is 20 ns), you should program 20 (0x14) when the System Time Nanoseconds register has an accuracy of 1 ns [Bit 9 (TSCTRLSSR) is set in TIMESTAMP_CONTROL]. When TSCTRLSSR is clear, the Nanoseconds register has a resolution of ~0.465 ns. In this case, you should program a value of 43 (0x2B) which is derived by 20 ns/0.465.
15-8 SNSINC	Sub-nanosecond Increment Value This field contains the sub-nanosecond increment value, represented in nanoseconds multiplied by 2^8. Sub-nanosecond Increment Value This field contains the sub-nanosecond increment value, represented in nanoseconds multiplied by 2^8. This value is accumulated with the sub-nanoseconds field of the subsecond register. For example, when TSCTRLSSR field in the TIMESTAMP_CONTROL register is set. and if the required increment is 5.3ns, then SSINC should be 0x05 and SNSINC should be 0x4C.
7-0 —	Reserved

43.8.1.329 System Time Seconds (MAC_SYSTEM_TIME_SECONDS)

The System Time Seconds register, along with System Time Nanoseconds register, indicates the current value of the system time maintained by the MAC. Though it is updated on a continuous basis, there is some delay from the actual time because of clock domain transfer latencies (from clk_ptp_ref_i to CSR clock).

Offset

Register	Offset
MAC_SYSTEM_TIME_SECONDS	B08h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TSS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TSS	Timestamp Second The value in this field indicates the current value in seconds of the System Time maintained by the MAC.

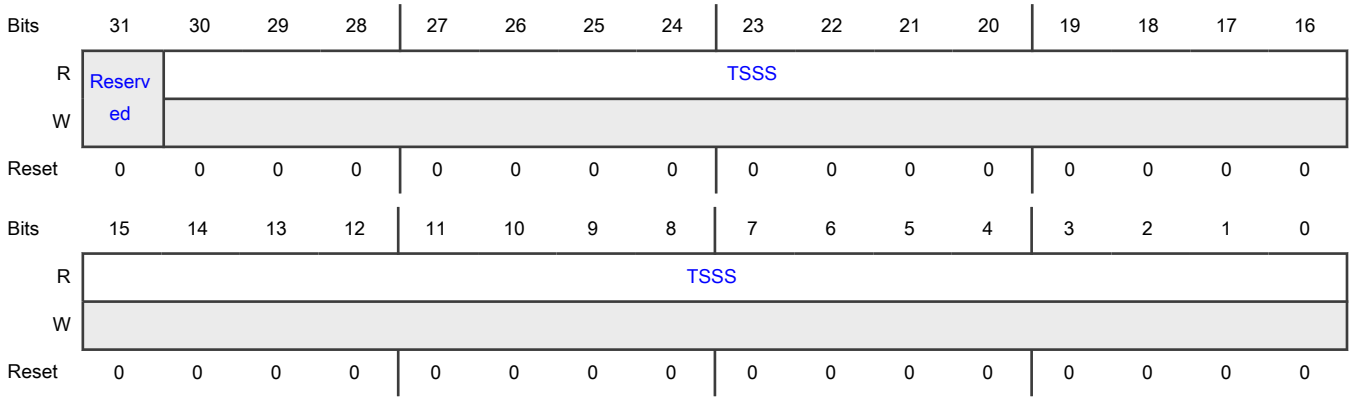
43.8.1.330 System Time Nanoseconds (MAC_SYSTEM_TIME_NANOSECONDS)

The System Time Nanoseconds register, along with System Time Seconds register, indicates the current value of the system time maintained by the MAC.

Offset

Register	Offset
MAC_SYSTEM_TIME_NANOSECONDS	B0Ch

Diagram



Fields

Field	Description
31 —	Reserved
30-0 TSSS	Timestamp Sub Seconds The value in this field has the sub-second representation of time, with an accuracy of 0. Timestamp Sub Seconds The value in this field has the sub-second representation of time, with an accuracy of 0.46 ns. When Bit 9 is set in TIMESTAMP_CONTROL, each bit represents 1 ns. The maximum value is 0x3B9A_C9FF after which it rolls-over to zero.

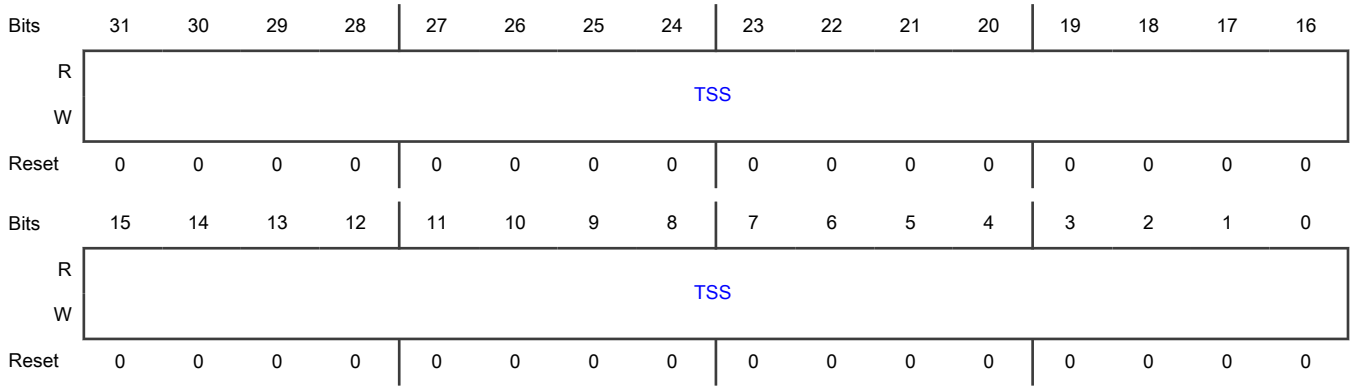
43.8.1.331 System Time Seconds Update (MAC_SYSTEM_TIME_SECONDS_UPDATE)

The System Time Seconds Update register, along with the System Time Nanoseconds Update register, initializes or updates the system time maintained by the MAC. You must write both registers before setting the TSINIT or TSUPDT bits in DWC_eqos_top_map/EQOS_MAC/MAC_TIMESTAMP_CONTROL.

Offset

Register	Offset
MAC_SYSTEM_TIME_SECONDS_UPDATE	B10h

Diagram



Fields

Field	Description
31-0 TSS	Timestamp Seconds The value in this field is the seconds part of the update. Timestamp Seconds The value in this field is the seconds part of the update. When ADDSUB is reset, this field must be programmed with the seconds part of the update value. When ADDSUB is set, this field must be programmed with the complement of the seconds part of the update value. For example, if 2.000000001 seconds need to be subtracted from the system time, the TSS field in the MAC_Timestamp_Seconds_Update register must be 0xFFFF_FFFE (that is, 2^32 - 2).

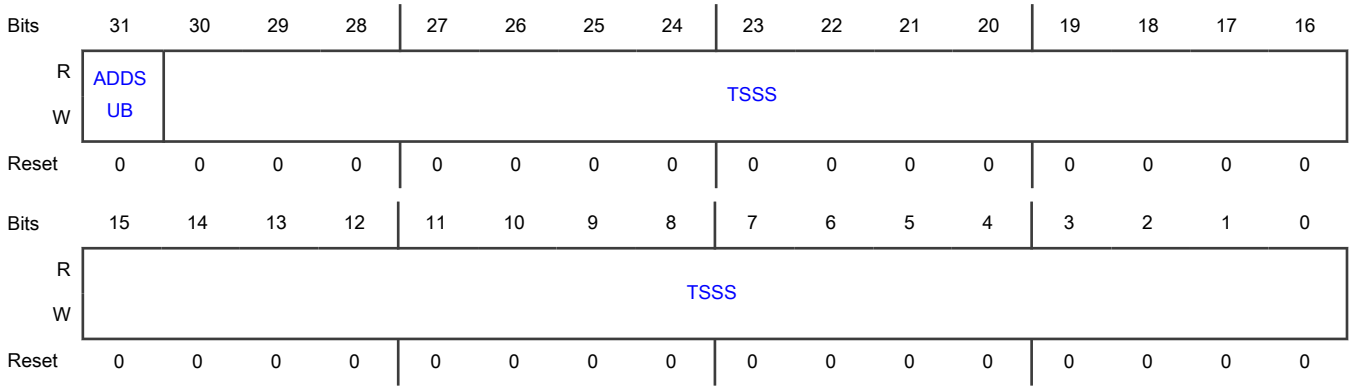
43.8.1.332 System Time Nanoseconds Update (MAC_SYSTEM_TIME_NANOSECONDS_UPDATE)

MAC System Time Nanoseconds Update register.

Offset

Register	Offset
MAC_SYSTEM_TIME_NANOSECONDS_UPDATE	B14h

Diagram



Fields

Field	Description
31 ADDSUB	Add or Subtract Time When this bit is set, the time value is subtracted with the contents of the update register. Add or Subtract Time When this bit is set, the time value is subtracted with the contents of the update register. When this bit is reset, the time value is added with the contents of the update register. 0 - Add time 1 - Subtract time
30-0 TSSS	Timestamp Sub Seconds The value in this field is the sub-seconds part of the update. Timestamp Sub Seconds The value in this field is the sub-seconds part of the update. When ADDSUB is reset, this field must be programmed with the sub-seconds part of the update value, with an accuracy based on the TSCTRLSSR bit of the TIMESTAMP_CONTROL register. When ADDSUB is set, this field must be programmed with the complement of the sub-seconds part of the update value as described below. When TSCTRLSSR bit in MAC_TIMESTAMP_CONTROL is set, the programmed value must be 10^9 - <sub-second value>. When TSCTRLSSR bit in MAC_TIMESTAMP_CONTROL is reset, the programmed value must be 2^31 - <sub-second_value>. When the TSCTRLSSR bit is reset in the TMAC_IMESTAMP_CONTROL register, each bit represents an accuracy of 0.46 ns. When the TSCTRLSSR bit is set in the MAC_TIMESTAMP_CONTROL register, each bit represents 1 ns and the programmed value should not exceed 0x3B9A_C9FF. For example, if 2.000000001 seconds need to be subtracted from the system time, then the TSSS field in the MAC_Timestamp_Nanoseconds_Update register must be 0x7FFF_FFFF (that is, 2^31 - 1), when TSCTRLSSR bit in MAC_TIMESTAMP_CONTROL is reset and 0x3B9A_C9FF (that is, 10^9 - 1), when TSCTRLSSR bit in MAC_TIMESTAMP_CONTROL is set.

43.8.1.333 Timestamp Addend (MAC_TIMESTAMP_ADDEND)

Timestamp Addend register. This register value is used only when the system time is configured for Fine Update mode (TSCFUPDT bit in the MAC_TIMESTAMP_CONTROL register). The content of this register is added to a 32-bit accumulator in every clock cycle (of clk_ptp_ref_i) and the system time is updated whenever the accumulator overflows.

Offset

Register	Offset
MAC_TIMESTAMP_ADDEND	B18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TSAR	Timestamp Addend Register This field indicates the 32-bit time value to be added to the Accumulator register to achieve time synchronization.

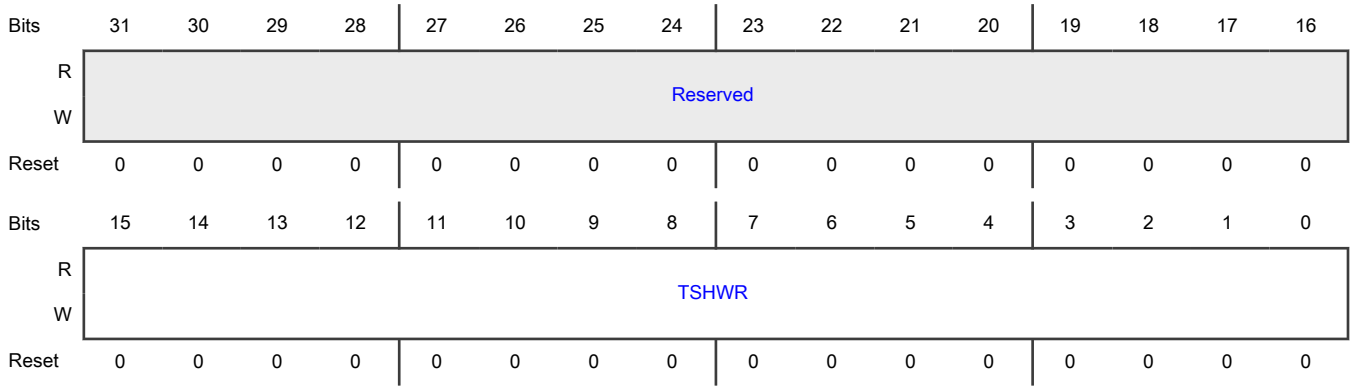
43.8.1.334 System Time - Higher Word Seconds (MAC_SYSTEM_TIME_HIGHER_WORD_SECONDS)

System Time - Higher Word Seconds register.

Offset

Register	Offset
MAC_SYSTEM_TIME_HIGHER_WORD_SECONDS	B1Ch

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 TSHWR	Timestamp Higher Word Register This field contains the most-significant 16-bits of timestamp seconds value. Timestamp Higher Word Register This field contains the most-significant 16-bits of timestamp seconds value. This register is optional. You can add this register by selecting the Add IEEE 1588 Higher Word Register option. This register is directly written to initialize the value and it is incremented when there is an overflow from 32-bits of the System Time Seconds register. Access restriction applies. Updated based on the event. Setting 1 sets. Setting 0 clears.

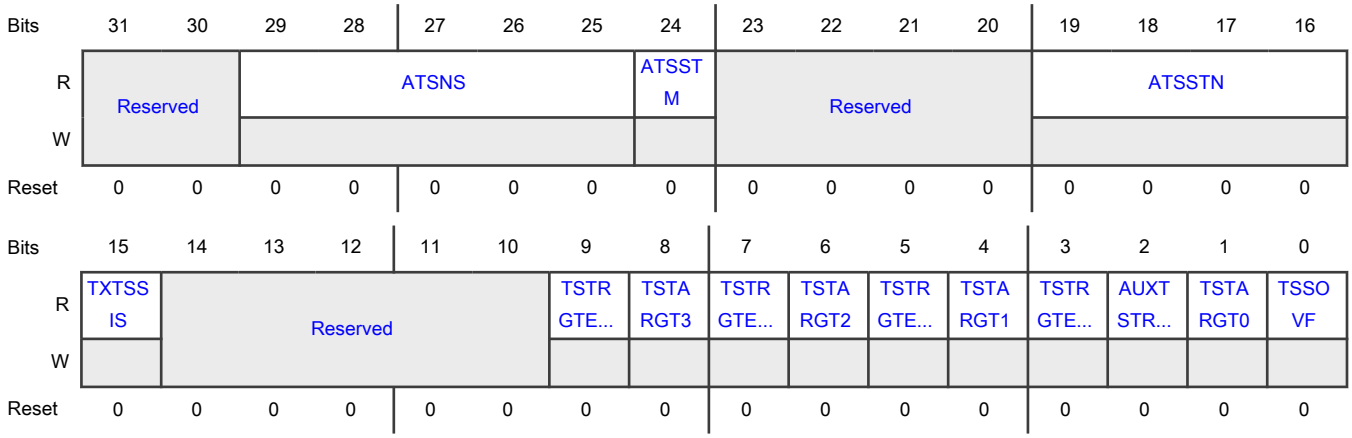
43.8.1.335 Timestamp Status (MAC_TIMESTAMP_STATUS)

Timestamp Status register. All bits except Bits[27:25] gets cleared when the application reads this register.

Offset

Register	Offset
MAC_TIMESTAMP_STATUS	B20h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-25 ATSNS	Number of Auxiliary Timestamp Snapshots This field indicates the number of Snapshots available in the FIFO. Number of Auxiliary Timestamp Snapshots This field indicates the number of Snapshots available in the FIFO. A value equal to the selected depth of FIFO (4, 8, or 16) indicates that the Auxiliary Snapshot FIFO is full. These bits are cleared (to 00000) when the Auxiliary snapshot FIFO clear bit is set. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected.
24 ATSSTM	Auxiliary Timestamp Snapshot Trigger Missed This bit is set when the Auxiliary timestamp snapshot FIFO is full and external trigger was set. Auxiliary Timestamp Snapshot Trigger Missed This bit is set when the Auxiliary timestamp snapshot FIFO is full and external trigger was set. This indicates that the latest snapshot is not stored in the FIFO. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected. 0 - Auxiliary Timestamp Snapshot Trigger Missed status not detected 1 - Auxiliary Timestamp Snapshot Trigger Missed status detected
23-20 —	Reserved
19-16 ATSSTN	Auxiliary Timestamp Snapshot Trigger Identifier These bits identify the Auxiliary trigger inputs for which the timestamp available in the Auxiliary Snapshot Register is applicable. Auxiliary Timestamp Snapshot Trigger Identifier These bits identify the Auxiliary trigger inputs for which the timestamp available in the Auxiliary Snapshot Register is applicable. When more than one bit is set at the same time, it means that corresponding auxiliary

Table continues on the next page...

Table continued from the previous page...

Field	Description
	triggers were sampled at the same clock. These bits are applicable only if the number of Auxiliary snapshots is more than one. One bit is assigned for each trigger as shown in the following list: - Bit 16: Auxiliary trigger 0 - Bit 17: Auxiliary trigger 1 - Bit 18: Auxiliary trigger 2 - Bit 19: Auxiliary trigger 3 The software can read this register to find the triggers that are set when the timestamp is taken. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15 TXTSSIS	Tx Timestamp Status Interrupt Status In non-EQOS_CORE configurations when drop transmit status is enabled in MTL, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and MAC_TX_TIMESTAMP_STATUS_SECONDS registers. Tx Timestamp Status Interrupt Status In non-EQOS_CORE configurations when drop transmit status is enabled in MTL, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and TMAC_X_TIMESTAMP_STATUS_SECONDS registers. When PTP offload feature is enabled, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and MAC_TX_TIMESTAMP_STATUS_SECONDS registers, for PTO generated Delay Request and Pdelay request packets. This bit is cleared when the MAC_TX_TIMESTAMP_STATUS_SECONDS register is read (or write to MAC_TX_TIMESTAMP_STATUS_SECONDS register when RCWE bit of CSR_SW_CTRL register is set). 0 - Tx Timestamp Status Interrupt status not detected 1 - Tx Timestamp Status Interrupt status detected
14-10 —	Reserved
9 TSTRGTERR3	Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS registers elapses. Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Error status not detected 1 - Timestamp Target Time Error status detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 TSTARGT3	Timestamp Target Time Reached for Target Time PPS3 When this bit is set, it indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS registers. Timestamp Target Time Reached for Target Time PPS3 When this bit is set, it indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS registers. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Reached for Target Time PPS3 status not detected 1 - Timestamp Target Time Reached for Target Time PPS3 status detected
7 TSTRGTERR2	Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS registers elapses. Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Error status not detected 1 - Timestamp Target Time Error status detected
6 TSTARGT2	Timestamp Target Time Reached for Target Time PPS2 When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS registers. Timestamp Target Time Reached for Target Time PPS2 When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS registers. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Reached for Target Time PPS2 status not detected 1 - Timestamp Target Time Reached for Target Time PPS2 status detected
5 TSTRGTERR1	Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS registers elapses. Timestamp Target Time Error

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This bit is set when the latest target time programmed in the MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Error status not detected 1 - Timestamp Target Time Error status detected
4 TSTARGT1	Timestamp Target Time Reached for Target Time PPS1 When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS registers. Timestamp Target Time Reached for Target Time PPS1 When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS registers. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Reached for Target Time PPS1 status not detected 1 - Timestamp Target Time Reached for Target Time PPS1 status detected
3 TSTRGTERR0	Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS0_Target_Time_Seconds and MAC_PPS0_Target_Time_Nanoseconds registers elapses. Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS0_Target_Time_Seconds and MAC_PPS0_TARGET_TIME_NANOSECONDS registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Error status not detected 1 - Timestamp Target Time Error status detected
2 AUXTSTRIG	Auxiliary Timestamp Trigger Snapshot This bit is set high when the auxiliary snapshot is written to the FIFO. Auxiliary Timestamp Trigger Snapshot This bit is set high when the auxiliary snapshot is written to the FIFO. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Auxiliary Timestamp Trigger Snapshot status not detected 1 - Auxiliary Timestamp Trigger Snapshot status detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 TSTARGT0	Timestamp Target Time Reached When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS0_Target_Time_Seconds and MAC_PPS0_Target_Time_Nanoseconds registers. Timestamp Target Time Reached When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS0_TARGET_TIME_SECONDS and MAC_PPS0_TARGET_TIME_NANOSECONDS registers. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Reached status not detected 1 - Timestamp Target Time Reached status detected
0 TSSOVF	Timestamp Seconds Overflow When this bit is set, it indicates that the seconds value of the timestamp (when supporting version 2 format) has overflowed beyond 32'hFFFF_FFFF. Timestamp Seconds Overflow When this bit is set, it indicates that the seconds value of the timestamp (when supporting version 2 format) has overflowed beyond 32'hFFFF_FFFF. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Seconds Overflow status not detected 1 - Timestamp Seconds Overflow status detected

43.8.1.336 Transmit Timestamp Status Nanoseconds (MAC_TX_TIMESTAMP_STATUS_NANOSECONDS)

This register contains the nanosecond part of timestamp captured for Transmit packets when Tx status is disabled.

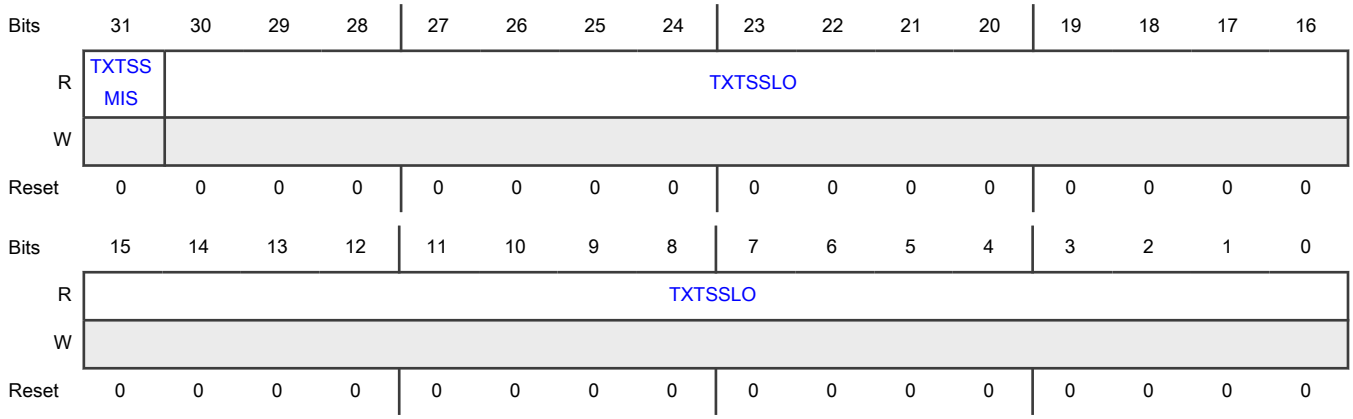
The MAC_TX_TIMESTAMP_STATUS_NANOSECONDS register, along with MAC_TX_TIMESTAMP_STATUS_SECONDS, gives the 64-bit timestamp captured for the PTP packet successfully transmitted by the MAC. This value is considered to be read by the application when the last byte of MAC_TX_TIMESTAMP_STATUS_NANOSECONDS is read. In the little-endian mode, this means when bits[31:24] are read; in big-endian mode, bits[7:0] are read.

If the application does not read these registers and timestamp of another packet is captured, then either the current timestamp is lost (overwritten) or the new timestamp is lost (dropped), depending on the setting of the TXTSSTSM bit of the MAC_TIMESTAMP_CONTROL register. The status bit TXTSC bit [15] in MAC_TIMESTAMP_STATUS register is set whenever the MAC transmitter captures the timestamp.

Offset

Register	Offset
MAC_TX_TIMESTAMP_STATUS_NANOSECONDS	B30h

Diagram



Fields

Field	Description
31 TXTSSMIS	Transmit Timestamp Status Missed When this bit is set, it indicates one of the following: - The timestamp of the current packet is ignored if TXTSSTSM bit of the TIMESTAMP_CONTROL register is reset - The timestamp of the previous packet is overwritten with timestamp of the current packet if TXTSSTSM bit of the MAC_TIMESTAMP_CONTROL register is set. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Transmit Timestamp Status Missed status not detected 1 - Transmit Timestamp Status Missed status detected
30-0 TXTSSLO	Transmit Timestamp Status Low This field contains the 31 bits of the Nanoseconds field of the Transmit packet's captured timestamp.

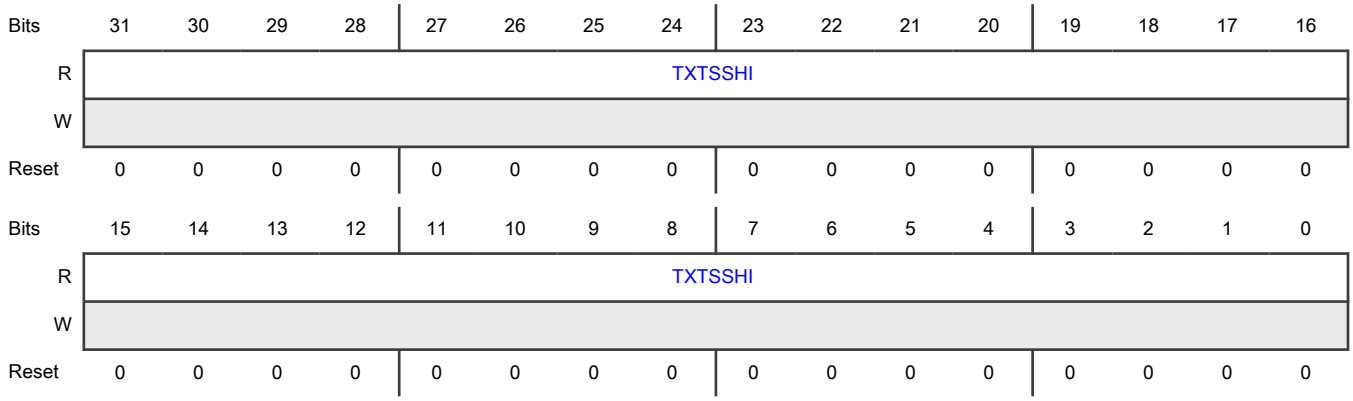
43.8.1.337 Transmit Timestamp Status Seconds (MAC_TX_TIMESTAMP_STATUS_SECONDS)

The register contains the higher 32 bits of the timestamp (in seconds) captured when a PTP packet is transmitted.

Offset

Register	Offset
MAC_TX_TIMESTAMP_STATUS_SECONDS	B34h

Diagram



Fields

Field	Description
31-0 TXTSSHI	Transmit Timestamp Status High This field contains the lower 32 bits of the Seconds field of Transmit packet's captured timestamp.

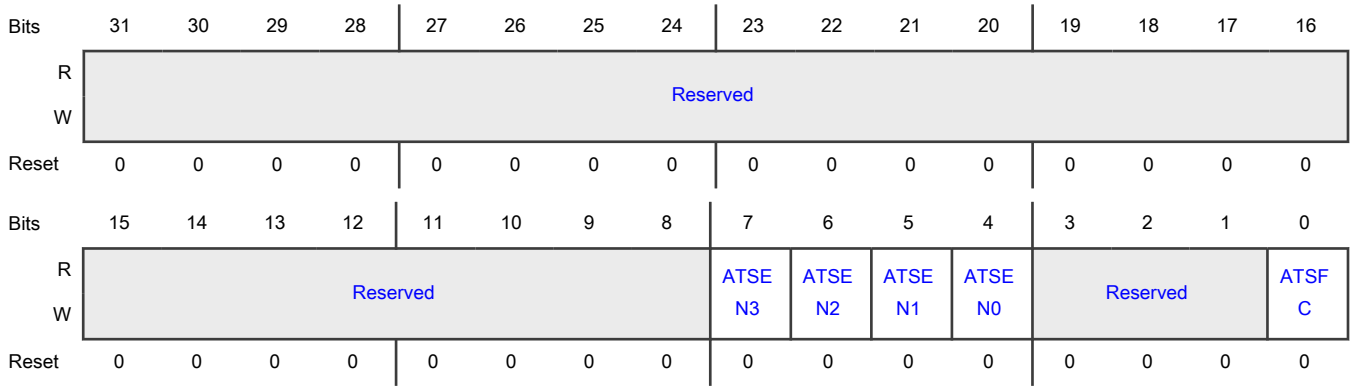
43.8.1.338 Auxiliary Timestamp Control (MAC_AUXILIARY_CONTROL)

The Auxiliary Timestamp Control register controls the Auxiliary Timestamp snapshot.

Offset

Register	Offset
MAC_AUXILIARY_CONTROL	B40h

Diagram



Fields

Field	Description
31-8 —	Reserved
7 ATSEN3	Auxiliary Snapshot 3 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 3. When this bit is set, the auxiliary snapshot of the event on ptp_aux_trig_i[3] input is enabled. When this bit is reset, the events on this input are ignored. 0 - Auxiliary Snapshot $i is disabled 1 - Auxiliary Snapshot $i is enabled
6 ATSEN2	Auxiliary Snapshot 2 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 2. When this bit is set, the auxiliary snapshot of the event on ptp_aux_trig_i[2] input is enabled. When this bit is reset, the events on this input are ignored. 0 - Auxiliary Snapshot $i is disabled 1 - Auxiliary Snapshot $i is enabled
5 ATSEN1	Auxiliary Snapshot 1 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 1. When this bit is set, the auxiliary snapshot of the event on ptp_aux_trig_i[1] input is enabled. When this bit is reset, the events on this input are ignored. 0 - Auxiliary Snapshot $i is disabled 1 - Auxiliary Snapshot $i is enabled
4 ATSEN0	Auxiliary Snapshot 0 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 0. When this bit is set, the auxiliary snapshot of the event on ptp_aux_trig_i[0] input is enabled. When this bit is reset, the events on this input are ignored. 0 - Auxiliary Snapshot $i is disabled 1 - Auxiliary Snapshot $i is enabled
3-1 —	Reserved
0 ATSFC	Auxiliary Snapshot FIFO Clear When set, this bit resets the pointers of the Auxiliary Snapshot FIFO. This bit is cleared when the pointers are reset and the FIFO is empty. When this bit is high, the auxiliary snapshots are stored in the FIFO. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Auxiliary Snapshot FIFO Clear is disabled 1 - Auxiliary Snapshot FIFO Clear is enabled

43.8.1.339 Auxiliary Timestamp Nanoseconds (MAC_AUXILIARY_TIMESTAMP_NANOSECONDS)

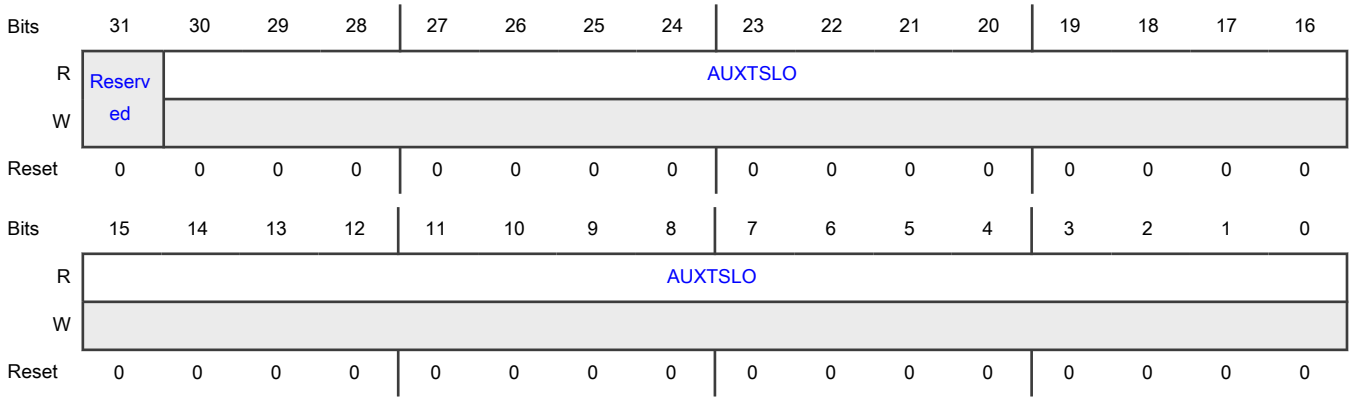
The Auxiliary Timestamp Nanoseconds register, along with MAC_Auxiliary_Timestamp_Seconds, gives the 64-bit timestamp stored as auxiliary snapshot. These two registers form the read port of a 64-bit wide FIFO with a depth of 4, 8, or 16 as selected while configuring the core.

You can store multiple snapshots in this FIFO. Bits[29:25] in MAC_TIMESTAMP_STATUS indicate the fill-level of the FIFO. The top of the FIFO is removed only when the last byte of MAC_AUXILIARY_TIMESTAMP_SECONDS register is read. In the little-endian mode, this means when Bits[31:24] are read and in big-endian mode, Bits[7:0] are read.

Offset

Register	Offset
MAC_AUXILIARY_TIMESTAMP_NANOSECONDS	B48h

Diagram



Fields

Field	Description
31 —	Reserved
30-0 AUXTSLO	Auxiliary Timestamp Contains the lower 31 bits (nanoseconds field) of the auxiliary timestamp.

43.8.1.340 Auxiliary Timestamp Seconds (MAC_AUXILIARY_TIMESTAMP_SECONDS)

The Auxiliary Timestamp - Seconds register contains the lower 32 bits of the Seconds field of the auxiliary timestamp register.

Offset

Register	Offset
MAC_AUXILIARY_TIMESTAMP_SECONDS	B4Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	AUXTSHI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AUXTSHI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 AUXTSHI	Auxiliary Timestamp Contains the lower 32 bits of the Seconds field of the auxiliary timestamp.

43.8.1.341 Timestamp Ingress Asymmetry Correction (MAC_TIMESTAMP_INGRESS_ASYM_CORR)

The MAC Timestamp Ingress Asymmetry Correction register contains the Ingress Asymmetry Correction value to be used while updating correction field in PDelay_Resp PTP messages.

Offset

Register	Offset
MAC_TIMESTAMP_INGRESS_ASYM_CORR	B50h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OSTIAC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OSTIAC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OSTIAC	One-Step Timestamp Ingress Asymmetry Correction This field contains the ingress path asymmetry value to be added to correctionField of Pdelay_Resp PTP packet. The programmed value should be in units of nanoseconds and multiplied by 2^16. For example, 2.5 ns is represented as 0x00028000. The value can also be negative, which is represented in 2's complement form with bit 31 representing the sign bit.

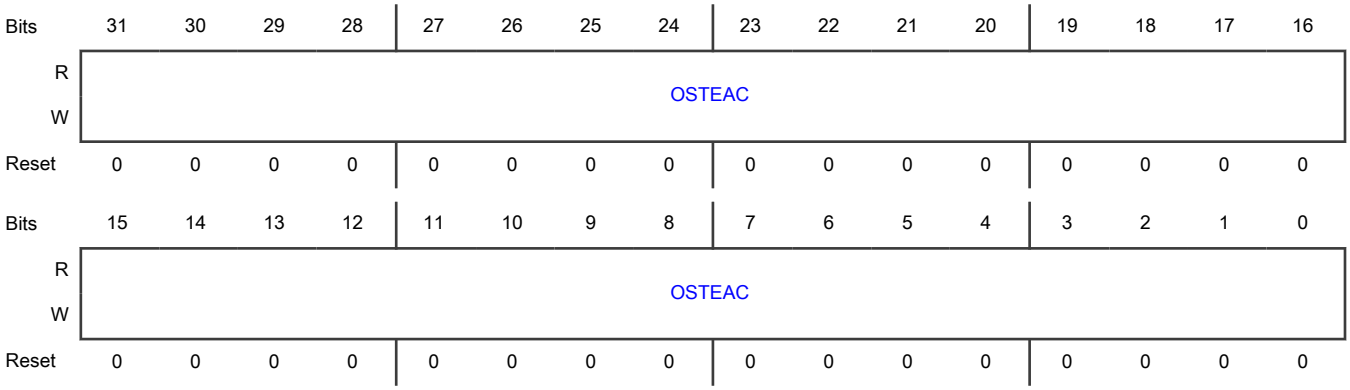
43.8.1.342 Timestamp Egress Asymmetry Correction (MAC_TIMESTAMP_EGRESS_ASYM_CORR)

The MAC Timestamp Egress Asymmetry Correction register contains the Egress Asymmetry Correction value to be used while updating the correction field in PDelay_Req PTP messages.

Offset

Register	Offset
MAC_TIMESTAMP_EGRESS_ASYM_CORR	B54h

Diagram



Fields

Field	Description
31-0 OSTEAC	One-Step Timestamp Egress Asymmetry Correction This field contains the egress path asymmetry value to be subtracted from correctionField of Pdelay_Resp PTP packet. The programmed value must be the negated value in units of nanoseconds multiplied by 2^16. For example, if the required correction is +2.5 ns, the programmed value must be 0xFFFD_8000, which is the 2's complement of 0x0002_8000(2.5 * 216). Similarly, if the required correction is -3.3 ns, the programmed value is 0x0003_4CCC (3.3 * 216).

43.8.1.343 Timestamp Ingress Correction Nanosecond (MAC_TIMESTAMP_INGRESS_CORR_NANOSECOND)

This register contains the correction value in nanoseconds to be used with the captured timestamp value in the ingress path.

Offset

Register	Offset
MAC_TIMESTAMP_INGRESS_CORR_NANOSECOND	B58h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSIC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSIC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TSIC	Timestamp Ingress Correction This field contains the ingress path correction value as defined by the Ingress Correction expression.

43.8.1.344 Timestamp Egress Correction Nanosecond (MAC_TIMESTAMP_EGRESS_CORR_NANOSECOND)

This register contains the correction value in nanoseconds to be used with the captured timestamp value in the egress path.

Offset

Register	Offset
MAC_TIMESTAMP_EGRESS_CORR_NANOSECOND	B5Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TSEC	Timestamp Egress Correction This field contains the nanoseconds part of the egress path correction value as defined by the Egress Correction expression.

43.8.1.345 Timestamp Ingress Correction Subnanosecond (MAC_TIMESTAMP_INGRESS_CORR_SUBNANOSEC)

This register contains the sub-nanosecond part of the correction value to be used with the captured timestamp value, for ingress direction.

Offset

Register	Offset
MAC_TIMESTAMP_INGRESS_CORR_SUBNANOSEC	B60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSICSNS								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-8 TSICSNS	Timestamp Ingress Correction, sub-nanoseconds This field contains the sub-nanoseconds part of the ingress path correction value as defined by the "Ingress Correction" expression.
7-0 —	Reserved

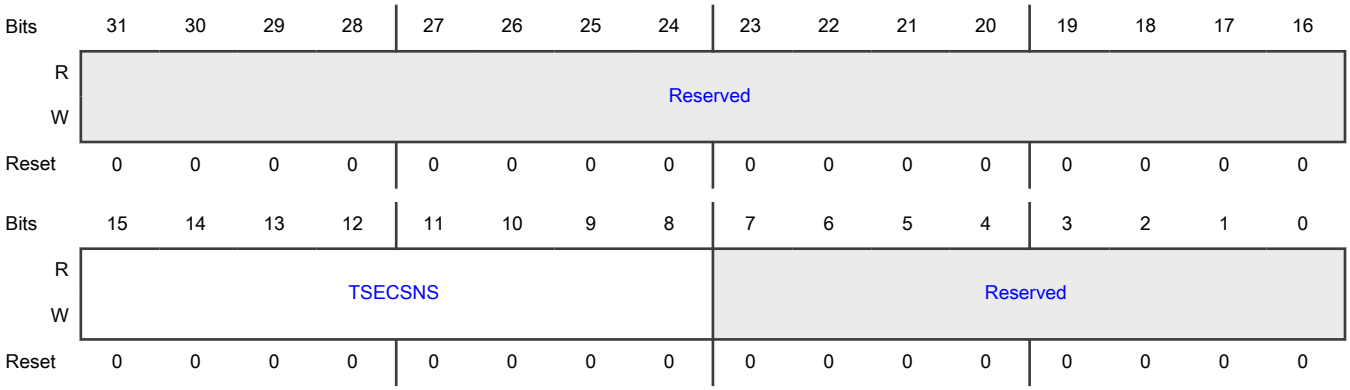
43.8.1.346 Timestamp Egress Correction Subnanosecond (MAC_TIMESTAMP_EGRESS_CORR_SUBNANOSEC)

This register contains the sub-nanosecond part of the correction value to be used with the captured timestamp value, for egress direction.

Offset

Register	Offset
MAC_TIMESTAMP_EGRESS_CORR_SUBNANOSEC	B64h

Diagram



Fields

Field	Description
31-16 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-8 TSECSNS	Timestamp Egress Correction, sub-nanoseconds This field contains the sub-nanoseconds part of the egress path correction value as defined by the "Egress Correction" expression.
7-0 —	Reserved

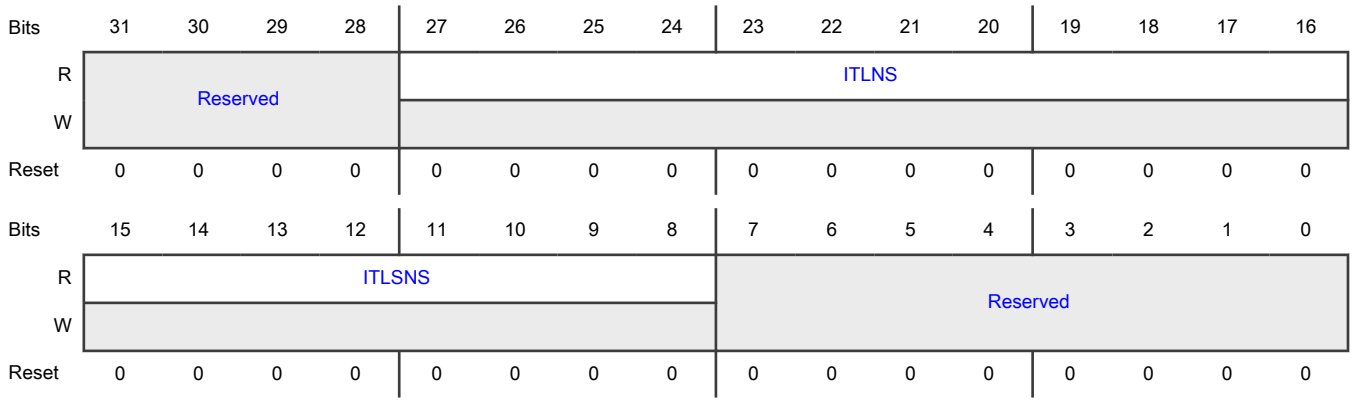
43.8.1.347 Timestamp Ingress Latency (MAC_TIMESTAMP_INGRESS_LATENCY)

This register holds the Ingress MAC latency.

Offset

Register	Offset
MAC_TIMESTAMP_INGRESS_LATENCY	B68h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 ITLNS	Ingress Timestamp Latency, in sub-nanoseconds This register holds the average latency in sub-nanoseconds between the input ports (phy_rxd_i) of MAC and the actual point (GMII) where the ingress timestamp is taken. Ingress correction value is computed as described in the section 7.1.2.4.1 of QoS Databook.
15-8	Ingress Timestamp Latency, in nanoseconds

Table continues on the next page...

Table continued from the previous page...

Field	Description
ITLSNS	This register holds the average latency in nanoseconds between the input ports (phy_rxd_i) of MAC and the actual point (GMII) where the ingress timestamp is taken. Ingress correction value is computed as described in the section 7.1.2.4.1 of QoS Databook.
7-0 —	Reserved

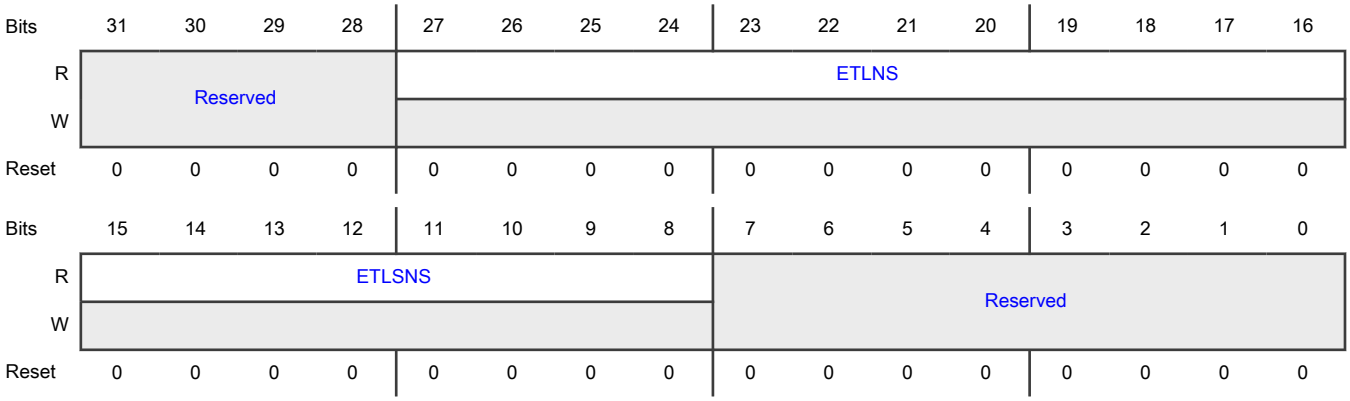
43.8.1.348 Timestamp Egress Latency (MAC_TIMESTAMP_EGRESS_LATENCY)

This register holds the Egress MAC latency.

Offset

Register	Offset
MAC_TIMESTAMP_EGRESS_LATENCY	B6Ch

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 ETLNS	Egress Timestamp Latency, in nanoseconds This register holds the average latency in nanoseconds between the actual point (GMII) where the egress timestamp is taken and the output ports (phy_txd_o) of the MAC. Ingress correction value is computed as described in the section 7.1.2.4.2 of QoS Databook.
15-8	Egress Timestamp Latency, in sub-nanoseconds

Table continues on the next page...

Table continued from the previous page...

Field	Description
ETLSNS	This register holds the average latency in sub-nanoseconds between the actual point (GMII) where the egress timestamp is taken and the output ports (phy_txd_o) of the MAC. Ingress correction value is computed as described in the section 7.1.2.4.2 of QoS Databook.
7-0 —	Reserved

43.8.1.349 PPS Control (MAC_PPS_CONTROL)

PPS Control register.

Bits[30:24] of this register are valid only when four Flexible PPS outputs are selected. Bits[22:16] are valid only when three or more Flexible PPS outputs are selected. Bits[14:8] are valid only when two or more Flexible PPS outputs are selected. Bits[6:4] are valid only when Flexible PPS feature is selected.

Offset

Register	Offset
MAC_PPS_CONTROL	B70h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MCGREN3	TRGTMODSEL3		Reserved	PPSCMD3				MCGREN2	TRGTMODSEL2		Reserved	PPSCMD2			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MCGREN1	TRGTMODSEL1		Reserved	PPSCMD1				MCGREN0	TRGTMODSEL0		PPSEN0	PPSCTRL_PPSCMD			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 MCGREN3	MCGR Mode Enable for PPS3 Output This field enables the 3rd PPS instance to operate in PPS or MCGR mode. MCGR Mode Enable for PPS3 Output This field enables the 3rd PPS instance to operate in PPS or MCGR mode. When set it operates in MCGR mode and on reset it operates in PPS mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
30-29 TRGTMODSEL3	Target Time Register Mode for PPS3 Output This field indicates the Target Time registers (MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS) mode for PPS3 output signal. 00 - Target Time registers are programmed only for generating the interrupt event. The Flexible PPS function must not be enabled in this mode, otherwise spurious transitions may be observed on the corresponding ptp_pps_o output port 01 - Reserved 10 - Target Time registers are programmed for generating the interrupt event and starting or stopping the PPS0 output signal generation 11 - Target Time registers are programmed only for starting or stopping the PPS0 output signal generation. No interrupt is asserted
28 —	Reserved
27-24 PPSCMD3	Flexible PPS3 Output Control This field controls the flexible PPS3 output (ptp_pps_o[3]) signal. Flexible PPS3 Output Control This field controls the flexible PPS3 output (ptp_pps_o[3]) signal. This field is similar to the PPSCMD0[2:0] field. If MCGREN3 is set, then PPSCMD3 indicated by these 4 bits [27:24] are taken as Presentation Time Control bits for media clock generation and recovery for comparator instance 3. This field is similar to the PPSCMD0 Presentation Time Control bits. If MCGREN3 is not set then only 3 bits from [26:24] is used as PPSCMD3 and the 4th bit is to be set as 0. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
23 MCGREN2	MCGR Mode Enable for PPS2 Output This field enables the 2nd PPS instance to operate in PPS or MCGR mode. MCGR Mode Enable for PPS2 Output This field enables the 2nd PPS instance to operate in PPS or MCGR mode. When set it operates in MCGR mode and on reset it operates in PPS mode. 0 - 2nd PPS instance is disabled to operate in PPS or MCGR mode 1 - 2nd PPS instance is enabled to operate in PPS or MCGR mode
22-21 TRGTMODSEL2	Target Time Register Mode for PPS2 Output This field indicates the Target Time registers (MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS) mode for PPS2 output signal. 00 - Target Time registers are programmed only for generating the interrupt event. The Flexible PPS function must not be enabled in this mode, otherwise spurious transitions may be observed on the corresponding ptp_pps_o output port 01 - Reserved 10 - Target Time registers are programmed for generating the interrupt event and starting or stopping the PPS0 output signal generation

Table continues on the next page...

Table continued from the previous page...

Field	Description
	11 - Target Time registers are programmed only for starting or stopping the PPS0 output signal generation. No interrupt is asserted
20 —	Reserved
19-16 PPSCMD2	Flexible PPS2 Output Control This field controls the flexible PPS2 output (ptp_pps_o[2]) signal. Flexible PPS2 Output Control This field controls the flexible PPS2 output (ptp_pps_o[2]) signal. This field is similar to the PPSCMD0 field. If MCGREN2 is set, then PPSCMD2 indicated by these 4 bits [19:16] are taken as Presentation Time Control bits for media clock generation and recovery for comparator instance 2. This field is similar to the PPSCMD0 Presentation Time Control bits. If MCGREN2 is not set then only 3 bits from [18:16] is used as PPSCMD2 and the 4th bit is to be set as 0. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
15 MCGREN1	MCGR Mode Enable for PPS1 Output This field enables the 1st PPS instance to operate in PPS or MCGR mode. MCGR Mode Enable for PPS1 Output This field enables the 1st PPS instance to operate in PPS or MCGR mode. When set it operates in MCGR mode and on reset it operates in PPS mode. 0 - 1st PPS instance is disabled to operate in PPS or MCGR mode 1 - 1st PPS instance is enabled to operate in PPS or MCGR mode
14-13 TRGTMODSEL1	Target Time Register Mode for PPS1 Output This field indicates the Target Time registers (MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS) mode for PPS1 output signal. 00 - Target Time registers are programmed only for generating the interrupt event. The Flexible PPS function must not be enabled in this mode, otherwise spurious transitions may be observed on the corresponding ptp_pps_o output port 01 - Reserved 10 - Target Time registers are programmed for generating the interrupt event and starting or stopping the PPS0 output signal generation 11 - Target Time registers are programmed only for starting or stopping the PPS0 output signal generation. No interrupt is asserted
12 —	Reserved
11-8 PPSCMD1	Flexible PPS1 Output Control This field controls the flexible PPS1 output (ptp_pps_o[1]) signal. Flexible PPS1 Output Control This field controls the flexible PPS1 output (ptp_pps_o[1]) signal. This field is similar to the PPSCMD0 field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If MCGREN1 is set, then PPSCMD1 indicated by these 4 bits [11:8] are taken as Presentation Time Control bits for media clock generation and recovery for comparator instance 1. This field is similar to the PPSCMD0 Presentation Time Control bits. If MCGREN1 is not set then only 3 bits from [10:8] is used as PPSCMD1 and the 4th bit is to be set as 0. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
7 MCGREN0	MCGR Mode Enable for PPS0 Output This field enables the 0th PPS instance to operate in PPS or MCGR mode. MCGR Mode Enable for PPS0 Output This field enables the 0th PPS instance to operate in PPS or MCGR mode. When set it operates in MCGR mode and on reset it operates in PPS mode. 0 - 0th PPS instance is enabled to operate in PPS mode 1 - 0th PPS instance is enabled to operate in MCGR mode
6-5 TRGTMODSEL0	Target Time Register Mode for PPS0 Output This field indicates the Target Time registers (MAC_PPS0_TARGET_TIME_SECONDS and MAC_PPS0_TARGET_TIME_NANOSECONDS) mode for PPS0 output signal: 00 - Target Time registers are programmed only for generating the interrupt event. The Flexible PPS function must not be enabled in this mode, otherwise spurious transitions may be observed on the corresponding ptp_pps_o output port 01 - Reserved 10 - Target Time registers are programmed for generating the interrupt event and starting or stopping the PPS0 output signal generation 11 - Target Time registers are programmed only for starting or stopping the PPS0 output signal generation. No interrupt is asserted
4 PPSEN0	Flexible PPS Output Mode Enable When this bit is set, Bits[3:0] function as PPSCMD. Flexible PPS Output Mode Enable When this bit is set, Bits[3:0] function as PPSCMD. When this bit is reset, Bits[3:0] function as PPSCTRL (Fixed PPS mode). 0 - Flexible PPS Output Mode is disabled 1 - Flexible PPS Output Mode is enabled
3-0 PPSCTRL_PPSCMD	PPS Output Frequency Control This field controls the frequency of the PPS0 output (ptp_pps_o[0]) signal. PPS Output Frequency Control This field controls the frequency of the PPS0 output (ptp_pps_o[0]) signal. The default value of PPSCTRL is 0000, and the PPS output is 1 pulse (of width clk_ptp_i) every second. For other values of PPSCTRL, the PPS output becomes a generated clock of following frequencies: - 0001: The binary rollover is 2 Hz, and the digital rollover is 1 Hz. - 0010: The binary rollover is 4 Hz, and the digital rollover is 2 Hz. - 0011: The binary rollover is 8 Hz, and the digital rollover is 4 Hz.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- 0100: The binary rollover is 16 Hz, and the digital rollover is 8 Hz. - .. - 1111: The binary rollover is 32.768 KHz and the digital rollover is 16.384 KHz. Note: In the binary rollover mode, the PPS output (ptp_pps_o) has a duty cycle of 50 percent with these frequencies. In the digital rollover mode, the PPS output frequency is an average number. The actual clock is of different frequency that gets synchronized every second. For example: - When PPSCTRL = 0001, the PPS (1 Hz) has a low period of 537 ms and a high period of 463 ms - When PPSCTRL = 0010, the PPS (2 Hz) is a sequence of One clock of 50 percent duty cycle and 537 ms period Second clock of 463 ms period (268 ms low and 195 ms high) - When PPSCTRL = 0011, the PPS (4 Hz) is a sequence of Three clocks of 50 percent duty cycle and 268 ms period Fourth clock of 195 ms period (134 ms low and 61 ms high) This behavior is because of the non-linear toggling of bits in the digital rollover mode in the MAC_System_Time_Nanoseconds register. or Flexible PPS Output (ptp_pps_o[0]) Control Programming these bits with a non-zero value instructs the MAC to initiate an event. When the command is transferred or synchronized to the PTP clock domain, these bits get cleared automatically. The software should ensure that these bits are programmed only when they are 'all-zero'. The following list describes the values of PPSCMD0: - 0000: No Command - 0001: START Single Pulse This command generates single pulse rising at the start point defined in MAC_PPS0_Target_Time_Seconds and MAC_PPS0_Target_Time_Nanoseconds register and of a duration defined in the PPS0 Width Register. - 0010: START Pulse Train This command generates the train of pulses rising at the start point defined in the Target Time Registers and of a duration defined in the PPS0 Width Register and repeated at interval defined in the PPS Interval Register. By default, the PPS pulse train is free-running unless stopped by the 'Stop Pulse train at time' or 'Stop Pulse Train immediately' commands. - 0011: Cancel START This command cancels the START Single Pulse and START Pulse Train commands if the system time has not crossed the programmed start time. - 0100: STOP Pulse train at time

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This command stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 0010) after the time programmed in the Target Time registers elapses. - 0101: STOP Pulse Train immediately This command immediately stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 0010). - 110: Cancel STOP Pulse train This command cancels the STOP pulse train at time command if the programmed stop time has not elapsed. The PPS pulse train becomes free-running on the successful execution of this command. - 0111-1111: Reserved or Presentation Time Control If MCGREN0 is set then these bits are treated as Presentation time control bits. The following list describes the values of PPSCMD0: - 0000: No Operation related to MCGR will be carried out. If set to this value in the mid of clock recovery or generation all the processing inputs will be flushed. - 0001: Capture the Presentation time at rising edge of mcg_pst_trig_i[0] into the MAC_PPS0_Target_Time_Seconds register - 0010: Capture the Presentation time at falling edge of mcg_pst_trig_i[0] into the MAC_PPS0_Target_Time_Seconds register - 0011: Capture the Presentation time at both edges of mcg_pst_trig_i[0] into the MAC_PPS0_Target_Time_Seconds register - 0100-1000: Reserved - 1001: Toggle output on compare - 1010: Pulse output low on compare for one PTP-clock cycle - 1011: Pulse output high on compare for one PTP-clock cycle - 1100-1111: Reserved

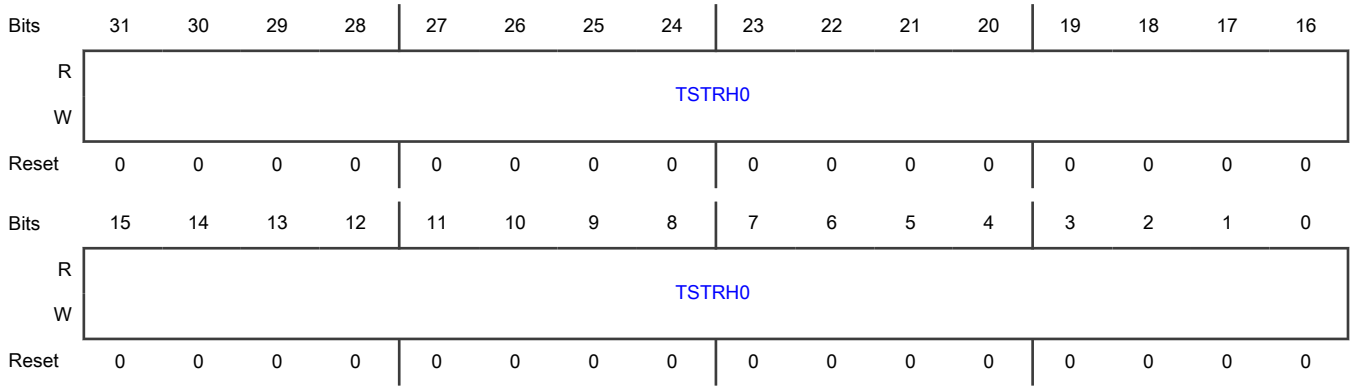
43.8.1.350 PPS0 Target Time Seconds (MAC_PPS0_TARGET_TIME_SECONDS)

The PPS0 Target Time Seconds register, along with PPS0 Target Time Nanoseconds register, is used to schedule an interrupt event [Bit 1 of MAC_TIMESTAMP_STATUS] when the system time exceeds the value programmed in these registers.

Offset

Register	Offset
MAC_PPS0_TARGET_TIME_SECONDS	B80h

Diagram



Fields

Field	Description
31-0 TSTRH0	PPS Target Time Seconds Register This field stores the time in seconds. PPS Target Time Seconds Register This field stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on Target Time mode selected for the corresponding PPS output in the MAC_PPS_CONTROL register.

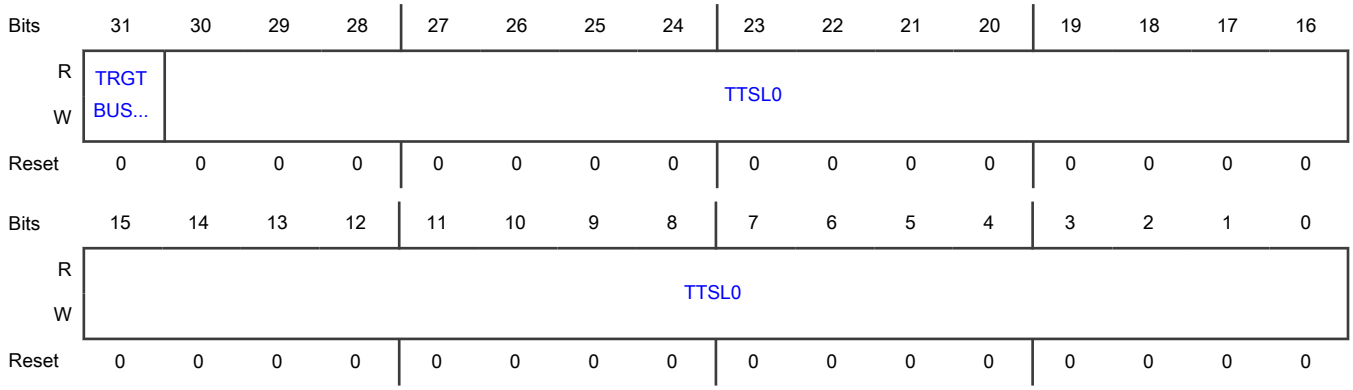
43.8.1.351 PPS0 Target Time Nanoseconds (MAC_PPS0_TARGET_TIME_NANOSECONDS)

PPS0 Target Time Nanoseconds register.

Offset

Register	Offset
MAC_PPS0_TARGET_TIME_NANOSECONDS	B84h

Diagram



Fields

Field	Description
31 TRGTBUSY0	PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the MAC_PPS_CONTROL register is programmed to 010 or 011. Programming the PPSCMD0 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. 0 - PPS Target Time Register Busy status is not detected 1 - PPS Target Time Register Busy is detected
30-0 TTSL0	Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the value in both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on the TRGTMODSEL0 field (Bits [6:5]) in PPS_CONTROL. When the TSCTRLSSR bit is reset in the MAC_TIMESTAMP_CONTROL register, this value should be (time in ns / 0.465). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. When the TSCTRLSSR bit is set in the MAC_TIMESTAMP_CONTROL register, this value should not exceed 0x3B9A_C9FF. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

43.8.1.352 PPS0 Interval (MAC_PPS0_INTERVAL)

The PPS0 Interval register contains the number of units of sub-second increment value between the rising edges of PPS0 signal output (ptp_pps_o[0]).

Offset

Register	Offset
MAC_PPS0_INTERVAL	B88h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSINT0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSINT0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PPSINT0	PPS Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output. PPS Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output. The interval is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between the rising edges of PPS0 signal output is 100 ns (that is, 5 units of sub-second increment value), you should program value 4 (5-1) in this register.

43.8.1.353 PPS0 Width (MAC_PPS0_WIDTH)

The PPS0 Width register contains the number of units of sub-second increment value between the rising and corresponding falling edges of PPS0 signal output (ptp_pps_o[0]).

Offset

Register	Offset
MAC_PPS0_WIDTH	B8Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSWIDTH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSWIDTH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PPSWIDTH0	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. The width is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub-second increment value), you should program value 3 (4-1) in this register. Note: The value programmed in this register must be lesser than the value programmed in MAC_PPS0_Interval.

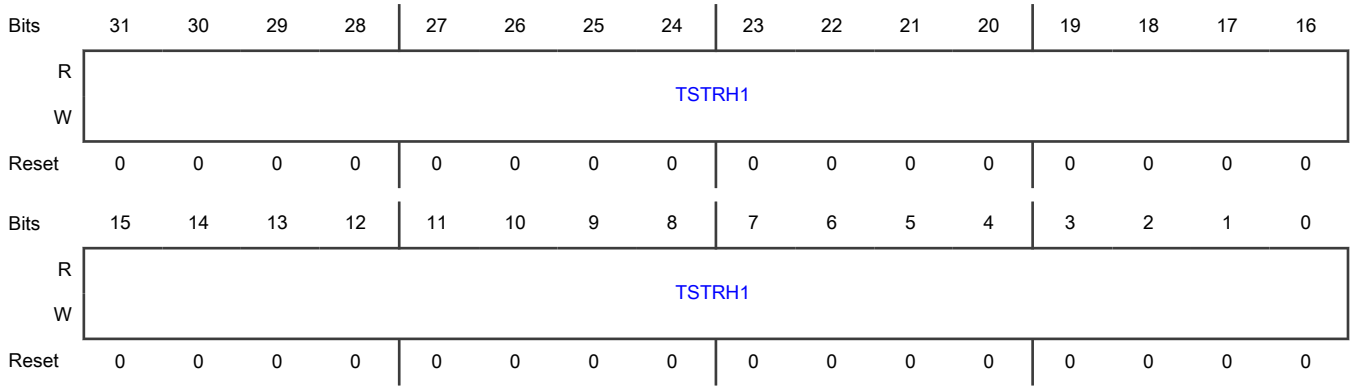
43.8.1.354 PPS1 Target Time Seconds (MAC_PPS1_TARGET_TIME_SECONDS)

The PPS1 Target Time Seconds register, along with PPS1 Target Time Nanoseconds register, is used to schedule an interrupt event [Bit 1 of TIMESTAMP_STATUS] when the system time exceeds the value programmed in these registers.

Offset

Register	Offset
MAC_PPS1_TARGET_TIME_SECONDS	B90h

Diagram



Fields

Field	Description
31-0 TSTRH1	PPS Target Time Seconds Register This field stores the time in seconds. PPS Target Time Seconds Register

Table continues on the next page...

Field	Description
	This field stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on Target Time mode selected for the corresponding PPS output in the PPS_CONTROL register. If DWC_EQOS_FLEXI_PPS_OUT_EN is enabled in the configuration and PTGE field of TIMESTAMP_CONTROL Register is set with Presentation time control set in recovery mode, then these bits indicate the TPT being programmed by the application and in generation mode it indicates the CPT generated at the sampled trigger.

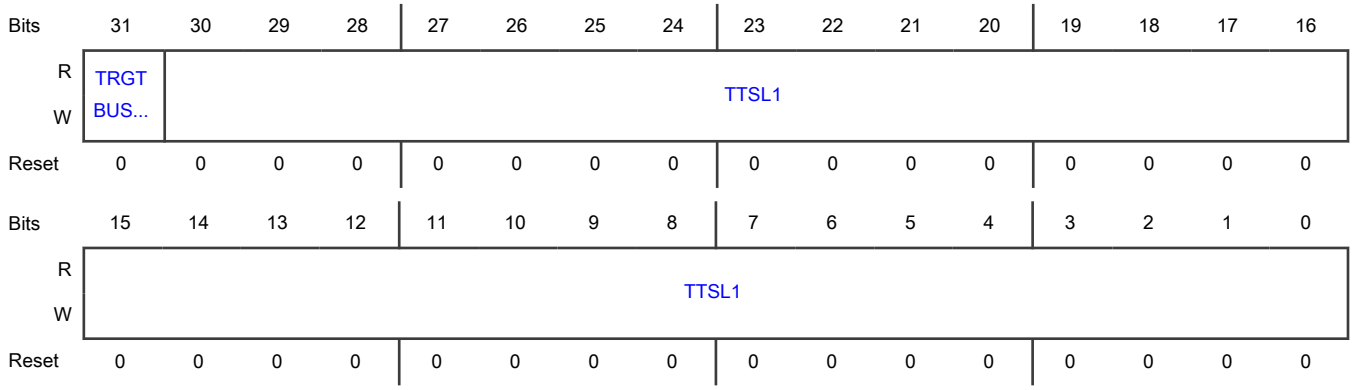
43.8.1.355 PPS1 Target Time Nanoseconds (MAC_PPS1_TARGET_TIME_NANOSECONDS)

PPS1 Target Time Nanoseconds register.

Offset

Register	Offset
MAC_PPS1_TARGET_TIME_NANOSECONDS	B94h

Diagram



Fields

Field	Description
31 TRGTBUSY1	PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. Programming the PPSCMD0 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - PPS Target Time Register Busy status is not detected 1 - PPS Target Time Register Busy is detected
30-0 TTSL1	Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the value in both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on the TRGTMODSEL0 field (Bits [6:5]) in PPS_CONTROL. When the TSCTRLSSR bit is reset in the TIMESTAMP_CONTROL register, this value should be (time in ns / 0.465). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. When the TSCTRLSSR bit is set in the TIMESTAMP_CONTROL register, this value should not exceed 0x3B9A_C9FF. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

43.8.1.356 PPS1 Interval (MAC_PPS1_INTERVAL)

The PPS1 Interval register contains the number of units of sub-second increment value between the rising edges of PPS1 signal output (ptp_pps_o[1]).

Offset

Register	Offset
MAC_PPS1_INTERVAL	B98h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSINT1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSINT1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PPSINT1	PPS Output Signal Interval These bits store the interval between the rising edges of PPS1 signal output. PPS Output Signal Interval These bits store the interval between the rising edges of PPS1 signal output. The interval is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between the rising edges of PPS1 signal output is 100 ns (that is, 5 units of sub-second increment value), you should program value 4 (5-1) in this register.

43.8.1.357 PPS1 Width (MAC_PPS1_WIDTH)

The PPS1 Width register contains the number of units of sub-second increment value between the rising and corresponding falling edges of PPS1 signal output (ptp_pps_o[1]).

Offset

Register	Offset
MAC_PPS1_WIDTH	B9Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSWIDTH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSWIDTH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PPSWIDTH1	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. The width is stored in terms of number of units of sub-second increment value.

Table continues on the next page...

Field	Description
	You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub-second increment value), you should program value 3 (4-1) in this register. Note: The value programmed in this register must be lesser than the value programmed in MAC_PPS0_Interval.

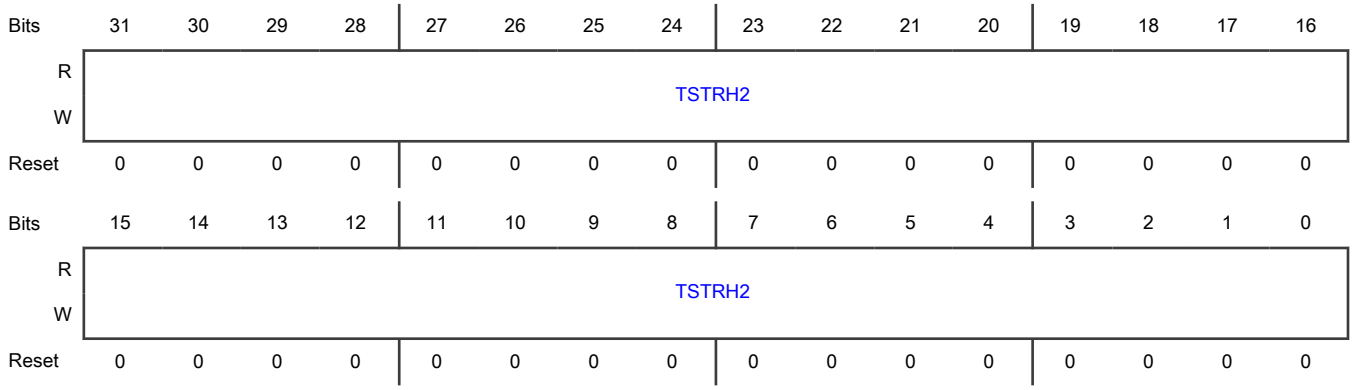
43.8.1.358 PPS2 Target Time Seconds (MAC_PPS2_TARGET_TIME_SECONDS)

The PPS2 Target Time Seconds register, along with PPS2 Target Time Nanoseconds register, is used to schedule an interrupt event [Bit 1 of TIMESTAMP_STATUS] when the system time exceeds the value programmed in these registers.

Offset

Register	Offset
MAC_PPS2_TARGET_TIME_SECONDS	BA0h

Diagram



Fields

Field	Description
31-0 TSTRH2	PPS Target Time Seconds Register This field stores the time in seconds. PPS Target Time Seconds Register This field stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on Target Time mode selected for the corresponding PPS output in the PPS_CONTROL register. If DWC_EQOS_FLEXI_PPS_OUT_EN is enabled in the configuration and PTGE field of TIMESTAMP_CONTROL Register is set with Presentation time control set in recovery mode, then these bits indicate the TPT being programmed by the application and in generation mode it indicates the CPT generated at the sampled trigger.

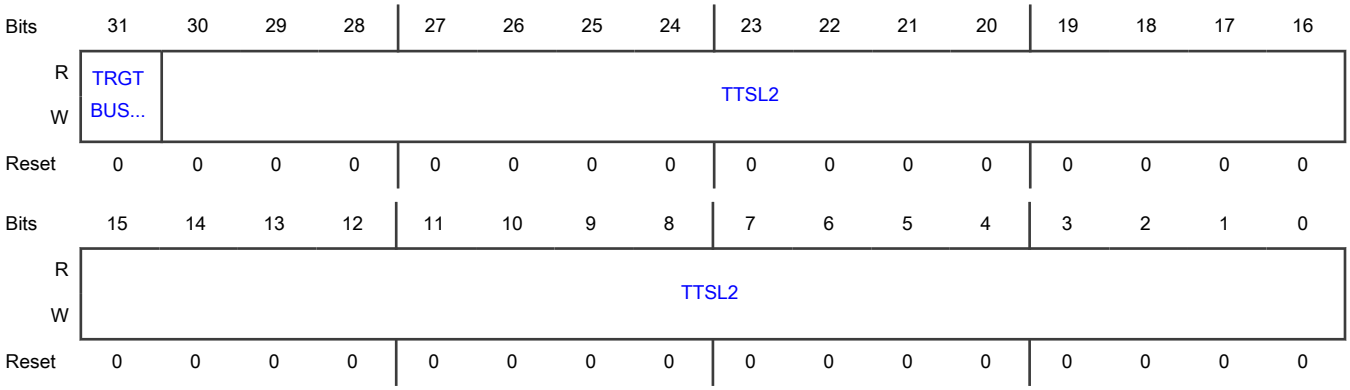
43.8.1.359 PPS2 Target Time Nanoseconds (MAC_PPS2_TARGET_TIME_NANOSECONDS)

PPS2 Target Time Nanoseconds register.

Offset

Register	Offset
MAC_PPS2_TARGET_TIME_NANOSECONDS	BA4h

Diagram



Fields

Field	Description
31 TRGTBUSY2	PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. Programming the PPSCMD0 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. 0 - PPS Target Time Register Busy status is not detected 1 - PPS Target Time Register Busy is detected
30-0 TTSL2	Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the value in both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on the TRGTMODSEL0 field (Bits [6:5]) in PPS_CONTROL.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When the TSCTRLSSR bit is reset in the TIMESTAMP_CONTROL register, this value should be (time in ns / 0.465). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. When the TSCTRLSSR bit is set in the TIMESTAMP_CONTROL register, this value should not exceed 0x3B9A_C9FF. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

43.8.1.360 PPS2 Interval (MAC_PPS2_INTERVAL)

The PPS2 Interval register contains the number of units of sub-second increment value between the rising edges of PPS2 signal output (ptp_pps_o[2]).

Offset

Register	Offset
MAC_PPS2_INTERVAL	BA8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSINT2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSINT2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PPSINT2	PPS Output Signal Interval These bits store the interval between the rising edges of PPS2 signal output. PPS Output Signal Interval These bits store the interval between the rising edges of PPS2 signal output. The interval is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between the rising edges of PPS2 signal output is 100 ns (that is, 5 units of sub-second increment value), you should program value 4 (5-1) in this register.

43.8.1.361 PPS2 Width (MAC_PPS2_WIDTH)

The PPS2 Width register contains the number of units of sub-second increment value between the rising and corresponding falling edges of PPS2 signal output (ptp_pps_o[2]).

Offset

Register	Offset
MAC_PPS2_WIDTH	BACh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSWIDTH2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSWIDTH2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PPSWIDTH2	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. The width is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub-second increment value), you should program value 3 (4-1) in this register. Note: The value programmed in this register must be lesser than the value programmed in MAC_PPS0_Interval.

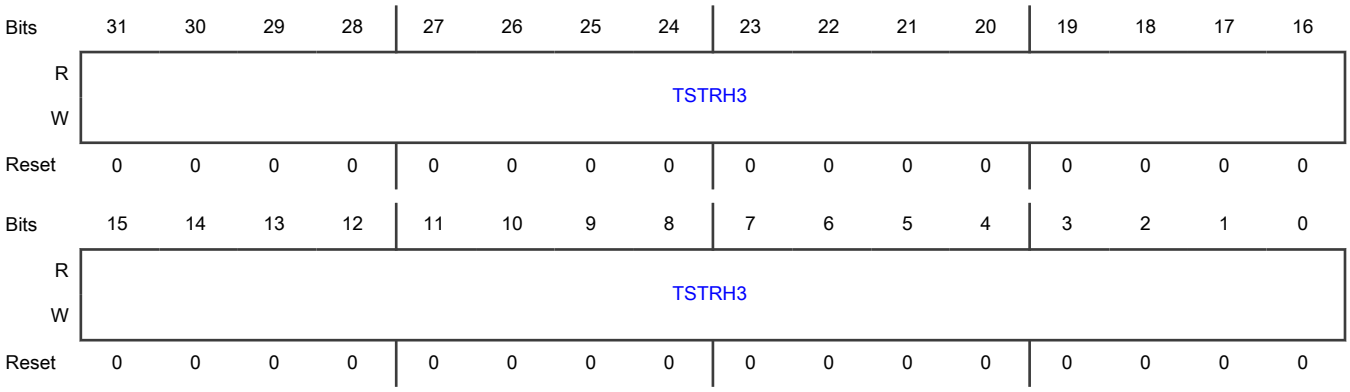
43.8.1.362 PPS3 Target Time Seconds (MAC_PPS3_TARGET_TIME_SECONDS)

The PPS3 Target Time Seconds register, along with PPS3 Target Time Nanoseconds register, is used to schedule an interrupt event [Bit 1 of TIMESTAMP_STATUS] when the system time exceeds the value programmed in these registers.

Offset

Register	Offset
MAC_PPS3_TARGET_TIME_SECONDS	BB0h

Diagram



Fields

Field	Description
31-0 TSTRH3	PPS Target Time Seconds Register This field stores the time in seconds. PPS Target Time Seconds Register This field stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on Target Time mode selected for the corresponding PPS output in the PPS_CONTROL register.

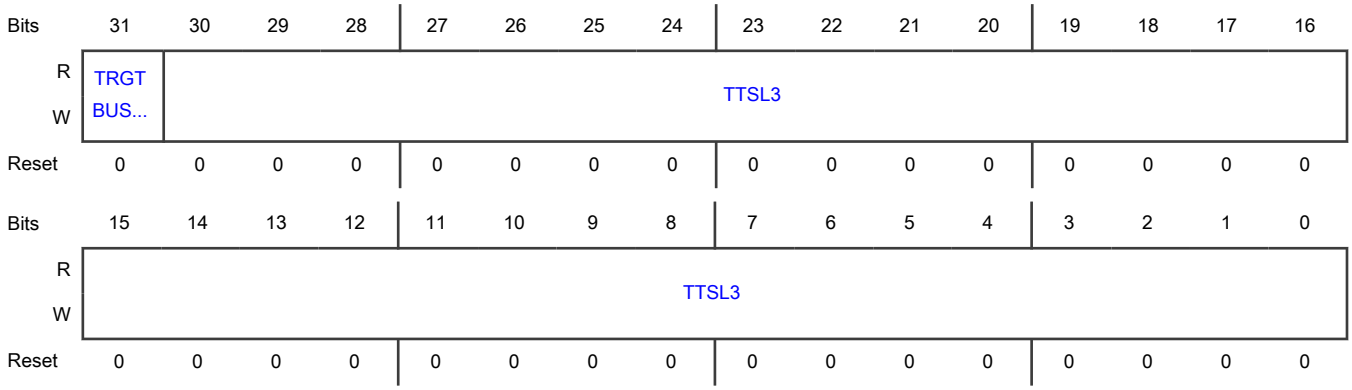
43.8.1.363 PPS3 Target Time Nanoseconds (MAC_PPS3_TARGET_TIME_NANOSECONDS)

PPS3 Target Time Nanoseconds register.

Offset

Register	Offset
MAC_PPS3_TARGET_TIME_NANOSECONDS	BB4h

Diagram



Fields

Field	Description
31 TRGTBUSY3	PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. Programming the PPSCMD0 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. 0 - PPS Target Time Register Busy status is not detected 1 - PPS Target Time Register Busy is detected
30-0 TTSL3	Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the value in both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on the TRGTMODSEL0 field (Bits [6:5]) in PPS_CONTROL. When the TSCTRLSSR bit is reset in the TIMESTAMP_CONTROL register, this value should be (time in ns / 0.465). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. When the TSCTRLSSR bit is set in the TIMESTAMP_CONTROL register, this value should not exceed 0x3B9A_C9FF. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

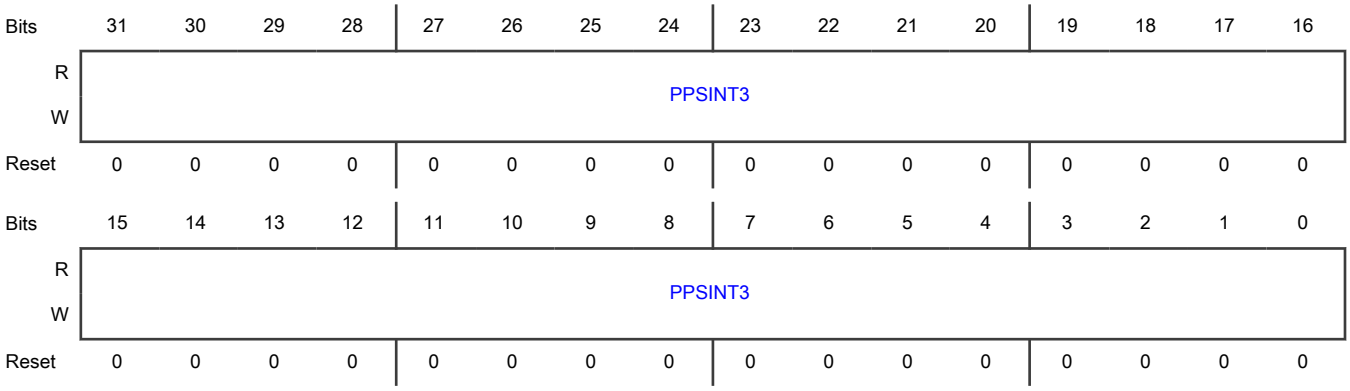
43.8.1.364 PPS3 Interval (MAC_PPS3_INTERVAL)

The PPS3 Interval register contains the number of units of sub-second increment value between the rising edges of PPS3 signal output (ptp_pps_o[3]).

Offset

Register	Offset
MAC_PPS3_INTERVAL	BB8h

Diagram



Fields

Field	Description
31-0 PPSINT3	PPS Output Signal Interval These bits store the interval between the rising edges of PPS3 signal output. PPS Output Signal Interval These bits store the interval between the rising edges of PPS3 signal output. The interval is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between the rising edges of PPS3 signal output is 100 ns (that is, 5 units of sub-second increment value), you should program value 4 (5-1) in this register.

43.8.1.365 PPS3 Width (MAC_PPS3_WIDTH)

The PPS3 Width register contains the number of units of sub-second increment value between the rising and corresponding falling edges of PPS3 signal output (ptp_pps_o[3]).

Offset

Register	Offset
MAC_PPS3_WIDTH	BBCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSWIDTH3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSWIDTH3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PPSWIDTH3	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. The width is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub-second increment value), you should program value 3 (4-1) in this register. Note: The value programmed in this register must be lesser than the value programmed in MAC_PPS0_Interval.

43.8.1.366 PTP Offload Engine Control (MAC_PTO_CONTROL)

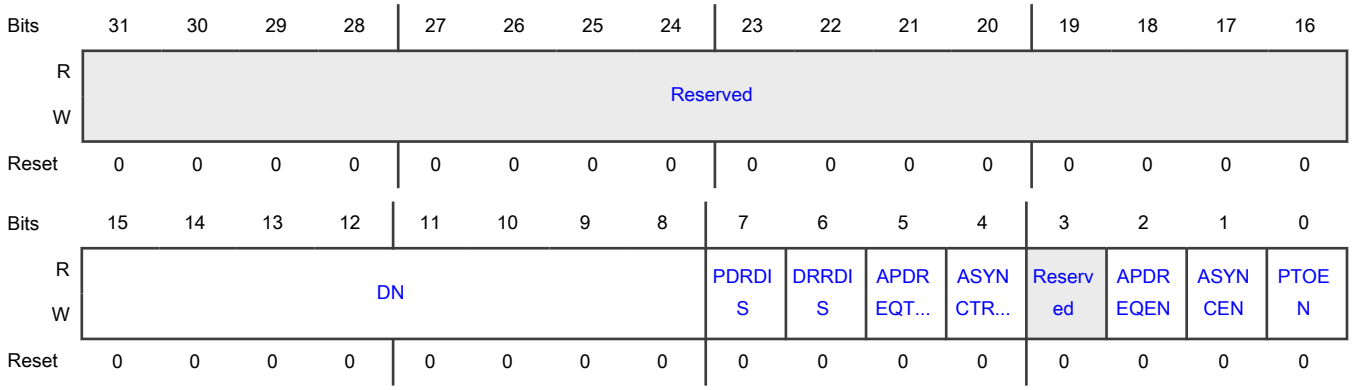
This register controls the PTP Offload Engine operation.

This register controls the PTP Offload Engine operation. This register is available only when the Enable PTP Timestamp Offload feature is selected.

Offset

Register	Offset
MAC_PTO_CONTROL	BC0h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 DN	Domain Number This field indicates the domain Number in which the PTP node is operating. Domain Number This field indicates the domain Number in which the PTP node is operating.
7 PDRDIS	Disable Peer Delay Response response generation When this bit is set, the Peer Delay Response (Pdelay_Resp) response is not be generated for received Peer Delay Request (Pdelay_Req) request packet, as required by the programmed mode. Disable Peer Delay Response response generation When this bit is set, the Peer Delay Response (Pdelay_Resp) response is not be generated for received Peer Delay Request (Pdelay_Req) request packet, as required by the programmed mode. Note: Setting this bit to 1 affects the normal PTP Offload operation and the time synchronization. So, this bit must be set only if there is problem with Pdelay_Resp generation in Hardware and/or Pdelay_Resp generation is handled by Software. 0 - Peer Delay Response response generation is enabled 1 - Peer Delay Response response generation is disabled
6 DRRDIS	Disable PTO Delay Request/Response response generation When this bit is set, the Delay Request and Delay response is not generated for received SYNC and Delay request packet respectively, as required by the programmed mode. Disable PTO Delay Request/Response response generation When this bit is set, the Delay Request and Delay response is not generated for received SYNC and Delay request packet respectively, as required by the programmed mode. 0 - PTO Delay Request/Response response generation is enabled 1 - PTO Delay Request/Response response generation is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
5 APDREQTRIG	Automatic PTP Pdelay_Req message Trigger When this bit is set, one PTP Pdelay_Req message is transmitted. Automatic PTP Pdelay_Req message Trigger When this bit is set, one PTP Pdelay_Req message is transmitted. This bit is automatically cleared after the PTP Pdelay_Req message is transmitted. The application should set the APDREQEN bit for this operation. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Automatic PTP Pdelay_Req message Trigger is disabled 1 - Automatic PTP Pdelay_Req message Trigger is enabled
4 ASYNCTRIG	Automatic PTP SYNC message Trigger When this bit is set, one PTP SYNC message is transmitted. Automatic PTP SYNC message Trigger When this bit is set, one PTP SYNC message is transmitted. This bit is automatically cleared after the PTP SYNC message is transmitted. The application should set the ASYNCEN bit for this operation. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Automatic PTP SYNC message Trigger is disabled 1 - Automatic PTP SYNC message Trigger is enabled
3 —	Reserved
2 APDREQEN	Automatic PTP Pdelay_Req message Enable When this bit is set, PTP Pdelay_Req message is generated periodically based on interval programmed or trigger from application, when the MAC is programmed to be in Peer-to-Peer Transparent mode. 0 - Automatic PTP Pdelay_Req message is disabled 1 - Automatic PTP Pdelay_Req message is enabled
1 ASYNCEN	Automatic PTP SYNC message Enable When this bit is set, PTP SYNC message is generated periodically based on interval programmed or trigger from application, when the MAC is programmed to be in Clock Master mode. 0 - Automatic PTP SYNC message is disabled 1 - Automatic PTP SYNC message is enabled
0 PTOEN	PTP Offload Enable When this bit is set, the PTP Offload feature is enabled. 0 - PTP Offload feature is disabled 1 - PTP Offload feature is enabled

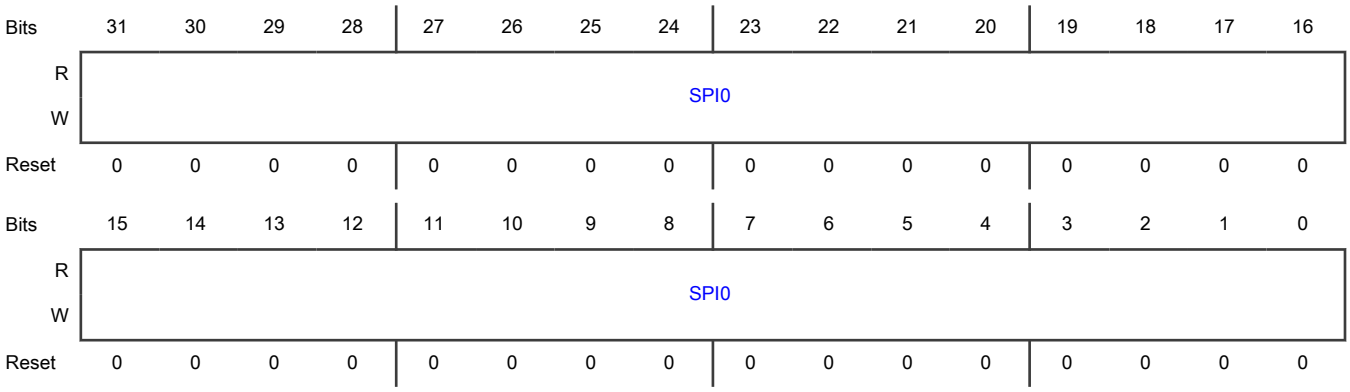
43.8.1.367 Source Port Identity 0 (MAC_SOURCE_PORT_IDENTITY0)

This register contains Bits[31:0] of the 80-bit Source Port Identity of the PTP node. This register is available only when the Enable PTP Timestamp Offload feature is selected.

Offset

Register	Offset
MAC_SOURCE_PORT_IDENTITY0	BC4h

Diagram



Fields

Field	Description
31-0 SPI0	Source Port Identity 0 This field indicates bits [31:0] of sourcePortIdentity of PTP node.

43.8.1.368 Source Port Identity 1 (MAC_SOURCE_PORT_IDENTITY1)

This register contains Bits[63:32] of the 80-bit Source Port Identity of the PTP node. This register is available only when the Enable PTP Timestamp Offload feature is selected.

Offset

Register	Offset
MAC_SOURCE_PORT_IDENTITY1	BC8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPI1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPI1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 SPI1	Source Port Identity 1 This field indicates bits [63:32] of sourcePortIdentity of PTP node.

43.8.1.369 Source Port Identity 2 (MAC_SOURCE_PORT_IDENTITY2)

This register contains Bits[79:64] of the 80-bit Source Port Identity of the PTP node. This register is available only when the Enable PTP Timestamp Offload feature is selected.

Offset

Register	Offset
MAC_SOURCE_PORT_IDENTITY2	BCCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPI2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 SPI2	Source Port Identity 2 This field indicates bits [79:64] of sourcePortIdentity of PTP node.

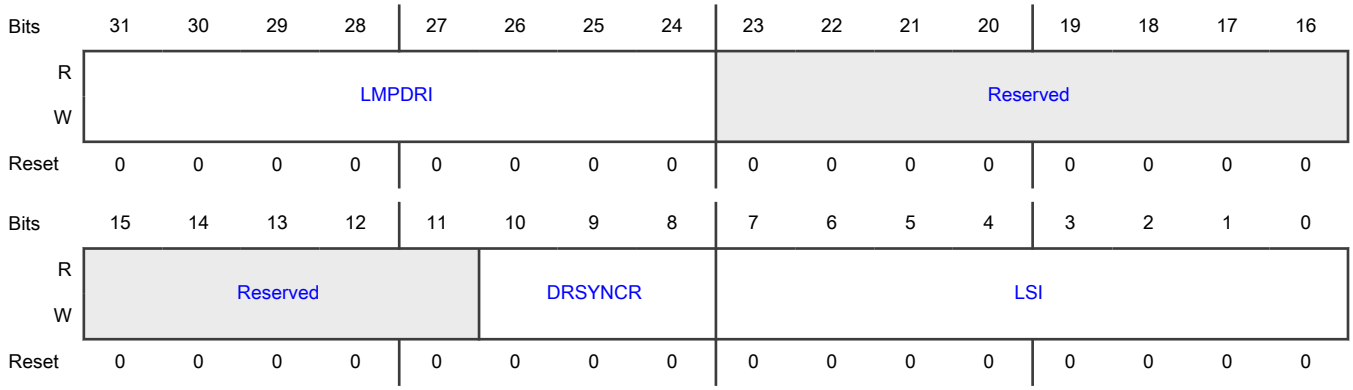
43.8.1.370 Log Message Interval (MAC_LOG_MESSAGE_INTERVAL)

This register contains the periodic intervals for automatic PTP packet generation. This register is available only when the Enable PTP Timestamp Offload feature is selected.

Offset

Register	Offset
MAC_LOG_MESSAGE_INTERVAL	BD0h

Diagram



Fields

Field	Description
31-24 LMPDRI	Log Min Pdelay_Req Interval This field indicates logMinPdelayReqInterval of PTP node. Log Min Pdelay_Req Interval This field indicates logMinPdelayReqInterval of PTP node. This is used to schedule the periodic Pdelay request packet transmission. Allowed values are -15 to 15.Negative value must be represented in 2's-complement form. For example, if the required value is -1, the value programmed must be 0xFF.
23-11 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
10-8 DRSYNCR	Delay_Req to SYNC Ratio In Slave mode, it is used for controlling frequency of Delay_Req messages transmitted. Delay_Req to SYNC Ratio In Slave mode, it is used for controlling frequency of Delay_Req messages transmitted. - 0: DelayReq generated for every received SYNC - 1: DelayReq generated every alternate reception of SYNC - 2: for every 4 SYNC messages - 3: for every 8 SYNC messages - 4: for every 16 SYNC messages - 5: for every 32 SYNC messages - 6-7: Reserved The master sends this information (logMinDelayReqInterval) in the DelayResp PTP messages to the slave. The DWC_ether_qos Receiver processes this value from the received DelayResp messages and updates this field accordingly. In the Slave mode, the host must not write/update this register unless it has to override the received value. In Master mode, the sum of this field and logSyncInterval (LSI) field is provided in the logMinDelayReqInterval field of the generated multicast Delay_Resp PTP message. Access restriction applies. Updated based on the event. Setting 1 sets. Setting 0 clears. 000 - DelayReq generated for every received SYNC 001 - DelayReq generated every alternate reception of SYNC 010 - for every 4 SYNC messages 011 - for every 8 SYNC messages 100 - for every 16 SYNC messages 101 - for every 32 SYNC messages 110 - Reserved
7-0 LSI	Log Sync Interval This field indicates the periodicity of the automatically generated SYNC message when the PTP node is Master. Log Sync Interval This field indicates the periodicity of the automatically generated SYNC message when the PTP node is Master. Allowed values are -15 to 15. Negative value must be represented in 2's-complement form. For example, if the required value is -1, the value programmed must be 0xFF.

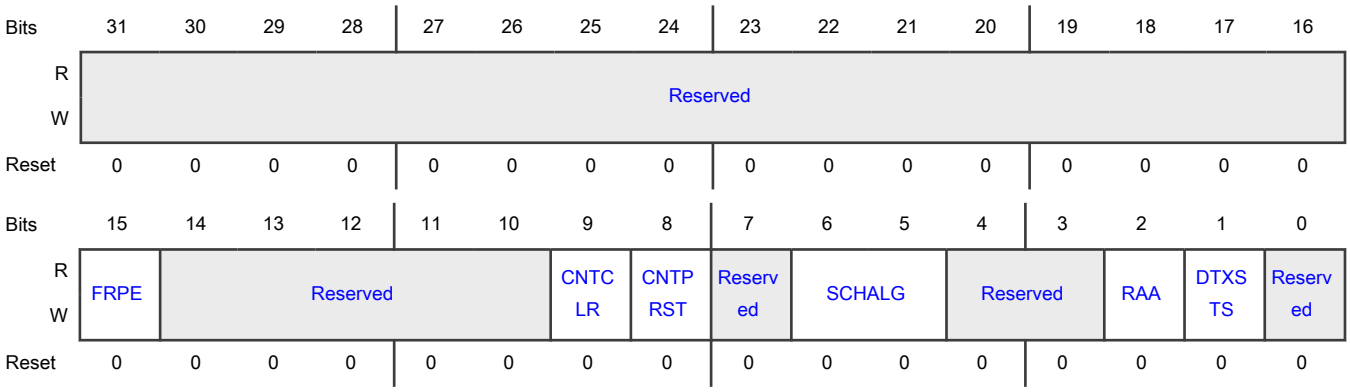
43.8.1.371 MTL Operation Mode (MTL_OPERATION_MODE)

The Operation Mode register establishes the Transmit and Receive operating modes and commands.

Offset

Register	Offset
MTL_OPERATION_MODE	C00h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 FRPE	Flexible Rx parser Enable When this bit is set to 1, the Programmable Rx Parser functionality is enabled. Flexible Rx parser Enable When this bit is set to 1, the Programmable Rx Parser functionality is enabled. When the Rx parser is disabled and if the Rx parser is in the middle of the parsing then it gets disabled only after completing the current packet parsing. When the Rx parser is enabled from disabled state then the Rx parser gets activated for the next immediate packet. 0 - Flexible Rx parser is disabled 1 - Flexible Rx parser is enabled
14-10 —	Reserved
9 CNTCLR	Counters Reset When this bit is set, all counters are reset. Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle. If this bit is set along with CNT_PRESET bit, CNT_PRESET has precedence. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Counters are not reset 1 - All counters are reset
8 CNTPRST	Counters Preset When this bit is set, - MTL_TxQ[0-7]_Underflow register is initialized/preset to 12'h7F0. Counters Preset When this bit is set, - MTL_TxQ[0-7]_Underflow register is initialized/preset to 12'h7F0. - Missed Packet and Overflow Packet counters in MTL_RxQ[0-7]_Missed_Packet_Overflow_Cnt register is initialized/preset to 12'h7F0. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Counters Preset is disabled 1 - Counters Preset is enabled
7 —	Reserved
6-5 SCHALG	Tx Scheduling Algorithm This field indicates the algorithm for Tx scheduling: 00 - WRR algorithm 01 - WFQ algorithm when DCB feature is selected.Otherwise, Reserved 10 - DWRR algorithm when DCB feature is selected.Otherwise, Reserved 11 - Strict priority algorithm
4-3 —	Reserved
2 RAA	Receive Arbitration Algorithm This field is used to select the arbitration algorithm for the Rx side. Receive Arbitration Algorithm This field is used to select the arbitration algorithm for the Rx side. - 0: Strict priority (SP) Queue 0 has the lowest priority and the last queue has the highest priority. - 1: Weighted Strict Priority (WSP) 0 - Strict priority (SP) 1 - Weighted Strict Priority (WSP)
1 DTXSTS	Drop Transmit Status When this bit is set, the Tx packet status received from the MAC is dropped in the MTL. Drop Transmit Status When this bit is set, the Tx packet status received from the MAC is dropped in the MTL. When this bit is reset, the Tx packet status received from the MAC is forwarded to the application.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Drop Transmit Status is disabled 1 - Drop Transmit Status is enabled
0 —	Reserved

43.8.1.372 FIFO Debug Access Control and Status (MTL_DBG_CTL)

The FIFO Debug Access Control and Status register controls the operation mode of FIFO debug access.

Offset

Register	Offset
MTL_DBG_CTL	C08h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved													Reserved		Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STSIE	PKTIE	FIFOSEL		FIFOWREN	FIFORDEN	RSTSEL	RSTALL	Reserved	PKTSTATE		Reserved	BYTEEN		DBGMOD	FDBGEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-19 —	Reserved
18-17 —	Reserved
16 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
15 STSIE	Transmit Status Available Interrupt Status Enable When this bit is set, an interrupt is generated when Transmit status is available in slave mode. 0 - Transmit Packet Available Interrupt Status is disabled 1 - Transmit Packet Available Interrupt Status is enabled
14 PKTIE	Receive Packet Available Interrupt Status Enable When this bit is set, an interrupt is generated when EOP of received packet is written to the Rx FIFO. 0 - Receive Packet Available Interrupt Status is disabled 1 - Receive Packet Available Interrupt Status is enabled
13-12 FIFOSEL	FIFO Selected for Access This field indicates the FIFO selected for debug access: 00 - Tx FIFO 01 - Tx Status FIFO (only read access when SLVMOD is set) 10 - TSO FIFO (cannot be accessed when SLVMOD is set) 11 - Rx FIFO
11 FIFOWREN	FIFO Write Enable When this bit is set, it enables the Write operation on selected FIFO when FIFO Debug Access is enabled. FIFO Write Enable When this bit is set, it enables the Write operation on selected FIFO when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0 - FIFO Write is disabled 1 - FIFO Write is enabled
10 FIFORDEN	FIFO Read Enable When this bit is set, it enables the Read operation on selected FIFO when FIFO Debug Access is enabled. FIFO Read Enable When this bit is set, it enables the Read operation on selected FIFO when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0 - FIFO Read is disabled 1 - FIFO Read is enabled
9 RSTSEL	Reset Pointers of Selected FIFO When this bit is set, the pointers of the currently-selected FIFO are reset when FIFO Debug Access is enabled. Reset Pointers of Selected FIFO When this bit is set, the pointers of the currently-selected FIFO are reset when FIFO Debug Access is enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0 - Reset Pointers of Selected FIFO is disabled 1 - Reset Pointers of Selected FIFO is enabled
8 RSTALL	Reset All Pointers When this bit is set, the pointers of all FIFOs are reset when FIFO Debug Access is enabled. Reset All Pointers When this bit is set, the pointers of all FIFOs are reset when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0 - Reset All Pointers is disabled 1 - Reset All Pointers is enabled
7 —	Reserved
6-5 PKTSTATE	Encoded Packet State This field is used to write the control information to the Tx FIFO or Rx FIFO. Encoded Packet State This field is used to write the control information to the Tx FIFO or Rx FIFO. Tx FIFO: - 00: Packet Data - 01: Control Word - 10: SOP Data - 11: EOP Data Rx FIFO: - 00: Packet Data - 01: Normal Status - 10: Last Status - 11: EOP 00 - Packet Data 01 - Control Word/Normal Status 10 - SOP Data/Last Status 11 - EOP Data/EOP
4	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
3-2 BYTEEN	Byte Enables This field indicates the number of data bytes valid in the data register during Write operation. Byte Enables This field indicates the number of data bytes valid in the data register during Write operation. This is valid only when PKTSTATE is 2'b10 (EOP) and Tx FIFO or Rx FIFO is selected. 00 - Byte 0 valid 01 - Byte 0 and Byte 1 are valid 10 - Byte 0, Byte 1, and Byte 2 are valid 11 - All four bytes are valid
1 DBGMOD	Debug Mode Access to FIFO When this bit is set, it indicates that the current access to the FIFO is read, write, and debug access. Debug Mode Access to FIFO When this bit is set, it indicates that the current access to the FIFO is read, write, and debug access. In this mode, the following access types are allowed: - Read and Write access to Tx FIFO, TSO FIFO, and Rx FIFO - Read access is allowed to Tx Status FIFO. When this bit is reset, it indicates that the current access to the FIFO is slave access bypassing the DMA. In this mode, the following access are allowed: - Write access to the Tx FIFO - Read access to the Rx FIFO and Tx Status FIFO 0 - Debug Mode Access to FIFO is disabled 1 - Debug Mode Access to FIFO is enabled
0 FDBGEN	FIFO Debug Access Enable When this bit is set, it indicates that the debug mode access to the FIFO is enabled. FIFO Debug Access Enable When this bit is set, it indicates that the debug mode access to the FIFO is enabled. When this bit is reset, it indicates that the FIFO can be accessed only through a master interface. 0 - FIFO Debug Access is disabled 1 - FIFO Debug Access is enabled

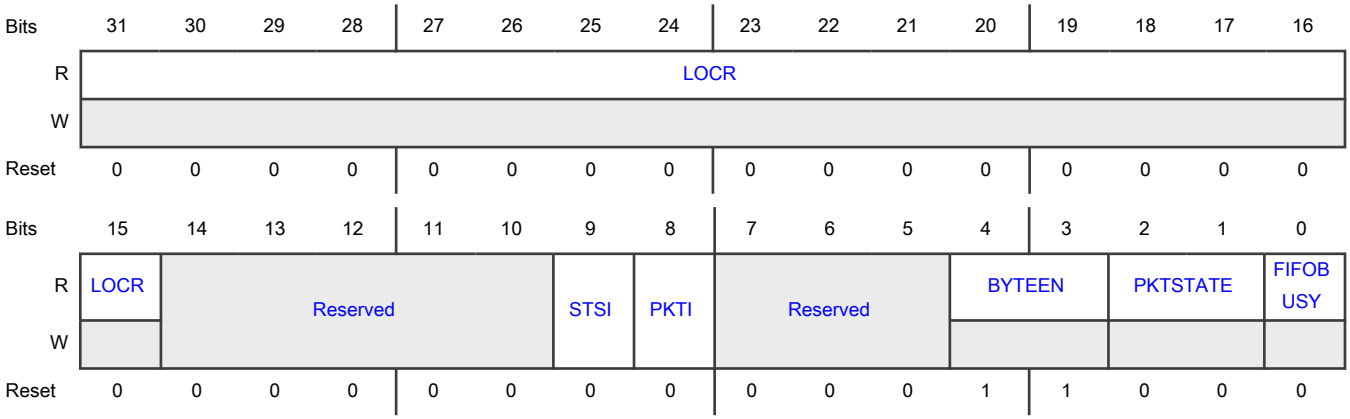
43.8.1.373 FIFO Debug Status (MTL_DBG_STS)

The FIFO Debug Status register contains the status of FIFO debug access.

Offset

Register	Offset
MTL_DBG_STS	C0Ch

Diagram



Fields

Field	Description
31-15 LOCR	Remaining Locations in the FIFO Slave Access Mode: This field indicates the space available in selected FIFO. Remaining Locations in the FIFO Slave Access Mode: This field indicates the space available in selected FIFO. Debug Access Mode: This field contains the Write or Read pointer value of the selected FIFO during Write or Read operation, respectively. Reset: In single Tx Queue configurations, (DWC_EQOS_TXFIFO_SIZE/(DWC_EQOS_DATAWIDTH/8)), Otherwise 0000H
14-10 —	Reserved
9 STSI	Transmit Status Available Interrupt Status When set, this bit indicates that the Slave mode Tx packet is transmitted, and the status is available in Tx Status FIFO. Transmit Status Available Interrupt Status When set, this bit indicates that the Slave mode Tx packet is transmitted, and the status is available in Tx Status FIFO. This bit is reset when 1 is written to this bit. 0 - Transmit Status Available Interrupt Status not detected 1 - Transmit Status Available Interrupt Status detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 PKTI	Receive Packet Available Interrupt Status When set, this bit indicates that MAC layer has written the EOP of received packet to the Rx FIFO. Receive Packet Available Interrupt Status When set, this bit indicates that MAC layer has written the EOP of received packet to the Rx FIFO. This bit is reset when 1 is written to this bit. 0 - Receive Packet Available Interrupt Status not detected 1 - Receive Packet Available Interrupt Status detected
7-5 —	Reserved
4-3 BYTEEN	Byte Enables This field indicates the number of data bytes valid in the data register during Read operation. Byte Enables This field indicates the number of data bytes valid in the data register during Read operation. This is valid only when PKTSTATE is 2'b10 (EOP) and Tx FIFO or Rx FIFO is selected. 00 - Byte 0 valid 01 - Byte 0 and Byte 1 are valid 10 - Byte 0, Byte 1, and Byte 2 are valid 11 - All four bytes are valid
2-1 PKTSTATE	Encoded Packet State This field is used to get the control or status information of the selected FIFO. Encoded Packet State This field is used to get the control or status information of the selected FIFO. Tx FIFO: - 00: Packet Data - 01: Control Word - 10: SOP Data - 11: EOP Data Rx FIFO: - 00: Packet Data - 01: Normal Status - 10: Last Status - 11: EOP This field is applicable only for Tx FIFO and Rx FIFO during Read operation. 00 - Packet Data 01 - Control Word/Normal Status 10 - SOP Data/Last Status

Table continues on the next page...

Table continued from the previous page...

Field	Description
	11 - EOP Data/EOP
0 FIFOBUSY	FIFO Busy When set, this bit indicates that a FIFO operation is in progress in the MAC and content of the following fields is not valid: - All other fields of this register - All fields of the MTL_FIFO_DEBUG_DATA register 0 - FIFO Busy not detected 1 - FIFO Busy detected

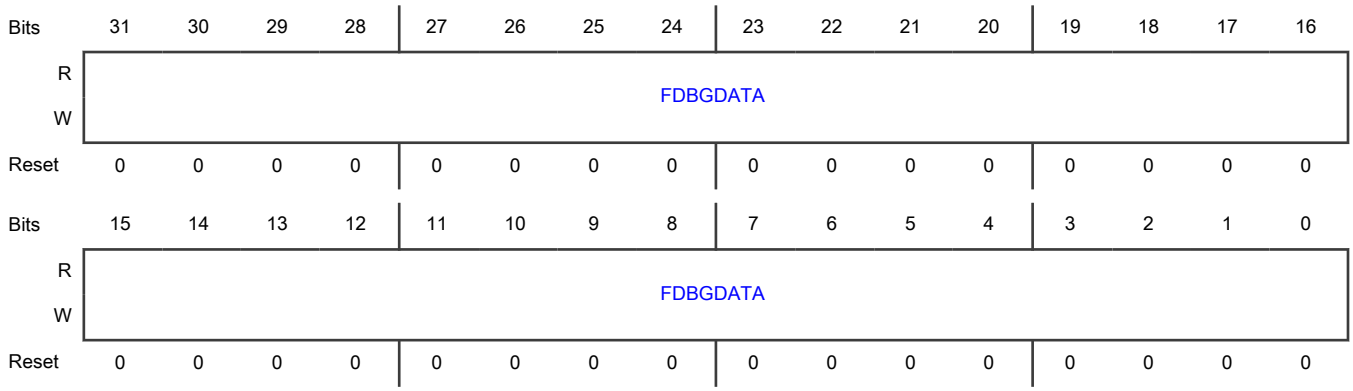
43.8.1.374 FIFO Debug Data (MTL_FIFO_DEBUG_DATA)

The FIFO Debug Data register contains the data to be written to or read from the FIFOs.

Offset

Register	Offset
MTL_FIFO_DEBUG_DATA	C10h

Diagram



Fields

Field	Description
31-0 FDBGDATA	FIFO Debug Data During debug or slave access write operation, this field contains the data to be written to the Tx FIFO, Rx FIFO, or TSO FIFO. FIFO Debug Data During debug or slave access write operation, this field contains the data to be written to the Tx FIFO, Rx FIFO, or TSO FIFO. During debug or slave access read operation, this field contains the data read from the Tx FIFO, Rx FIFO, TSO FIFO, or Tx Status FIFO.

43.8.1.375 MTL Interrupt Status (MTL_INTERRUPT_STATUS)

The software driver (application) reads this register during interrupt service routine or polling to determine the interrupt status of MTL queues and the MAC.

Offset

Register	Offset
MTL_INTERRUPT_STATUS	C20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								MTLPIS	Reserved				ESTIS	DBGIS	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Reserved	Reserved	Reserved	Q4IS	Q3IS	Q2IS	Q1IS	Q0IS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23 MTLPIS	MTL Rx Parser Interrupt Status This bit indicates that there is an interrupt from Rx Parser Block. MTL Rx Parser Interrupt Status This bit indicates that there is an interrupt from Rx Parser Block. To reset this bit, the application must read the MTL_RXP_INTERRUPT_STATUS register to get the exact cause of the interrupt and clear its source. 0 - MTL Rx Parser Interrupt status not detected 1 - MTL Rx Parser Interrupt status detected
22-19 —	Reserved
18 ESTIS	EST (TAS- 802. EST (TAS- 802.1Qbv) Interrupt Status This bit indicates an interrupt event during the operation of 802.1Qbv. To reset this bit, the application must clear the error/event that has caused the Interrupt.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - EST (TAS- 802.1Qbv) Interrupt status not detected 1 - EST (TAS- 802.1Qbv) Interrupt status detected
17 DBGIS	Debug Interrupt status This bit indicates an interrupt event during the slave access. Debug Interrupt status This bit indicates an interrupt event during the slave access. To reset this bit, the application must read the FIFO Debug Access Status register to get the exact cause of the interrupt and clear its source. 0 - Debug Interrupt status not detected 1 - Debug Interrupt status detected
16 —	Reserved
15-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 Q4IS	Queue 4 Interrupt status This bit indicates that there is an interrupt from Queue 4. Queue 4 Interrupt status This bit indicates that there is an interrupt from Queue 4. To reset this bit, the application must read the MTL_Q4_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0 - Queue 4 Interrupt status not detected 1 - Queue 4 Interrupt status detected
3 Q3IS	Queue 3 Interrupt status This bit indicates that there is an interrupt from Queue 3. Queue 3 Interrupt status This bit indicates that there is an interrupt from Queue 3. To reset this bit, the application must read the MTL_Q3_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0 - Queue 3 Interrupt status not detected 1 - Queue 3 Interrupt status detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 Q2IS	Queue 2 Interrupt status This bit indicates that there is an interrupt from Queue 2. Queue 2 Interrupt status This bit indicates that there is an interrupt from Queue 2. To reset this bit, the application must read the MTL_Q2_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0 - Queue 2 Interrupt status not detected 1 - Queue 2 Interrupt status detected
1 Q1IS	Queue 1 Interrupt status This bit indicates that there is an interrupt from Queue 1. Queue 1 Interrupt status This bit indicates that there is an interrupt from Queue 1. To reset this bit, the application must read the MTL_Q1_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0 - Queue 1 Interrupt status not detected 1 - Queue 1 Interrupt status detected
0 Q0IS	Queue 0 Interrupt status This bit indicates that there is an interrupt from Queue 0. Queue 0 Interrupt status This bit indicates that there is an interrupt from Queue 0. To reset this bit, the application must read Queue 0 Interrupt Control and Status register to get the exact cause of the interrupt and clear its source. 0 - Queue 0 Interrupt status not detected 1 - Queue 0 Interrupt status detected

43.8.1.376 Receive Queue and DMA Channel Mapping 0 (MTL_RXQ_DMA_MAP0)

The Receive Queue and DMA Channel Mapping 0 register is reserved in EQOS-CORE and EQOS-MTL configurations.

Offset

Register	Offset
MTL_RXQ_DMA_MAP0	C30h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			Q3DDMACH	Reserved	Q3MDMACH			Reserved			Q2DDMACH	Reserved	Q2MDMACH		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			Q1DDMACH	Reserved	Q1MDMACH			Reserved			Q0DDMACH	Reserved	Q0MDMACH		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 Q3DDMACH	Queue 3 Enabled for Dynamic (per packet) DMA Channel Selection When set, this bit indicates that the packets received in Queue 3 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 3 Enabled for Dynamic (per packet) DMA Channel Selection When set, this bit indicates that the packets received in Queue 3 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 3 are routed to the DMA Channel programmed in the Q3MDMACH field (Bits[26:24]). 0 - Queue 3 disabled for DA-based DMA Channel Selection 1 - Queue 3 enabled for DA-based DMA Channel Selection
27 —	Reserved
26-24 Q3MDMACH	Queue 3 Mapped to DMA Channel This field controls the routing of the received packet in Queue 3 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q3DDMACH field is reset. Queue 3 Mapped to DMA Channel This field controls the routing of the received packet in Queue 3 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q3DDMACH field is reset. Note: The width of this field depends on the number of RX DMA channels and not all the values may be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the others are reserved
23-21 —	Reserved
20 Q2DDMACH	Queue 2 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 2 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 2 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 2 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 2 are routed to the DMA Channel programmed in the Q2MDMACH field (Bits[18:16]). 0 - Queue 2 disabled for DA-based DMA Channel Selection 1 - Queue 2 enabled for DA-based DMA Channel Selection
19 —	Reserved
18-16 Q2MDMACH	Queue 2 Mapped to DMA Channel This field controls the routing of the received packet in Queue 2 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q2DDMACH field is reset.
15-13 —	Reserved
12 Q1DDMACH	Queue 1 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 1 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 1 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 1 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When reset, this bit indicates that the packets received in Queue 1 are routed to the DMA Channel programmed in the Q1MDMACH field (Bits[10:8]). 0 - Queue 1 disabled for DA-based DMA Channel Selection 1 - Queue 1 enabled for DA-based DMA Channel Selection
11 —	Reserved
10-8 Q1MDMACH	Queue 1 Mapped to DMA Channel This field controls the routing of the received packet in Queue 1 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q1DDMACH field is reset. Queue 1 Mapped to DMA Channel This field controls the routing of the received packet in Queue 1 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q1DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values may be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the other bits are reserved.
7-5 —	Reserved
4 Q0DDMACH	Queue 0 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 0 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 0 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 0 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 0 are routed to the DMA Channel programmed in the Q0MDMACH field. 0 - Queue 0 disabled for DA-based DMA Channel Selection

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Queue 0 enabled for DA-based DMA Channel Selection
3 —	Reserved
2-0 Q0MDMACH	Queue 0 Mapped to DMA Channel This field controls the routing of the packet received in Queue 0 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q0DDMACH field is reset. Queue 0 Mapped to DMA Channel This field controls the routing of the packet received in Queue 0 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q0DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values may be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the other bits are reserved.

43.8.1.377 Receive Queue and DMA Channel Mapping 1 (MTL_RXQ_DMA_MAP1)

The Receive Queue and DMA Channel Mapping 1 register is reserved in EQOS-CORE and EQOS-MTL configurations.

Offset

Register	Offset
MTL_RXQ_DMA_MAP1	C34h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			Reserved	Reserved	Reserved			Reserved			Reserved	Reserved	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			Reserved	Reserved	Reserved			Reserved			Q4DDMACH	Reserved	Q4MDMACH		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 —	Reserved
27 —	Reserved
26-24 —	Reserved
23-21 —	Reserved
20 —	Reserved
19 —	Reserved
18-16 —	Reserved
15-13 —	Reserved
12 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
11 —	Reserved
10-8 —	Reserved
7-5 —	Reserved
4 Q4DDMACH	Queue 4 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 4 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 4 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 4 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 4 are routed to the DMA Channel programmed in the Q4MDMACH field. 0 - Queue 4 disabled for DA-based DMA Channel Selection 1 - Queue 4 enabled for DA-based DMA Channel Selection
3 —	Reserved
2-0 Q4MDMACH	Queue 4 Mapped to DMA Channel This field controls the routing of the packet received in Queue 4 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q4DDMACH field is reset. Queue 4 Mapped to DMA Channel This field controls the routing of the packet received in Queue 4 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q4DDMACH field is reset.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	The width of this field depends on the number of RX DMA channels and not all the values may be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the other bits are reserved.

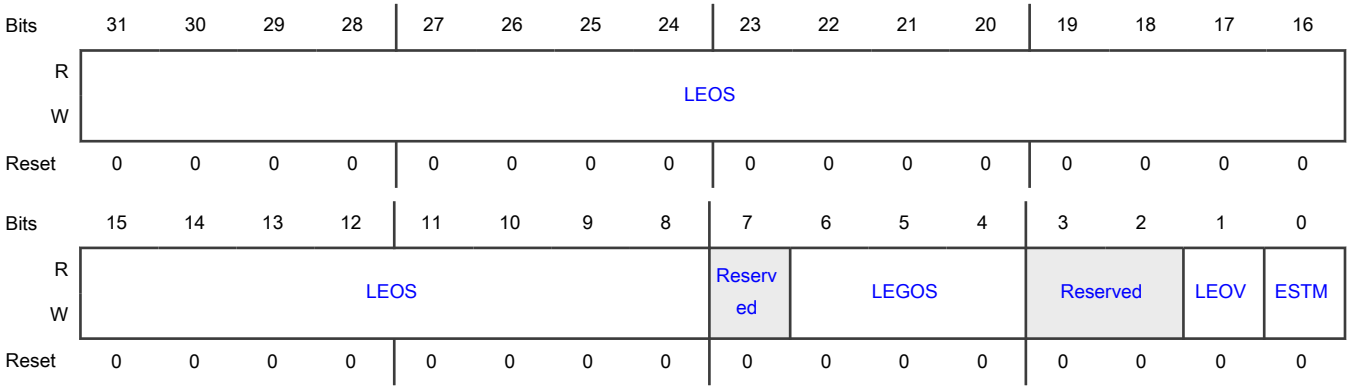
43.8.1.378 Time Based Scheduling Control (MTL_TBS_CTRL)

This register controls the operation of Time Based Scheduling.

Offset

Register	Offset
MTL_TBS_CTRL	C40h

Diagram



Fields

Field	Description
31-8 LEOS	Launch Expiry Offset The value in units of 256 nanoseconds that has to be added to the Launch time to compute the Launch Expiry time. Launch Expiry Offset The value in units of 256 nanoseconds that has to be added to the Launch time to compute the Launch Expiry time. Value valid only when LEOV is set. Max value: 999,999,999 ns, additionally should be smaller than CTR-1 value when ESTM mode is set since this value is a modulo CTR value.
7 —	Reserved
6-4	Launch Expiry GSN Offset The number GSN slots that has to be added to the Launch GSN to compute the Launch Expiry time.

Table continues on the next page...

Table continued from the previous page...

Field	Description
LEGOS	Launch Expiry GSN Offset The number GSN slots that has to be added to the Launch GSN to compute the Launch Expiry time. Value valid only when LEOV is set.
3-2 —	Reserved
1 LEOV	Launch Expiry Offset Valid When set indicates the LEOS field is valid. Launch Expiry Offset Valid When set indicates the LEOS field is valid. When not set, indicates the Launch Expiry Offset is not valid and the MTL must not check for Launch expiry time. 0 - LEOS field is invalid 1 - LEOS field is valid
0 ESTM	EST offset Mode When this bit is set, the Launch Time value used in Time Based Scheduling is interpreted as an EST offset value and is added to the Base Time Register (BTR) of the current list. EST offset Mode When this bit is set, the Launch Time value used in Time Based Scheduling is interpreted as an EST offset value and is added to the Base Time Register (BTR) of the current list. When reset, the Launch Time value is used as an absolute value that should be compared with the System time [39:8]. 0 - EST offset Mode is disabled 1 - EST offset Mode is enabled

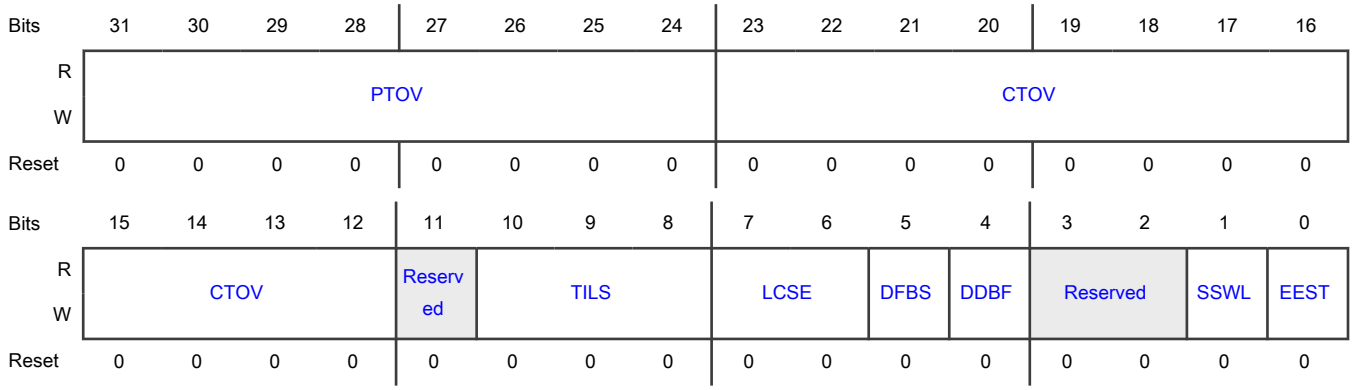
43.8.1.379 Enhancements to Scheduled Transmission Control (MTL_EST_CONTROL)

This register controls the operation of Enhancements to Scheduled Transmission (IEEE802.1Qbv).

Offset

Register	Offset
MTL_EST_CONTROL	C50h

Diagram



Fields

Field	Description
31-24 PTOV	PTP Time Offset Value The value of PTP Clock period multiplied by 6 in nanoseconds. PTP Time Offset Value The value of PTP Clock period multiplied by 6 in nanoseconds. This value is needed to avoid transmission overruns at the beginning of the installation of a new GCL.
23-12 CTOV	Current Time Offset Value Provides a 12 bit time offset value in nano second that is added to the current time to compensate for all the implementation pipeline delays such as the CDC sync delay, buffering delays, data path delays etc. Current Time Offset Value Provides a 12 bit time offset value in nano second that is added to the current time to compensate for all the implementation pipeline delays such as the CDC sync delay, buffering delays, data path delays etc. This offset helps to ensure that the impact of gate controls is visible on the line exactly at the pre-determined schedule (or as close to the schedule as possible).
11 —	Reserved
10-8 TILS	Time Interval Left Shift Amount This field provides the left shift amount for the programmed Time Interval values used in the Gate Control Lists. Time Interval Left Shift Amount This field provides the left shift amount for the programmed Time Interval values used in the Gate Control Lists. - 000: No left shift needed (equal to x1ns) - 001: Left shift TI by 1 bit (equal to x2ns) - 010: Left shift TI by 2 bits (equal to x4ns) - . - . - 100: Left shift TI by 7 bits (equal to x128ns)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Based on the configuration one or more bits of this field should be treated as Reserved/Read-Only.
7-6 LCSE	Loop Count to report Scheduling Error Programmable number of GCL list iterations before reporting an HLBS error defined in EST_STATUS register. 00 - 4 iterations 01 - 8 iterations 10 - 16 iterations 11 - 32 iterations
5 DFBS	Drop Frames causing Scheduling Error When set frames reported to cause HOL Blocking due to not getting scheduled (HLBS field of EST_STATUS register) after 4,8,16,32 (based on LCSE field of this register) GCL iterations are dropped. 0 - Do not Drop Frames causing Scheduling Error 1 - Drop Frames causing Scheduling Error
4 DDBF	Do not Drop frames during Frame Size Error When set, frames are not be dropped during Head-of-Line blocking due to Frame Size Error (HLBF field of MTL_EST_STATUS register). 0 - Drop frames during Frame Size Error 1 - Do not Drop frames during Frame Size Error
3-2 —	Reserved
1 SSWL	Switch to S/W owned list When set indicates that the software has programmed that list that it currently owns (SWOL) and the hardware should switch to the new list based on the new BTR. Switch to S/W owned list When set indicates that the software has programmed that list that it currently owns (SWOL) and the hardware should switch to the new list based on the new BTR. Hardware clears this bit when the switch to the SWOL happens to indicate the completion of the switch or when an BTR error (BTRE in Status register) is set. When BTRE is set this bit is cleared but SWOL is not updated as the switch was not successful. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Switch to S/W owned list is disabled 1 - Switch to S/W owned list is enabled
0 EEST	Enable EST When reset, the gate control list processing is halted and all gates are assumed to be in Open state. Enable EST When reset, the gate control list processing is halted and all gates are assumed to be in Open state. Should be set for the hardware to start processing the gate control lists. During the toggle from 0 to 1, the gate control list processing starts only after the SSWL bit it set. When DWC_EQOS_ASP_ECC is selected during the configuration, if any uncorrectable error is detected in the EST memory the hardware will reset this bit and disable the EST function.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - EST is disabled 1 - EST is enabled

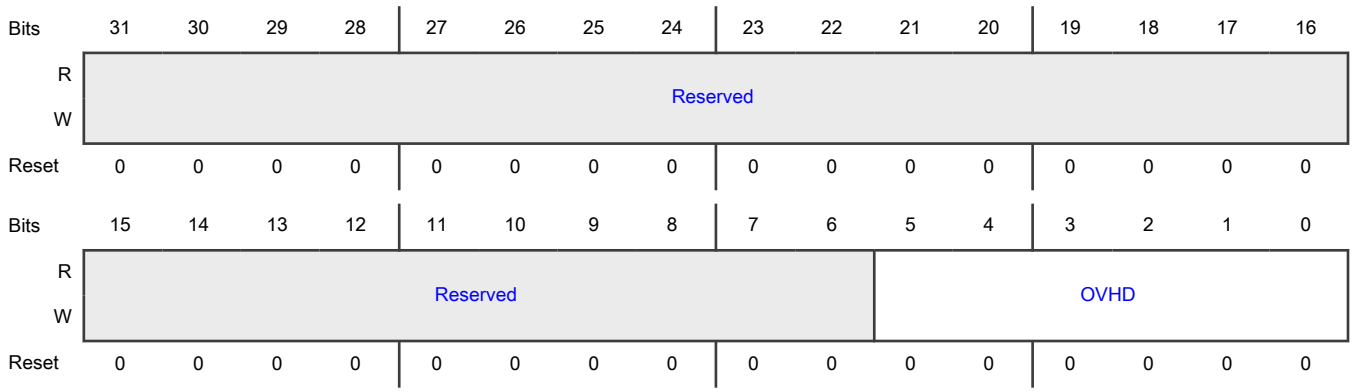
43.8.1.380 MTL_EST_EXT_CONTROL (MTL_EST_EXT_CONTROL)

This register indicates the number of overhead bytes for EST related scheduling.

Offset

Register	Offset
MTL_EST_EXT_CONTROL	C54h

Diagram



Fields

Field	Description
31-6 —	Reserved
5-0 OVHD	Overhead Bytes Value This field indicates the fixed overhead for every packet to account for EST Scheduler Delay, IPG or EIPG, and Preamble bytes.

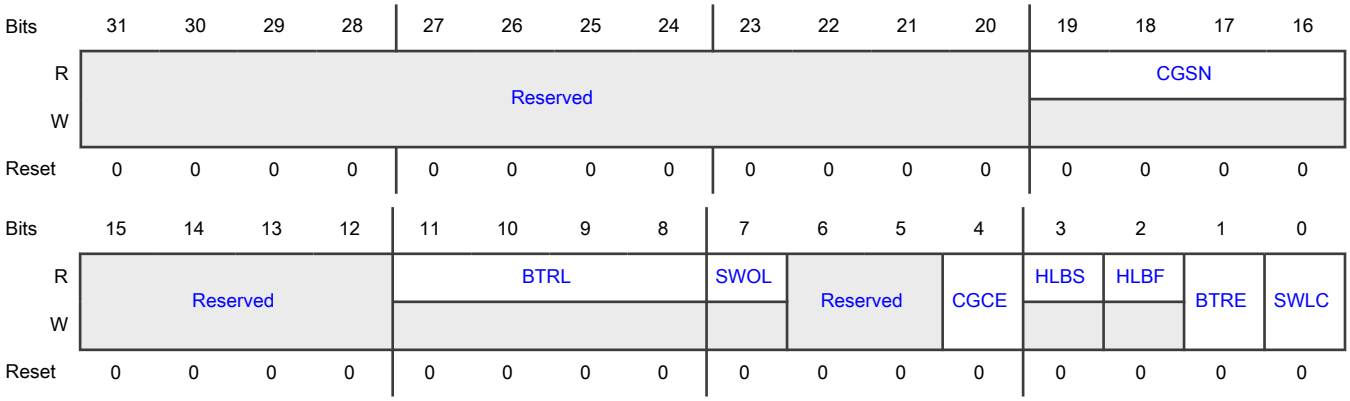
43.8.1.381 Enhancements to Scheduled Transmission Status (MTL_EST_STATUS)

This register provides Status related to Enhancements to Scheduled Transmission (IEEE802.1Qbv).

Offset

Register	Offset
MTL_EST_STATUS	C58h

Diagram



Fields

Field	Description
31-20 —	Reserved
19-16 CGSN	Current GCL Slot Number Indicates the slot number of the GCL list. Current GCL Slot Number Indicates the slot number of the GCL list. Slot number is a modulo 16 count of the GCL List loops executed so far. Even if a new GCL list is installed, the count is incremental.
15-12 —	Reserved
11-8 BTRL	BTR Error Loop Count Provides the minimum count (N) for which the equation Current Time =< New BTR + (N * New Cycle Time) becomes true. BTR Error Loop Count Provides the minimum count (N) for which the equation Current Time =< New BTR + (N * New Cycle Time) becomes true. N = "1111" indicates the iterations exceeded the value of 8 and the hardware was not able to update New BTR to be equal to or greater than Current Time. Software intervention is needed to update the New BTR. Value cleared when BTRE field of this register is cleared.
7 SWOL	S/W owned list When '0' indicates Gate control list number "0" is owned by software and when "1" indicates the Gate Control list "1" is owned by the software. S/W owned list

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When '0' indicates Gate control list number "0" is owned by software and when "1" indicates the Gate Control list "1" is owned by the software. Any reads/writes by the software (using indirect access via GCL_Control) is directed to the list indicated by this value by default. The inverse of this value is treated as HWOL. R/W operations performed by hardware are directed to the list pointed by HWOL by default. 0 - Gate control list number "0" is owned by software 1 - Gate control list number "1" is owned by software
6-5 —	Reserved
4 CGCE	Constant Gate Control Error This error occurs when the list length (LLR) is 1 and the programmed Time Interval (TI) value after the optional Left Shifting is less than or equal to the Cycle Time (CTR). Constant Gate Control Error This error occurs when the list length (LLR) is 1 and the programmed Time Interval (TI) value after the optional Left Shifting is less than or equal to the Cycle Time (CTR). The above programming implies Gates are either always Closed or always Open based on the Gate Control values; the same effect can be achieved by other simpler (non TSN) programming mechanisms. Since the implementation does not support such a programming an error is reported. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Constant Gate Control Error not detected 1 - Constant Gate Control Error detected
3 HLBS	Head-Of-Line Blocking due to Scheduling Set when the frame is not able to win arbitration and get scheduled even after 4 iterations of the GCL. Head-Of-Line Blocking due to Scheduling Set when the frame is not able to win arbitration and get scheduled even after 4 iterations of the GCL. Indicates to software a potential programming error. The one hot encoded values of the Queue Numbers that are not able to make progress are indicated in the MTL_EST_SCH_ERROR register. Bit cleared when MTL_EST_SCH_ERROR register is all zeroes. 0 - Head-Of-Line Blocking due to Scheduling not detected 1 - Head-Of-Line Blocking due to Scheduling detected
2 HLBF	Head-Of-Line Blocking due to Frame Size Set when HOL Blocking is noticed on one or more Queues as a result of none of the Time Intervals of gate open in the GCL being greater than or equal to the duration needed for frame size (or frame fragment size when preemption is enabled) transmission. Head-Of-Line Blocking due to Frame Size Set when HOL Blocking is noticed on one or more Queues as a result of none of the Time Intervals of gate open in the GCL being greater than or equal to the duration needed for frame size (or frame fragment size when preemption is enabled) transmission. The one hot encoded Queue numbers that are experiencing HLBF are indicated in the MTL_EST_FRM_SIZE_ERROR register. Additionally, the first Queue number that experienced HLBF along with the frame size is captured in MTL_EST_FRM_SIZE_CAPTURE register. Bit cleared when MTL_EST_FRAME_SIZE_ ERROR register is all zeros.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Head-Of-Line Blocking due to Frame Size not detected 1 - Head-Of-Line Blocking due to Frame Size detected
1 BTRE	BTR Error When "1" indicates a programming error in the BTR of SWOL where the programmed value is less than current time. BTR Error When "1" indicates a programming error in the BTR of SWOL where the programmed value is less than current time. If the BTRL = "1111", SWOL is not updated and Software should reprogram the BTR to a value greater than current time and then set SSWL to reinitiate the switch to SWOL. Else if the value of BTRL < "1111", SWOL is updated and this field indicates the number of iterations (of + CycleTime) taken by hardware to update the BTR to a value greater than Current Time. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - BTR Error not detected 1 - BTR Error detected
0 SWLC	Switch to S/W owned list Complete When "1" indicates the hardware has successfully switched to the SWOL, and the SWOL bit has been updated to that effect. Switch to S/W owned list Complete When "1" indicates the hardware has successfully switched to the SWOL, and the SWOL bit has been updated to that effect. Cleared when the SSWL of MTL_EST_CONTROL register transitions from 0 to 1, or on a software write. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Switch to S/W owned list Complete not detected 1 - Switch to S/W owned list Complete detected

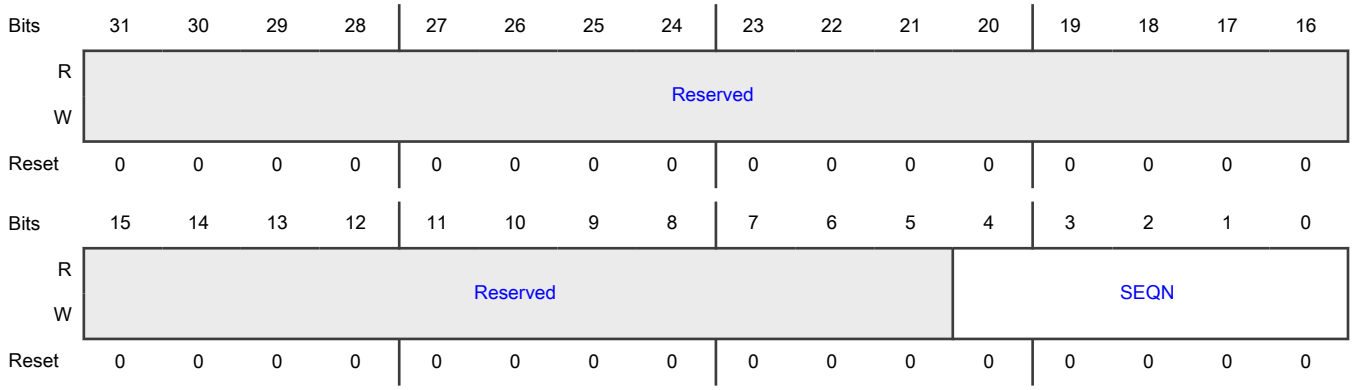
43.8.1.382 EST Scheduling Error (MTL_EST_SCH_ERROR)

This register provides the One Hot encoded Queue Numbers that are having the Scheduling related error (timeout).

Offset

Register	Offset
MTL_EST_SCH_ERROR	C60h

Diagram



Fields

Field	Description
31-5 —	Reserved
4-0 SEQN	Schedule Error Queue Number The One Hot Encoded Queue Numbers that have experienced error/timeout described in HLBS field of status register. Schedule Error Queue Number The One Hot Encoded Queue Numbers that have experienced error/timeout described in HLBS field of status register. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

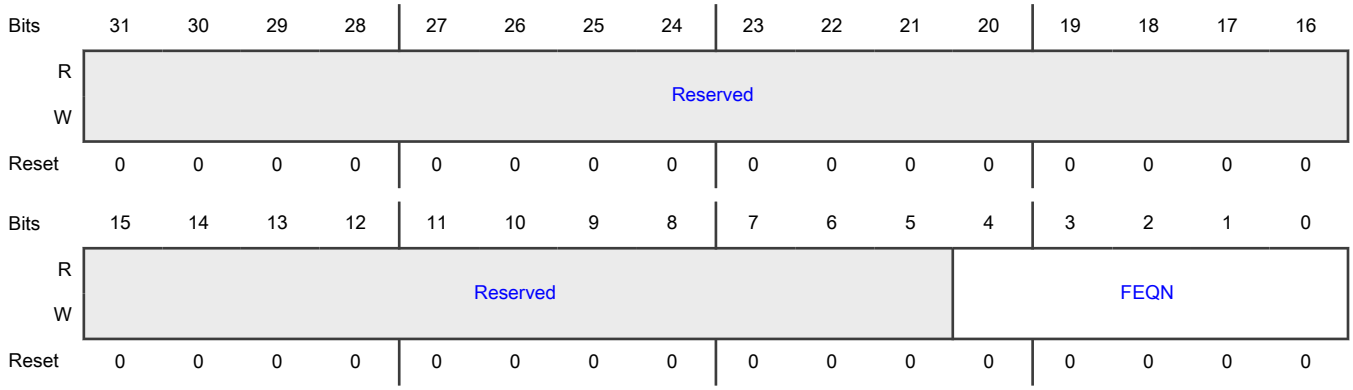
43.8.1.383 EST Frame Size Error (MTL_EST_FRM_SIZE_ERROR)

This register provides the One Hot encoded Queue Numbers that are having the Frame Size related error.

Offset

Register	Offset
MTL_EST_FRM_SIZE_ERROR	C64h

Diagram



Fields

Field	Description
31-5 —	Reserved
4-0 FEQN	Frame Size Error Queue Number The One Hot Encoded Queue Numbers that have experienced error described in HLBF field of status register. Frame Size Error Queue Number The One Hot Encoded Queue Numbers that have experienced error described in HLBF field of status register. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

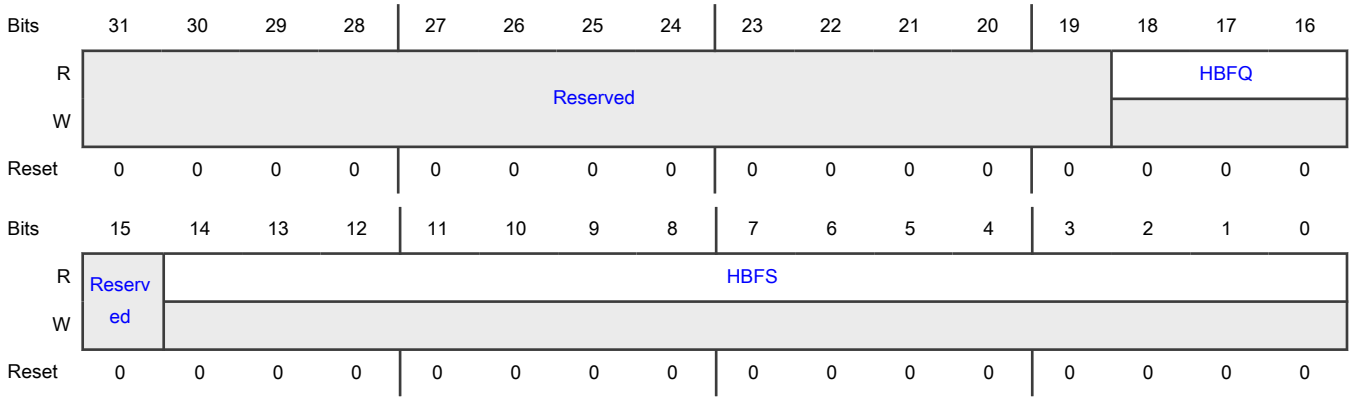
43.8.1.384 EST Frame Size Capture (MTL_EST_FRM_SIZE_CAPTURE)

This register captures the Frame Size and Queue Number of the first occurrence of the Frame Size related error. Up on clearing it captures the data of immediate next occurrence of a similar error.

Offset

Register	Offset
MTL_EST_FRM_SIZE_CAPTURE	C68h

Diagram



Fields

Field	Description
31-19 —	Reserved
18-16 HBFQ	Queue Number of HLBF Captures the binary value of the of the first Queue (number) experiencing HLBF error (see HLBF field of status register). Queue Number of HLBF Captures the binary value of the of the first Queue (number) experiencing HLBF error (see HLBF field of status register). Value once written is not altered by any subsequent queue errors of similar nature. Once cleared the queue number of the next occurring HLBF error is captured. Width is based on the number of Tx Queues configured; remaining bits are Read-Only. Cleared when MTL_EST_FRM_SIZE_ERROR register is all zeros.
15 —	Reserved
14-0 HBFS	Frame Size of HLBF Captures the Frame Size of the dropped frame related to queue number indicated in HBFQ field of this register. Frame Size of HLBF Captures the Frame Size of the dropped frame related to queue number indicated in HBFQ field of this register. Contents of this register should be considered invalid, if this field is zero. Cleared when MTL_EST_FRM_SIZE_ERROR register is all zeros.

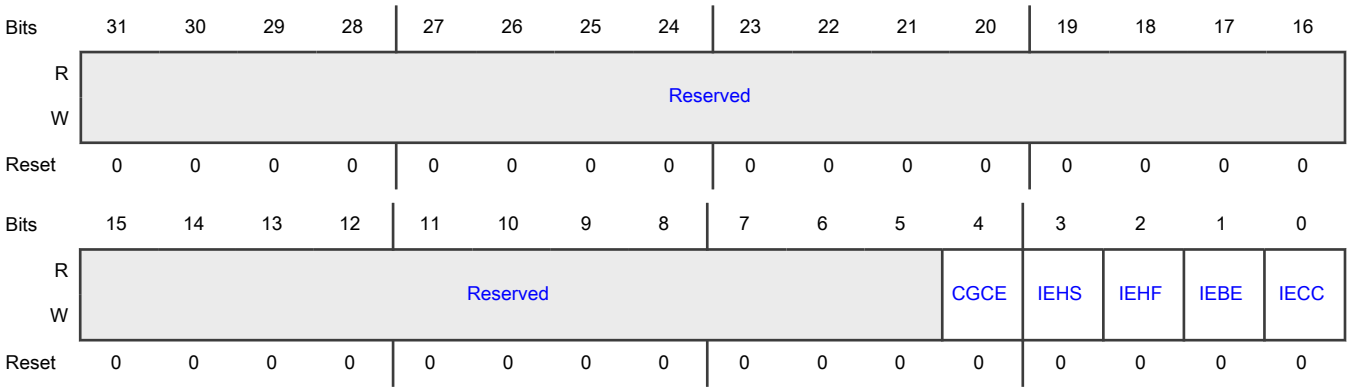
43.8.1.385 EST Interrupt Enable (MTL_EST_INTR_ENABLE)

This register implements the Interrupt Enable bits for the various events that generate an interrupt. Bit positions have a 1 to 1 correlation with the status bit positions in MTL_ETS_STATUS register.

Offset

Register	Offset
MTL_EST_INTR_ENABLE	C70h

Diagram



Fields

Field	Description
31-5 —	Reserved
4 CGCE	Interrupt Enable for CGCE When set, generates interrupt when the Constant Gate Control Error occurs and is indicated in the status. Interrupt Enable for CGCE When set, generates interrupt when the Constant Gate Control Error occurs and is indicated in the status. When reset this event does not generate an interrupt 0 - Interrupt for CGCE is disabled 1 - Interrupt for CGCE is enabled
3 IEHS	Interrupt Enable for HLBS When set, generates interrupt when the Head-of-Line Blocking due to Scheduling issue and is indicated in the status. Interrupt Enable for HLBS When set, generates interrupt when the Head-of-Line Blocking due to Scheduling issue and is indicated in the status. When reset this event does not generate an interrupt. 0 - Interrupt for HLBS is disabled 1 - Interrupt for HLBS is enabled
2 IEHF	Interrupt Enable for HLBF When set, generates interrupt when the Head-of-Line Blocking due to Frame Size error occurs and is indicated in the status. Interrupt Enable for HLBF

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When set, generates interrupt when the Head-of-Line Blocking due to Frame Size error occurs and is indicated in the status. When reset this event does not generate an interrupt. 0 - Interrupt for HLBF is disabled 1 - Interrupt for HLBF is enabled
1 IEBE	Interrupt Enable for BTR Error When set, generates interrupt when the BTR Error occurs and is indicated in the status. Interrupt Enable for BTR Error When set, generates interrupt when the BTR Error occurs and is indicated in the status. When reset this event does not generate an interrupt. 0 - Interrupt for BTR Error is disabled 1 - Interrupt for BTR Error is enabled
0 IECC	Interrupt Enable for Switch List When set, generates interrupt when the configuration change is successful and the hardware has switched to the new list. Interrupt Enable for Switch List When set, generates interrupt when the configuration change is successful and the hardware has switched to the new list. When reset this event does not generate an interrupt. 0 - Interrupt for Switch List is disabled 1 - Interrupt for Switch List is enabled

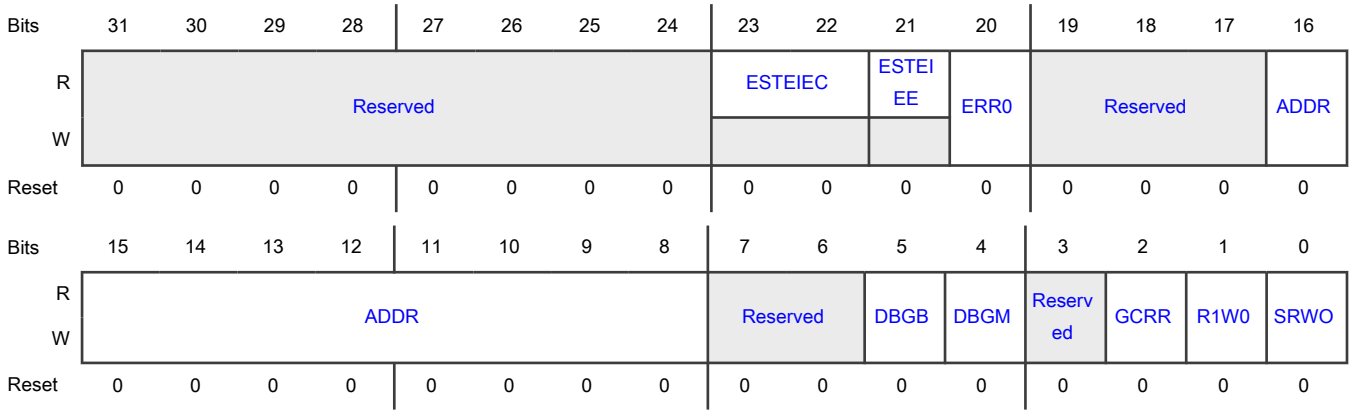
43.8.1.386 EST GCL Control (MTL_EST_GCL_CONTROL)

This register provides the control information for reading/writing to the Gate Control lists.

Offset

Register	Offset
MTL_EST_GCL_CONTROL	C80h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-22 ESTEIEC	ECC Inject Error Control for EST Memory When EIEE bit of this register is set, following are the errors inserted based on the value encoded in this field. ECC Inject Error Control for EST Memory When EIEE bit of this register is set, following are the errors inserted based on the value encoded in this field. This filed will be valid only if DWC_EQOS_ASP_ECC feature is selected during the configuration, else it will be reserved. 00 - Insert 1 bit error 01 - Insert 2 bit errors 10 - Insert 3 bit errors 11 - Insert 1 bit error in address field
21 ESTEIEE	EST ECC Inject Error Enable When set along with EEST bit of MTL_EST_CONTROL register, enables the ECC error injection feature. EST ECC Inject Error Enable When set along with EEST bit of MTL_EST_CONTROL register, enables the ECC error injection feature. When reset, disables the ECC error injection feature. 0 - EST ECC Inject Error is disabled 1 - EST ECC Inject Error is enabled
20 ERR0	When set indicates the last write operation was aborted as software writes to GCL and GCL registers is prohibited when SSWL bit of MTL_EST_CONTROL Register is set. When set indicates the last write operation was aborted as software writes to GCL and GCL registers is prohibited when SSWL bit of MTL_EST_CONTROL Register is set. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - ERR0 is disabled 1 - ERR1 is enabled
19-17 —	Reserved
16-8 ADDR	Gate Control List Address: (GCLA when GCRR is "0"). Gate Control List Address: (GCLA when GCRR is "0"). Provides the address (row number) of the Gate Control List at which the R/W operation has to be performed. By default the Gate Control List pointed by SWOL of MTL_EST_STATUS is selected for R/W, however if the DBGM bit of this register is set, a debug mode access is given to R/W from DBGB. The width of this field is dependent on DWC_EQOS_EST_DEP; unused bits should be treated as read only. Gate Control list Related Registers Address: (GCRA when GCRR is "1"). By default the GCL related register set pointed by SWOL of MTL_EST_STATUS is selected for R/W, however if the DBGM bit of this register is set, a debug mode access is given to R/W from DBGB. Lower 3 bits are only used in this mode, higher order bits are treated as dont cares. - 000: BTR Low (31:0) - 001: BTR High (63:31) - 010: CTR Low (31:0) - 011: CTR High (39:32) - 100: TER (31:0) - 101: LLR (n:0) (where n is (log{DWC_EQOS_EST_DEP} / log2)) - Others: Reserved
7-6 —	Reserved
5 DBGB	Debug Mode Bank Select When set to "0" indicates R/W in debug mode should be directed to Bank 0 (GCL0 and corresponding Time related registers). Debug Mode Bank Select When set to "0" indicates R/W in debug mode should be directed to Bank 0 (GCL0 and corresponding Time related registers). When set to "1" indicates R/W in debug mode should be directed to Bank 1 (GCL1 and corresponding Time related registers). This value is used when DBGM is set and overrides by value of SWOL which is normally used. 0 - R/W in debug mode should be directed to Bank 0 1 - R/W in debug mode should be directed to Bank 1
4 DBGM	Debug Mode When set to "1" indicates R/W in debug mode where the memory bank (for GCL and Time related registers) is explicitly provided by DBGB value, when set to "0" SWOL bit is used to determine which bank to use. 0 - Debug Mode is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Debug Mode is enabled
3 —	Reserved
2 GCRR	Gate Control Related Registers When set to "1" indicates the R/W access is for the GCL related registers (BTR, CTR, TER, LLR) whose address is provided by GCRA. Gate Control Related Registers When set to "1" indicates the R/W access is for the GCL related registers (BTR, CTR, TER, LLR) whose address is provided by GCRA. When "0" indicates R/W should be directed to GCL from the address provided by GCLA. 0 - Gate Control Related Registers are disabled 1 - Gate Control Related Registers are enabled
1 R1W0	Read '1', Write '0': When set to '1': Read Operation When set to '0': Write Operation. 0 - Write Operation 1 - Read Operation
0 SRWO	Start Read/Write Op When set indicates a Read/Write Op has started and is in progress. Start Read/Write Op When set indicates a Read/Write Op has started and is in progress. When reset by hardware indicates the R/W Op has completed or an error has occurred (when bit 20 is set) Reads: Data can be read from MTL_EST_GCL_DATA register after this bit is reset Writes: MTL_EST_GCL_DATA should be programmed with write data before setting SRWO. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Start Read/Write Op disabled 1 - Start Read/Write Op enabled

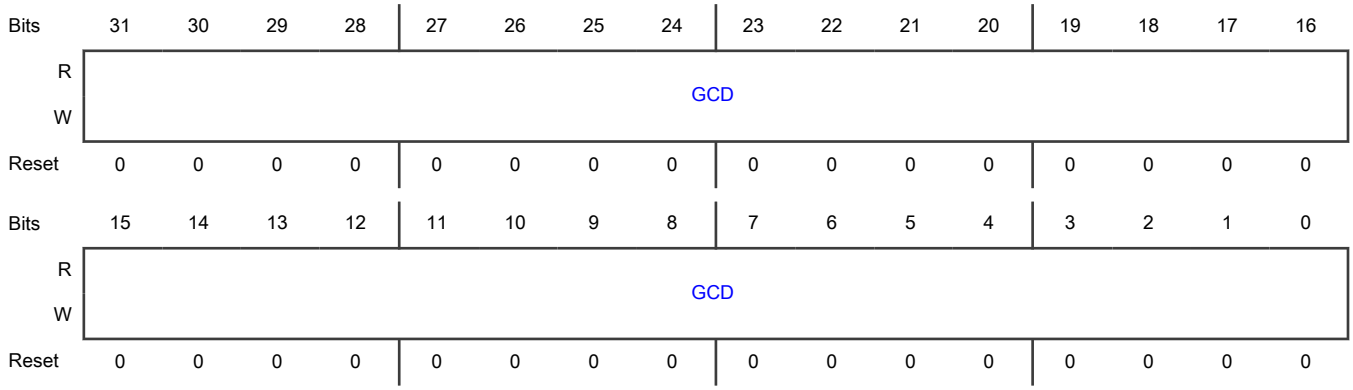
43.8.1.387 EST GCL Data (MTL_EST_GCL_DATA)

This register holds the read data or write data in case of reads and writes respectively.

Offset

Register	Offset
MTL_EST_GCL_DATA	C84h

Diagram



Fields

Field	Description
31-0 GCD	Gate Control Data The data corresponding to the address selected in the MTL_GCL_CONTROL register. Gate Control Data The data corresponding to the address selected in the MTL_GCL_CONTROL register. Used for both Read and Write operations.

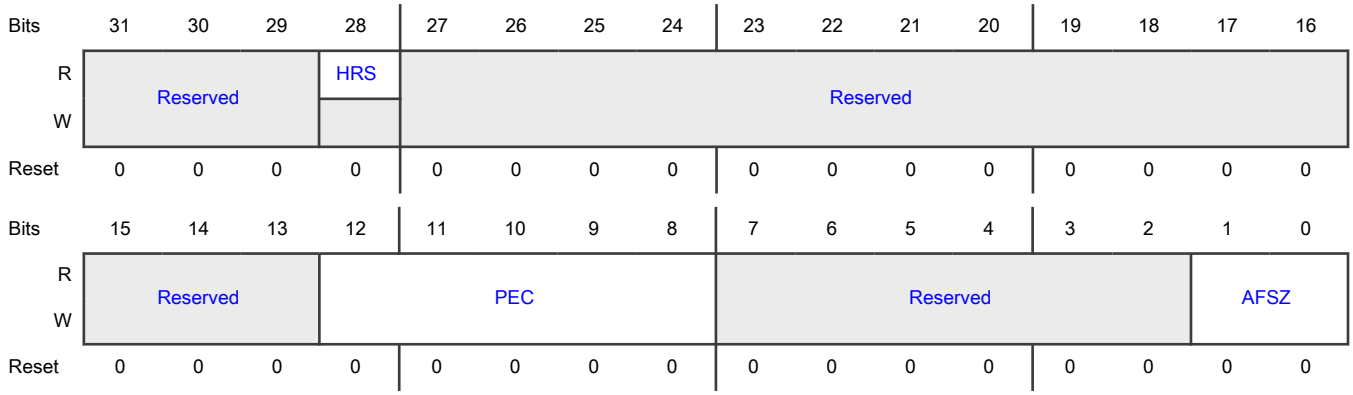
43.8.1.388 Frame Preemption Control and Status (MTL_FPE_CTRL_STS)

This register controls the operation of, and provides status for Frame Preemption (IEEE802.1Qbu/802.3br).

Offset

Register	Offset
MTL_FPE_CTRL_STS	C90h

Diagram



Fields

Field	Description
31-29 —	Reserved
28 HRS	Hold/Release Status - 1: Indicates a Set-and-Hold-MAC operation was last executed and the pMAC is in Hold State. Hold/Release Status - 1: Indicates a Set-and-Hold-MAC operation was last executed and the pMAC is in Hold State. - 0: Indicates a Set-and-Release-MAC operation was last executed and the pMAC is in Release State. 0 - Indicates a Set-and-Release-MAC operation was last executed and the pMAC is in Release State 1 - Indicates a Set-and-Hold-MAC operation was last executed and the pMAC is in Hold State
27-16 —	Reserved
15-13 —	Reserved
12-8 PEC	Preemption Classification When set indicates the corresponding Queue must be classified as preemptable, when '0' Queue is classified as express. Preemption Classification When set indicates the corresponding Queue must be classified as preemptable, when '0' Queue is classified as express. When both EST (Qbv) and Preemption are enabled, Queue-0 is always assumed to be preemptable. When EST (Qbv) is enabled Queues categorized as preemptable here are always assumed to be in "Open" state in the Gate Control List.
7-2 —	Reserved
1-0 AFSZ	Additional Fragment Size used to indicate, in units of 64 bytes, the minimum number of bytes over 64 bytes required in non-final fragments of preempted frames. Additional Fragment Size used to indicate, in units of 64 bytes, the minimum number of bytes over 64 bytes required in non-final fragments of preempted frames. The minimum non-final fragment size is (AFSZ +1) * 64 bytes

43.8.1.389 Frame Preemption Hold and Release Advance (MTL_FPE_ADVANCE)

This register holds the Hold and Release Advance time.

Offset

Register	Offset
MTL_FPE_ADVANCE	C94h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RADV															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HADV															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 RADV	Release Advance The maximum time in nanoseconds that can elapse between issuing a RELEASE to the MAC and the MAC being ready to resume transmission of preemptable frames, in the absence of there being any express frames available for transmission.
15-0 HADV	Hold Advance The maximum time in nanoseconds that can elapse between issuing a HOLD to the MAC and the MAC ceasing to transmit any preemptable frame that is in the process of transmission or any preemptable frames that are queued for transmission.

43.8.1.390 RXP Control Status (MTL_RXP_CONTROL_STATUS)

The RXP Control Status register establishes the operating mode of Rx Parser and provides some status.

Offset

Register	Offset
MTL_RXP_CONTROL_STATUS	CA0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXPI	Reserved							NPE							
W																
Reset	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								NVE							
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Fields

Field	Description
31 RXPI	RX Parser in Idle state This status bit is set to 1 when the Rx parser is in Idle State and waiting for a new packet for processing. RX Parser in Idle state This status bit is set to 1 when the Rx parser is in Idle State and waiting for a new packet for processing. This bit is used as a handshake with software when parser gets disables. After disabling, when bit is set then software can update the Rx parser instruction table. 0 - RX Parser not in Idle state 1 - RX Parser in Idle state
30-24 —	Reserved
23-16 NPE	Number of parsable entries in the Instruction table This control indicates the number of parsable entries in the Instruction Memory. Number of parsable entries in the Instruction table This control indicates the number of parsable entries in the Instruction Memory. This is used in Rx parser logic to detect programming Error. In case number of parsed entries for a packet is more than this entry then NPEOVIS bit in the MTL_RXP_INTERRUPT_CONTROL_STATUS register is set.
15-8 —	Reserved
7-0 NVE	Number of valid entries in the Instruction table This control indicates the number of valid entries in the Instruction Memory. Number of valid entries in the Instruction table This control indicates the number of valid entries in the Instruction Memory. This is used in Rx parser logic to detect any programming Error. In case while parsing Table address (memory address) found to be more than this entry then NVEOVIS bit in the MTL_RXP_INTERRUPT_CONTROL_STATUS register is set. Note: The minimum value of this should be 2.

43.8.1.391 RXP Interrupt Control Status (MTL_RXP_INTERRUPT_CONTROL_STATUS)

The RXP Interrupt Control Status register provides enable control for the interrupts and provides interrupt status.

Offset

Register	Offset
MTL_RXP_INTERRUPT_CONTROL_STATUS	CA4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved												PDRFIE	FOOVIE	NPEOVIE	NVEOVIE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												PDRFIS	FOOVIS	NPEOVIS	NVEOVIS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-20 —	Reserved
19 PDRFIE	Packet Drop due to RF Interrupt Enable When this bit is set, the PDRFIS interrupt is enabled. Packet Drop due to RF Interrupt Enable When this bit is set, the PDRFIS interrupt is enabled. When this bit is reset, the PDRFIS interrupt is disabled. 0 - Packet Drop due to RF Interrupt is disabled 1 - Packet Drop due to RF Interrupt is enabled
18 FOOVIE	Frame Offset Overflow Interrupt Enable When this bit is set, the FOOVIS interrupt is enabled. Frame Offset Overflow Interrupt Enable When this bit is set, the FOOVIS interrupt is enabled. When this bit is reset, the FOOVIS interrupt is disabled. 0 - Frame Offset Overflow Interrupt is disabled 1 - Frame Offset Overflow Interrupt is enabled
17 NPEOVIE	Number of Parsable Entries Overflow Interrupt Enable When this bit is set, the NPEOVIS interrupt is enabled. Number of Parsable Entries Overflow Interrupt Enable When this bit is set, the NPEOVIS interrupt is enabled. When this bit is reset, the NPEOVIS interrupt is disabled. 0 - Number of Parsable Entries Overflow Interrupt is disabled 1 - Number of Parsable Entries Overflow Interrupt is enabled
16 NVEOVIE	Number of Valid Entries Overflow Interrupt Enable When this bit is set, the NVEOVIS interrupt is enabled. Number of Valid Entries Overflow Interrupt Enable When this bit is set, the NVEOVIS interrupt is enabled. When this bit is reset, the NVEOVIS interrupt is disabled.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Number of Valid Entries Overflow Interrupt is disabled 1 - Number of Valid Entries Overflow Interrupt is enabled
15-4 —	Reserved
3 PDRFIS	Packet Dropped due to RF Interrupt Status If the Rx Parser result says to drop the packet by setting RF=1 in the instruction memory, then this bit is set to 1. Packet Dropped due to RF Interrupt Status If the Rx Parser result says to drop the packet by setting RF=1 in the instruction memory, then this bit is set to 1. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Packet Dropped due to RF Interrupt Status not detected 1 - Packet Dropped due to RF Interrupt Status detected
2 FOOVIS	Frame Offset Overflow Interrupt Status While parsing if the Instruction table entry's 'Frame Offset' found to be more than EOF offset, then then this bit is set. Frame Offset Overflow Interrupt Status While parsing if the Instruction table entry's 'Frame Offset' found to be more than EOF offset, then then this bit is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Frame Offset Overflow Interrupt Status not detected 1 - Frame Offset Overflow Interrupt Status detected
1 NPEOVIS	Number of Parsable Entries Overflow Interrupt Status While parsing a packet if the number of parsed entries found to be more than NPE[] (Number of Parseable Entries in MTL_RXP_CONTROL register),then this bit is set to 1. Number of Parsable Entries Overflow Interrupt Status While parsing a packet if the number of parsed entries found to be more than NPE[] (Number of Parseable Entries in MTL_RXP_CONTROL register),then this bit is set to 1. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Number of Parsable Entries Overflow Interrupt Status not detected 1 - Number of Parsable Entries Overflow Interrupt Status detected
0 NVEOVIS	Number of Valid Entries Overflow Interrupt Status While parsing if the Instruction address found to be more than NVE (Number of Valid Entries in MTL_RXP_CONTROL register), then this bit is set to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Number of Valid Entries Overflow Interrupt Status While parsing if the Instruction address found to be more than NVE (Number of Valid Entries in MTL_RXP_CONTROL register), then this bit is set to 1. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Number of Valid Entries Overflow Interrupt Status not detected 1 - Number of Valid Entries Overflow Interrupt Status detected

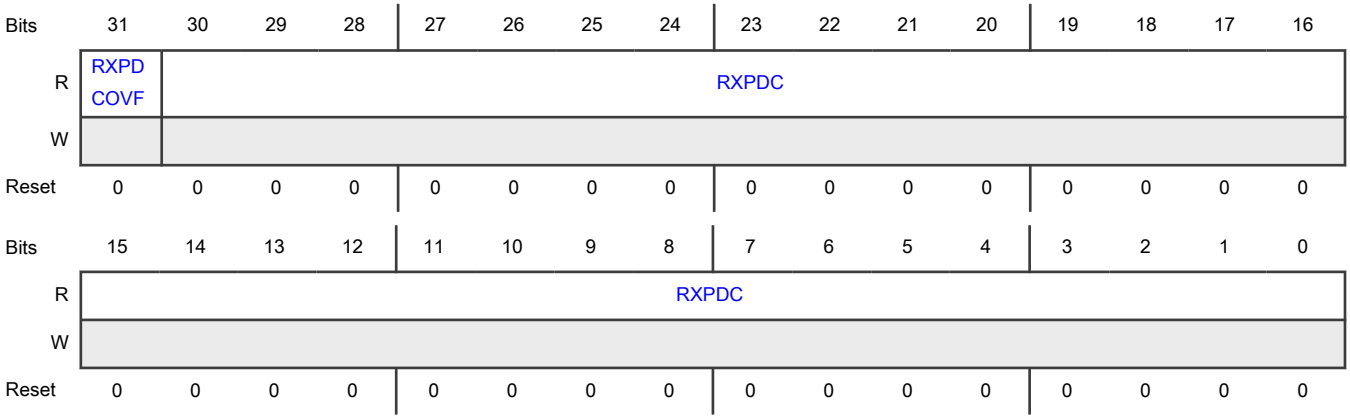
43.8.1.392 RXP Drop Count (MTL_RXP_DROP_CNT)

The MTL_RXP_DROP_CNT register provides the drop count of Rx Parser initiated drops.

Offset

Register	Offset
MTL_RXP_DROP_CNT	CA8h

Diagram



Fields

Field	Description
31 RXPDCOVF	Rx Parser Drop Counter Overflow Bit When set, this bit indicates that the MTL_RXP_DROP_CNT (RXPDC) Counter field crossed the maximum limit. Rx Parser Drop Counter Overflow Bit When set, this bit indicates that the MTL_RXP_DROP_CNT (RXPDC) Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Drop count overflow not occurred

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Rx Parser Drop count overflow occurred
30-0 RXPDC	Rx Parser Drop count This 31-bit counter is implemented whenever a Rx Parser Drops a packet due to RF =1. Rx Parser Drop count This 31-bit counter is implemented whenever a Rx Parser Drops a packet due to RF =1. The counter is cleared when the register is read.

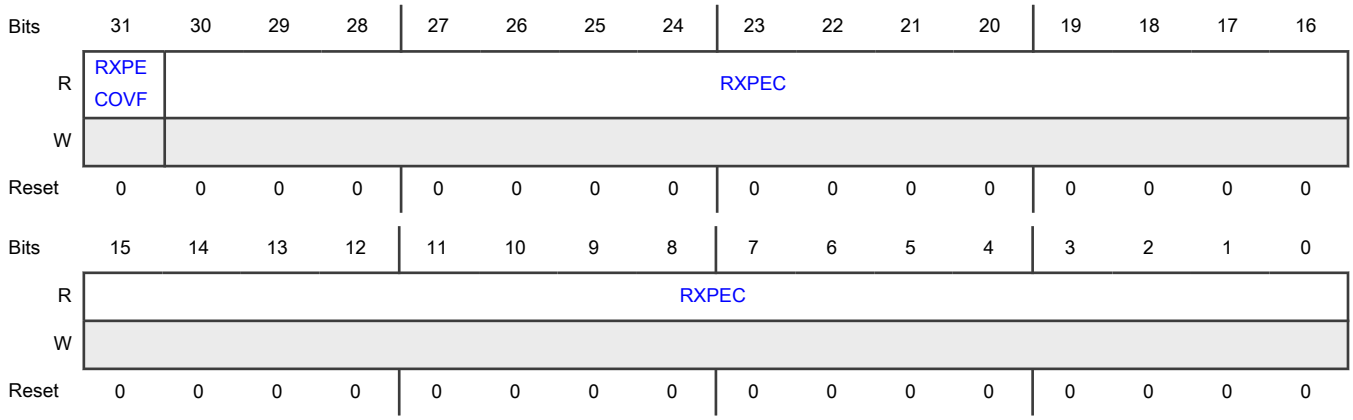
43.8.1.393 RXP Error Count (MTL_RXP_ERROR_CNT)

The RXP_ERROR_CNT register provides the Rx Parser related error occurrence count.

Offset

Register	Offset
MTL_RXP_ERROR_CNT	CACh

Diagram



Fields

Field	Description
31 RXPECOVF	Rx Parser Error Counter Overflow Bit When set, this bit indicates that the MTL_RXP_ERROR_CNT (RXPEC) Counter field crossed the maximum limit. Rx Parser Error Counter Overflow Bit When set, this bit indicates that the MTL_RXP_ERROR_CNT (RXPEC) Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Error count overflow not occurred

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Rx Parser Error count overflow occurred
30-0 RXPEC	Rx Parser Error count This 31-bit counter is implemented whenever a Rx Parser encounters following Error scenarios - Entry address >= NVE[] - Number Parsed Entries >= NPE[] - Entry address > EOF data entry address The counter is cleared when the register is read.

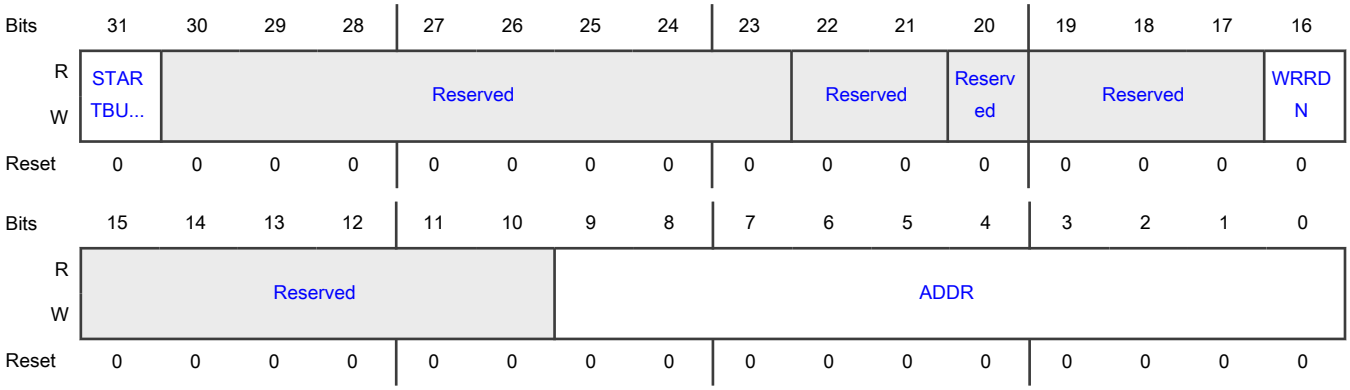
43.8.1.394 RXP Indirect Access Control and Status (MTL_RXP_INDIRECT_ACC_CONTROL_STATUS)

The RXP_INDIRECT_ACC_CONTROL_STATUS register provides the Indirect Access control and status for Rx Parser memory.

Offset

Register	Offset
MTL_RXP_INDIRECT_ACC_CONTROL_STATUS	CB0h

Diagram



Fields

Field	Description
31 STARTBUSY	FRP Instruction Table Access Busy When this bit is set to 1 by the software then it indicates to start the Read/Write operation from/to the Rx Parser Memory. FRP Instruction Table Access Busy When this bit is set to 1 by the software then it indicates to start the Read/Write operation from/to the Rx Parser Memory. Software should read this bit as 0 before issuing read or write request to access the Parser Memory Instructions.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This bit when set to 1 indicates that hardware is busy until its gets cleared by hardware and software should not issue any read or write operation. 0 - hardware not busy 1 - hardware is busy (Read/Write operation from/to the Rx Parser Memory)
30-23 —	Reserved
22-21 —	Reserved
20 —	Reserved
19-17 —	Reserved
16 WRRDN	Read Write Control When this bit is set to 1 indicates the write operation to the Rx Parser Memory. Read Write Control When this bit is set to 1 indicates the write operation to the Rx Parser Memory. When this bit is set to 0 indicates the read operation to the Rx Parser Memory. 0 - Read operation to the Rx Parser Memory 1 - Write operation to the Rx Parser Memory
15-10 —	Reserved
9-0 ADDR	FRP Instruction Table Offset Address This field indicates the ADDR of the 32-bit entry in Rx parser instruction table. FRP Instruction Table Offset Address This field indicates the ADDR of the 32-bit entry in Rx parser instruction table. Each entry has 128-bit (4x32-bit words). The X depends on the configurable DWC_EQOS_FRP_ENTRIES If DWC_EQOS_FRP_ENTRIES == 256 , then X = 9 If DWC_EQOS_FRP_ENTRIES = 128 , then X = 8 IF DWC_EQOS_FRP_ENTRIES = 64, then X = 7 This is can be written by the software before issuing any Read/Write command. The hardware auto-increments this field after the read-write operation gets completed.

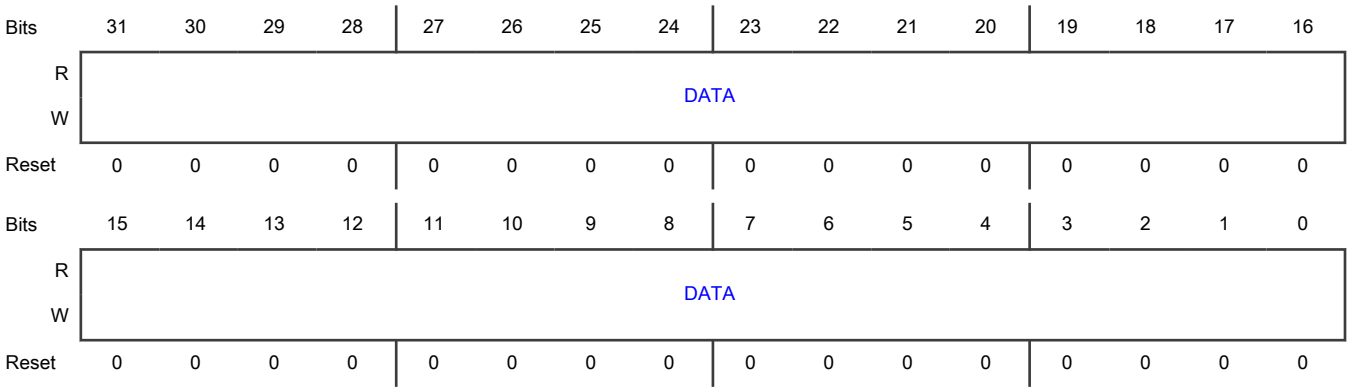
43.8.1.395 RXP Indirect Access Data (MTL_RXP_INDIRECT_ACC_DATA)

The RXP_INDIRECT_ACC_DATA registers holds the data associated to Indirect Access to Rx Parser memory.

Offset

Register	Offset
MTL_RXP_INDIRECT_ACC_DATA	CB4h

Diagram



Fields

Field	Description
31-0 DATA	FRP Instruction Table Write/Read Data Software should write this register before issuing any write command. FRP Instruction Table Write/Read Data Software should write this register before issuing any write command. The hardware provides the read data from the Rx Parser Memory for read operation when STARTBUSY =0 after read command.

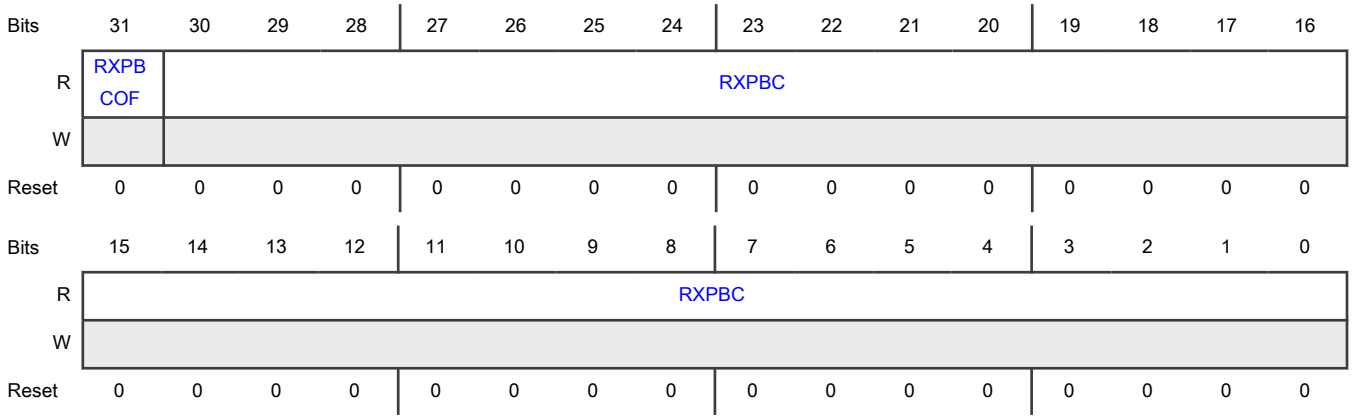
43.8.1.396 MTL_RXP_BYPASS_CNT (MTL_RXP_BYPASS_CNT)

The MTL_RXP_BYPASS_CNT register provides the bypass count of Rx Parser.

Offset

Register	Offset
MTL_RXP_BYPASS_CNT	CB8h

Diagram



Fields

Field	Description
31 RXPBCOF	Rx Parser bypass Counter Overflow Bit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Indicates that MTL_RXP_BYPASS_CNT[RXPBC] counter field has not crossed the maximum limit 1 - Indicates that MTL_RXP_BYPASS_CNT[RXPBC] counter field has crossed the maximum limit
30-0 RXPBC	Rx Parser Bypass Count This 31-bit counter is implemented when a Rx Parser bypass a packet due to AF=1 and RF=1. The counter is cleared when the register is read.

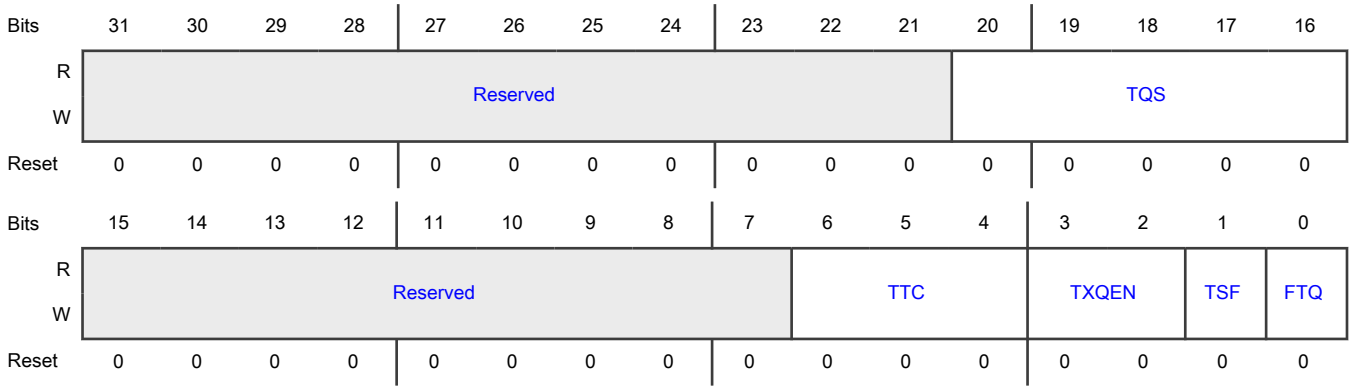
43.8.1.397 Queue 0 Transmit Operation Mode (MTL_TXQ0_OPERATION_MODE)

The Queue 0 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

Offset

Register	Offset
MTL_TXQ0_OPERATION_MODE	D00h

Diagram



Fields

Field	Description
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000 - 32 001 - 64 010 - 96 011 - 128 100 - 192

Table continues on the next page...

Table continued from the previous page...

Field	Description
	101 - 256 110 - 384 111 - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Reserved - 2'b10: Enabled - 2'b11: Reserved This field is Read Only in Single Queue configurations and Read Write in Multiple Queue configurations. Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00 - Not enabled 01 - Enable in AV mode (Reserved in non-AV) 10 - Enabled 11 - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0 - Transmit Store and Forward is disabled 1 - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ0_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Flush Transmit Queue is disabled 1 - Flush Transmit Queue is enabled

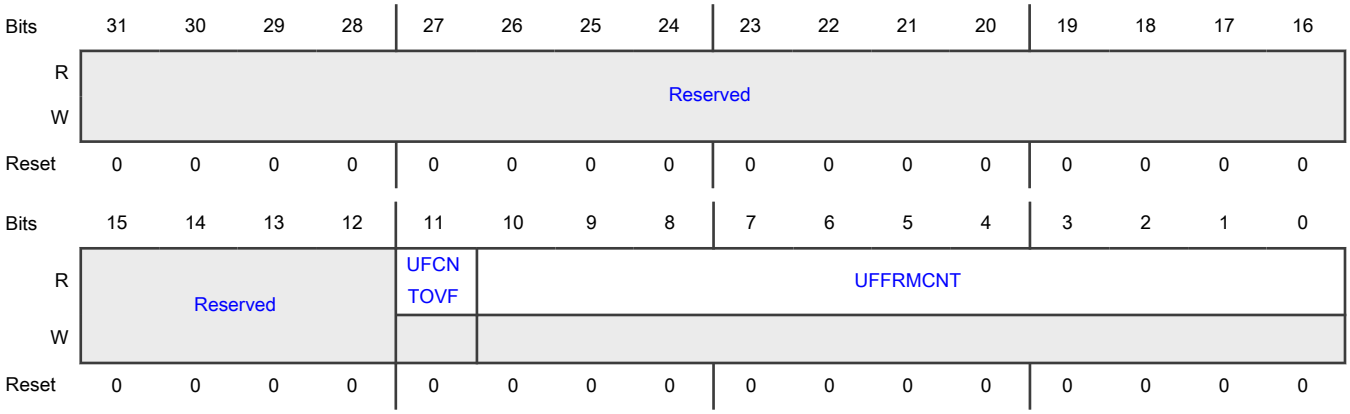
43.8.1.398 Queue 0 Underflow Counter (MTL_TXQ0_UNDERFLOW)

The Queue 0 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

Offset

Register	Offset
MTL_TXQ0_UNDERFLOW	D04h

Diagram



Fields

Field	Description
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Overflow not detected for Underflow Packet Counter 1 - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

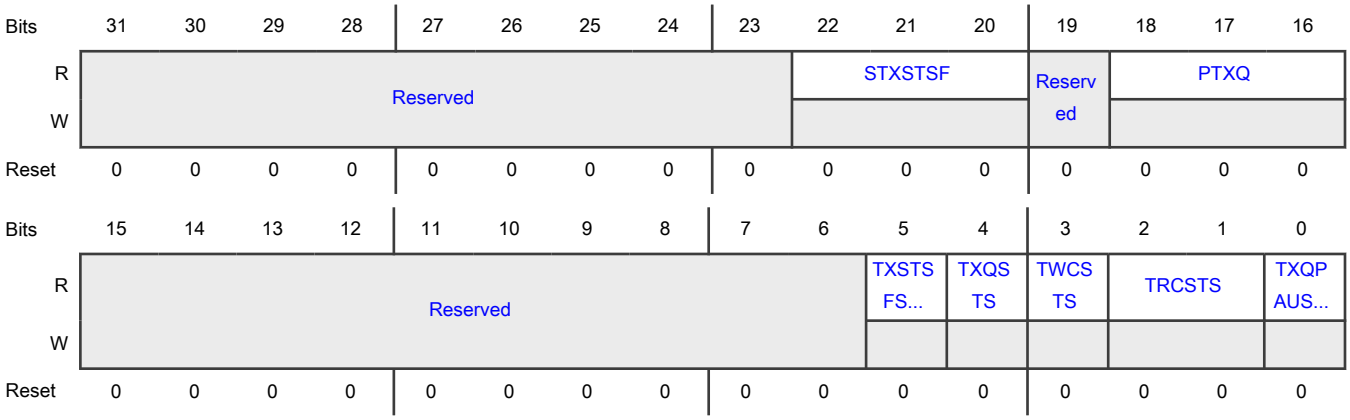
43.8.1.399 Queue 0 Transmit Debug (MTL_TXQ0_DEBUG)

The Queue 0 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

Offset

Register	Offset
MTL_TXQ0_DEBUG	D08h

Diagram



Fields

Field	Description
31-23 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0 - MTL Tx Status FIFO Full status is not detected 1 - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0 - MTL Tx Queue Not Empty status is not detected 1 - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0 - MTL Tx Queue Write Controller status is not detected 1 - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00 - Idle state 01 - Read state (transferring data to the MAC transmitter) 10 - Waiting for pending Tx Status from the MAC transmitter

Table continues on the next page...

Table continued from the previous page...

Field	Description
	11 - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0 - Transmit Queue in Pause status is not detected 1 - Transmit Queue in Pause status is detected

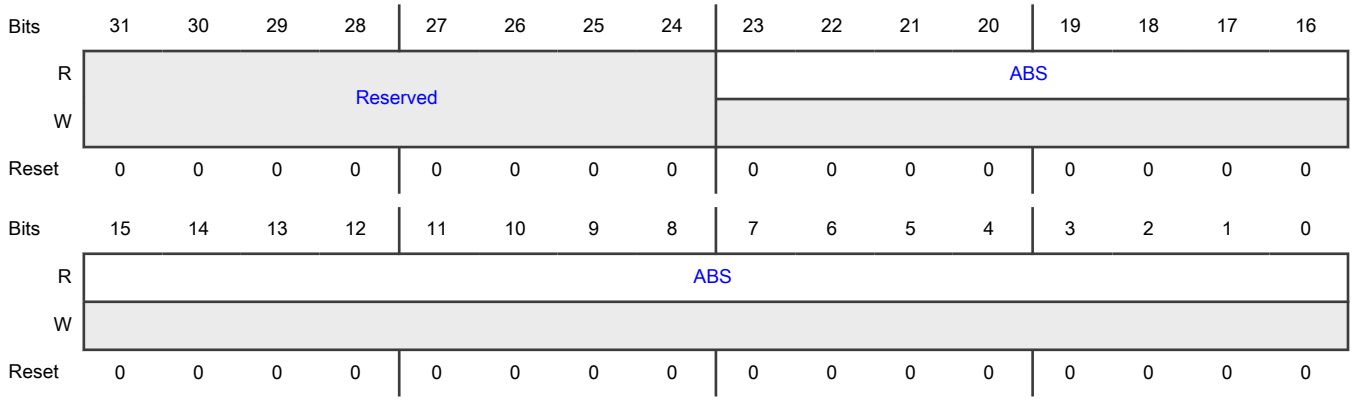
43.8.1.400 Queue 0 ETS Status (MTL_TXQ0_ETS_STATUS)

The Queue 0 ETS Status register provides the average traffic transmitted in Queue 0.

Offset

Register	Offset
MTL_TXQ0_ETS_STATUS	D14h

Diagram



Fields

Field	Description
31-24	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. When the DCB operation is enabled for Queue 0, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

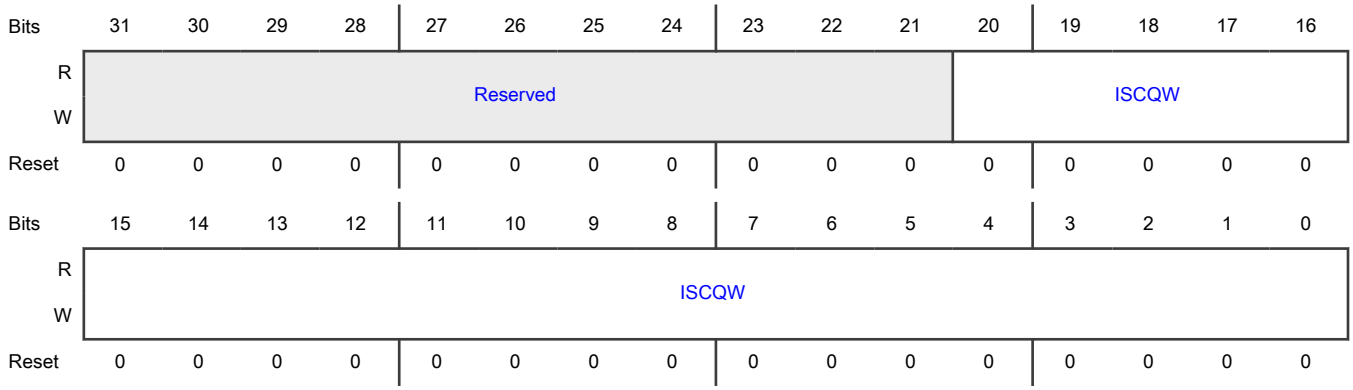
43.8.1.401 Queue 0 Quantum or Weights (MTL_TXQ0_QUANTUM_WEIGHT)

The Queue 0 Quantum or Weights register contains the quantum value for Deficit Weighted Round Robin (DWRR), weights for the Weighted Round Robin (WRR), and Weighted Fair Queuing (WFQ) for Queue 0.

Offset

Register	Offset
MTL_TXQ0_QUANTUM_WEIGHT	D18h

Diagram



Fields

Field	Description
31-21 —	Reserved
20-0 ISCQW	Quantum or Weights When the DCB operation is enabled with DWRR algorithm for Queue 0 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. Quantum or Weights

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When the DCB operation is enabled with DWRR algorithm for Queue 0 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. When DCB operation is enabled with WFQ algorithm for Queue 0 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. The higher the programmed weights lesser the bandwidth allocated for the particular Transmit Queue. This is because the weights are used to compute the packet finish time (weights*packet_size). Lesser the finish time, higher the probability of the packet getting scheduled first and using more bandwidth. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 0 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero.

43.8.1.402 Queue 0 Interrupt Control Status (MTL_Q0_INTERRUPT_CONTROL_STATUS)

This register contains the interrupt enable and status bits for the queue 0 interrupts.

Offset

Register	Offset
MTL_Q0_INTERRUPT_CONTROL_STATUS	D2Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved							RXOIE	Reserved							RXOVFIS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						ABPSIE	TXUIE	Reserved						ABPSIS	TXUNFIS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-25 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0 - Receive Queue Overflow Interrupt is disabled 1 - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Queue Overflow Interrupt Status not detected 1 - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9 ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0 - Average Bits Per Slot Interrupt is disabled 1 - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0 - Transmit Queue Underflow Interrupt Status is disabled 1 - Transmit Queue Underflow Interrupt Status is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Average Bits Per Slot Interrupt Status not detected 1 - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Queue Underflow Interrupt Status not detected 1 - Transmit Queue Underflow Interrupt Status detected

43.8.1.403 Queue 0 Receive Operation Mode (MTL_RXQ0_OPERATION_MODE)

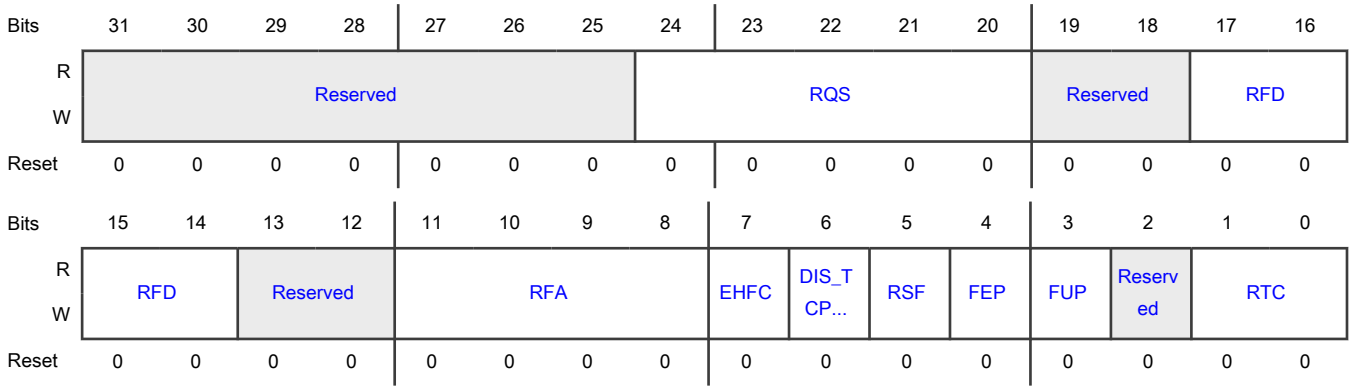
The Queue 0 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

Offset

Register	Offset
MTL_RXQ0_OPERATION_MODE	D30h

Diagram



Fields

Field	Description
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ...

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0 - Hardware Flow Control is disabled 1 - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0 - Dropping of TCP/IP Checksum Error Packets is enabled 1 - Dropping of TCP/IP Checksum Error Packets is disabled
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0 - Receive Queue Store and Forward is disabled 1 - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0 - Forward Error Packets is disabled 1 - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0 - Forward Undersized Good Packets is disabled 1 - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	00 - 64 01 - 32 10 - 96 11 - 128

43.8.1.404 Queue 0 Missed Packet and Overflow Counter (MTL_RXQ0_MISSED_PACKET_OVERFLOW_CNT)

The Queue 0 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

Offset

Register	Offset
MTL_RXQ0_MISSED_PACKET_OVERFLOW_CNT	D34h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				MISCNTO...	MISPKTCNT										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				OVFCNTO...	OVFPKTCNT										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Missed Packet Counter overflow not detected 1 - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow Counter overflow not detected 1 - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

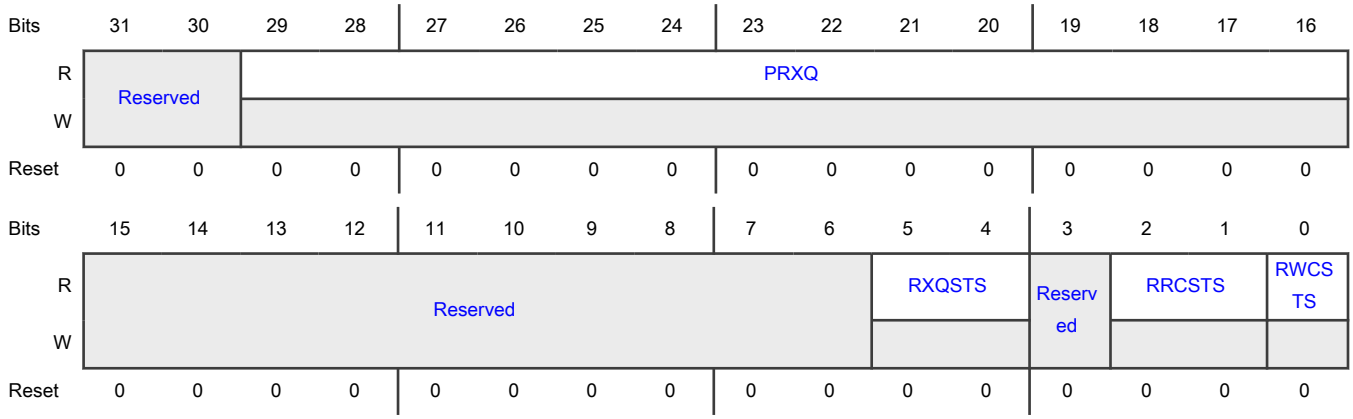
43.8.1.405 Queue 0 Receive Debug (MTL_RXQ0_DEBUG)

The Queue 0 Receive Debug register gives the debug status of various blocks related to the Receive queue.

Offset

Register	Offset
MTL_RXQ0_DEBUG	D38h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00 - Rx Queue empty 01 - Rx Queue fill-level below flow-control deactivate threshold 10 - Rx Queue fill-level above flow-control activate threshold 11 - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00 - Idle state 01 - Reading packet data 10 - Reading packet status (or timestamp) 11 - Flushing the packet data and status
0	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue.

Table continues on the next page...

Table continued from the previous page...

Field	Description
RWCSTS	0 - MTL Rx Queue Write Controller Active Status not detected 1 - MTL Rx Queue Write Controller Active Status detected

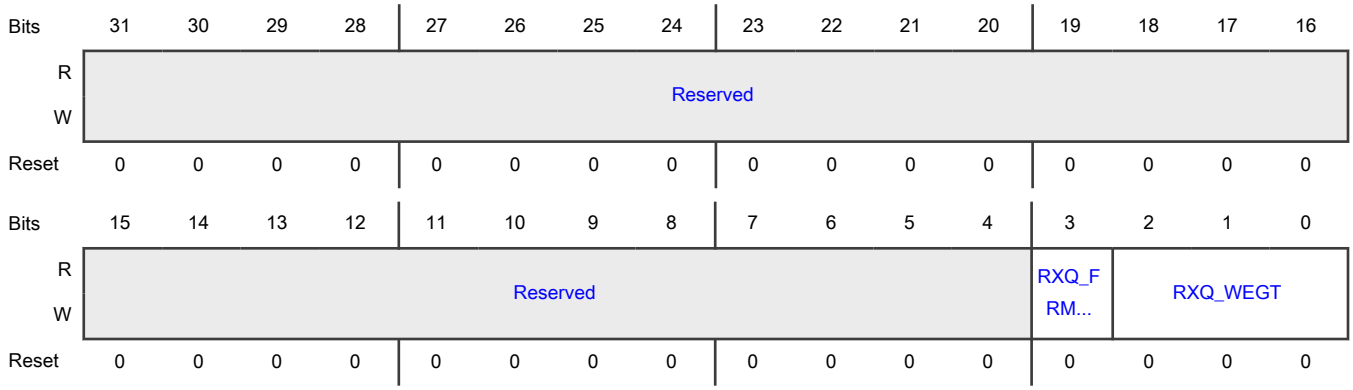
43.8.1.406 Queue 0 Receive Control (MTL_RXQ0_CONTROL)

The Queue 0 Receive Control register controls the receive arbitration and passing of received packets to the application.

Offset

Register	Offset
MTL_RXQ0_CONTROL	D3Ch

Diagram



Fields

Field	Description
31-4 —	Reserved
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[])

Table continues on the next page...

Table continued from the previous page...

Field	Description
	or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0 - Receive Queue Packet Arbitration is disabled 1 - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

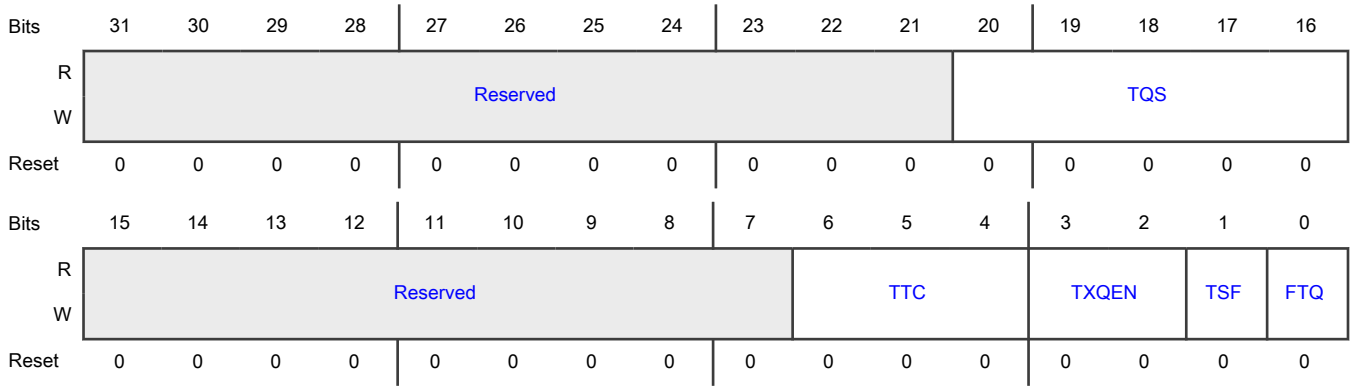
43.8.1.407 Queue 1 Transmit Operation Mode (MTL_TXQ1_OPERATION_MODE)

The Queue 1 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

Offset

Register	Offset
MTL_TXQ1_OPERATION_MODE	D40h

Diagram



Fields

Field	Description
31-21	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000 - 32 001 - 64 010 - 96 011 - 128 100 - 192 101 - 256 110 - 384 111 - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Enable in AV mode - 2'b10: Enabled - 2'b11: Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00 - Not enabled 01 - Enable in AV mode (Reserved in non-AV) 10 - Enabled 11 - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0 - Transmit Store and Forward is disabled 1 - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ1_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flush Transmit Queue is disabled 1 - Flush Transmit Queue is enabled

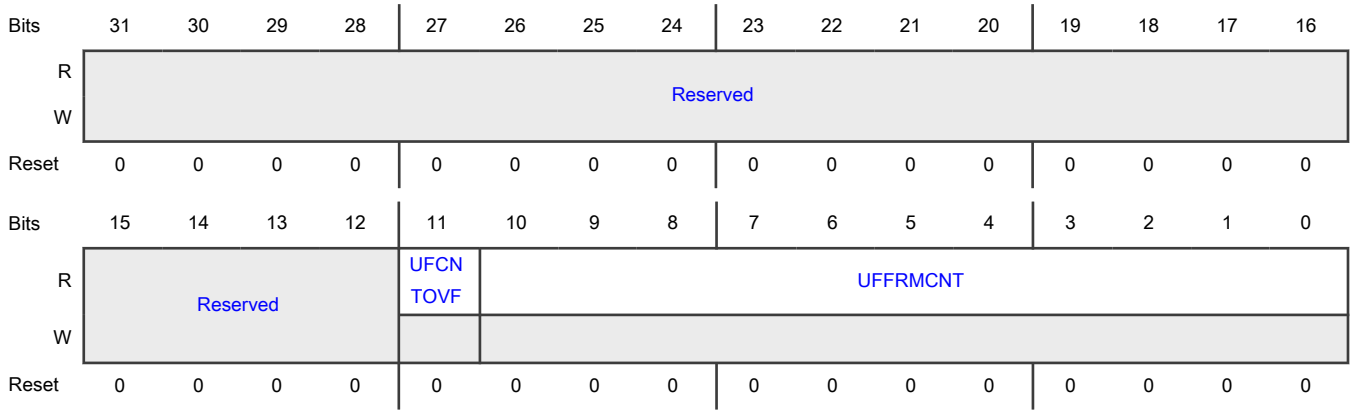
43.8.1.408 Queue 1 Underflow Counter (MTL_TXQ1_UNDERFLOW)

The Queue 1 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

Offset

Register	Offset
MTL_TXQ1_UNDERFLOW	D44h

Diagram



Fields

Field	Description
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow not detected for Underflow Packet Counter 1 - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

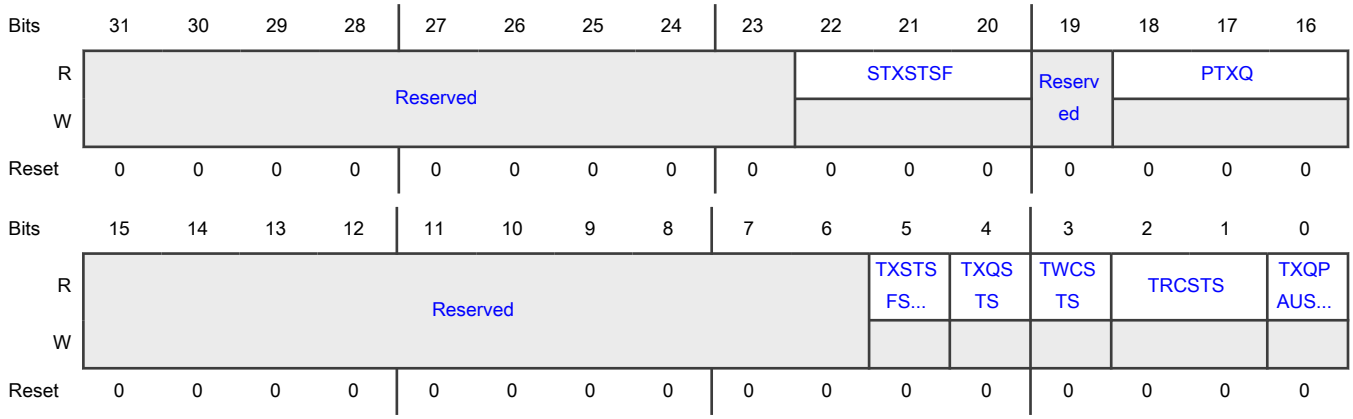
43.8.1.409 Queue 1 Transmit Debug (MTL_TXQ1_DEBUG)

The Queue 1 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

Offset

Register	Offset
MTL_TXQ1_DEBUG	D48h

Diagram



Fields

Field	Description
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0 - MTL Tx Status FIFO Full status is not detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0 - MTL Tx Queue Not Empty status is not detected 1 - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0 - MTL Tx Queue Write Controller status is not detected 1 - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00 - Idle state 01 - Read state (transferring data to the MAC transmitter) 10 - Waiting for pending Tx Status from the MAC transmitter 11 - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0 - Transmit Queue in Pause status is not detected 1 - Transmit Queue in Pause status is detected

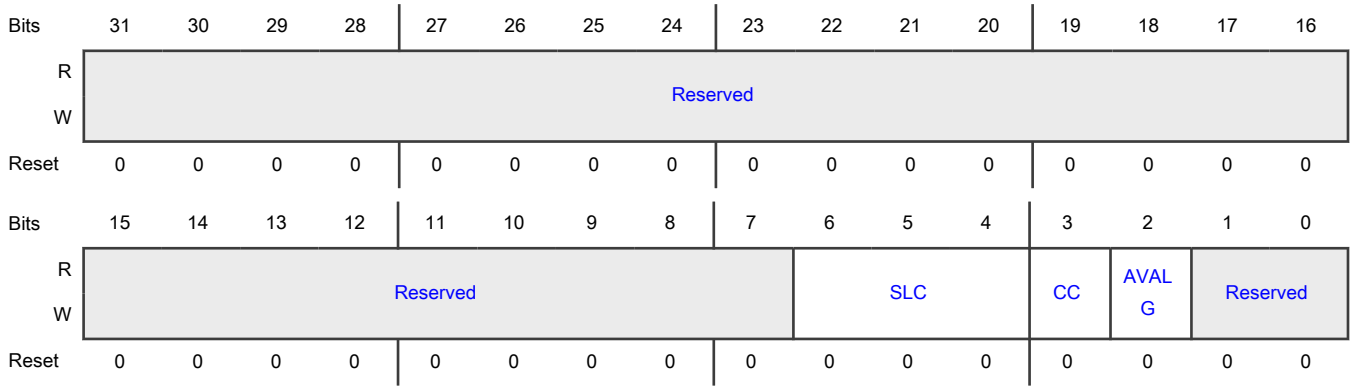
43.8.1.410 Queue 1 ETS Control (MTL_TXQ1_ETS_CONTROL)

The Queue 1 ETS Control register controls the enhanced transmission selection operation.

Offset

Register	Offset
MTL_TXQ1_ETS_CONTROL	D50h

Diagram



Fields

Field	Description
31-7 —	Reserved
6-4 SLC	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH1_Slot_Interval register) over which the average transmitted bits per slot, provided in the MTL_TXQ1_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 000 - 1 slot 001 - 2 slots 010 - 4 slots 011 - 8 slots 100 - 16 slots 101 - Reserved
3 CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. The credit accumulates even when there is no packet waiting in Channel 1 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 1. When there is no packet waiting in Channel 1 and other channel is transmitting, no credit is accumulated. 0 - Credit Control is disabled 1 - Credit Control is enabled
2 AVALG	AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic. When reset, strict priority is selected. 0 - CBS Algorithm is disabled 1 - CBS Algorithm is enabled
1-0 —	Reserved

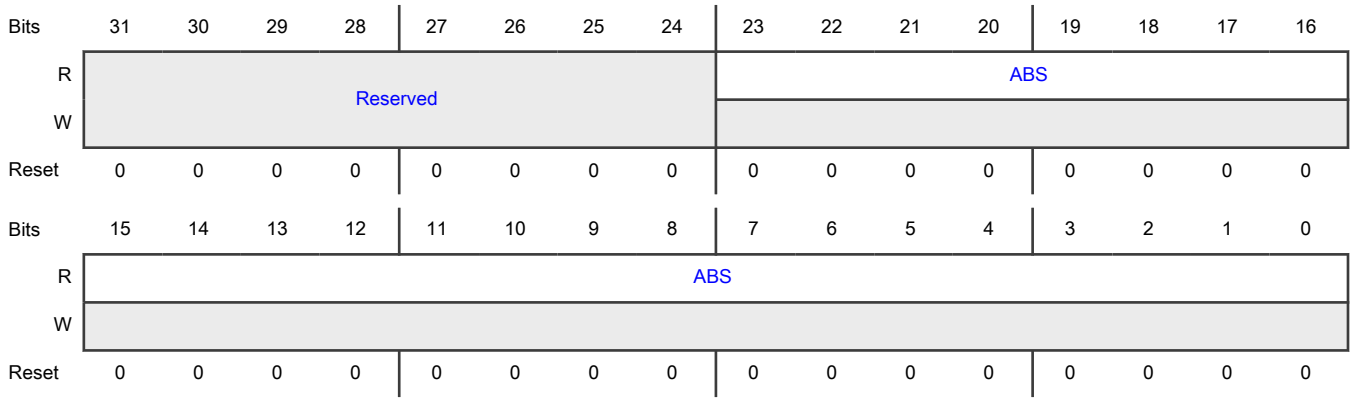
43.8.1.411 Queue 1 ETS Status (MTL_TXQ1_ETS_STATUS)

The Queue 1 ETS Status register provides the average traffic transmitted in Queue 1.

Offset

Register	Offset
MTL_TXQ1_ETS_STATUS	D54h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0	Average Bits per Slot This field contains the average transmitted bits per slot.

Table continues on the next page...

Table continued from the previous page...

Field	Description
ABS	Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQ1_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

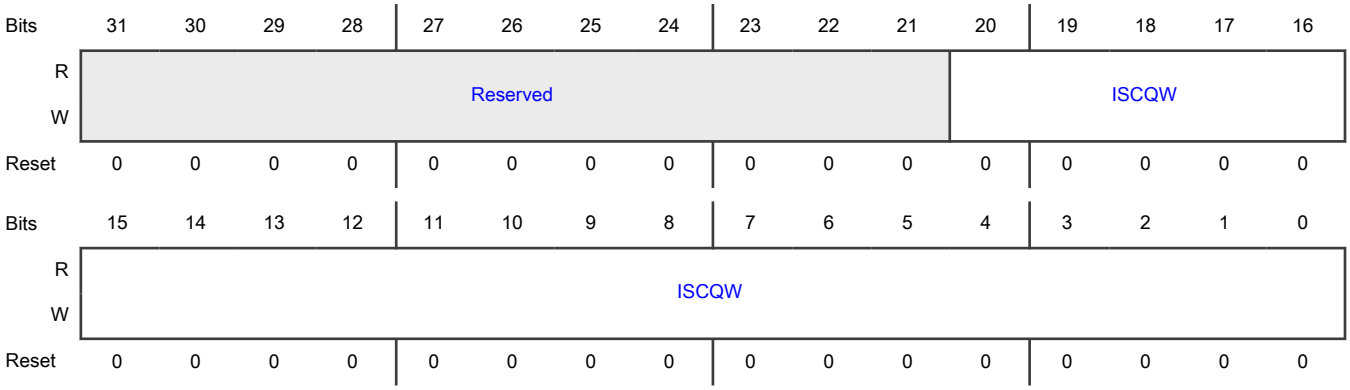
43.8.1.412 Queue 1 idleSlopeCredit, Quantum or Weights (MTL_TXQ1_QUANTUM_WEIGHT)

The Queue 1 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 1.

Offset

Register	Offset
MTL_TXQ1_QUANTUM_WEIGHT	D58h

Diagram



Fields

Field	Description
31-21 —	Reserved
20-0 ISCQW	idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. idleSlopeCredit, Quantum or Weights

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero. - Quantum When the DCB operation is enabled with DWRR algorithm for Queue 1 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. - Weights When DCB operation is enabled with WFQ algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero. - Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is MTL_OPERATION_MODE register. - Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size)) - Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

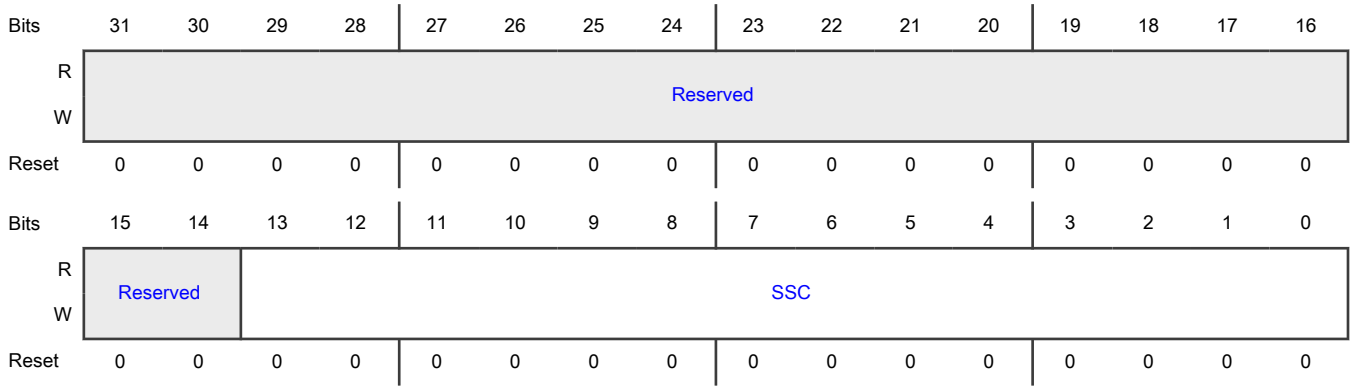
43.8.1.413 Queue 1 sendSlopeCredit (MTL_TXQ1_SENDSLOPECREDIT)

The sendSlopeCredit register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

Offset

Register	Offset
MTL_TXQ1_SENDSLOPECREDIT	D5Ch

Diagram



Fields

Field	Description
31-14 —	Reserved
13-0 SSC	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns, 8 ns and 3.2 ns for 100 Mbps, 1000 Mbps and 2500 Mbps respectively) when the credit is decreasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 1 is selected for transmission.

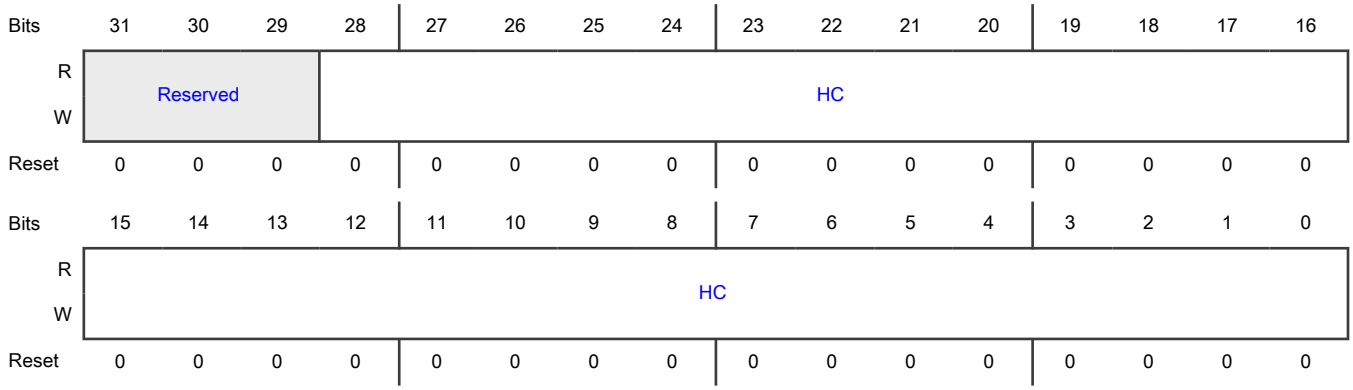
43.8.1.414 Queue 1 hiCredit (MTL_TXQ1_HICREDIT)

The hiCredit register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

Offset

Register	Offset
MTL_TXQ1_HICREDIT	D60h

Diagram



Fields

Field	Description
31-29 —	Reserved
28-0 HC	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 * 1,024 = 134,217,728 or 0x0800_0000.

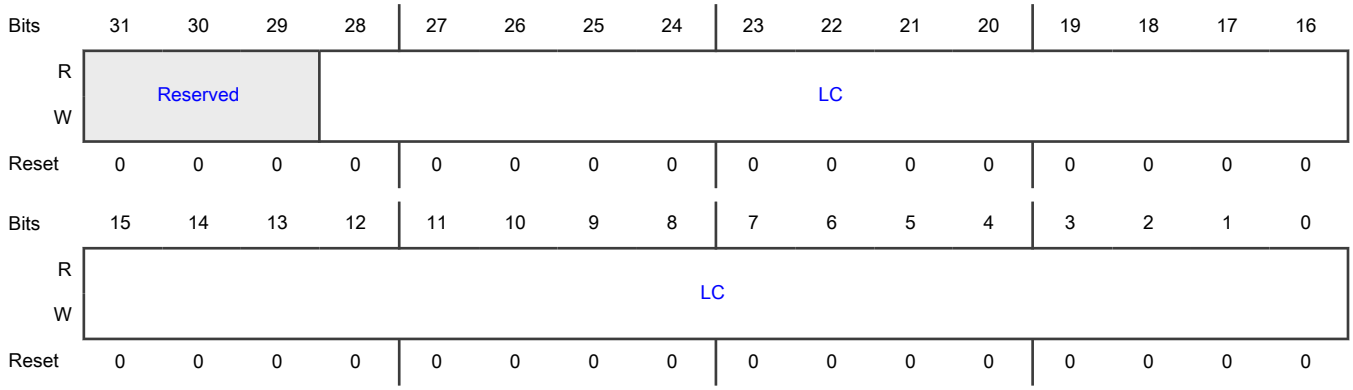
43.8.1.415 Queue 1 loCredit (MTL_TXQ1_LOCREDIT)

The loCredit register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

Offset

Register	Offset
MTL_TXQ1_LOCREDIT	D64h

Diagram



Fields

Field	Description
31-29 —	Reserved
28-0 LC	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192*2) * 8 * 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.

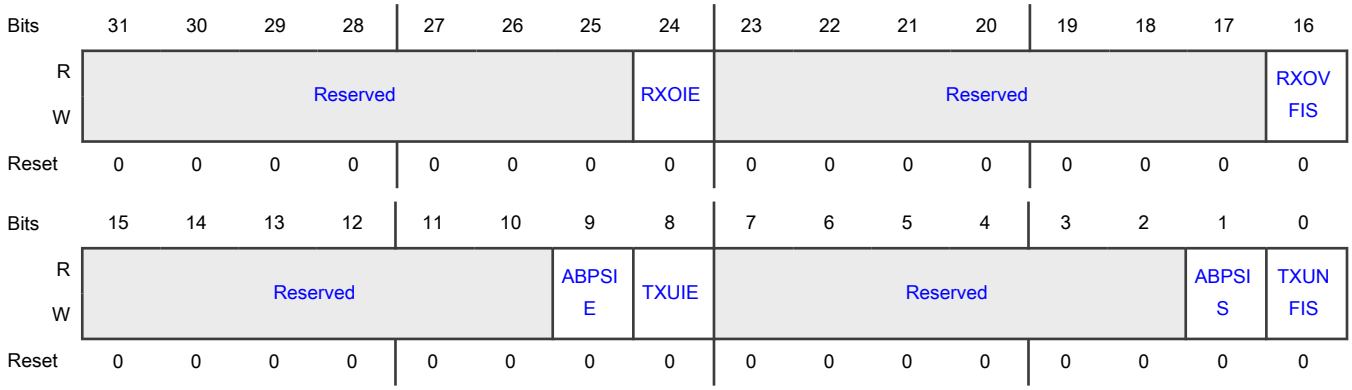
43.8.1.416 Queue 1 Interrupt Control Status (MTL_Q1_INTERRUPT_CONTROL_STATUS)

This register contains the interrupt enable and status bits for the queue 1 interrupts.

Offset

Register	Offset
MTL_Q1_INTERRUPT_CONTROL_STATUS	D6Ch

Diagram



Fields

Field	Description
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0 - Receive Queue Overflow Interrupt is disabled 1 - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Queue Overflow Interrupt Status not detected 1 - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated.

Table continues on the next page...

Table continued from the previous page...

Field	Description
ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0 - Average Bits Per Slot Interrupt is disabled 1 - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0 - Transmit Queue Underflow Interrupt Status is disabled 1 - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Average Bits Per Slot Interrupt Status not detected 1 - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Queue Underflow Interrupt Status not detected 1 - Transmit Queue Underflow Interrupt Status detected

43.8.1.417 Queue 1 Receive Operation Mode (MTL_RXQ1_OPERATION_MODE)

The Queue 1 Receive Operation Mode register establishes the Receive queue operating modes and command.

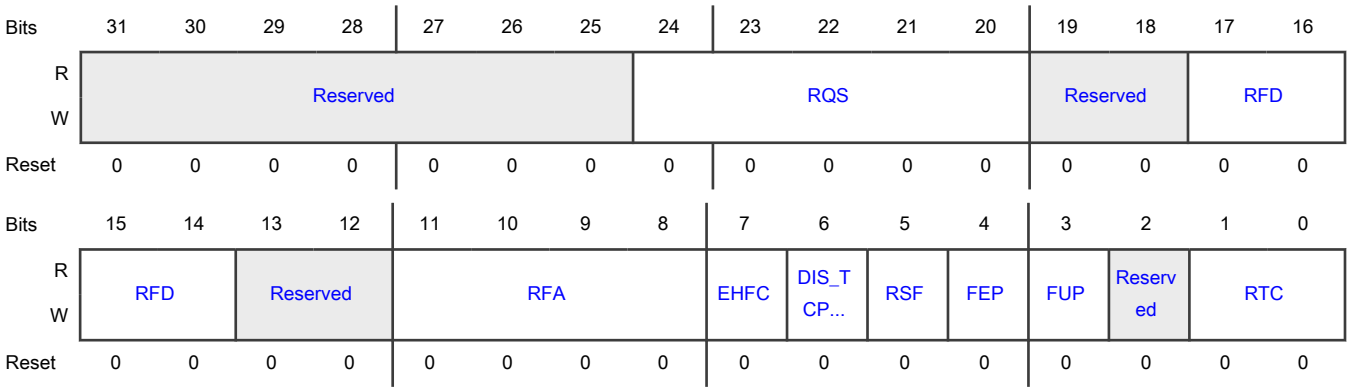
The Queue 1 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

Offset

Register	Offset
MTL_RXQ1_OPERATION_MODE	D70h

Diagram



Fields

Field	Description
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0 - Hardware Flow Control is disabled 1 - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is reset, all error packets are dropped if the FEP bit is reset. 0 - Dropping of TCP/IP Checksum Error Packets is enabled 1 - Dropping of TCP/IP Checksum Error Packets is disabled
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0 - Receive Queue Store and Forward is disabled 1 - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0 - Forward Error Packets is disabled 1 - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0 - Forward Undersized Good Packets is disabled 1 - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00 - 64 01 - 32 10 - 96 11 - 128

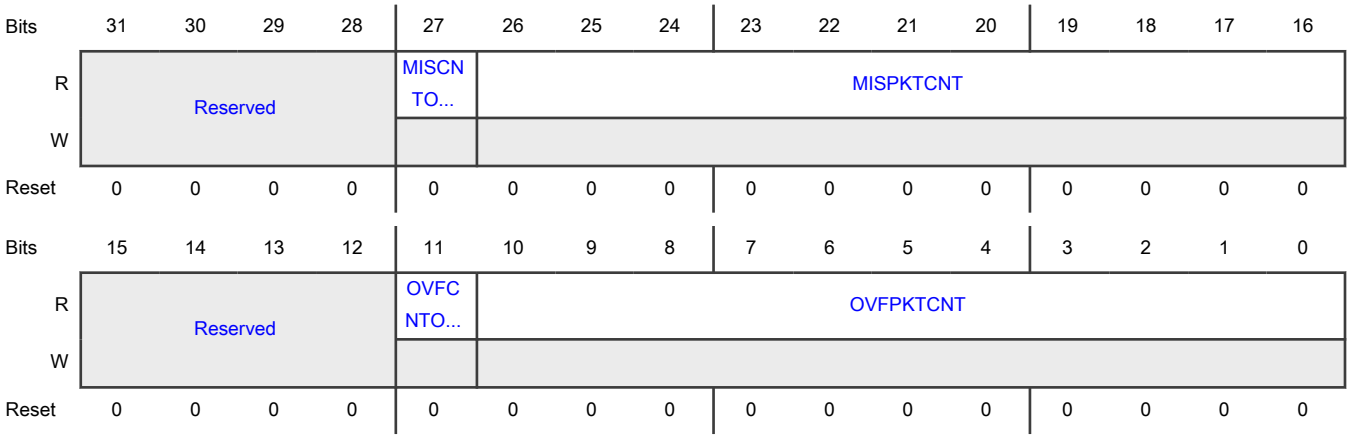
43.8.1.418 Queue 1 Missed Packet and Overflow Counter (MTL_RXQ1_MISSED_PACKET_OVERFLOW_CNT)

The Queue 1 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

Offset

Register	Offset
MTL_RXQ1_MISSED_PACKET_OVERFLOW_CNT	D74h

Diagram



Fields

Field	Description
31-28 —	Reserved
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Missed Packet Counter overflow not detected 1 - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow Counter overflow not detected 1 - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

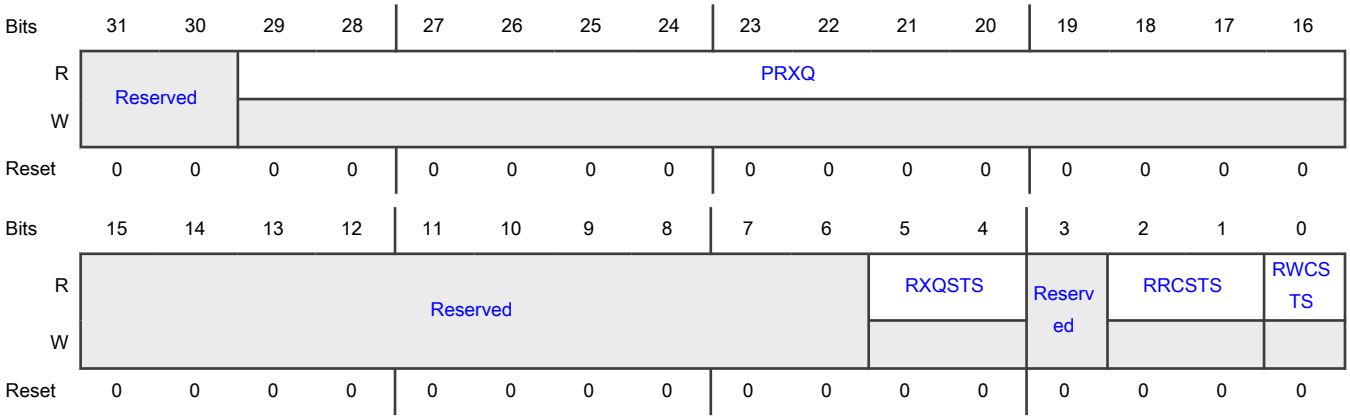
43.8.1.419 Queue 1 Receive Debug (MTL_RXQ1_DEBUG)

The Queue 1 Receive Debug register gives the debug status of various blocks related to the Receive queue.

Offset

Register	Offset
MTL_RXQ1_DEBUG	D78h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00 - Rx Queue empty 01 - Rx Queue fill-level below flow-control deactivate threshold 10 - Rx Queue fill-level above flow-control activate threshold 11 - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00 - Idle state

Table continues on the next page...

Table continued from the previous page...

Field	Description
	01 - Reading packet data 10 - Reading packet status (or timestamp) 11 - Flushing the packet data and status
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0 - MTL Rx Queue Write Controller Active Status not detected 1 - MTL Rx Queue Write Controller Active Status detected

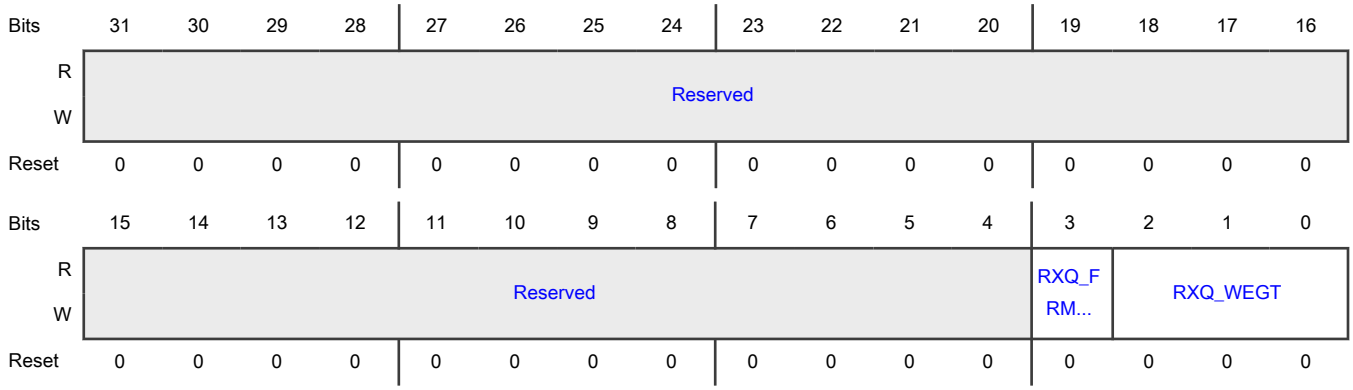
43.8.1.420 Queue 1 Receive Control (MTL_RXQ1_CONTROL)

The Queue 1 Receive Control register controls the receive arbitration and passing of received packets to the application.

Offset

Register	Offset
MTL_RXQ1_CONTROL	D7Ch

Diagram



Fields

Field	Description
31-4 —	Reserved
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0 - Receive Queue Packet Arbitration is disabled 1 - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 1. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 1. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

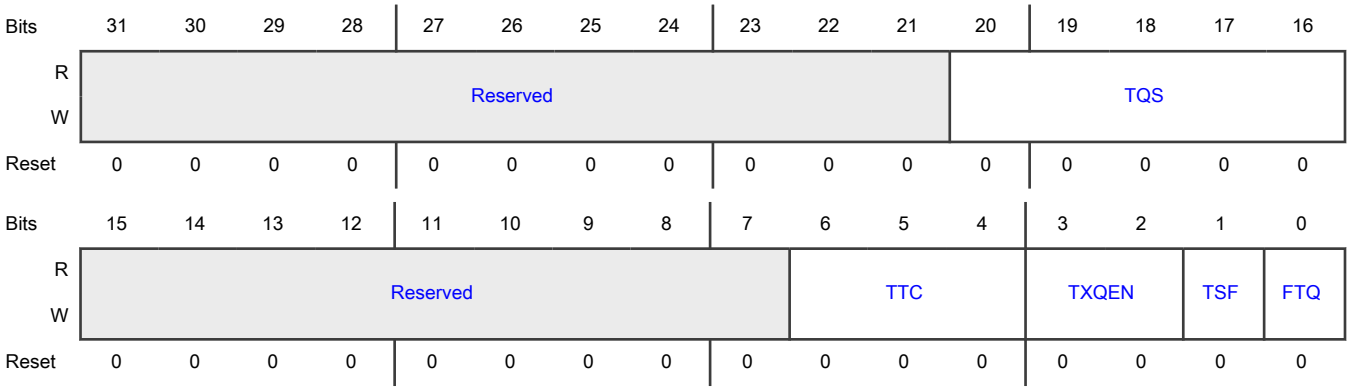
43.8.1.421 Queue 2 Transmit Operation Mode (MTL_TXQ2_OPERATION_MODE)

The Queue 2 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

Offset

Register	Offset
MTL_TXQ2_OPERATION_MODE	D80h

Diagram



Fields

Field	Description
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000 - 32 001 - 64 010 - 96 011 - 128 100 - 192 101 - 256 110 - 384 111 - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Enable in AV mode - 2'b10: Enabled - 2'b11: Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00 - Not enabled 01 - Enable in AV mode (Reserved in non-AV) 10 - Enabled 11 - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0 - Transmit Store and Forward is disabled 1 - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ2_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flush Transmit Queue is disabled 1 - Flush Transmit Queue is enabled

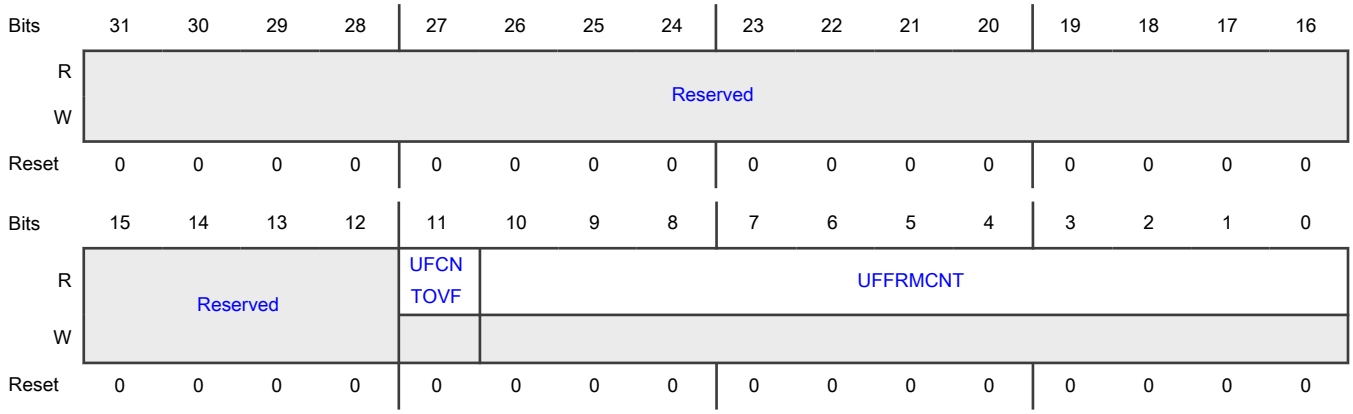
43.8.1.422 Queue 2 Underflow Counter (MTL_TXQ2_UNDERFLOW)

The Queue 2 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

Offset

Register	Offset
MTL_TXQ2_UNDERFLOW	D84h

Diagram



Fields

Field	Description
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow not detected for Underflow Packet Counter 1 - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

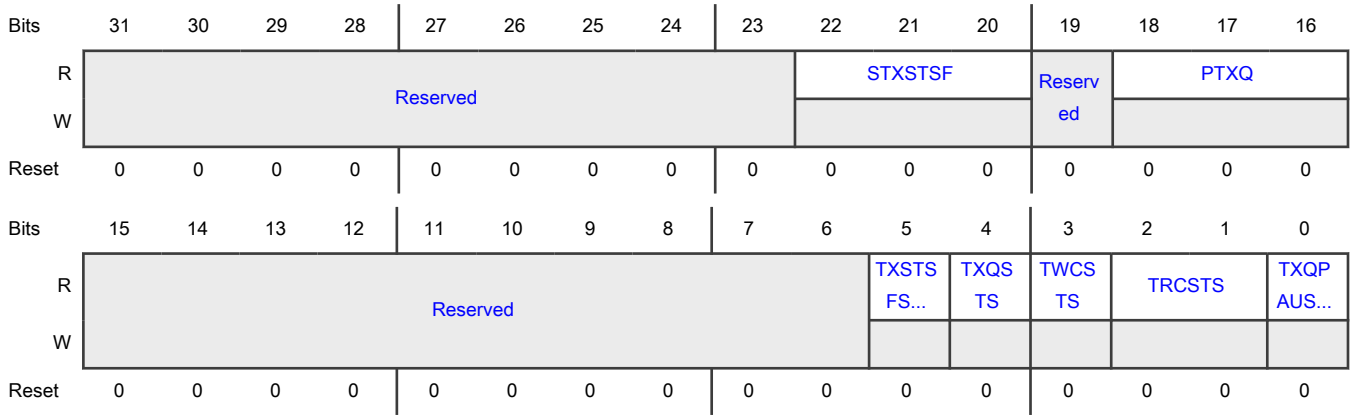
43.8.1.423 Queue 2 Transmit Debug (MTL_TXQ2_DEBUG)

The Queue 2 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

Offset

Register	Offset
MTL_TXQ2_DEBUG	D88h

Diagram



Fields

Field	Description
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0 - MTL Tx Status FIFO Full status is not detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0 - MTL Tx Queue Not Empty status is not detected 1 - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0 - MTL Tx Queue Write Controller status is not detected 1 - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00 - Idle state 01 - Read state (transferring data to the MAC transmitter) 10 - Waiting for pending Tx Status from the MAC transmitter 11 - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0 - Transmit Queue in Pause status is not detected 1 - Transmit Queue in Pause status is detected

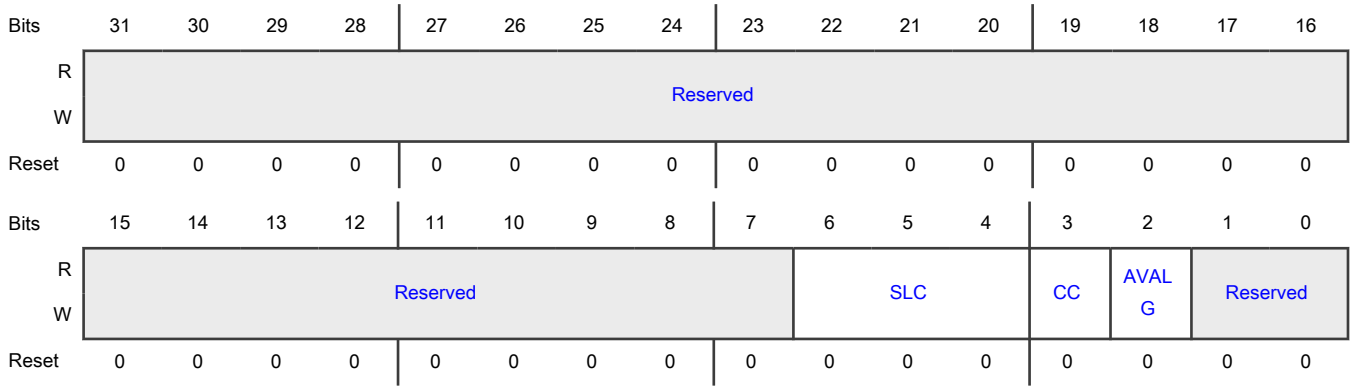
43.8.1.424 Queue 2 ETS Control (MTL_TXQ2_ETS_CONTROL)

The Queue 2 ETS Control register controls the enhanced transmission selection operation.

Offset

Register	Offset
MTL_TXQ2_ETS_CONTROL	D90h

Diagram



Fields

Field	Description
31-7 —	Reserved
6-4 SLC	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH2_SLOT_INTERVAL register) over which the average transmitted bits per slot, provided in the MTL_TXQ2_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 000 - 1 slot 001 - 2 slots 010 - 4 slots 011 - 8 slots 100 - 16 slots 101 - Reserved
3 CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 2. Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 2. The credit accumulates even when there is no packet waiting in Channel 2 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 2. When there is no packet waiting in Channel 2 and other channel is transmitting, no credit is accumulated. 0 - Credit Control is disabled 1 - Credit Control is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 AVALG	AV Algorithm When Queue 2 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 2 traffic. AV Algorithm When Queue 2 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 2 traffic. When reset, strict priority is selected. 0 - CBS Algorithm is disabled 1 - CBS Algorithm is enabled
1-0 —	Reserved

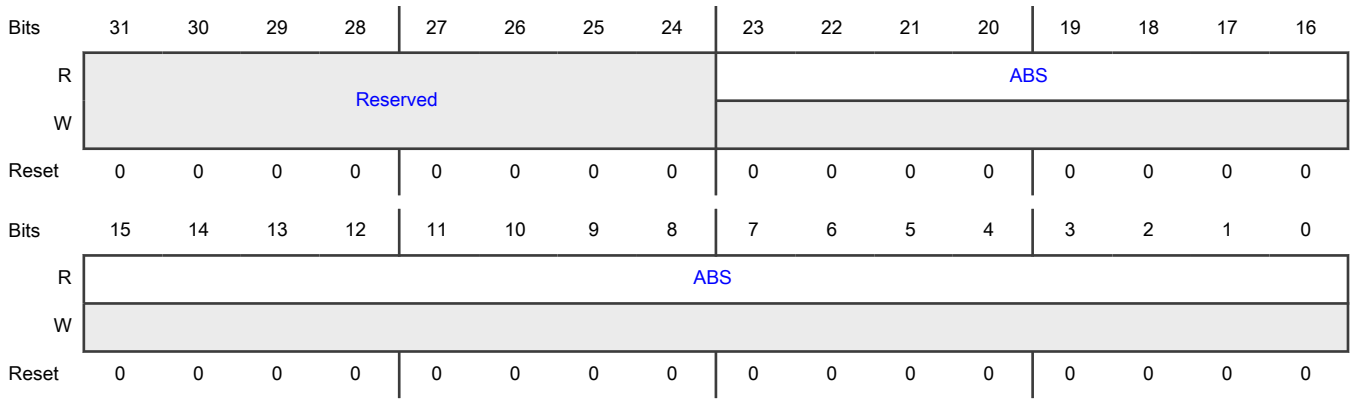
43.8.1.425 Queue 2 ETS Status (MTL_TXQ2_ETS_STATUS)

The Queue 2 ETS Status register provides the average traffic transmitted in Queue 2.

Offset

Register	Offset
MTL_TXQ2_ETS_STATUS	D94h

Diagram



Fields

Field	Description
31-24	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQ2_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

43.8.1.426 Queue 2 idleSlopeCredit, Quantum or Weights (MTL_TXQ2_QUANTUM_WEIGHT)

The Queue 2 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 2.

Offset

Register	Offset
MTL_TXQ2_QUANTUM_WEIGHT	D98h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												ISCQW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ISCQW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-21 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
20-0 ISCQW	idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 2. idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 2. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero. - Quantum When the DCB operation is enabled with DWRR algorithm for Queue 2 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. - Weights When DCB operation is enabled with WFQ algorithm for Queue 2 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 2 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero. - Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is MTL_OPERATION_MODE register. - Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size)) - Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

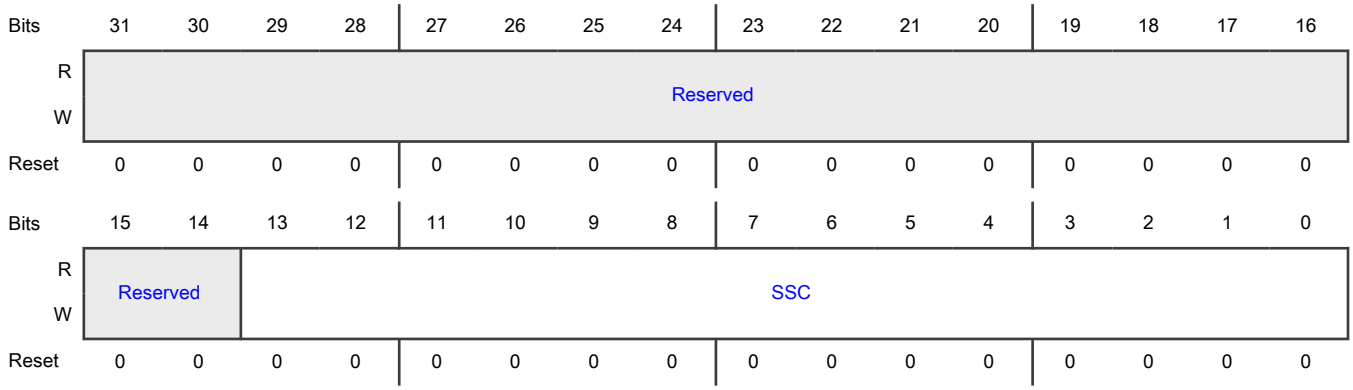
43.8.1.427 Queue 2 sendSlopeCredit (MTL_TXQ2_SENDSLOPECREDIT)

The sendSlopeCredit register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

Offset

Register	Offset
MTL_TXQ2_SENDSLOPECREDIT	D9Ch

Diagram



Fields

Field	Description
31-14 —	Reserved
13-0 SSC	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 2. sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 2. This is the rate of change of credit in bits per cycle (40 ns, 8 ns and 3.2 ns for 100 Mbps, 1000 Mbps and 2500 Mbps respectively) when the credit is decreasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 2 is selected for transmission.

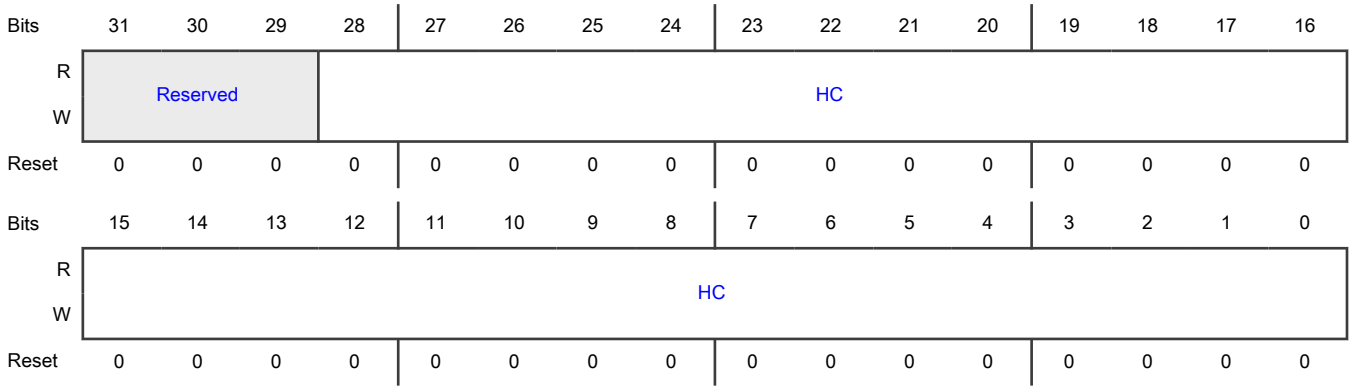
43.8.1.428 Queue 2 hiCredit (MTL_TXQ2_HICREDIT)

The hiCredit register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

Offset

Register	Offset
MTL_TXQ2_HICREDIT	DA0h

Diagram



Fields

Field	Description
31-29 —	Reserved
28-0 HC	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 * 1,024 = 134,217,728 or 0x0800_0000.

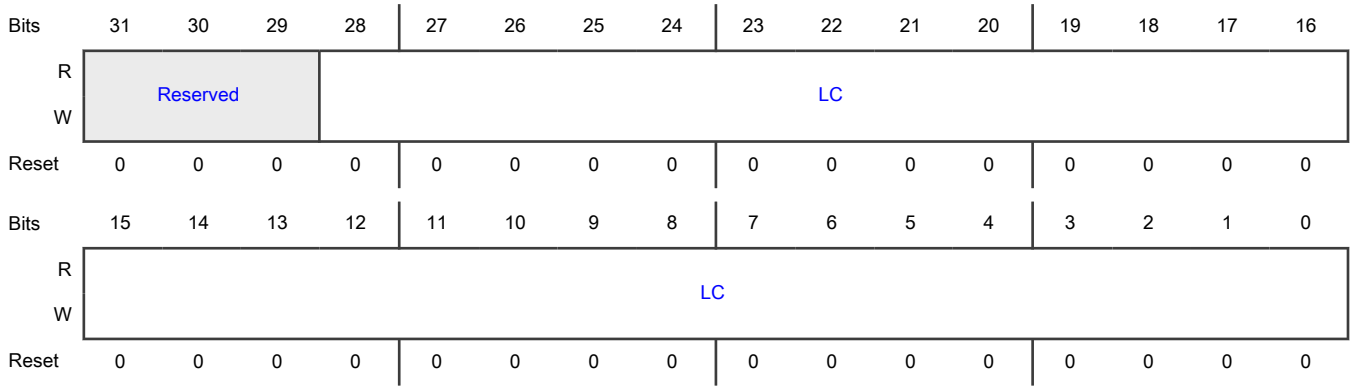
43.8.1.429 Queue 2 loCredit (MTL_TXQ2_LOCREDIT)

The loCredit register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

Offset

Register	Offset
MTL_TXQ2_LOCREDIT	DA4h

Diagram



Fields

Field	Description
31-29 —	Reserved
28-0 LC	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192*2) * 8 * 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.

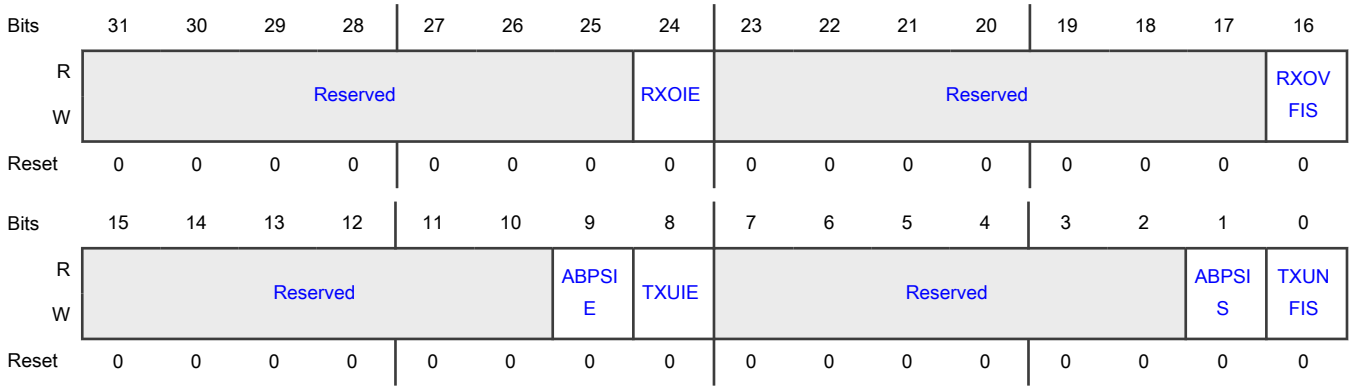
43.8.1.430 Queue 2 Interrupt Control Status (MTL_Q2_INTERRUPT_CONTROL_STATUS)

This register contains the interrupt enable and status bits for the queue 2 interrupts.

Offset

Register	Offset
MTL_Q2_INTERRUPT_CONTROL_STATUS	DACh

Diagram



Fields

Field	Description
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0 - Receive Queue Overflow Interrupt is disabled 1 - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Queue Overflow Interrupt Status not detected 1 - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated.

Table continues on the next page...

Table continued from the previous page...

Field	Description
ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0 - Average Bits Per Slot Interrupt is disabled 1 - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0 - Transmit Queue Underflow Interrupt Status is disabled 1 - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Average Bits Per Slot Interrupt Status not detected 1 - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Queue Underflow Interrupt Status not detected 1 - Transmit Queue Underflow Interrupt Status detected

43.8.1.431 Queue 2 Receive Operation Mode (MTL_RXQ2_OPERATION_MODE)

The Queue 2 Receive Operation Mode register establishes the Receive queue operating modes and command.

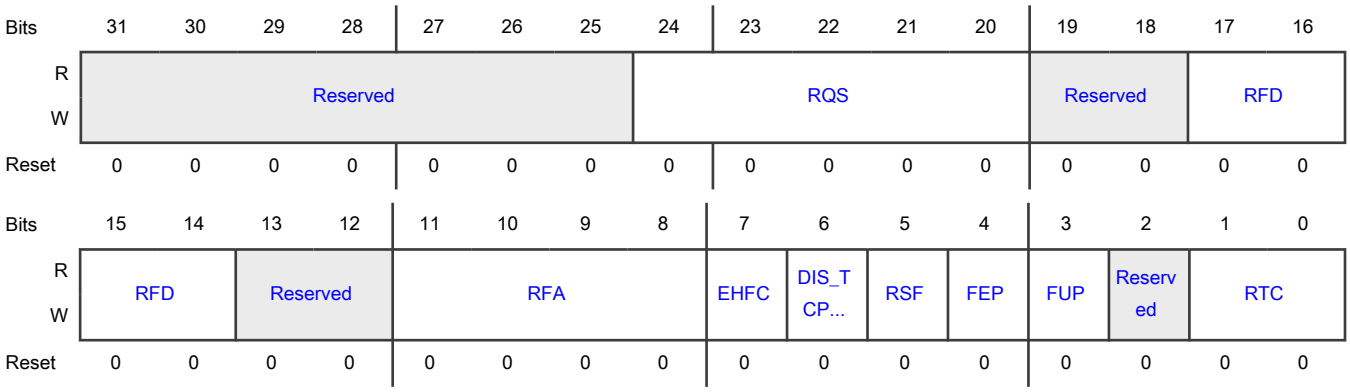
The Queue 2 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

Offset

Register	Offset
MTL_RXQ2_OPERATION_MODE	DB0h

Diagram



Fields

Field	Description
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0 - Hardware Flow Control is disabled 1 - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is reset, all error packets are dropped if the FEP bit is reset. 0 - Dropping of TCP/IP Checksum Error Packets is enabled 1 - Dropping of TCP/IP Checksum Error Packets is disabled
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0 - Receive Queue Store and Forward is disabled 1 - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0 - Forward Error Packets is disabled 1 - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0 - Forward Undersized Good Packets is disabled 1 - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00 - 64 01 - 32 10 - 96 11 - 128

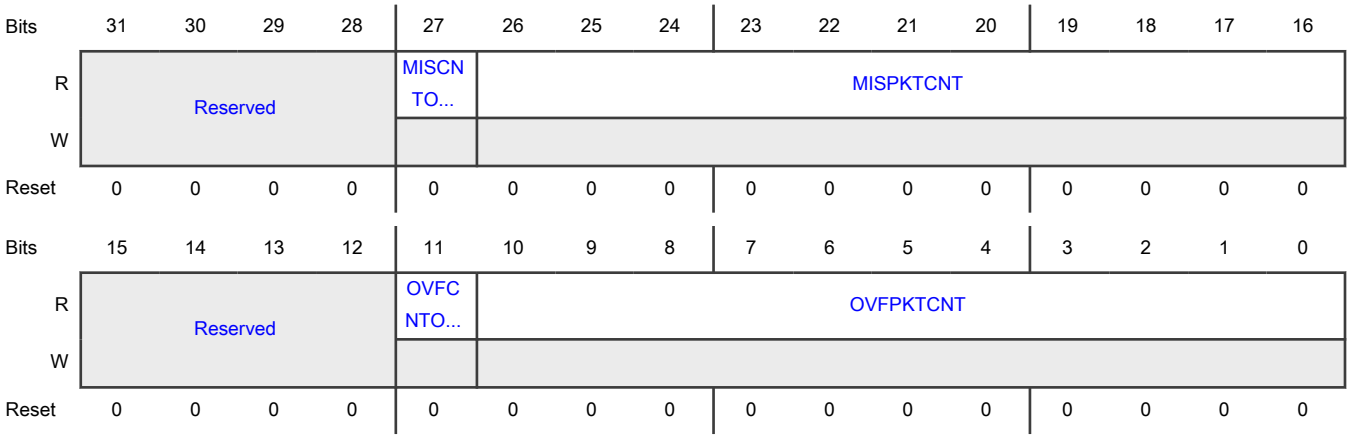
43.8.1.432 Queue 2 Missed Packet and Overflow Counter (MTL_RXQ2_MISSED_PACKET_OVERFLOW_CNT)

The Queue 2 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

Offset

Register	Offset
MTL_RXQ2_MISSED_PACKET_OVERFLOW_CNT	DB4h

Diagram



Fields

Field	Description
31-28 —	Reserved
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Missed Packet Counter overflow not detected 1 - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow Counter overflow not detected 1 - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

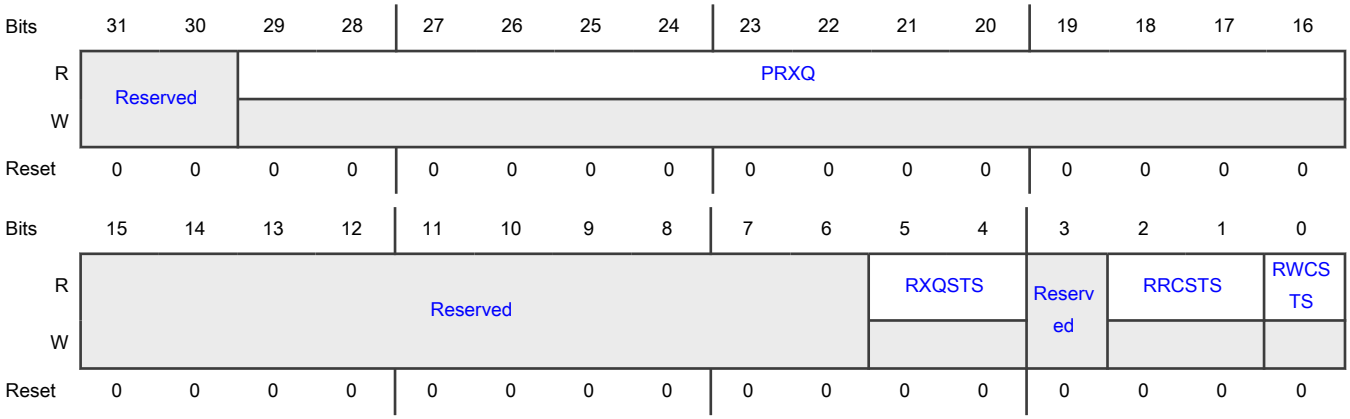
43.8.1.433 Queue 2 Receive Debug (MTL_RXQ2_DEBUG)

The Queue 2 Receive Debug register gives the debug status of various blocks related to the Receive queue.

Offset

Register	Offset
MTL_RXQ2_DEBUG	DB8h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00 - Rx Queue empty 01 - Rx Queue fill-level below flow-control deactivate threshold 10 - Rx Queue fill-level above flow-control activate threshold 11 - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00 - Idle state

Table continues on the next page...

Table continued from the previous page...

Field	Description
	01 - Reading packet data 10 - Reading packet status (or timestamp) 11 - Flushing the packet data and status
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0 - MTL Rx Queue Write Controller Active Status not detected 1 - MTL Rx Queue Write Controller Active Status detected

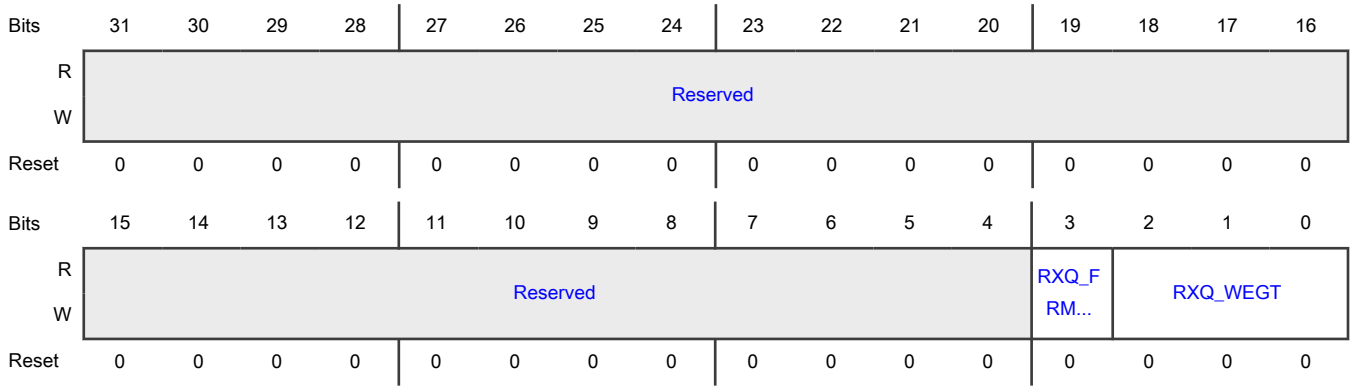
43.8.1.434 Queue 2 Receive Control (MTL_RXQ2_CONTROL)

The Queue 2 Receive Control register controls the receive arbitration and passing of received packets to the application.

Offset

Register	Offset
MTL_RXQ2_CONTROL	DBCh

Diagram



Fields

Field	Description
31-4 —	Reserved
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0 - Receive Queue Packet Arbitration is disabled 1 - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 2. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 2. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

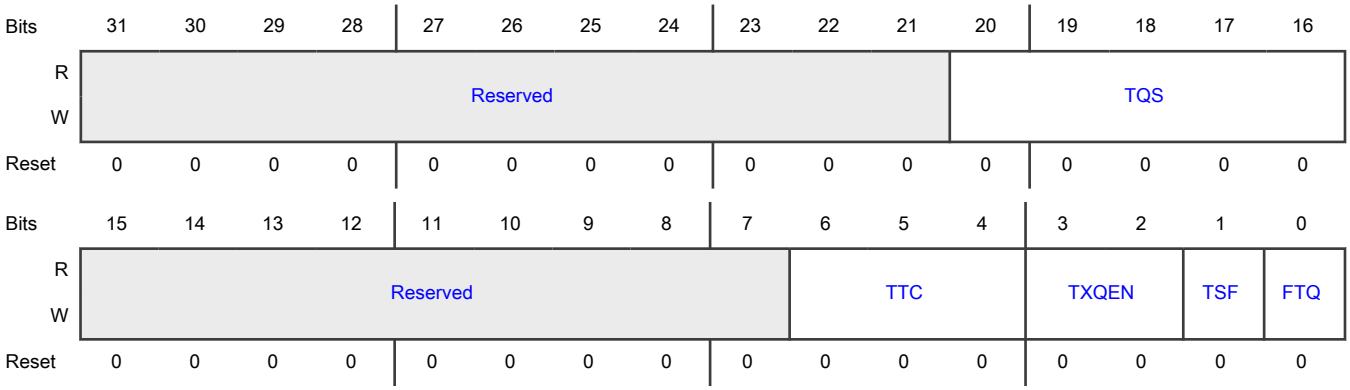
43.8.1.435 Queue 3 Transmit Operation Mode (MTL_TXQ3_OPERATION_MODE)

The Queue 3 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

Offset

Register	Offset
MTL_TXQ3_OPERATION_MODE	DC0h

Diagram



Fields

Field	Description
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000 - 32 001 - 64 010 - 96 011 - 128 100 - 192 101 - 256 110 - 384 111 - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Enable in AV mode - 2'b10: Enabled - 2'b11: Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00 - Not enabled 01 - Enable in AV mode (Reserved in non-AV) 10 - Enabled 11 - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0 - Transmit Store and Forward is disabled 1 - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ3_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flush Transmit Queue is disabled 1 - Flush Transmit Queue is enabled

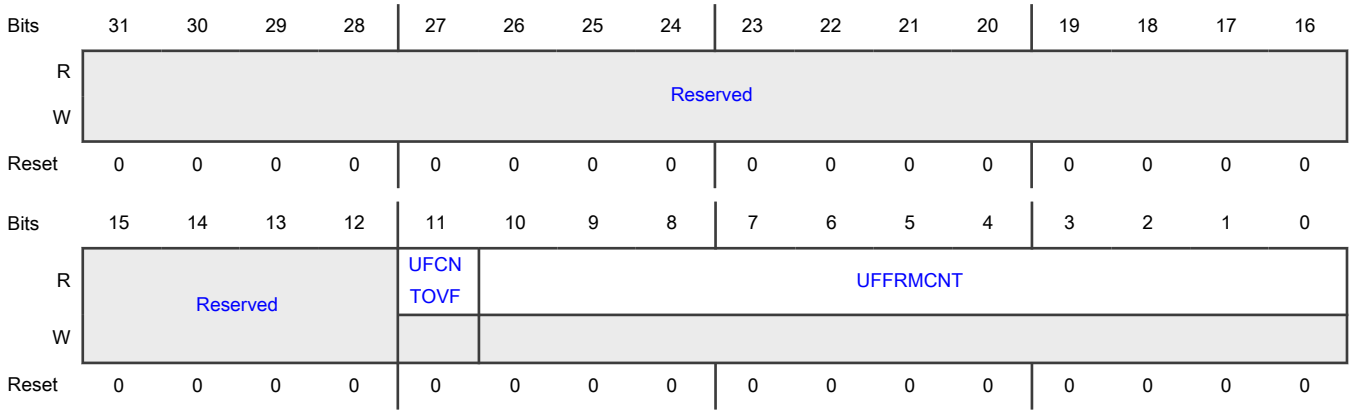
43.8.1.436 Queue 3 Underflow Counter (MTL_TXQ3_UNDERFLOW)

The Queue 3 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

Offset

Register	Offset
MTL_TXQ3_UNDERFLOW	DC4h

Diagram



Fields

Field	Description
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow not detected for Underflow Packet Counter 1 - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

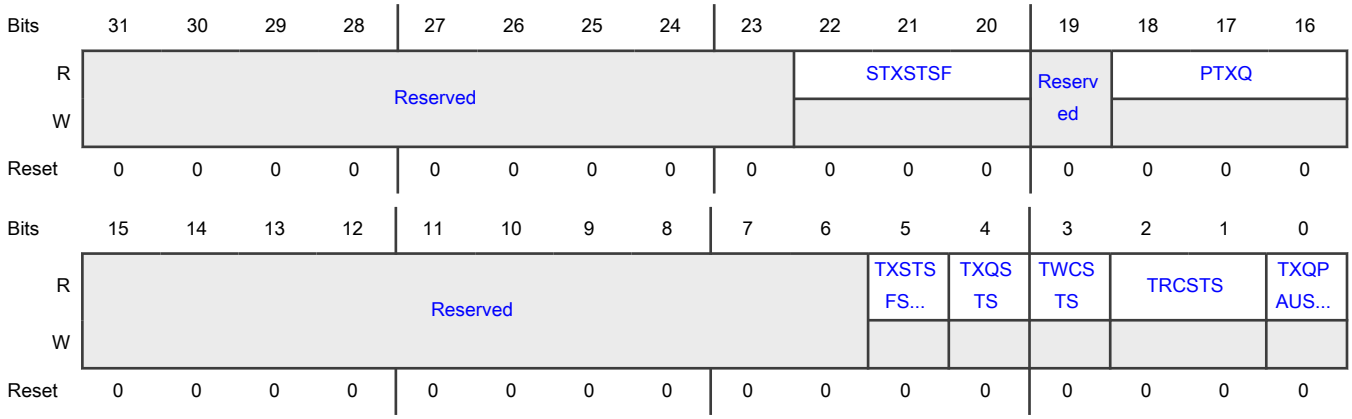
43.8.1.437 Queue 3 Transmit Debug (MTL_TXQ3_DEBUG)

The Queue 3 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

Offset

Register	Offset
MTL_TXQ3_DEBUG	DC8h

Diagram



Fields

Field	Description
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0 - MTL Tx Status FIFO Full status is not detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0 - MTL Tx Queue Not Empty status is not detected 1 - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0 - MTL Tx Queue Write Controller status is not detected 1 - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00 - Idle state 01 - Read state (transferring data to the MAC transmitter) 10 - Waiting for pending Tx Status from the MAC transmitter 11 - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0 - Transmit Queue in Pause status is not detected 1 - Transmit Queue in Pause status is detected

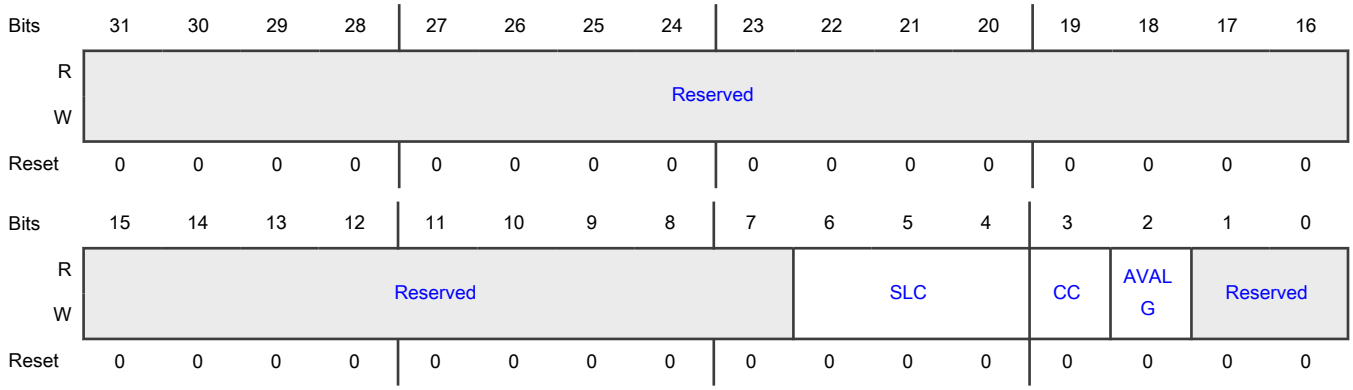
43.8.1.438 Queue 3 ETS Control (MTL_TXQ3_ETS_CONTROL)

The Queue 3 ETS Control register controls the enhanced transmission selection operation.

Offset

Register	Offset
MTL_TXQ3_ETS_CONTROL	DD0h

Diagram



Fields

Field	Description
31-7 —	Reserved
6-4 SLC	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH3_SLOT_INTERVAL register) over which the average transmitted bits per slot, provided in the MTL_TXQ3_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 000 - 1 slot 001 - 2 slots 010 - 4 slots 011 - 8 slots 100 - 16 slots 101 - Reserved
3 CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 3. Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 3. The credit accumulates even when there is no packet waiting in Channel 3 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 3. When there is no packet waiting in Channel 3 and other channel is transmitting, no credit is accumulated. 0 - Credit Control is disabled 1 - Credit Control is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 AVALG	AV Algorithm When Queue 3 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 3 traffic. AV Algorithm When Queue 3 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 3 traffic. When reset, strict priority is selected. 0 - CBS Algorithm is disabled 1 - CBS Algorithm is enabled
1-0 —	Reserved

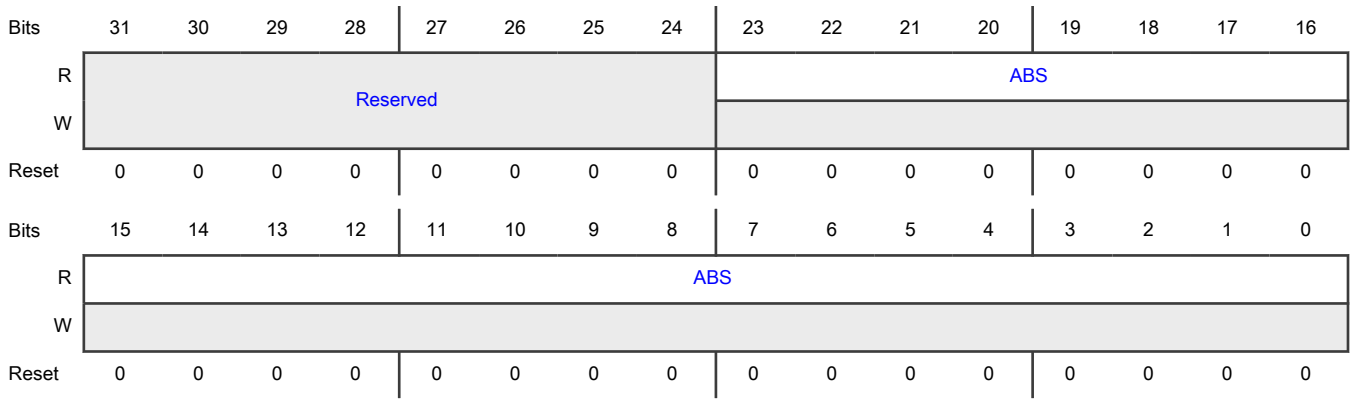
43.8.1.439 Queue 3 ETS Status (MTL_TXQ3_ETS_STATUS)

The Queue 3 ETS Status register provides the average traffic transmitted in Queue 3.

Offset

Register	Offset
MTL_TXQ3_ETS_STATUS	DD4h

Diagram



Fields

Field	Description
31-24	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQ3_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

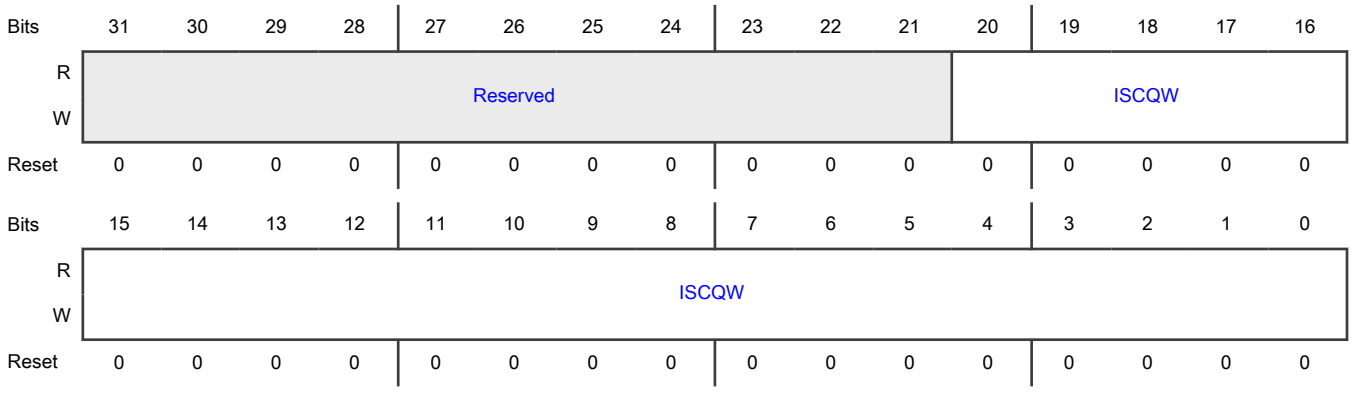
43.8.1.440 Queue 3 idleSlopeCredit, Quantum or Weights (MTL_TXQ3_QUANTUM_WEIGHT)

The Queue 3 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 3.

Offset

Register	Offset
MTL_TXQ3_QUANTUM_WEIGHT	DD8h

Diagram



Fields

Field	Description
31-21 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
20-0 ISCQW	idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 3. idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 3. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero. - Quantum When the DCB operation is enabled with DWRR algorithm for Queue 3 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. - Weights When DCB operation is enabled with WFQ algorithm for Queue 3 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 3 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero. - Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is OPERATION_MODE register. - Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size)) - Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

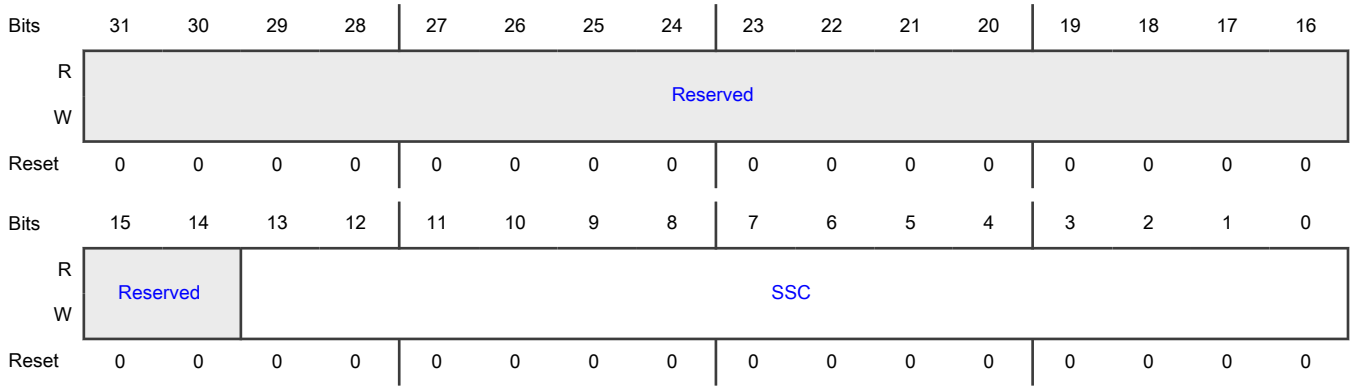
43.8.1.441 Queue 3 sendSlopeCredit (MTL_TXQ3_SENDSLOPECREDIT)

The sendSlopeCredit register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

Offset

Register	Offset
MTL_TXQ3_SENDSLOPECREDIT	DDCh

Diagram



Fields

Field	Description
31-14 —	Reserved
13-0 SSC	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 3. sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 3. This is the rate of change of credit in bits per cycle (40 ns, 8 ns and 3.2 ns for 100 Mbps, 1000 Mbps and 2500 Mbps respectively) when the credit is decreasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 3 is selected for transmission.

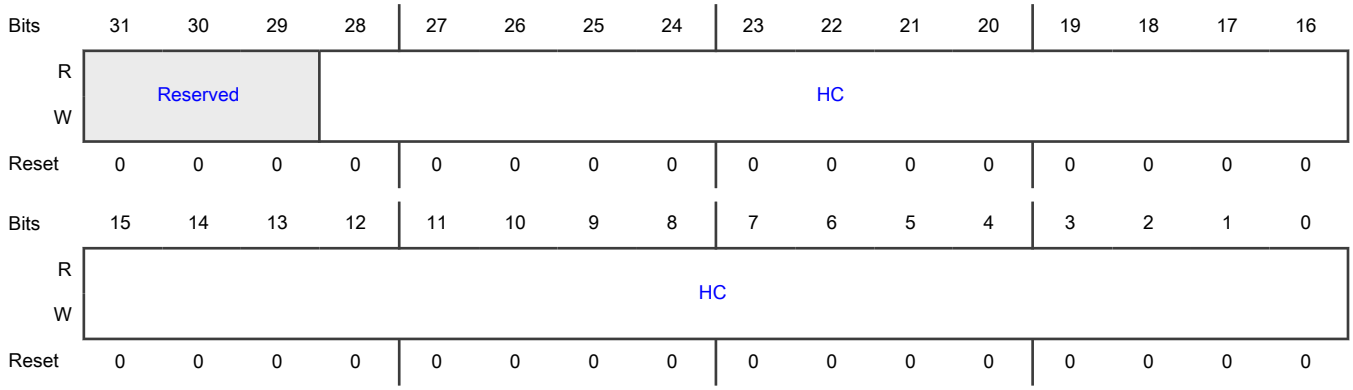
43.8.1.442 Queue 3 hiCredit (MTL_TXQ3_HICREDIT)

The hiCredit register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

Offset

Register	Offset
MTL_TXQ3_HICREDIT	DE0h

Diagram



Fields

Field	Description
31-29 —	Reserved
28-0 HC	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 * 1,024 = 134,217,728 or 0x0800_0000.

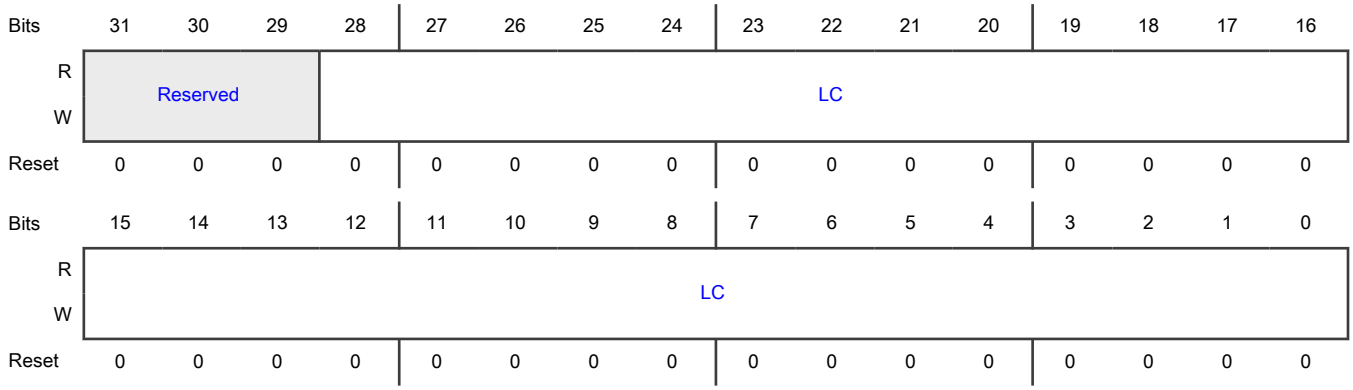
43.8.1.443 Queue 3 loCredit (MTL_TXQ3_LOCREDIT)

The loCredit register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

Offset

Register	Offset
MTL_TXQ3_LOCREDIT	DE4h

Diagram



Fields

Field	Description
31-29 —	Reserved
28-0 LC	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192*2) * 8 * 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.

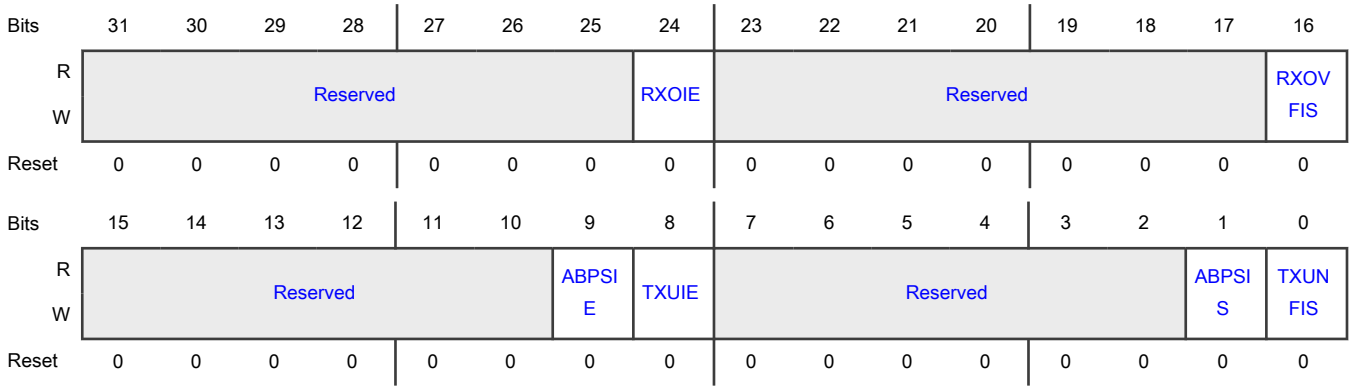
43.8.1.444 Queue 3 Interrupt Control Status (MTL_Q3_INTERRUPT_CONTROL_STATUS)

This register contains the interrupt enable and status bits for the queue 3 interrupts.

Offset

Register	Offset
MTL_Q3_INTERRUPT_CONTROL_STATUS	DECh

Diagram



Fields

Field	Description
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0 - Receive Queue Overflow Interrupt is disabled 1 - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Queue Overflow Interrupt Status not detected 1 - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated.

Table continues on the next page...

Table continued from the previous page...

Field	Description
ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0 - Average Bits Per Slot Interrupt is disabled 1 - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0 - Transmit Queue Underflow Interrupt Status is disabled 1 - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Average Bits Per Slot Interrupt Status not detected 1 - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Queue Underflow Interrupt Status not detected 1 - Transmit Queue Underflow Interrupt Status detected

43.8.1.445 Queue 3 Receive Operation Mode (MTL_RXQ3_OPERATION_MODE)

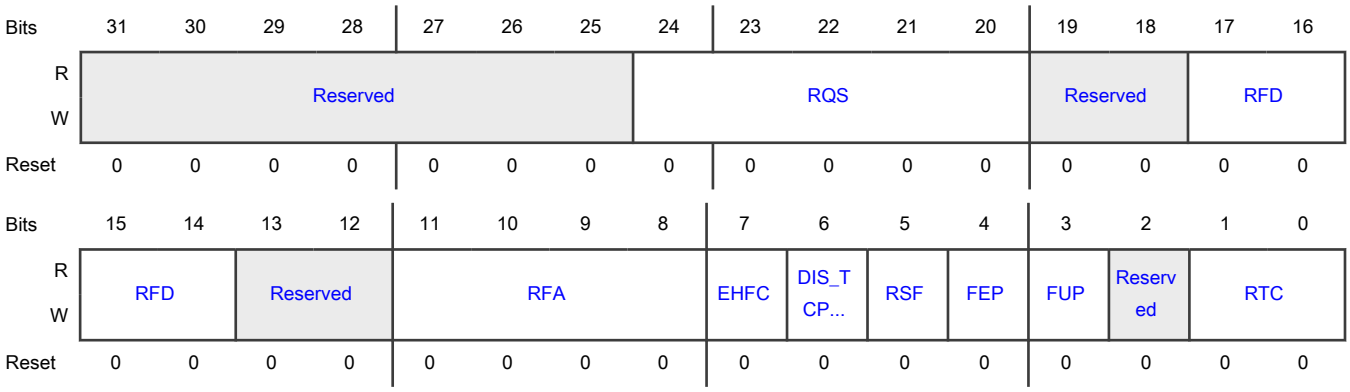
The Queue 3 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

Offset

Register	Offset
MTL_RXQ3_OPERATION_MODE	DF0h

Diagram



Fields

Field	Description
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0 - Hardware Flow Control is disabled 1 - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is reset, all error packets are dropped if the FEP bit is reset. 0 - Dropping of TCP/IP Checksum Error Packets is enabled 1 - Dropping of TCP/IP Checksum Error Packets is disabled
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0 - Receive Queue Store and Forward is disabled 1 - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0 - Forward Error Packets is disabled 1 - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0 - Forward Undersized Good Packets is disabled 1 - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00 - 64 01 - 32 10 - 96 11 - 128

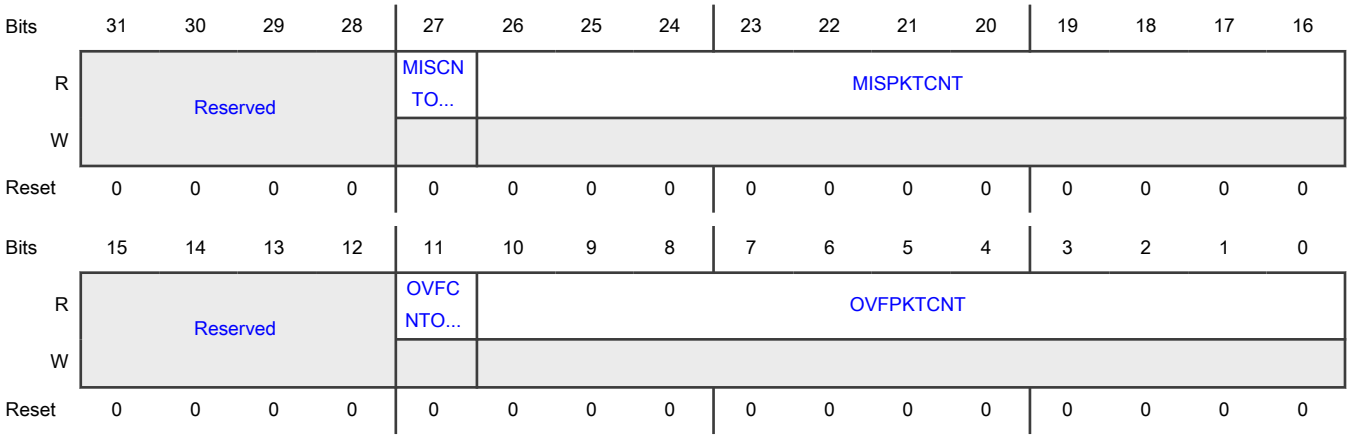
43.8.1.446 Queue 3 Missed Packet and Overflow Counter (MTL_RXQ3_MISSED_PACKET_OVERFLOW_CNT)

The Queue 3 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

Offset

Register	Offset
MTL_RXQ3_MISSED_PACKET_OVERFLOW_CNT	DF4h

Diagram



Fields

Field	Description
31-28 —	Reserved
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Missed Packet Counter overflow not detected 1 - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow Counter overflow not detected 1 - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

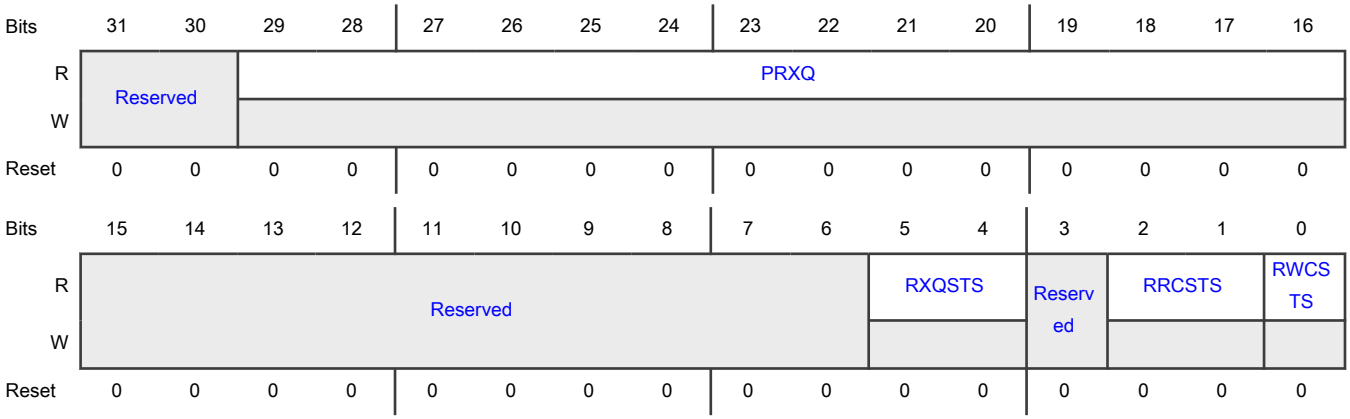
43.8.1.447 Queue 3 Receive Debug (MTL_RXQ3_DEBUG)

The Queue 3 Receive Debug register gives the debug status of various blocks related to the Receive queue.

Offset

Register	Offset
MTL_RXQ3_DEBUG	DF8h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00 - Rx Queue empty 01 - Rx Queue fill-level below flow-control deactivate threshold 10 - Rx Queue fill-level above flow-control activate threshold 11 - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00 - Idle state

Table continues on the next page...

Table continued from the previous page...

Field	Description
	01 - Reading packet data 10 - Reading packet status (or timestamp) 11 - Flushing the packet data and status
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0 - MTL Rx Queue Write Controller Active Status not detected 1 - MTL Rx Queue Write Controller Active Status detected

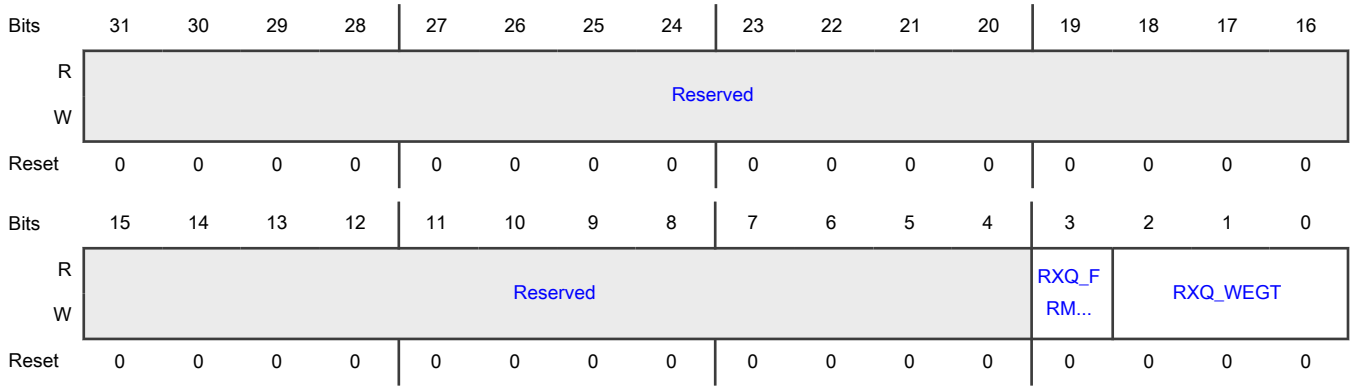
43.8.1.448 Queue 3 Receive Control (MTL_RXQ3_CONTROL)

The Queue 3 Receive Control register controls the receive arbitration and passing of received packets to the application.

Offset

Register	Offset
MTL_RXQ3_CONTROL	DFCh

Diagram



Fields

Field	Description
31-4 —	Reserved
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0 - Receive Queue Packet Arbitration is disabled 1 - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 3. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 3. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

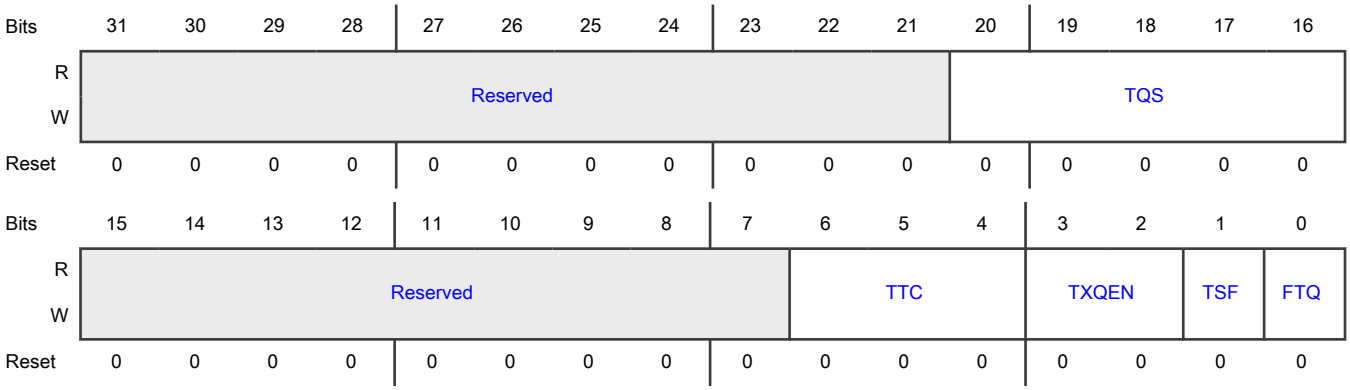
43.8.1.449 Queue 4 Transmit Operation Mode (MTL_TXQ4_OPERATION_MODE)

The Queue 4 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

Offset

Register	Offset
MTL_TXQ4_OPERATION_MODE	E00h

Diagram



Fields

Field	Description
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000 - 32 001 - 64 010 - 96 011 - 128 100 - 192 101 - 256 110 - 384 111 - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Enable in AV mode - 2'b10: Enabled - 2'b11: Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00 - Not enabled 01 - Enable in AV mode (Reserved in non-AV) 10 - Enabled 11 - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0 - Transmit Store and Forward is disabled 1 - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ4_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flush Transmit Queue is disabled 1 - Flush Transmit Queue is enabled

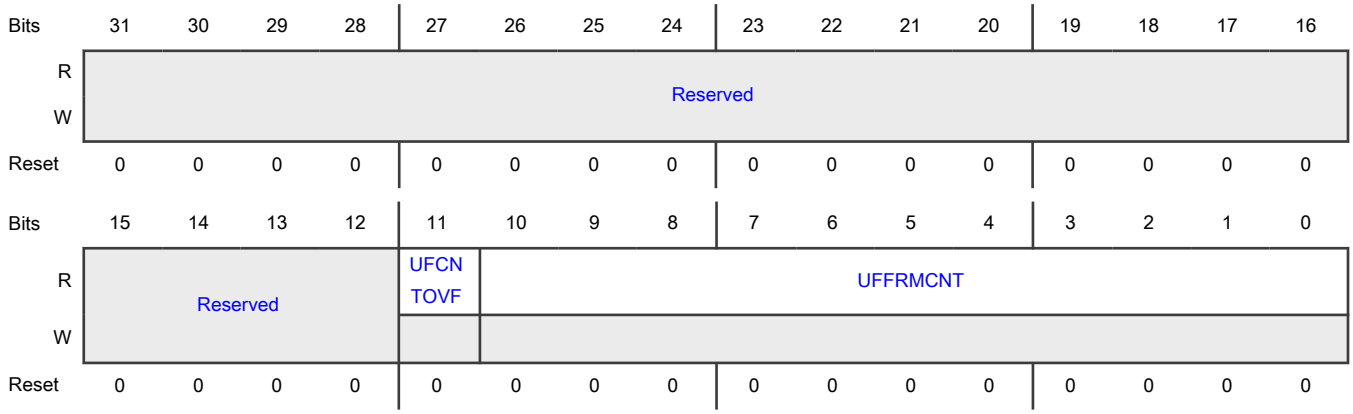
43.8.1.450 Queue 4 Underflow Counter (MTL_TXQ4_UNDERFLOW)

The Queue 4 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

Offset

Register	Offset
MTL_TXQ4_UNDERFLOW	E04h

Diagram



Fields

Field	Description
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow not detected for Underflow Packet Counter 1 - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

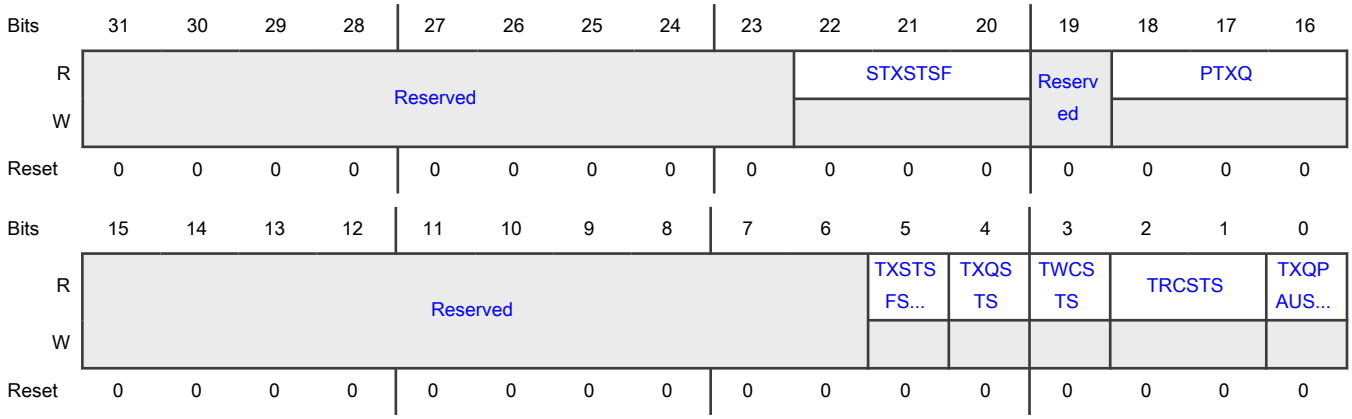
43.8.1.451 Queue 4 Transmit Debug (MTL_TXQ4_DEBUG)

The Queue 4 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

Offset

Register	Offset
MTL_TXQ4_DEBUG	E08h

Diagram



Fields

Field	Description
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0 - MTL Tx Status FIFO Full status is not detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0 - MTL Tx Queue Not Empty status is not detected 1 - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0 - MTL Tx Queue Write Controller status is not detected 1 - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00 - Idle state 01 - Read state (transferring data to the MAC transmitter) 10 - Waiting for pending Tx Status from the MAC transmitter 11 - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0 - Transmit Queue in Pause status is not detected 1 - Transmit Queue in Pause status is detected

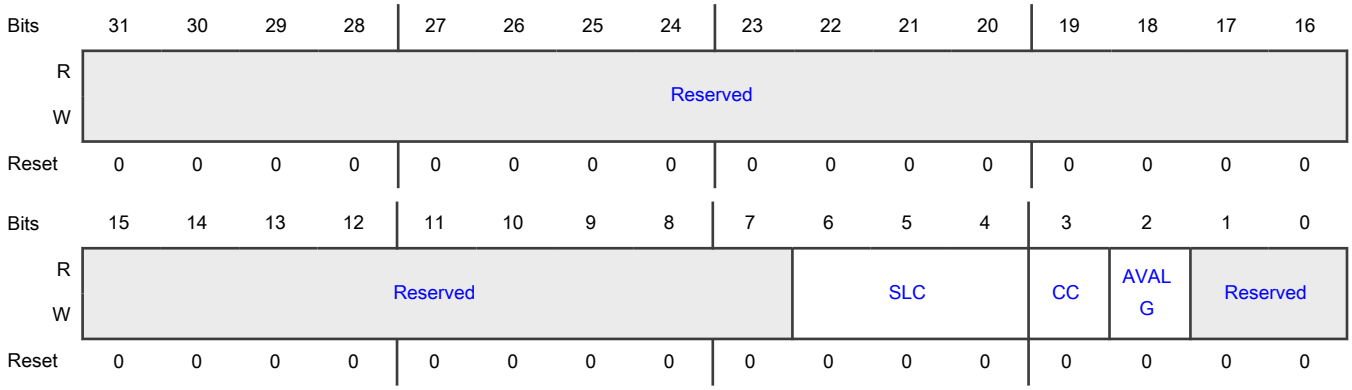
43.8.1.452 Queue 4 ETS Control (MTL_TXQ4_ETS_CONTROL)

The Queue 4 ETS Control register controls the enhanced transmission selection operation.

Offset

Register	Offset
MTL_TXQ4_ETS_CONTROL	E10h

Diagram



Fields

Field	Description
31-7 —	Reserved
6-4 SLC	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH4_SLOT_INTERVAL register) over which the average transmitted bits per slot, provided in the MTL_TXQ4_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 000 - 1 slot 001 - 2 slots 010 - 4 slots 011 - 8 slots 100 - 16 slots 101 - Reserved
3 CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 4. Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 4. The credit accumulates even when there is no packet waiting in Channel 4 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 4. When there is no packet waiting in Channel 4 and other channel is transmitting, no credit is accumulated. 0 - Credit Control is disabled 1 - Credit Control is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 AVALG	AV Algorithm When Queue 4 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 4 traffic. AV Algorithm When Queue 4 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 4 traffic. When reset, strict priority is selected. 0 - CBS Algorithm is disabled 1 - CBS Algorithm is enabled
1-0 —	Reserved

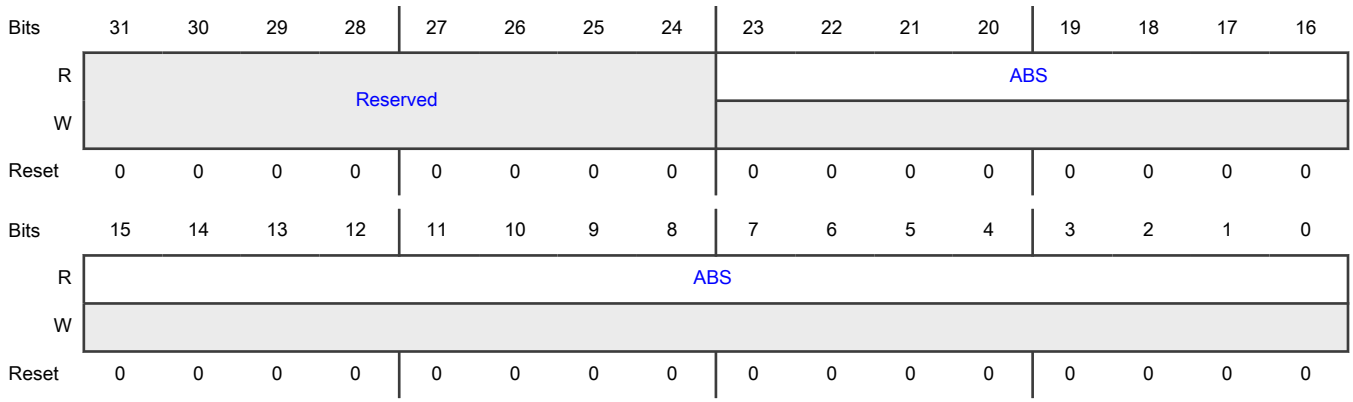
43.8.1.453 Queue 4 ETS Status (MTL_TXQ4_ETS_STATUS)

The Queue 4 ETS Status register provides the average traffic transmitted in Queue 4.

Offset

Register	Offset
MTL_TXQ4_ETS_STATUS	E14h

Diagram



Fields

Field	Description
31-24	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQ4_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

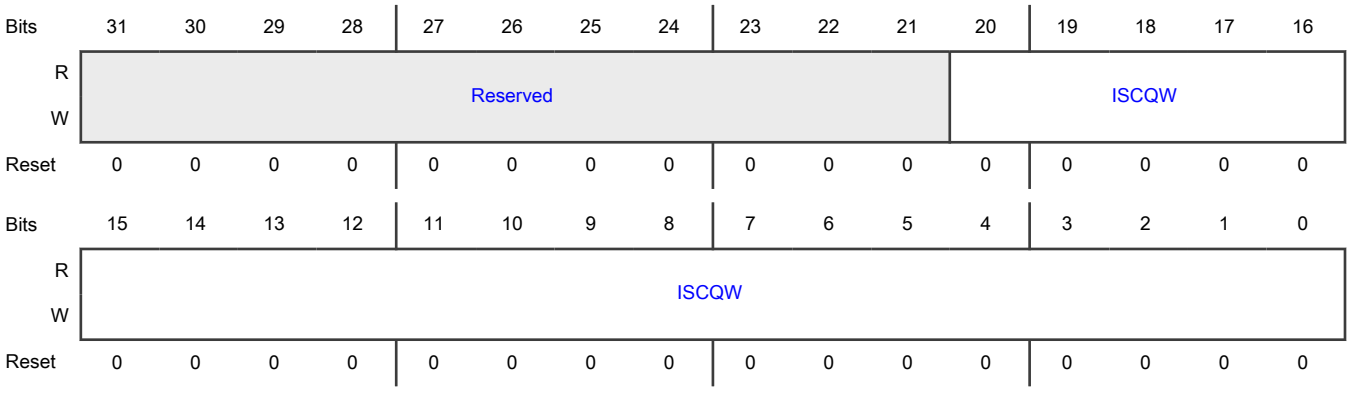
43.8.1.454 Queue 4 idleSlopeCredit, Quantum or Weights (MTL_TXQ4_QUANTUM_WEIGHT)

The Queue 4 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 4.

Offset

Register	Offset
MTL_TXQ4_QUANTUM_WEIGHT	E18h

Diagram



Fields

Field	Description
31-21 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
20-0 ISCQW	idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 4. idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 4. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero. - Quantum When the DCB operation is enabled with DWRR algorithm for Queue 4 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. - Weights When DCB operation is enabled with WFQ algorithm for Queue 4 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 4 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero. - Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is OPERATION_MODE register. - Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size)) - Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

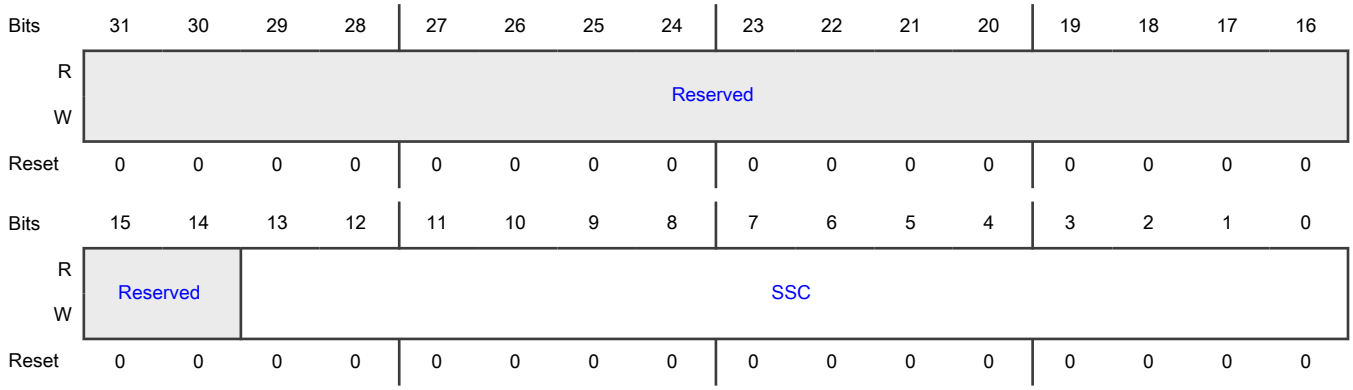
43.8.1.455 Queue 4 sendSlopeCredit (MTL_TXQ4_SENDSLOPECREDIT)

The sendSlopeCredit register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

Offset

Register	Offset
MTL_TXQ4_SENDSLOPECREDIT	E1Ch

Diagram



Fields

Field	Description
31-14 —	Reserved
13-0 SSC	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 4. sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 4. This is the rate of change of credit in bits per cycle (40 ns, 8 ns and 3.2 ns for 100 Mbps, 1000 Mbps and 2500 Mbps respectively) when the credit is decreasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 4 is selected for transmission.

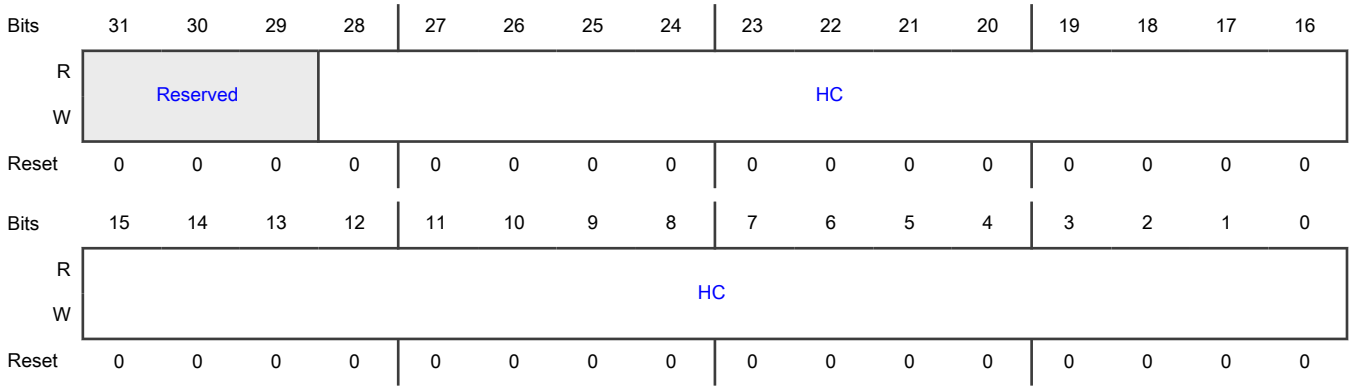
43.8.1.456 Queue 4 hiCredit (MTL_TXQ4_HICREDIT)

The hiCredit register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

Offset

Register	Offset
MTL_TXQ4_HICREDIT	E20h

Diagram



Fields

Field	Description
31-29 —	Reserved
28-0 HC	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 * 1,024 = 134,217,728 or 0x0800_0000.

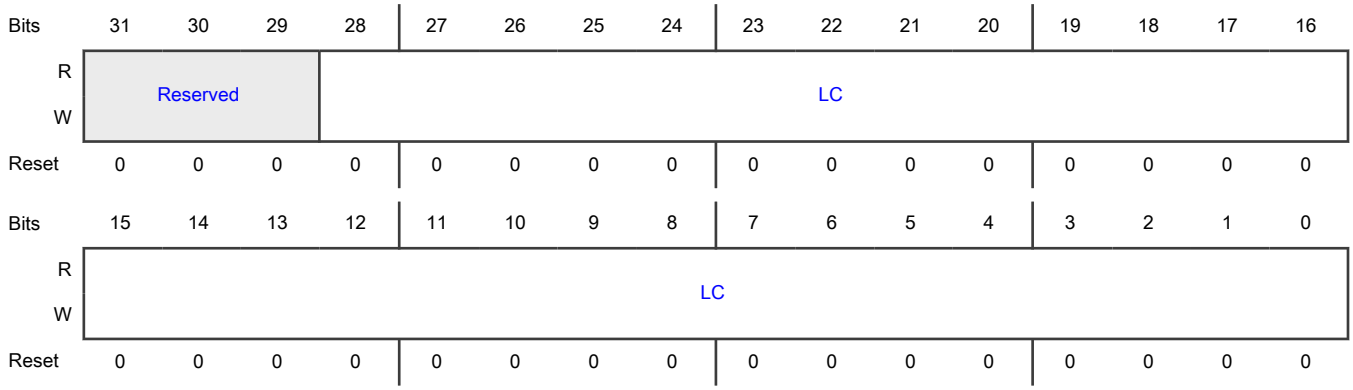
43.8.1.457 Queue 4 loCredit (MTL_TXQ4_LOCREDIT)

The loCredit register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

Offset

Register	Offset
MTL_TXQ4_LOCREDIT	E24h

Diagram



Fields

Field	Description
31-29 —	Reserved
28-0 LC	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192*2) * 8 * 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.

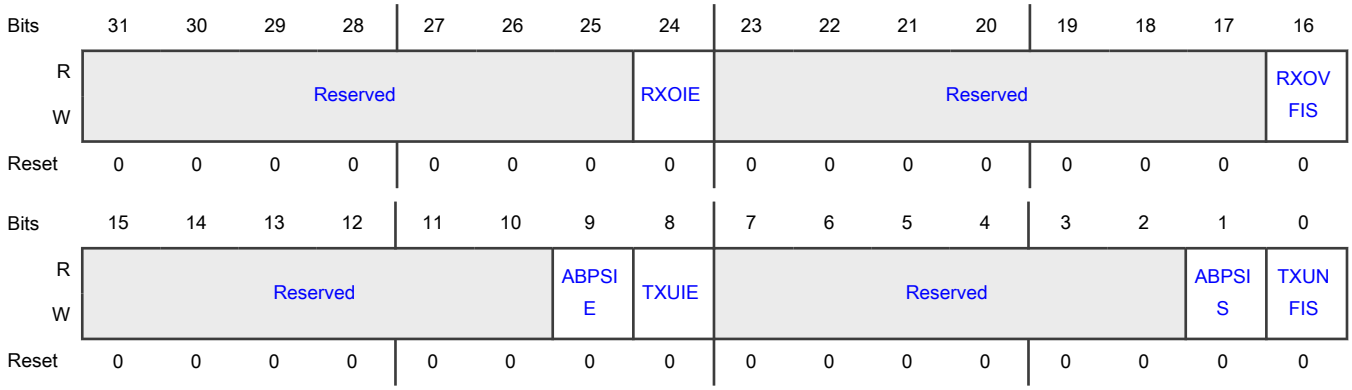
43.8.1.458 Queue 4 Interrupt Control Status (MTL_Q4_INTERRUPT_CONTROL_STATUS)

This register contains the interrupt enable and status bits for the queue 4 interrupts.

Offset

Register	Offset
MTL_Q4_INTERRUPT_CONTROL_STATUS	E2Ch

Diagram



Fields

Field	Description
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0 - Receive Queue Overflow Interrupt is disabled 1 - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Queue Overflow Interrupt Status not detected 1 - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated.

Table continues on the next page...

Table continued from the previous page...

Field	Description
ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0 - Average Bits Per Slot Interrupt is disabled 1 - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0 - Transmit Queue Underflow Interrupt Status is disabled 1 - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Average Bits Per Slot Interrupt Status not detected 1 - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Queue Underflow Interrupt Status not detected 1 - Transmit Queue Underflow Interrupt Status detected

43.8.1.459 Queue 4 Receive Operation Mode (MTL_RXQ4_OPERATION_MODE)

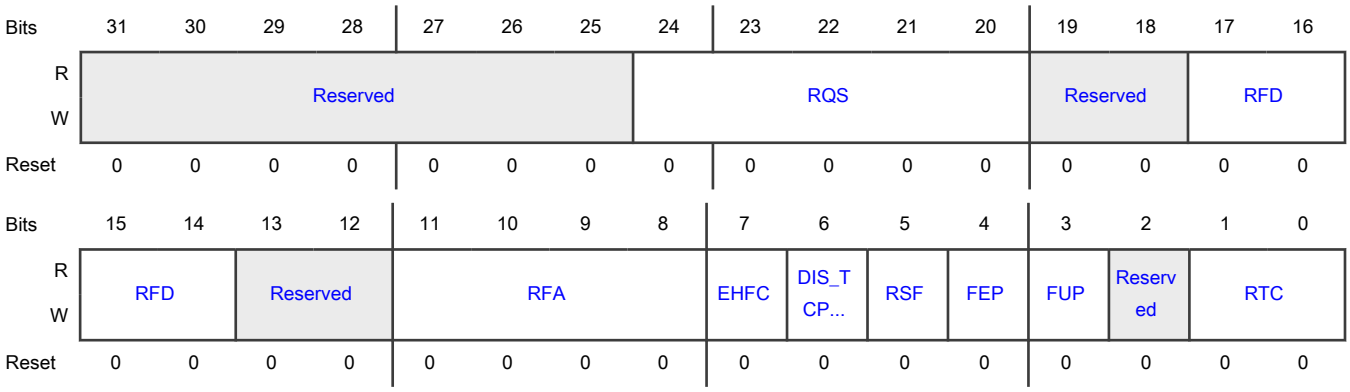
The Queue 4 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

Offset

Register	Offset
MTL_RXQ4_OPERATION_MODE	E30h

Diagram



Fields

Field	Description
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0 - Hardware Flow Control is disabled 1 - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is reset, all error packets are dropped if the FEP bit is reset. 0 - Dropping of TCP/IP Checksum Error Packets is enabled 1 - Dropping of TCP/IP Checksum Error Packets is disabled
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0 - Receive Queue Store and Forward is disabled 1 - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0 - Forward Error Packets is disabled 1 - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0 - Forward Undersized Good Packets is disabled 1 - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00 - 64 01 - 32 10 - 96 11 - 128

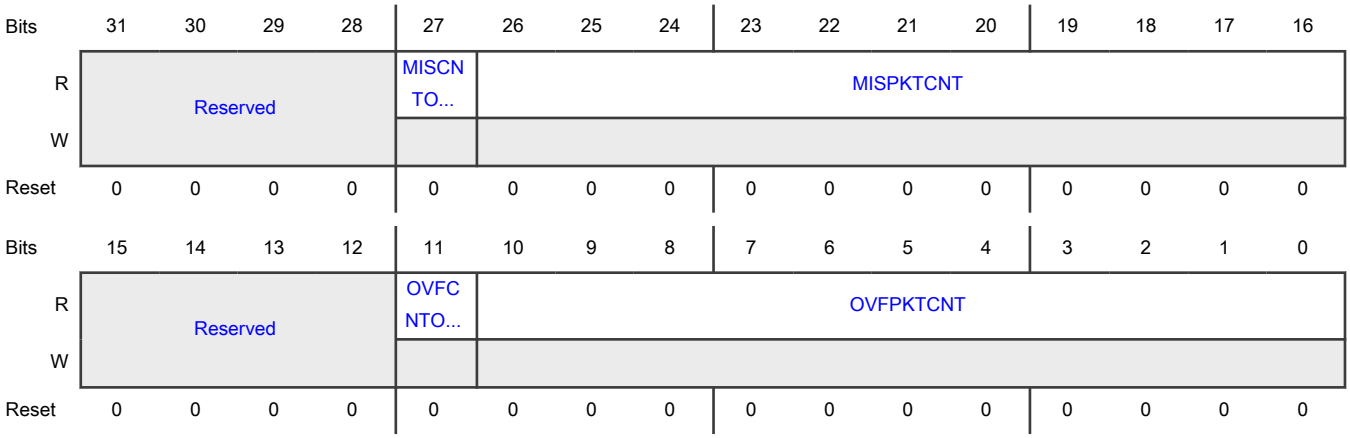
43.8.1.460 Queue 4 Missed Packet and Overflow Counter (MTL_RXQ4_MISSED_PACKET_OVERFLOW_CNT)

The Queue 4 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

Offset

Register	Offset
MTL_RXQ4_MISSED_PACKET_OVERFLOW_CNT	E34h

Diagram



Fields

Field	Description
31-28 —	Reserved
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Missed Packet Counter overflow not detected 1 - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow Counter overflow not detected 1 - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

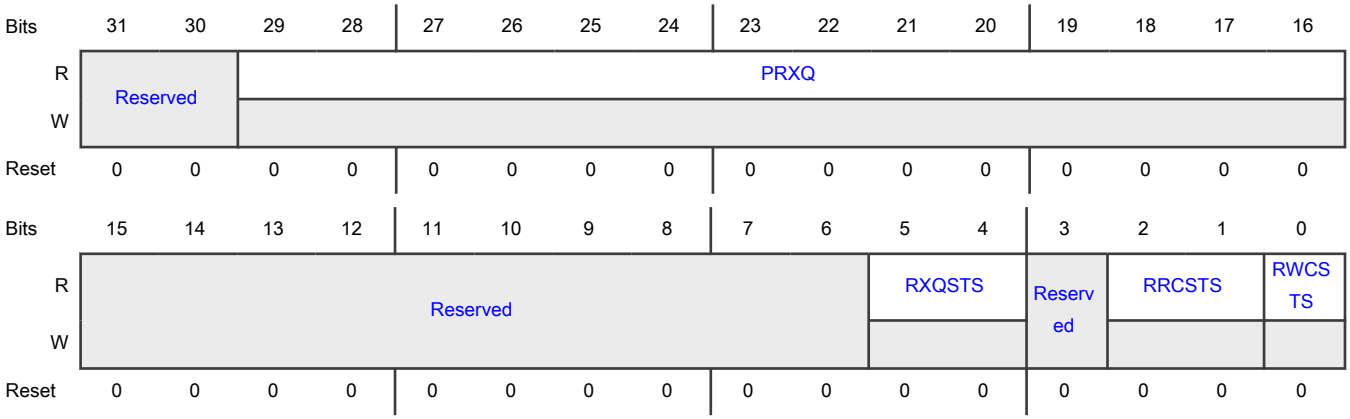
43.8.1.461 Queue 4 Receive Debug (MTL_RXQ4_DEBUG)

The Queue 4 Receive Debug register gives the debug status of various blocks related to the Receive queue.

Offset

Register	Offset
MTL_RXQ4_DEBUG	E38h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00 - Rx Queue empty 01 - Rx Queue fill-level below flow-control deactivate threshold 10 - Rx Queue fill-level above flow-control activate threshold 11 - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00 - Idle state

Table continues on the next page...

Table continued from the previous page...

Field	Description
	01 - Reading packet data 10 - Reading packet status (or timestamp) 11 - Flushing the packet data and status
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0 - MTL Rx Queue Write Controller Active Status not detected 1 - MTL Rx Queue Write Controller Active Status detected

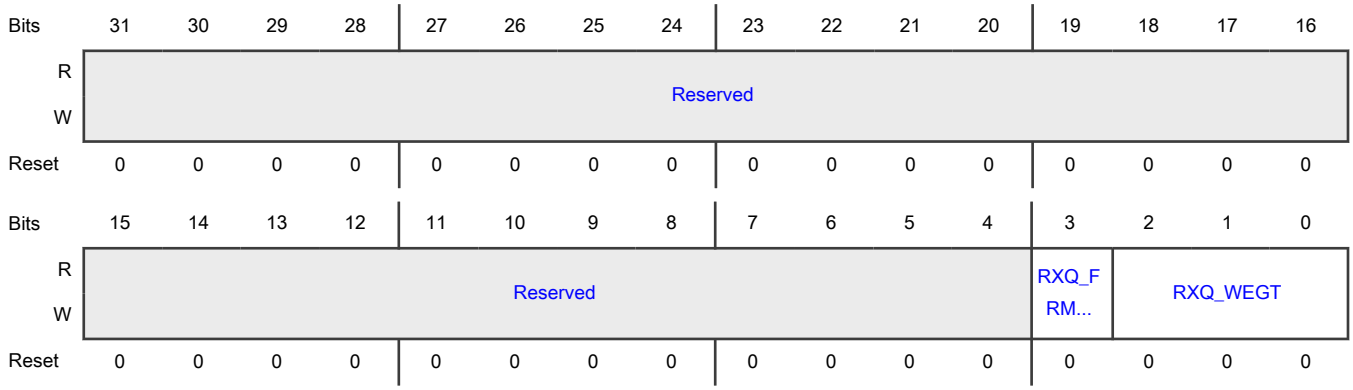
43.8.1.462 Queue 4 Receive Control (MTL_RXQ4_CONTROL)

The Queue 4 Receive Control register controls the receive arbitration and passing of received packets to the application.

Offset

Register	Offset
MTL_RXQ4_CONTROL	E3Ch

Diagram



Fields

Field	Description
31-4 —	Reserved
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0 - Receive Queue Packet Arbitration is disabled 1 - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 4. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 4. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

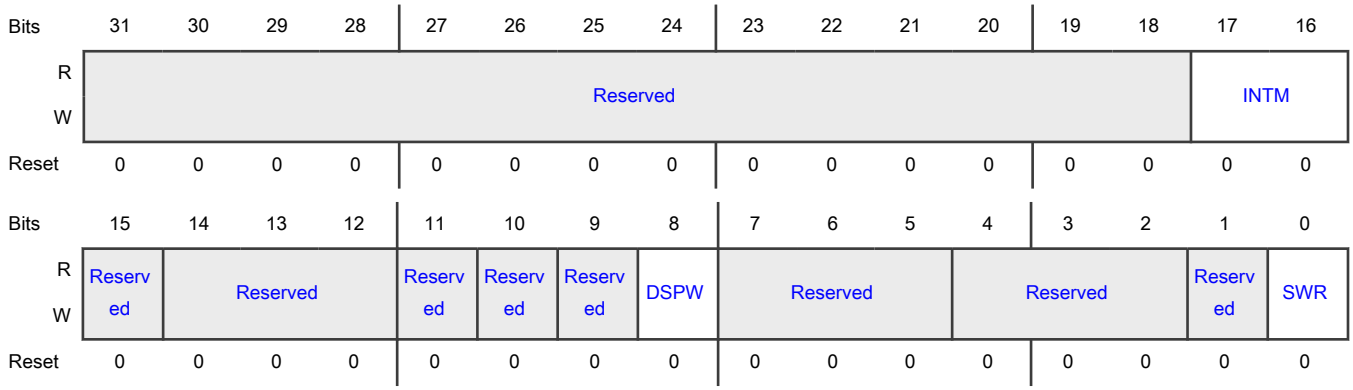
43.8.1.463 DMA Bus Mode (DMA_MODE)

The Bus Mode register establishes the bus operating modes for the DMA.

Offset

Register	Offset
DMA_MODE	1000h

Diagram



Fields

Field	Description
31-18 —	Reserved
17-16 INTM	Interrupt Mode This field defines the interrupt mode of DWC_ether_qos. Interrupt Mode This field defines the interrupt mode of DWC_ether_qos. The behavior of the following outputs changes depending on the following settings: - sbd_perch_tx_intr_o[] (Transmit Per Channel Interrupt) - sbd_perch_rx_intr_o[] (Receive Per Channel Interrupt) - sbd_intr_o (Common Interrupt) It also changes the behavior of the RI/TI bits in the STATUS. - 00: sbd_perch_* are pulse signals for each TX/RX packet transfer completion events (irrespective of whether corresponding interrupts are enabled) for which IOC bits are enabled in descriptor. sbd_intr_o is also asserted when corresponding interrupts are enabled and cleared only when software clears the corresponding RI/TI status bits. - 01: sbd_perch_* are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. The sbd_intr_o is not asserted for these TX/RX packet transfer completion events. - 10: sbd_perch_* are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. However, the signal is asserted again if the same event occurred again before it was cleared. The sbd_intr_o is not asserted for these TX/RX packet transfer completion events. - 11: Reserved For more details please refer Table "DWC_ether_qos Transfer Complete Interrupt Behavior". 00 - See above description 01 - See above description 10 - See above description 11 - Reserved
15 —	Reserved
14-12 —	Reserved
11 —	Reserved
10	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
9 —	Reserved
8 DSPW	Descriptor Posted Write When this bit is set to 0, the descriptor writes are always non-posted. Descriptor Posted Write When this bit is set to 0, the descriptor writes are always non-posted. When this bit is set to 1, the descriptor writes are non-posted only when IOC (Interrupt on completion) is set in last descriptor, otherwise the descriptor writes are always posted. 0 - Descriptor Posted Write is disabled 1 - Descriptor Posted Write is enabled
7-5 —	Reserved
4-2 —	Reserved
1 —	Reserved
0 SWR	Software Reset When this bit is set, the MAC and the DMA controller reset the logic and all internal registers of the DMA, MTL, and MAC. Software Reset When this bit is set, the MAC and the DMA controller reset the logic and all internal registers of the DMA, MTL, and MAC. This bit is automatically cleared after the reset operation is complete in all DWC_ether_qos clock domains. Before reprogramming any DWC_ether_qos register, a value of zero should be read in this bit. This bit must be read at least 4 CSR clock cycles after it is written to 1. Note: The reset operation is complete only when all resets in all active clock domains are de-asserted. Therefore, it is essential that all PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion. The time to complete the software reset operation depends on the frequency of the slowest active clock. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Software Reset is disabled 1 - Software Reset is enabled

43.8.1.464 DMA System Bus Mode (DMA_SYSBUS_MODE)

The System Bus mode register controls the behavior of the AHB or AXI master. It mainly controls burst splitting and number of outstanding requests.

Offset

Register	Offset
DMA_SYSBUS_MODE	1004h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN_LPI	LPI_XIT...	Reserved		WR_OSR_LMT				Reserved				RD_OSR_LMT			
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	ONEKBBE	AAL	Reserved	AALE	Reserved		Reserved	Reserved	Reserved	Reserved	BLEN16	BLEN8	BLEN4	FB
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 EN_LPI	Enable Low Power Interface (LPI) When set to 1, this bit enables the LPI mode supported by the EQOS-AXI configuration and accepts the LPI request from the AXI System Clock controller. Enable Low Power Interface (LPI) When set to 1, this bit enables the LPI mode supported by the EQOS-AXI configuration and accepts the LPI request from the AXI System Clock controller. When set to 0, this bit disables the LPI mode and always denies the LPI request from the AXI System Clock controller. 0 - Low Power Interface (LPI) is disabled 1 - Low Power Interface (LPI) is enabled
30 LPI_XIT_PKT	Unlock on Magic Packet or Remote Wake-Up Packet When set to 1, this bit enables the AXI master to come out of the LPI mode only when the magic packet or remote wake-up packet is received. Unlock on Magic Packet or Remote Wake-Up Packet When set to 1, this bit enables the AXI master to come out of the LPI mode only when the magic packet or remote wake-up packet is received. When set to 0, this bit enables the AXI master to come out of the LPI mode when any packet is received. 0 - Unlock on Magic Packet or Remote Wake-Up Packet is disabled 1 - Unlock on Magic Packet or Remote Wake-Up Packet is enabled
29-28 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
27-24 WR_OSR_LMT	AXI Maximum Write Outstanding Request Limit This value limits the maximum outstanding request on the AXI write interface. AXI Maximum Write Outstanding Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT + 1 Note: - Bit 26 is reserved if DWC_ETHER_QOS_AXI_MAX_WR_REQ = 4 - Bit 27 is reserved if DWC_ETHER_QOS_AXI_MAX_WR_REQ!= 16
23-20 —	Reserved
19-16 RD_OSR_LMT	AXI Maximum Read Outstanding Request Limit This value limits the maximum outstanding request on the AXI read interface. AXI Maximum Read Outstanding Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT + 1 Note: - Bit 18 is reserved if parameter DWC_ETHER_QOS_AXI_MAX_RD_REQ = 4 - Bit 19 is reserved if parameter DWC_ETHER_QOS_AXI_MAX_RD_REQ!= 16
15 —	Reserved
14 —	Reserved
13 ONEKBBE	1 KB Boundary Crossing Enable for the EQOS-AXI Master When set, the burst transfers performed by the EQOS-AXI master do not cross 1 KB boundary. 1 KB Boundary Crossing Enable for the EQOS-AXI Master When set, the burst transfers performed by the EQOS-AXI master do not cross 1 KB boundary. When reset, the burst transfers performed by the EQOS-AXI master do not cross 4 KB boundary. 0 - 1 KB Boundary Crossing for the EQOS-AXI Master Beats is disabled 1 - 1 KB Boundary Crossing for the EQOS-AXI Master Beats is enabled
12 AAL	Address-Aligned Beats When this bit is set to 1, the EQOS-AXI or EQOS-AHB master performs address-aligned burst transfers on Read and Write channels. 0 - Address-Aligned Beats is disabled 1 - Address-Aligned Beats is enabled
11	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
10 AALE	Automatic AXI LPI enable When set to 1, enables the AXI master to enter into LPI state when there is no activity in the DWC_ether_qos for number of system clock cycles programmed in the LPIEI field of DMA_AXI_LPI_ENTRY_INTERVAL register. 0 - Automatic AXI LPI is disabled 1 - Automatic AXI LPI is enabled
9-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 BLEN16	AXI Burst Length 16 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 16 on the AXI interface. AXI Burst Length 16 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 16 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. 0 - No effect 1 - AXI Burst Length 16
2 BLEN8	AXI Burst Length 8 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 8 on the AXI interface. AXI Burst Length 8 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 8 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. 0 - No effect 1 - AXI Burst Length 8

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 BLEN4	AXI Burst Length 4 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 4 on the AXI interface. AXI Burst Length 4 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 4 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. 0 - No effect 1 - AXI Burst Length 4
0 FB	Fixed Burst Length When this bit is set to 1, the EQOS-AXI master initiates burst transfers of specified lengths as given below. Fixed Burst Length When this bit is set to 1, the EQOS-AXI master initiates burst transfers of specified lengths as given below. - Burst transfers of fixed burst lengths as indicated by the BLEN256, BLEN128, BLEN64, BLEN32, BLEN16, BLEN8, or BLEN4 field - Burst transfers of length 1 When this bit is set to 0, the EQOS-AXI master initiates burst transfers that are equal to or less than the maximum allowed burst length programmed in Bits[7:1]. 0 - Fixed Burst Length is disabled 1 - Fixed Burst Length is enabled

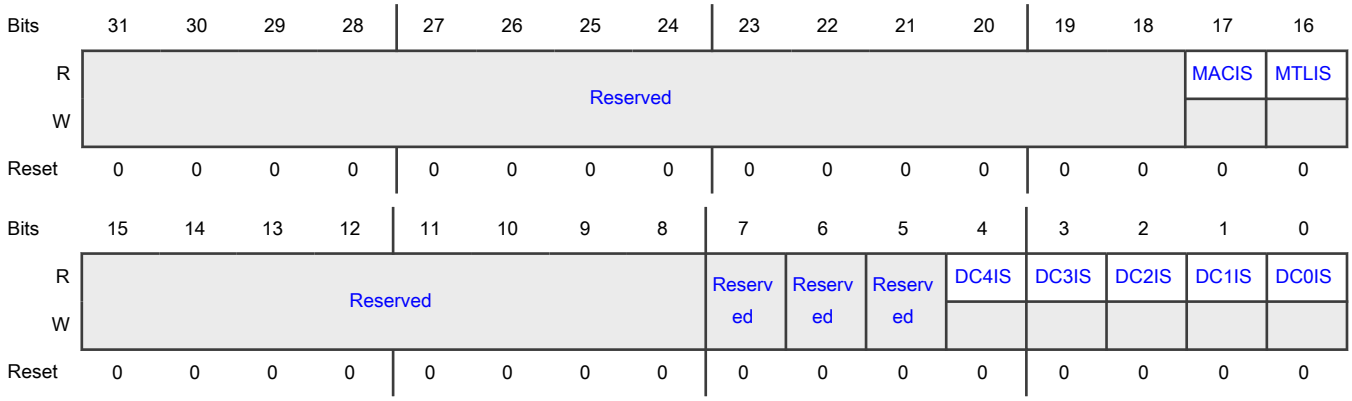
43.8.1.465 DMA Interrupt Status (DMA_INTERRUPT_STATUS)

The application reads this Interrupt Status register during interrupt service routine or polling to determine the interrupt status of DMA channels, MTL queues, and the MAC.

Offset

Register	Offset
DMA_INTERRUPT_STATUS	1008h

Diagram



Fields

Field	Description
31-18 —	Reserved
17 MACIS	MAC Interrupt Status This bit indicates an interrupt event in the MAC. MAC Interrupt Status This bit indicates an interrupt event in the MAC. To reset this bit to 1'b0, the software must read the corresponding register in the MAC to get the exact cause of the interrupt and clear its source. 0 - MAC Interrupt Status not detected 1 - MAC Interrupt Status detected
16 MTLIS	MTL Interrupt Status This bit indicates an interrupt event in the MTL. MTL Interrupt Status This bit indicates an interrupt event in the MTL. To reset this bit to 1'b0, the software must read the corresponding register in the MTL to get the exact cause of the interrupt and clear its source. 0 - MTL Interrupt Status not detected 1 - MTL Interrupt Status detected
15-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 DC4IS	DMA Channel 4 Interrupt Status This bit indicates an interrupt event in DMA Channel 4. DMA Channel 4 Interrupt Status This bit indicates an interrupt event in DMA Channel 4. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 4 to get the exact cause of the interrupt and clear its source. 0 - DMA Channel 4 Interrupt Status not detected 1 - DMA Channel 4 Interrupt Status detected
3 DC3IS	DMA Channel 3 Interrupt Status This bit indicates an interrupt event in DMA Channel 3. DMA Channel 3 Interrupt Status This bit indicates an interrupt event in DMA Channel 3. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 3 to get the exact cause of the interrupt and clear its source. 0 - DMA Channel 3 Interrupt Status not detected 1 - DMA Channel 3 Interrupt Status detected
2 DC2IS	DMA Channel 2 Interrupt Status This bit indicates an interrupt event in DMA Channel 2. DMA Channel 2 Interrupt Status This bit indicates an interrupt event in DMA Channel 2. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 2 to get the exact cause of the interrupt and clear its source. 0 - DMA Channel 2 Interrupt Status not detected 1 - DMA Channel 2 Interrupt Status detected
1 DC1IS	DMA Channel 1 Interrupt Status This bit indicates an interrupt event in DMA Channel 1. DMA Channel 1 Interrupt Status This bit indicates an interrupt event in DMA Channel 1. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 1 to get the exact cause of the interrupt and clear its source. 0 - DMA Channel 1 Interrupt Status not detected 1 - DMA Channel 1 Interrupt Status detected
0 DC0IS	DMA Channel 0 Interrupt Status This bit indicates an interrupt event in DMA Channel 0. DMA Channel 0 Interrupt Status This bit indicates an interrupt event in DMA Channel 0. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 0 to get the exact cause of the interrupt and clear its source. 0 - DMA Channel 0 Interrupt Status not detected 1 - DMA Channel 0 Interrupt Status detected

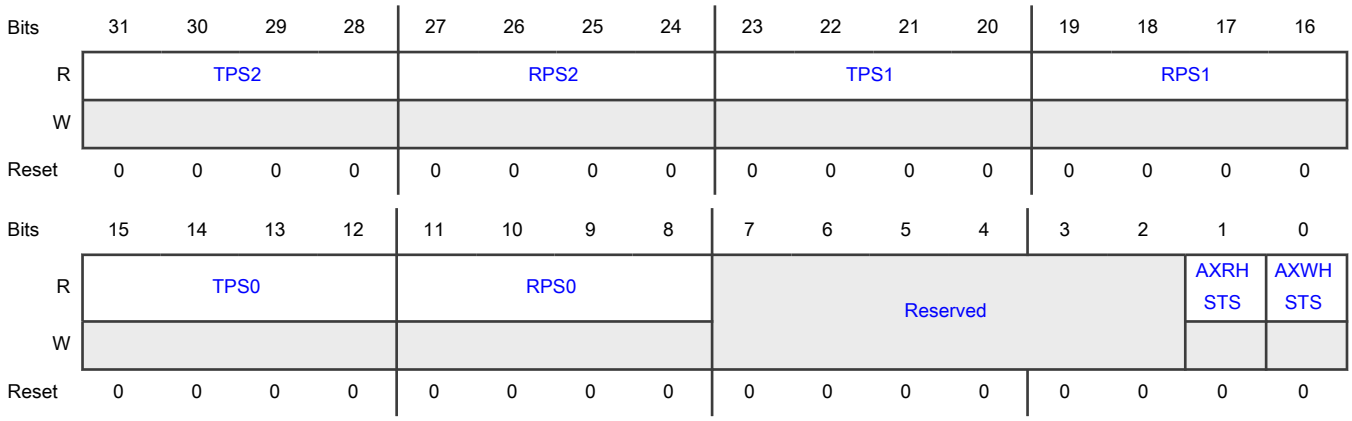
43.8.1.466 DMA Debug Status 0 (DMA_DEBUG_STATUS0)

The Debug Status 0 register gives the Receive and Transmit process status for DMA Channel 0-Channel 2 for debugging purpose.

Offset

Register	Offset
DMA_DEBUG_STATUS0	100Ch

Diagram



Fields

Field	Description
31-28 TPS2	DMA Channel 2 Transmit Process State This field indicates the Tx DMA FSM state for Channel 2. DMA Channel 2 Transmit Process State This field indicates the Tx DMA FSM state for Channel 2. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Transmit Command issued) 0001 - Running (Fetching Tx Transfer Descriptor) 0010 - Running (Waiting for status) 0011 - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100 - Timestamp write state 0101 - Reserved for future use 0110 - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111 - Running (Closing Tx Descriptor)
27-24 RPS2	DMA Channel 2 Receive Process State This field indicates the Rx DMA FSM state for Channel 2. DMA Channel 2 Receive Process State This field indicates the Rx DMA FSM state for Channel 2. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Receive Command issued)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0001 - Running (Fetching Rx Transfer Descriptor) 0010 - Reserved for future use 0011 - Running (Waiting for Rx packet) 0100 - Suspended (Rx Descriptor Unavailable) 0101 - Running (Closing the Rx Descriptor) 0110 - Timestamp write state 0111 - Running (Transferring the received packet data from the Rx buffer to the system memory)
23-20 TPS1	DMA Channel 1 Transmit Process State This field indicates the Tx DMA FSM state for Channel 1. DMA Channel 1 Transmit Process State This field indicates the Tx DMA FSM state for Channel 1. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Transmit Command issued) 0001 - Running (Fetching Tx Transfer Descriptor) 0010 - Running (Waiting for status) 0011 - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100 - Timestamp write state 0101 - Reserved for future use 0110 - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111 - Running (Closing Tx Descriptor)
19-16 RPS1	DMA Channel 1 Receive Process State This field indicates the Rx DMA FSM state for Channel 1. DMA Channel 1 Receive Process State This field indicates the Rx DMA FSM state for Channel 1. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Receive Command issued) 0001 - Running (Fetching Rx Transfer Descriptor) 0010 - Reserved for future use 0011 - Running (Waiting for Rx packet) 0100 - Suspended (Rx Descriptor Unavailable) 0101 - Running (Closing the Rx Descriptor) 0110 - Timestamp write state 0111 - Running (Transferring the received packet data from the Rx buffer to the system memory)
15-12 TPS0	DMA Channel 0 Transmit Process State This field indicates the Tx DMA FSM state for Channel 0. DMA Channel 0 Transmit Process State

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field indicates the Tx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Transmit Command issued) 0001 - Running (Fetching Tx Transfer Descriptor) 0010 - Running (Waiting for status) 0011 - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100 - Timestamp write state 0101 - Reserved for future use 0110 - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111 - Running (Closing Tx Descriptor)
11-8 RPS0	DMA Channel 0 Receive Process State This field indicates the Rx DMA FSM state for Channel 0. DMA Channel 0 Receive Process State This field indicates the Rx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Receive Command issued) 0001 - Running (Fetching Rx Transfer Descriptor) 0010 - Reserved for future use 0011 - Running (Waiting for Rx packet) 0100 - Suspended (Rx Descriptor Unavailable) 0101 - Running (Closing the Rx Descriptor) 0110 - Timestamp write state 0111 - Running (Transferring the received packet data from the Rx buffer to the system memory)
7-2 —	Reserved
1 AXRHSTS	AXI Master Read Channel Status When high, this bit indicates that the read channel of the AXI master is active, and it is transferring the data. 0 - AXI Master Read Channel Status not detected 1 - AXI Master Read Channel Status detected
0 AXWHSTS	AXI Master Write Channel When high, this bit indicates that the write channel of the AXI master is active, and it is transferring data. 0 - AXI Master Write Channel or AHB Master Status not detected 1 - AXI Master Write Channel or AHB Master Status detected

43.8.1.467 DMA Debug Status 1 (DMA_DEBUG_STATUS1)

The Debug Status1 register gives the Receive and Transmit process status for DMA Channel 3-Channel 6.

Offset

Register	Offset
DMA_DEBUG_STATUS1	1010h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				Reserved				Reserved				Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TPS4				RPS4				TPS3				RPS3			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-24 —	Reserved
23-20 —	Reserved
19-16 —	Reserved
15-12 TPS4	DMA Channel 4 Transmit Process State This field indicates the Tx DMA FSM state for Channel 4. DMA Channel 4 Transmit Process State This field indicates the Tx DMA FSM state for Channel 4. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Transmit Command issued) 0001 - Running (Fetching Tx Transfer Descriptor)

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0010 - Running (Waiting for status) 0011 - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100 - Timestamp write state 0101 - Reserved for future use 0110 - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111 - Running (Closing Tx Descriptor)
11-8 RPS4	DMA Channel 4 Receive Process State This field indicates the Rx DMA FSM state for Channel 4. DMA Channel 4 Receive Process State This field indicates the Rx DMA FSM state for Channel 4. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Receive Command issued) 0001 - Running (Fetching Rx Transfer Descriptor) 0010 - Reserved for future use 0011 - Running (Waiting for Rx packet) 0100 - Suspended (Rx Descriptor Unavailable) 0101 - Running (Closing the Rx Descriptor) 0110 - Timestamp write state 0111 - Running (Transferring the received packet data from the Rx buffer to the system memory)
7-4 TPS3	DMA Channel 3 Transmit Process State This field indicates the Tx DMA FSM state for Channel 3. DMA Channel 3 Transmit Process State This field indicates the Tx DMA FSM state for Channel 3. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Transmit Command issued) 0001 - Running (Fetching Tx Transfer Descriptor) 0010 - Running (Waiting for status) 0011 - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100 - Timestamp write state 0101 - Reserved for future use 0110 - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111 - Running (Closing Tx Descriptor)
3-0 RPS3	DMA Channel 3 Receive Process State This field indicates the Rx DMA FSM state for Channel 3. DMA Channel 3 Receive Process State This field indicates the Rx DMA FSM state for Channel 3.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Receive Command issued) 0001 - Running (Fetching Rx Transfer Descriptor) 0010 - Reserved for future use 0011 - Running (Waiting for Rx packet) 0100 - Suspended (Rx Descriptor Unavailable) 0101 - Running (Closing the Rx Descriptor) 0110 - Timestamp write state 0111 - Running (Transferring the received packet data from the Rx buffer to the system memory)

43.8.1.468 AXI LPI Entry Interval Control (DMA_AXI_LPI_ENTRY_INTERVAL)

This register is used to control the AXI LPI entry interval.

Offset

Register	Offset
DMA_AXI_LPI_ENTRY_INTERVAL	1040h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												LPIEI			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
3-0 LPIEI	LPI Entry Interval Contains the number of system clock cycles, multiplied by 64, to wait for an activity in the DWC_ether_qos to enter into the AXI low power state 0 indicates 64 clock cycles

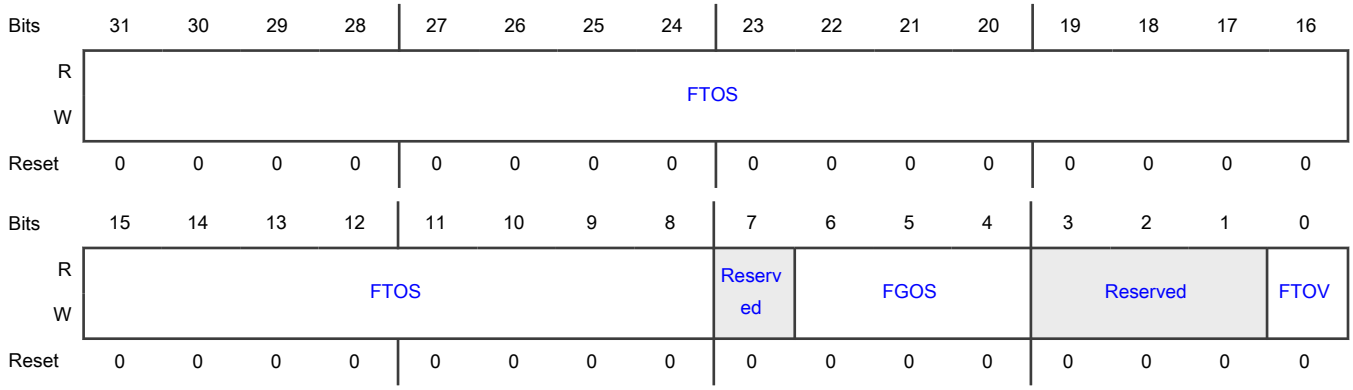
43.8.1.469 DMA_TBS_CTRL0 (DMA_TBS_CTRL0)

This register is used to control the TBS attributes.

Offset

Register	Offset
DMA_TBS_CTRL0	1050h

Diagram



Fields

Field	Description
31-8 FTOS	Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR-1 value when ESTM mode is set since this value is a modulo CTR value.
7 —	Reserved
6-4 FGOS	Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.
3-1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
0 FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0 - Fetch Time Offset is invalid 1 - Fetch Time Offset is valid

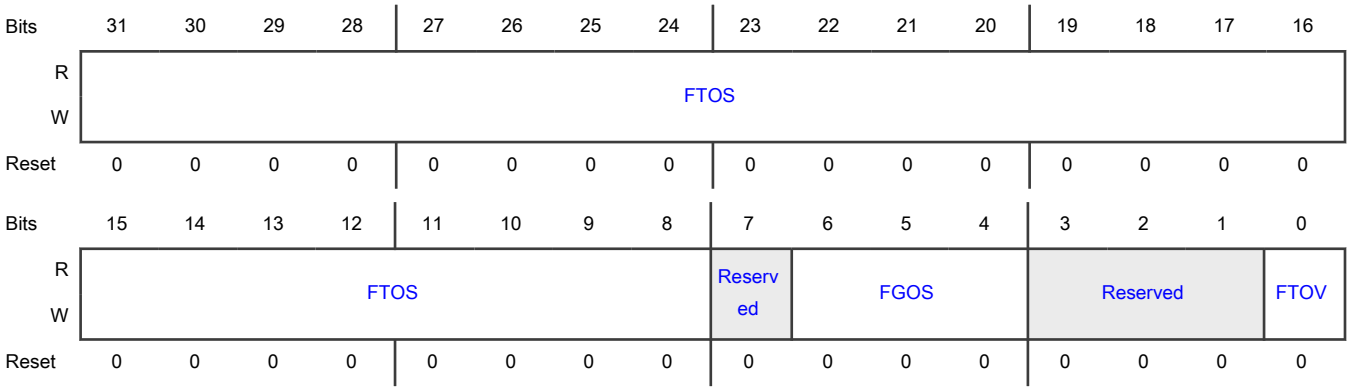
43.8.1.470 DMA_TBS_CTRL1 (DMA_TBS_CTRL1)

This register is used to control the TBS attributes.

Offset

Register	Offset
DMA_TBS_CTRL1	1054h

Diagram



Fields

Field	Description
31-8 FTOS	Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR-1 value when ESTM mode is set since this value is a modulo CTR value.
7 —	Reserved
6-4	Fetch GSN Offset

Table continues on the next page...

Table continued from the previous page...

Field	Description
FGOS	The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.
3-1 —	Reserved
0 FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0 - Fetch Time Offset is invalid 1 - Fetch Time Offset is valid

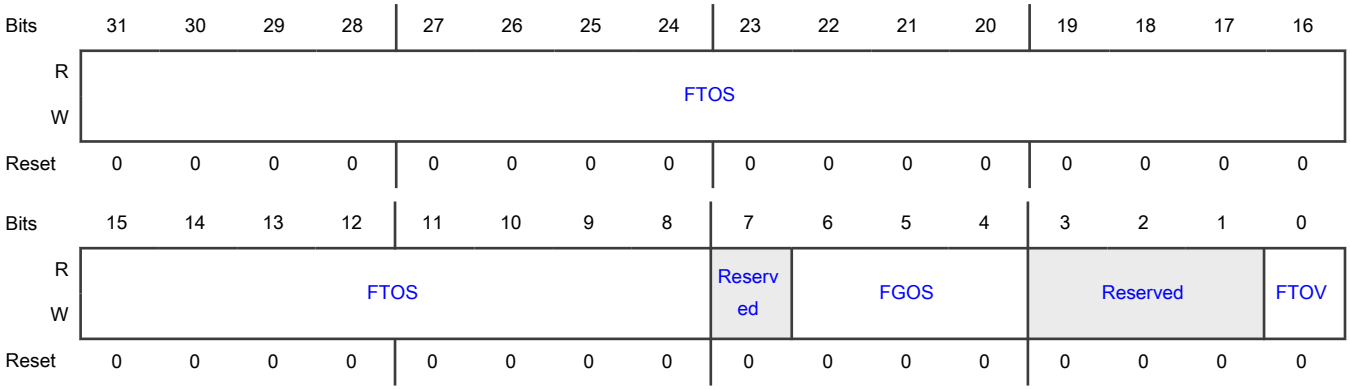
43.8.1.471 DMA_TBS_CTRL2 (DMA_TBS_CTRL2)

This register is used to control the TBS attributes.

Offset

Register	Offset
DMA_TBS_CTRL2	1058h

Diagram



Fields

Field	Description
31-8 FTOS	Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR-1 value when ESTM mode is set since this value is a modulo CTR value.

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 —	Reserved
6-4 FGOS	Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.
3-1 —	Reserved
0 FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0 - Fetch Time Offset is invalid 1 - Fetch Time Offset is valid

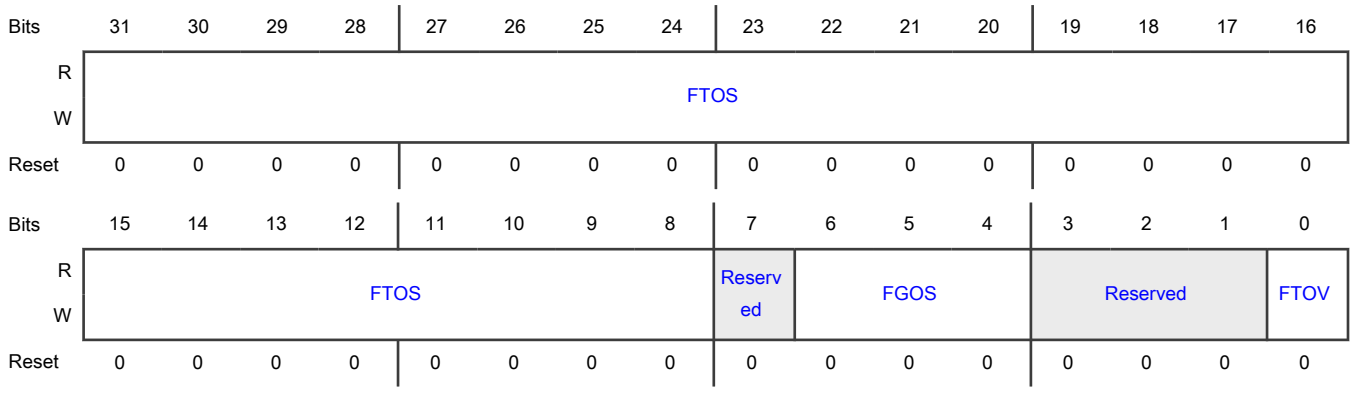
43.8.1.472 DMA_TBS_CTRL3 (DMA_TBS_CTRL3)

This register is used to control the TBS attributes.

Offset

Register	Offset
DMA_TBS_CTRL3	105Ch

Diagram



Fields

Field	Description
31-8 FTOS	Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR-1 value when ESTM mode is set since this value is a modulo CTR value.
7 —	Reserved
6-4 FGOS	Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.
3-1 —	Reserved
0 FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0 - Fetch Time Offset is invalid 1 - Fetch Time Offset is valid

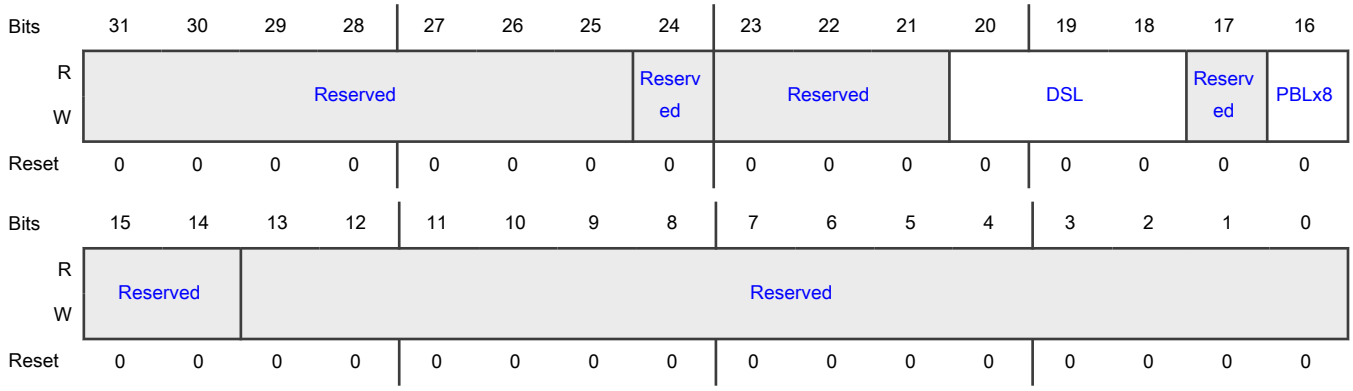
43.8.1.473 DMA Channel 0 Control (DMA_CH0_CONTROL)

The DMA Channel 0 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

Offset

Register	Offset
DMA_CH0_CONTROL	1100h

Diagram



Fields

Field	Description
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH0_TX_CONTROL and Bits[21:16] in DMA_CH0_RX_CONTROL is multiplied by eight times. 8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH0_TX_CONTROL and Bits[21:16] in DMA_CH0_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0 - 8xPBL mode is disabled 1 - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

43.8.1.474 DMA Channel 0 Transmit Control (DMA_CH0_TX_CONTROL)

The DMA Channel 0 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

Offset

Register	Offset
DMA_CH0_TX_CONTROL	1104h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			EDSE	Reserved				Reserved	Reserved	TxPBL					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IPBL	Reserved		Reserved	Reserved							OSF	Reserved			ST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0 - Enhanced Descriptor is disabled 1 - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in CONTROL register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ0_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0 - Ignore PBL Requirement is disabled 1 - Ignore PBL Requirement is enabled
14-13 —	Reserved
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0 - Operate on Second Packet disabled 1 - Operate on Second Packet enabled
3-1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH0_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH0_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0 - Stop Transmission Command 1 - Start Transmission Command

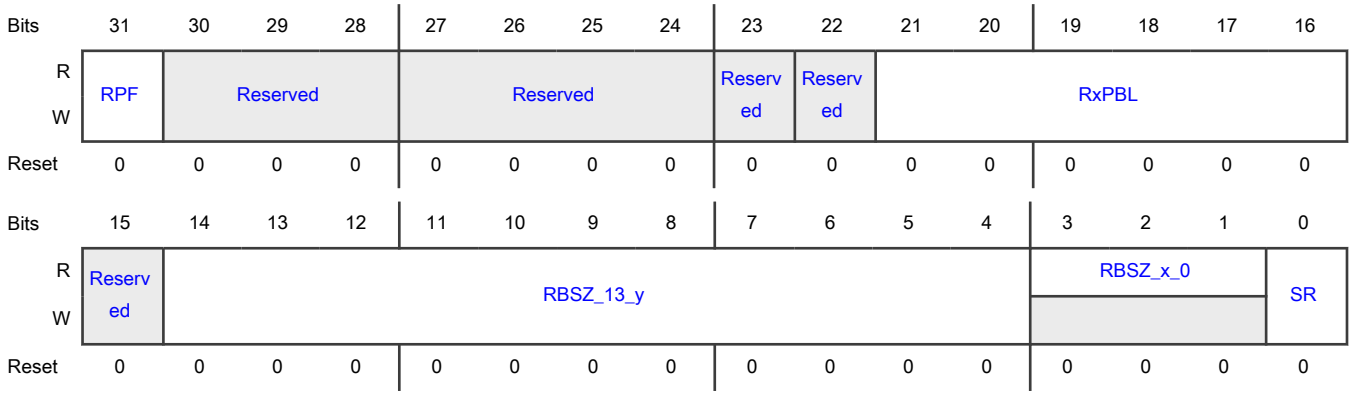
43.8.1.475 DMA Channel 0 Receive Control (DMA_CH0_RX_CONTROL)

The DMA Channel 0 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

Offset

Register	Offset
DMA_CH0_RX_CONTROL	1108h

Diagram



Fields

Field	Description
31 RPF	Rx Packet Flush. Rx Packet Flush. When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0 - Rx Packet Flush is disabled 1 - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application

Table continues on the next page...

Table continued from the previous page...

Field	Description
	bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the DMA_CH0_CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ0_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1 RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH0_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH0_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH0_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0 - Stop Receive 1 - Start Receive

43.8.1.476 Channel 0 Tx Descriptor List Address register (DMA_CH0_TXDESC_LIST_ADDRESS)

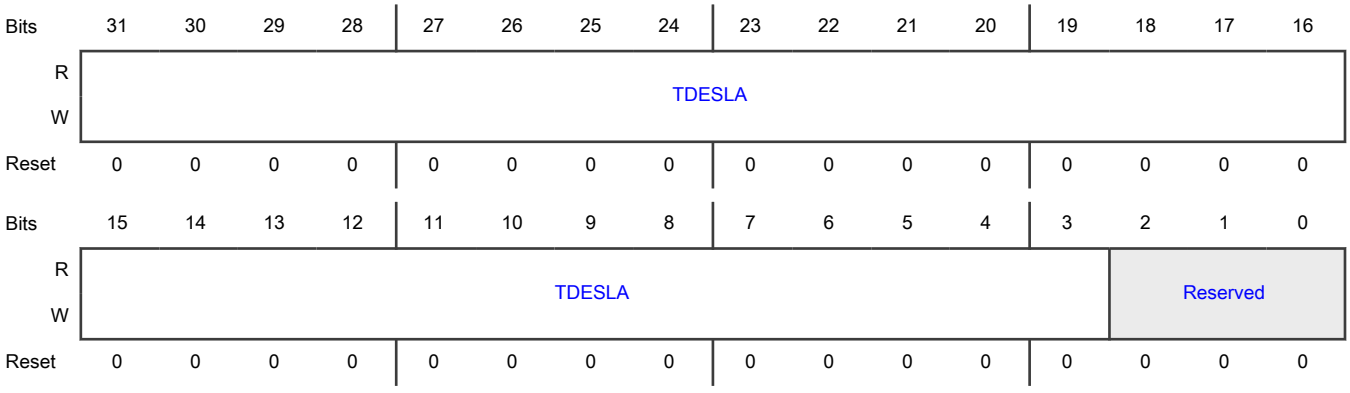
The Channel 0 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

Offset

Register	Offset
DMA_CH0_TXDESC_LIST_ADDRESS	1114h

Diagram



Fields

Field	Description
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

43.8.1.477 Channel 0 Rx Descriptor List Address register (DMA_CH0_RXDESC_LIST_ADDRESS)

The Channel 0 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

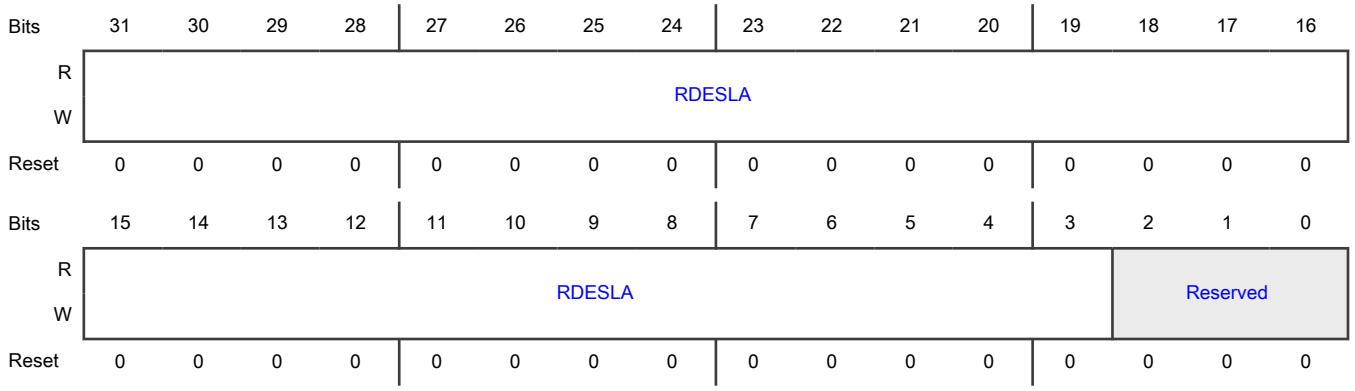
This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

Offset

Register	Offset
DMA_CH0_RXDESC_LIST_ADDRESS	111Ch

Diagram



Fields

Field	Description
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

43.8.1.478 Channel 0 Tx Descriptor Tail Pointer (DMA_CH0_TXDESC_TAIL_POINTER)

The Channel 0 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

Offset

Register	Offset
DMA_CH0_TXDESC_TAIL_POINTER	1120h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDTP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDTP													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

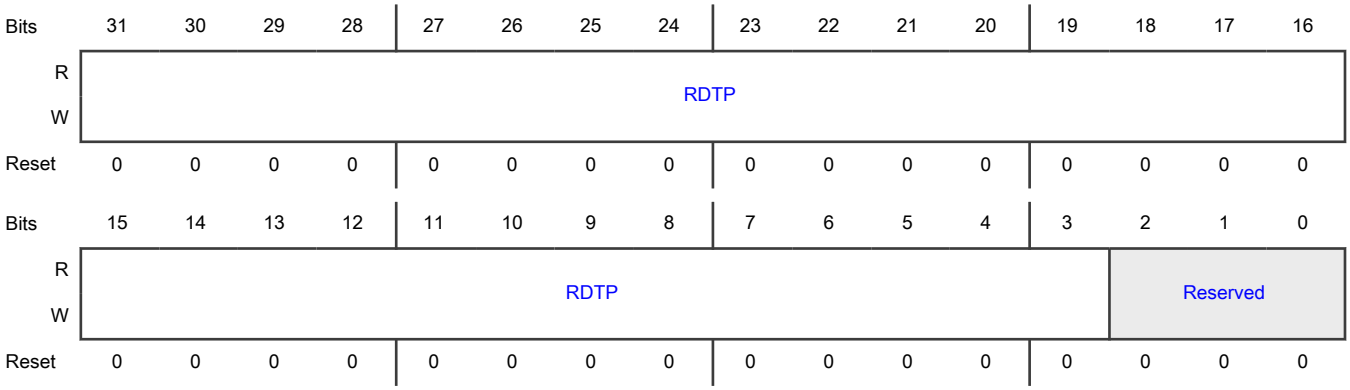
43.8.1.479 Channel 0 Rx Descriptor Tail Pointer (DMA_CH0_RXDESC_TAIL_POINTER)

The Channel 0 Rx Descriptor Tail Pointer points to an offset from the base and indicates the location of the last valid descriptor.

Offset

Register	Offset
DMA_CH0_RXDESC_TAIL_POINTER	1128h

Diagram



Fields

Field	Description
31-3 RDTP	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

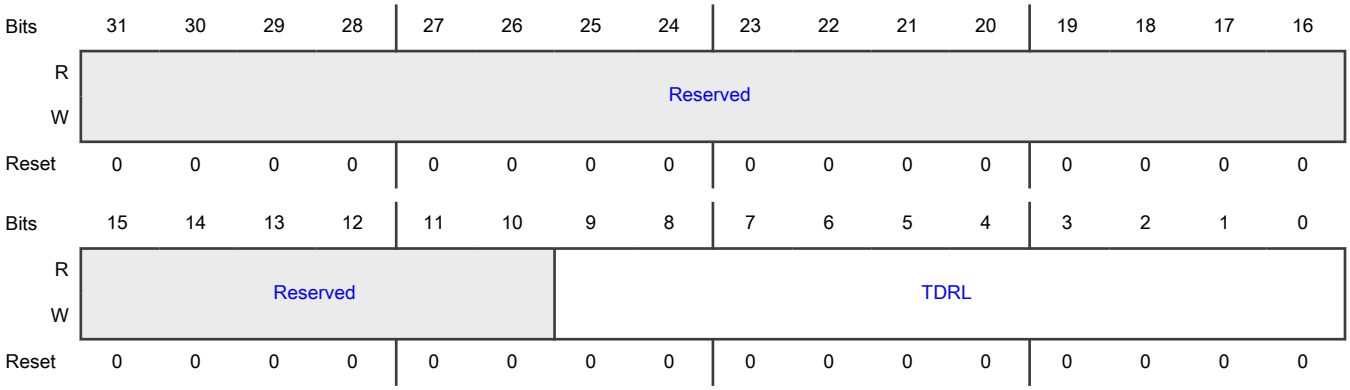
43.8.1.480 Channel 0 Tx Descriptor Ring Length (DMA_CH0_TXDESC_RING_LENGTH)

The Channel 0 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

Offset

Register	Offset
DMA_CH0_TXDESC_RING_LENGTH	112Ch

Diagram



Fields

Field	Description
31-10	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. We recommend a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

43.8.1.481 Channel 0 Receive Control 2 register (DMA_CH0_RX_CONTROL2)

The Channel 0 Receive Control register controls Rx Descriptor Ring Length and Alternate Rx Buffer Size.

Offset

Register	Offset
DMA_CH0_RX_CONTROL2	1130h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								ARBS							Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RDRL							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23-17 ARBS	Alternate Receive Buffer Size Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header

Table continues on the next page...

Table continued from the previous page...

Field	Description
	data. The maximum alternate buffer is limited to 1020, 1016 or 1008-bytes depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). When ARBS=0, Rx Buffer1 and Rx Buffer2 sizes are based on DMA_CH0_RX_CONTROL[RBSZ]. Width of ARBS field is 8, 7 or 6-bits depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively).
16-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

43.8.1.482 Channel 0 Interrupt Enable (DMA_CH0_INTERRUPT_ENABLE)

The Channel 0 Interrupt Enable register enables the interrupts reported by the Status register.

Offset

Register	Offset
DMA_CH0_INTERRUPT_ENABLE	1134h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	Reserved			TBUE	TXSE	TIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0 - Normal Interrupt Summary is disabled 1 - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled. 0 - Abnormal Interrupt Summary is disabled 1 - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0 - Context Descriptor Error is disabled 1 - Context Descriptor Error is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0 - Fatal Bus Error is disabled 1 - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0 - Early Receive Interrupt is disabled 1 - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0 - Early Transmit Interrupt is disabled 1 - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0 - Receive Watchdog Timeout is disabled 1 - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0 - Receive Stopped is disabled 1 - Receive Stopped is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0 - Receive Buffer Unavailable is disabled 1 - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0 - Receive Interrupt is disabled 1 - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0 - Transmit Buffer Unavailable is disabled 1 - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0 - Transmit Stopped is disabled 1 - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0 - Transmit Interrupt is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Transmit Interrupt is enabled

43.8.1.483 Channel 0 Receive Interrupt Watchdog Timer (DMA_CH0_RX_INTERRUPT_WATCHDOG_TIMER)

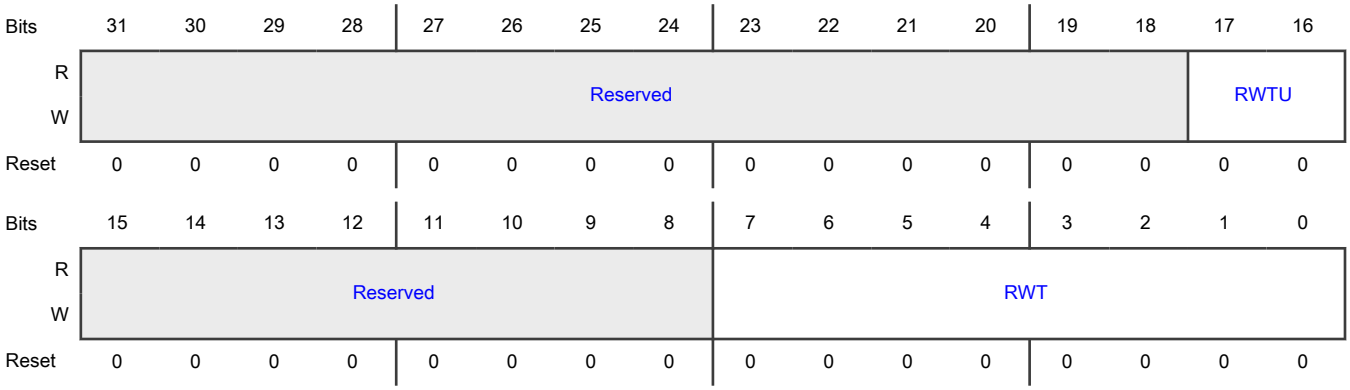
The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA.

The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA_CH0_Status register.

Offset

Register	Offset
DMA_CH0_RX_INTERRUPT_WATCHDOG_TIMER	1138h

Diagram



Fields

Field	Description
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- 2'b10: 1024 - 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8 —	Reserved
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

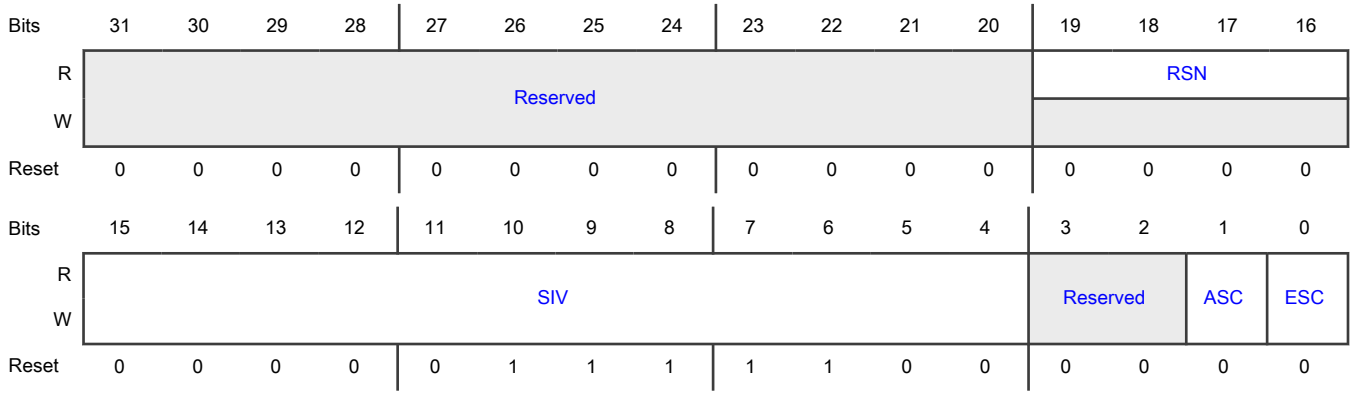
43.8.1.484 Channel 0 Slot Function Control and Status (DMA_CH0_SLOT_FUNCTION_CONTROL_STATUS)

The Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

Offset

Register	Offset
DMA_CH0_SLOT_FUNCTION_CONTROL_STATUS	113Ch

Diagram



Fields

Field	Description
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0 - Advance Slot Check is disabled 1 - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or - ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0 - Slot Comparison is disabled 1 - Slot Comparison is enabled

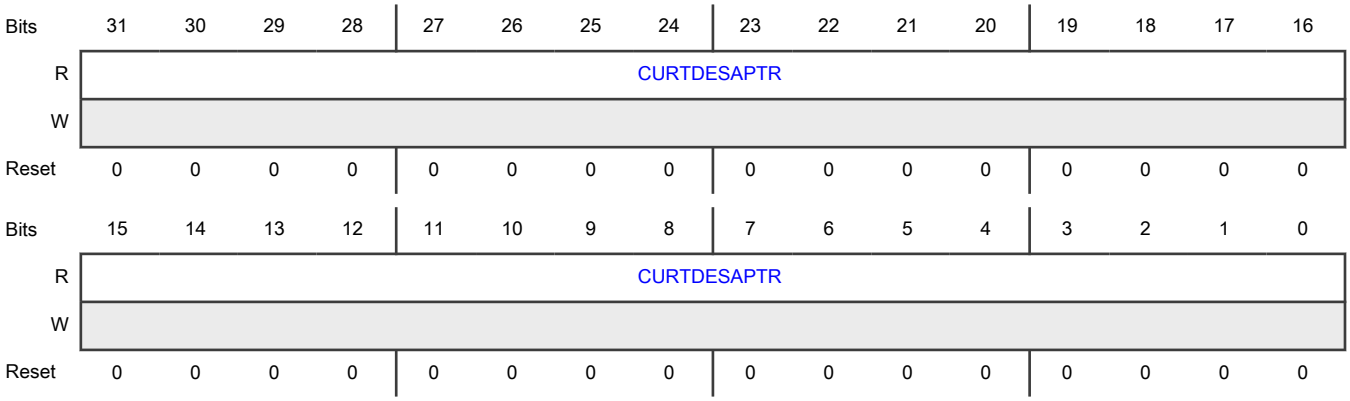
43.8.1.485 Channel 0 Current Application Transmit Descriptor (DMA_CH0_CURRENT_APP_TXDESC)

The Channel 0 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

Offset

Register	Offset
DMA_CH0_CURRENT_APP_TXDESC	1144h

Diagram



Fields

Field	Description
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

43.8.1.486 Channel 0 Current Application Receive Descriptor (DMA_CH0_CURRENT_APP_RXDESC)

The Channel 0 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

Offset

Register	Offset
DMA_CH0_CURRENT_APP_RXDESC	114Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CURRDESAPTR	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

43.8.1.487 Channel 0 Current Application Transmit Buffer Address (DMA_CH0_CURRENT_APP_TXBUFFER)

The Channel 0 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

Offset

Register	Offset
DMA_CH0_CURRENT_APP_TXBUFFER	1154h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

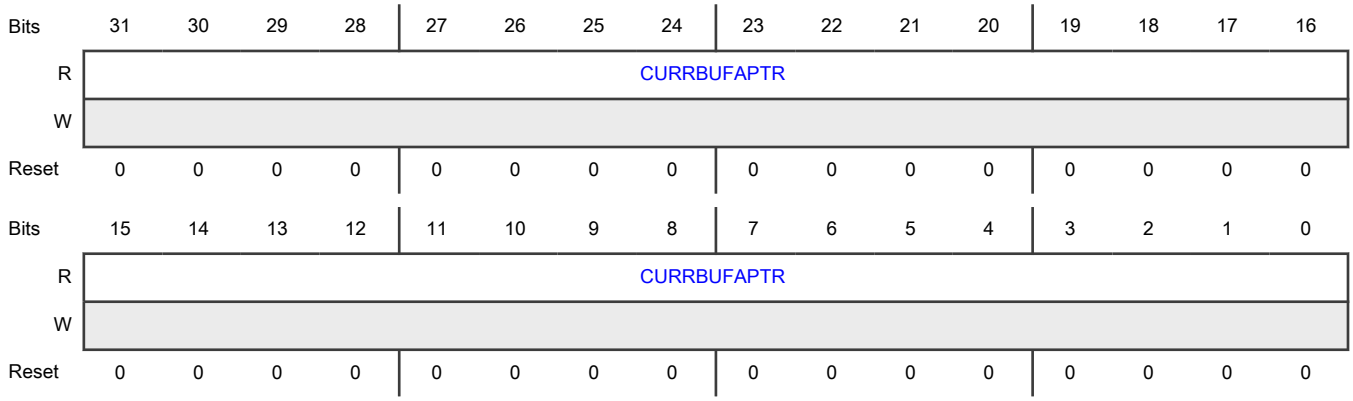
43.8.1.488 Channel 0 Current Application Receive Buffer Address (DMA_CH0_CURRENT_APP_RXBUFFER)

The Channel 0 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

Offset

Register	Offset
DMA_CH0_CURRENT_APP_RXBUFFER	115Ch

Diagram



Fields

Field	Description
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

43.8.1.489 DMA Channel 0 Status (DMA_CH0_STATUS)

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

Offset

Register	Offset
DMA_CH0_STATUS	1160h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB		TEB			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH0_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH0_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Normal Interrupt Summary status not detected 1 - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH0_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Abnormal Interrupt Summary status not detected 1 - Abnormal Interrupt Summary status detected
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all ones descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all ones descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Context Descriptor Error status not detected 1 - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Fatal Bus Error status not detected 1 - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Receive Interrupt status not detected 1 - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Transmit Interrupt status not detected 1 - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0 - Receive Watchdog Timeout status not detected 1 - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Process Stopped status not detected 1 - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode,

Table continues on the next page...

Table continued from the previous page...

Field	Description
	the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Buffer Unavailable status not detected 1 - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Interrupt status not detected 1 - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Buffer Unavailable status not detected 1 - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Process Stopped status not detected 1 - Transmit Process Stopped status detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Interrupt status not detected 1 - Transmit Interrupt status detected

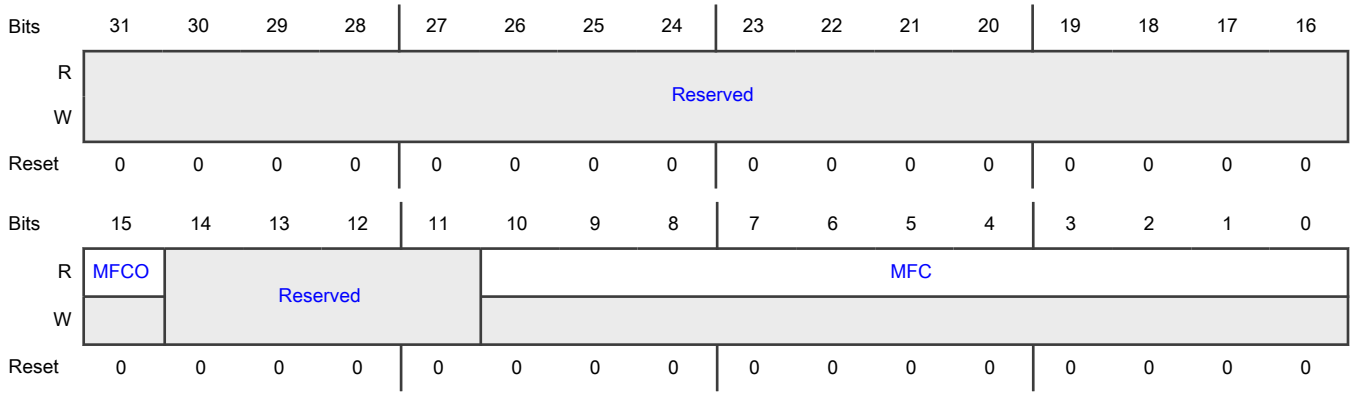
43.8.1.490 Channel 0 Missed Frame Counter (DMA_CH0_MISS_FRAME_CNT)

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in RX_CONTROL register.

Offset

Register	Offset
DMA_CH0_MISS_FRAME_CNT	1164h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Miss Frame Counter overflow not occurred 1 - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH0_RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH0_RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

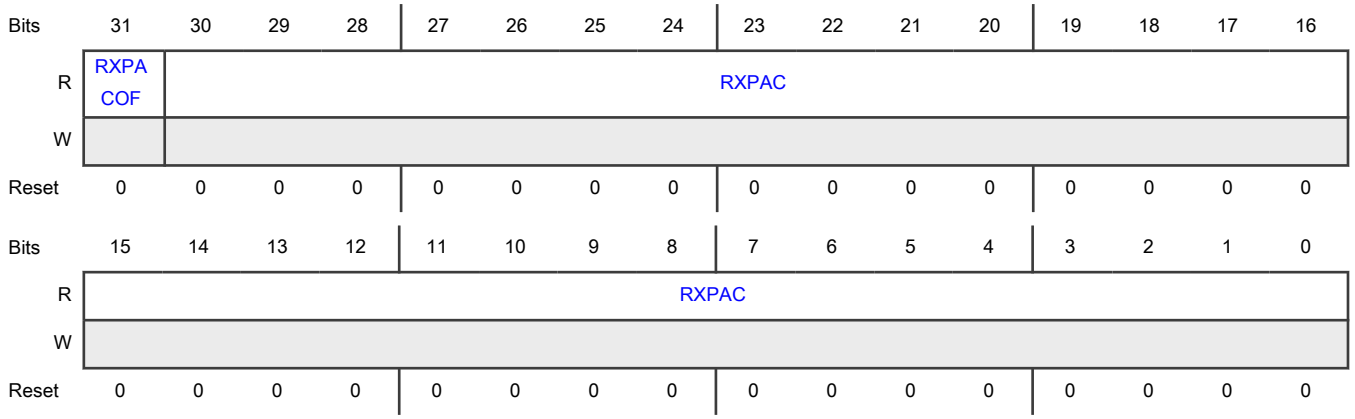
43.8.1.491 Channel 0 RXP Frames Accepted Counter (DMA_CH0_RXP_ACCEPT_CNT)

The RXP_ACCEPT_CNT register provides the count of the number of frames accepted by Rx Parser.

Offset

Register	Offset
DMA_CH0_RXP_ACCEPT_CNT	1168h

Diagram



Fields

Field	Description
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Accept Counter overflow not occurred 1 - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

43.8.1.492 DMA Channel 1 Control (DMA_CH1_CONTROL)

The DMA Channel 1 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

Offset

Register	Offset
DMA_CH1_CONTROL	1180h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved							Reserved	Reserved				DSL			Reserved	PBLx8
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in TX_CONTROL and Bits[21:16] in DMA_CH1_RX_CONTROL is multiplied by eight times. 8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in TX_CONTROL and Bits[21:16] in DMA_CH1_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0 - 8xPBL mode is disabled 1 - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

43.8.1.493 DMA Channel 1 Transmit Control (DMA_CH1_TX_CONTROL)

The DMA Channel 1 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

Offset

Register	Offset
DMA_CH1_TX_CONTROL	1184h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			EDSE	Reserved				Reserved	Reserved	TxPBL					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IPBL	Reserved		Reserved	Reserved							OSF	Reserved			ST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0 - Enhanced Descriptor is disabled 1 - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in CONTROL register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ1_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0 - Ignore PBL Requirement is disabled 1 - Ignore PBL Requirement is enabled
14-13 —	Reserved
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0 - Operate on Second Packet disabled 1 - Operate on Second Packet enabled
3-1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH1_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH1_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH1_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0 - Stop Transmission Command 1 - Start Transmission Command

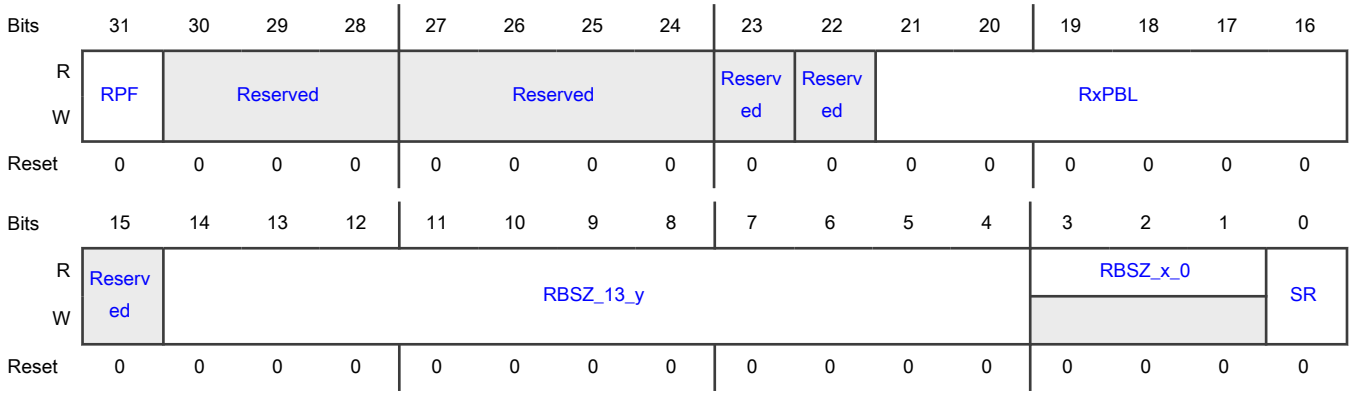
43.8.1.494 DMA Channel 1 Receive Control (DMA_CH1_RX_CONTROL)

The DMA Channel 1 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

Offset

Register	Offset
DMA_CH1_RX_CONTROL	1188h

Diagram



Fields

Field	Description
31 RPF	Rx Packet Flush. Rx Packet Flush. When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0 - Rx Packet Flush is disabled 1 - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application

Table continues on the next page...

Table continued from the previous page...

Field	Description
	bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ1_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1 RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH1_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH1_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH1_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0 - Stop Receive 1 - Start Receive

43.8.1.495 Channel 1 Tx Descriptor List Address (DMA_CH1_TXDESC_LIST_ADDRESS)

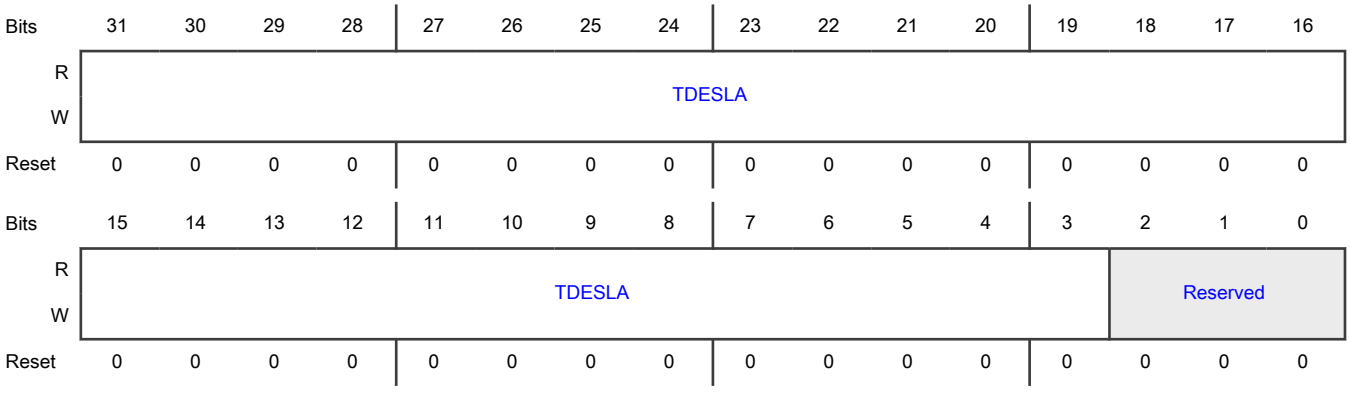
The Channel 1 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in DMA_CH1_TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

Offset

Register	Offset
DMA_CH1_TXDESC_LIST_ADDRESS	1194h

Diagram



Fields

Field	Description
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

43.8.1.496 Channel 1 Rx Descriptor List Address (DMA_CH1_RXDESC_LIST_ADDRESS)

The Channel 1 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

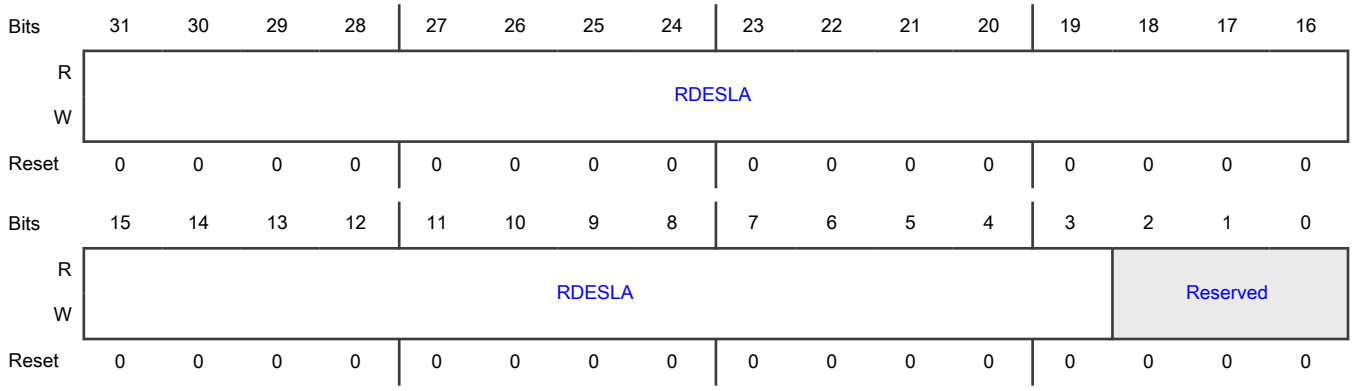
This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in DMA_CH1_RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

Offset

Register	Offset
DMA_CH1_RXDESC_LIST_ADDRESS	119Ch

Diagram



Fields

Field	Description
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

43.8.1.497 Channel 1 Tx Descriptor Tail Pointer (DMA_CH1_TXDESC_TAIL_POINTER)

The Channel 1 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

Offset

Register	Offset
DMA_CH1_TXDESC_TAIL_POINTER	11A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDTP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDTP													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

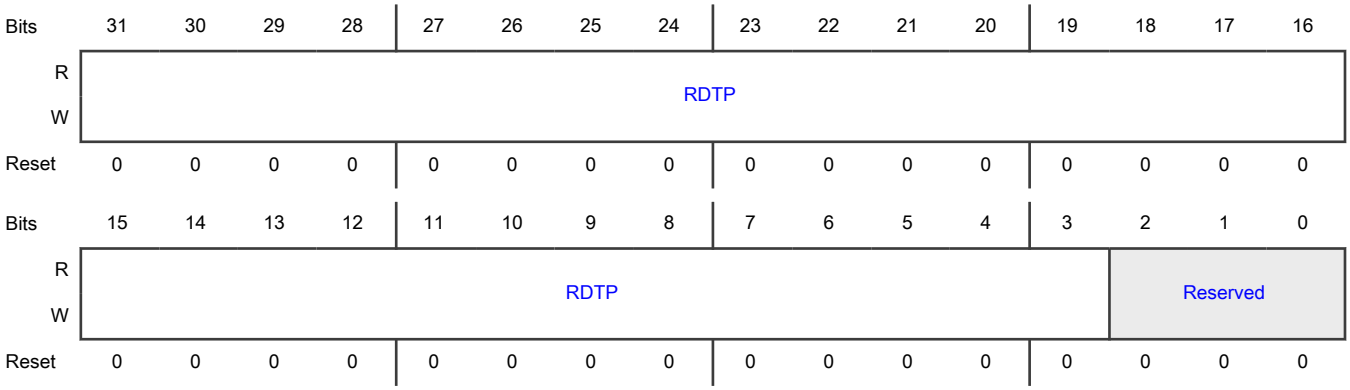
43.8.1.498 Channel 1 Rx Descriptor Tail Pointer (DMA_CH1_RXDESC_TAIL_POINTER)

The Channel 1 Rx Descriptor Tail Pointer points to an offset from the base and indicates the location of the last valid descriptor.

Offset

Register	Offset
DMA_CH1_RXDESC_TAIL_POINTER	11A8h

Diagram



Fields

Field	Description
31-3 RDTP	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

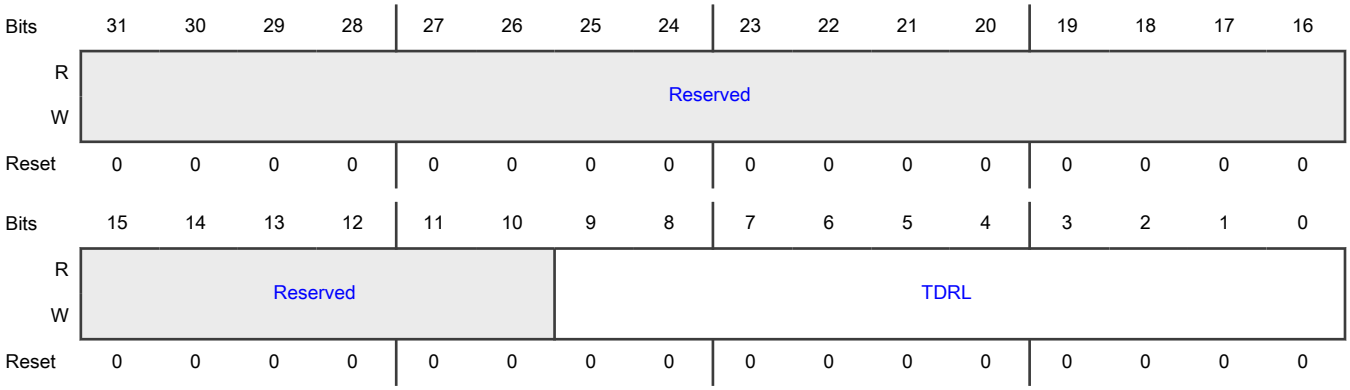
43.8.1.499 Channel 1 Tx Descriptor Ring Length (DMA_CH1_TXDESC_RING_LENGTH)

The Channel 1 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

Offset

Register	Offset
DMA_CH1_TXDESC_RING_LENGTH	11ACh

Diagram



Fields

Field	Description
31-10	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. We recommend a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

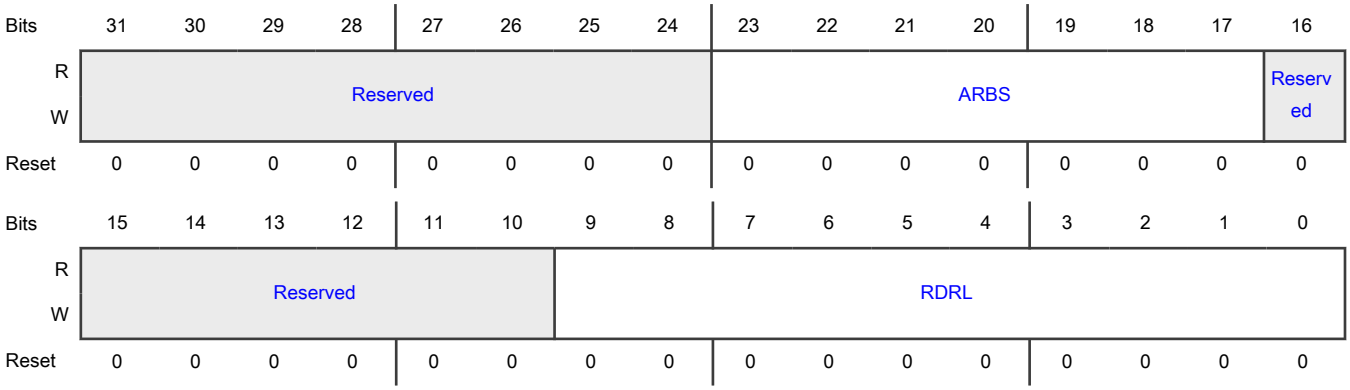
43.8.1.500 Channel 1 Receive Control 2 register (DMA_CH1_RX_CONTROL2)

The Channel 1 Receive Control register controls Rx Descriptor Ring Length and Alternate Rx Buffer Size.

Offset

Register	Offset
DMA_CH1_RX_CONTROL2	11B0h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-17 ARBS	Alternate Receive Buffer Size Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header

Table continues on the next page...

Table continued from the previous page...

Field	Description
	data. The maximum alternate buffer is limited to 1020, 1016 or 1008-bytes depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). When ARBS=0, Rx Buffer1 and Rx Buffer2 sizes are based on DMA_CH1_RX_CONTROL[RBSZ]. Width of ARBS field is 8, 7 or 6-bits depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively).
16-10 ——	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

43.8.1.501 Channel 1 Interrupt Enable (DMA_CH1_INTERRUPT_ENABLE)

The Channel 1 Interrupt Enable register enables the interrupts reported by the Status register.

Offset

Register	Offset
DMA_CH1_INTERRUPT_ENABLE	11B4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	Reserved			TBUE	TXSE	TIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0 - Normal Interrupt Summary is disabled 1 - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled. 0 - Abnormal Interrupt Summary is disabled 1 - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0 - Context Descriptor Error is disabled 1 - Context Descriptor Error is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0 - Fatal Bus Error is disabled 1 - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0 - Early Receive Interrupt is disabled 1 - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0 - Early Transmit Interrupt is disabled 1 - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0 - Receive Watchdog Timeout is disabled 1 - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0 - Receive Stopped is disabled 1 - Receive Stopped is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0 - Receive Buffer Unavailable is disabled 1 - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0 - Receive Interrupt is disabled 1 - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0 - Transmit Buffer Unavailable is disabled 1 - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0 - Transmit Stopped is disabled 1 - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0 - Transmit Interrupt is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Transmit Interrupt is enabled

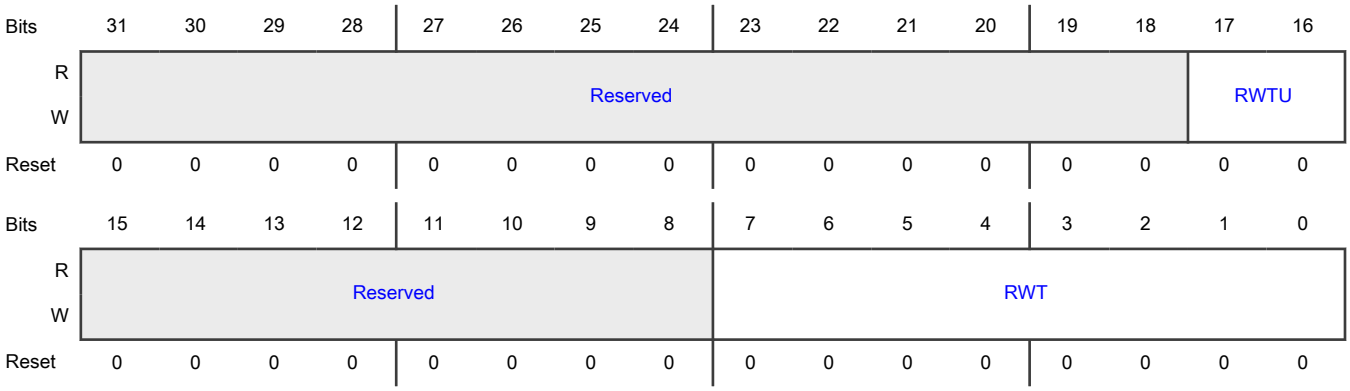
43.8.1.502 Channel 1 Receive Interrupt Watchdog Timer (DMA_CH1_RX_INTERRUPT_WATCHDOG_TIMER)

The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA_CH1_Status register.

Offset

Register	Offset
DMA_CH1_RX_INTERRUPT_WATCHDOG_TIMER	11B8h

Diagram



Fields

Field	Description
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8 —	Reserved
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

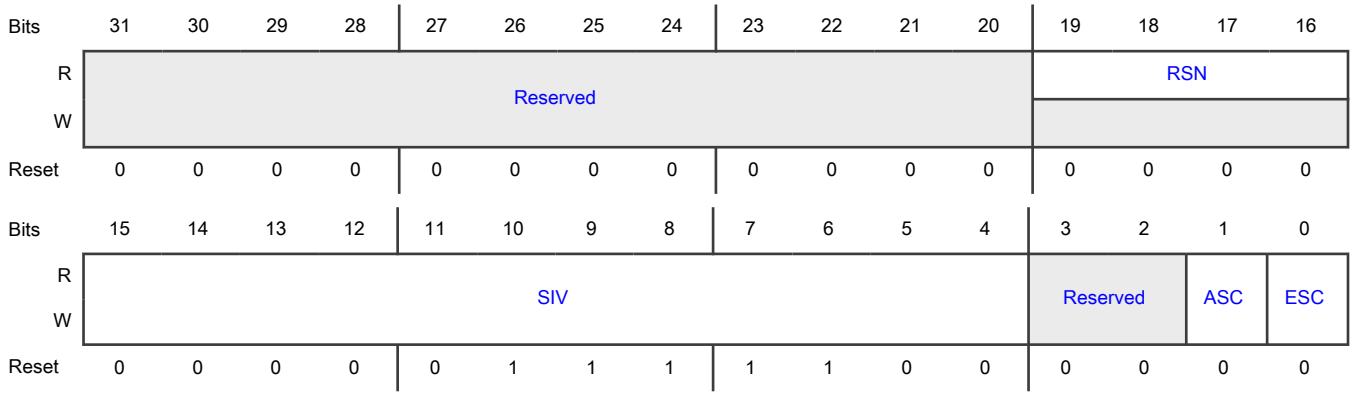
43.8.1.503 Channel 1 Slot Function Control and Status (DMA_CH1_SLOT_FUNCTION_CONTROL_STATUS)

The Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

Offset

Register	Offset
DMA_CH1_SLOT_FUNCTION_CONTROL_STATUS	11BCh

Diagram



Fields

Field	Description
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0 - Advance Slot Check is disabled 1 - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or - ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0 - Slot Comparison is disabled 1 - Slot Comparison is enabled

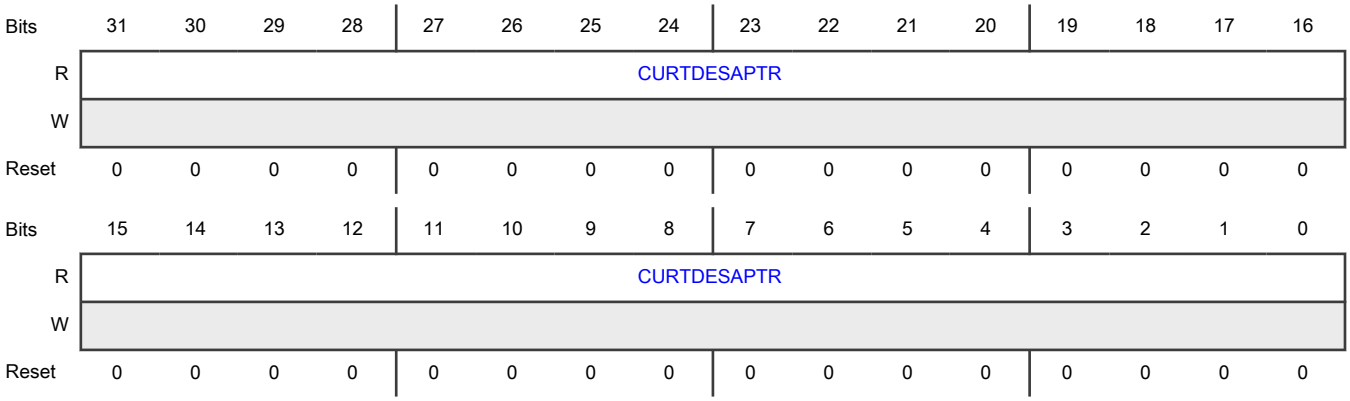
43.8.1.504 Channel 1 Current Application Transmit Descriptor (DMA_CH1_CURRENT_APP_TXDESC)

The Channel 1 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

Offset

Register	Offset
DMA_CH1_CURRENT_APP_TXDESC	11C4h

Diagram



Fields

Field	Description
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

43.8.1.505 Channel 1 Current Application Receive Descriptor (DMA_CH1_CURRENT_APP_RXDESC)

The Channel 1 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

Offset

Register	Offset
DMA_CH1_CURRENT_APP_RXDESC	11CCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CURRDESAPTR	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

43.8.1.506 Channel 1 Current Application Transmit Buffer Address (DMA_CH1_CURRENT_APP_TXBUFFER)

The Channel 1 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

Offset

Register	Offset
DMA_CH1_CURRENT_APP_TXBUFFER	11D4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

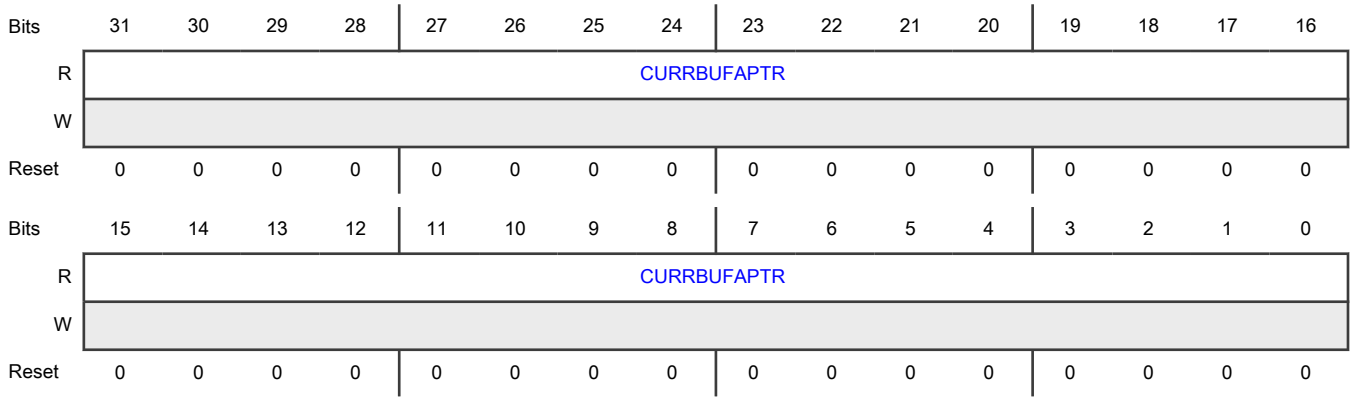
43.8.1.507 Channel 1 Current Application Receive Buffer Address (DMA_CH1_CURRENT_APP_RXBUFFER)

The Channel 1 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

Offset

Register	Offset
DMA_CH1_CURRENT_APP_RXBUFFER	11DCh

Diagram



Fields

Field	Description
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

43.8.1.508 DMA Channel 1 Status (DMA_CH1_STATUS)

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

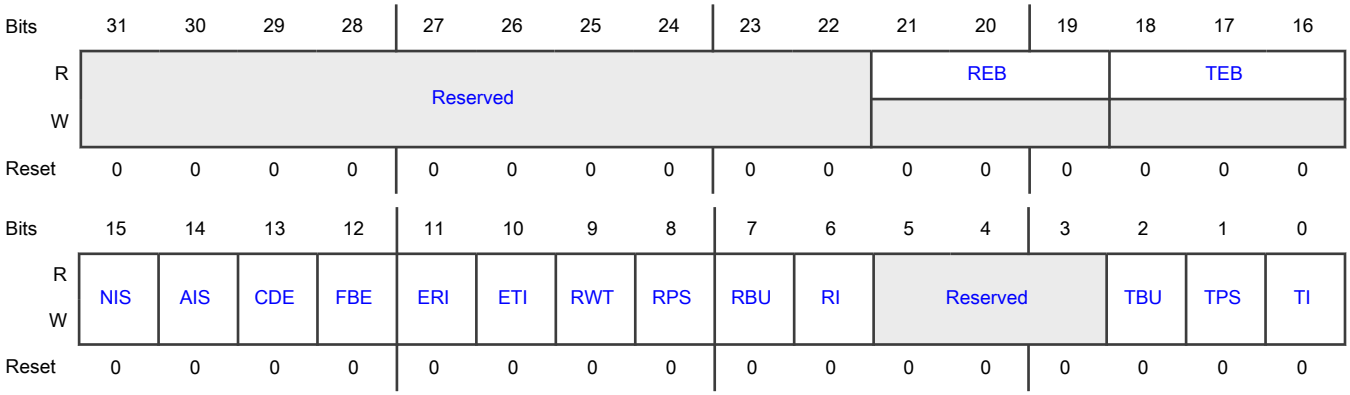
The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

Offset

Register	Offset
DMA_CH1_STATUS	11E0h

Diagram



Fields

Field	Description
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH1_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH1_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Normal Interrupt Summary status not detected 1 - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Abnormal Interrupt Summary status not detected 1 - Abnormal Interrupt Summary status detected
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Context Descriptor Error status not detected 1 - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Fatal Bus Error status not detected 1 - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Receive Interrupt status not detected 1 - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Transmit Interrupt status not detected 1 - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0 - Receive Watchdog Timeout status not detected 1 - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Process Stopped status not detected 1 - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode,

Table continues on the next page...

Table continued from the previous page...

Field	Description
	the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Buffer Unavailable status not detected 1 - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Interrupt status not detected 1 - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Buffer Unavailable status not detected 1 - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Process Stopped status not detected 1 - Transmit Process Stopped status detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Interrupt status not detected 1 - Transmit Interrupt status detected

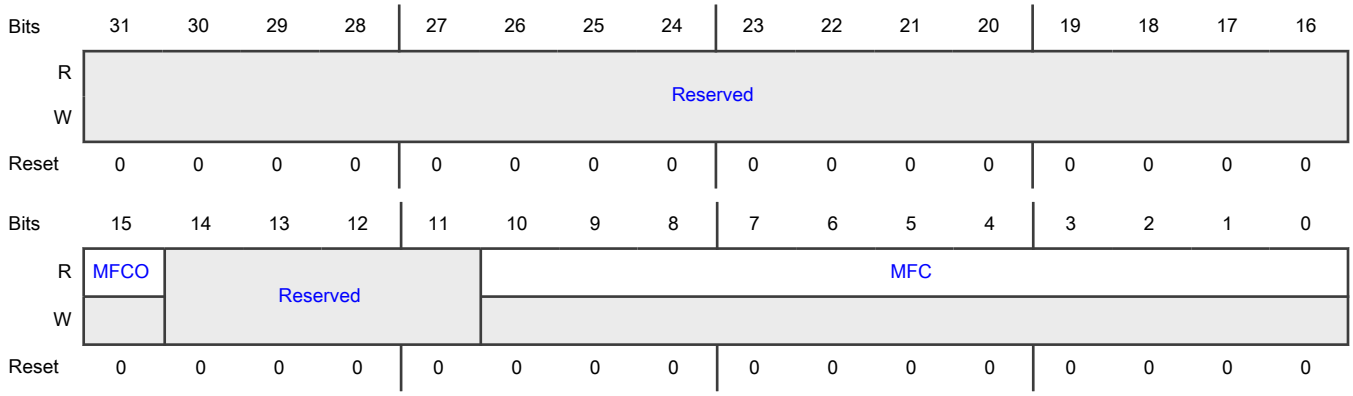
43.8.1.509 Channel 1 Missed Frame Counter (DMA_CH1_MISS_FRAME_CNT)

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in RX_CONTROL register.

Offset

Register	Offset
DMA_CH1_MISS_FRAME_CNT	11E4h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Miss Frame Counter overflow not occurred 1 - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH1_RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

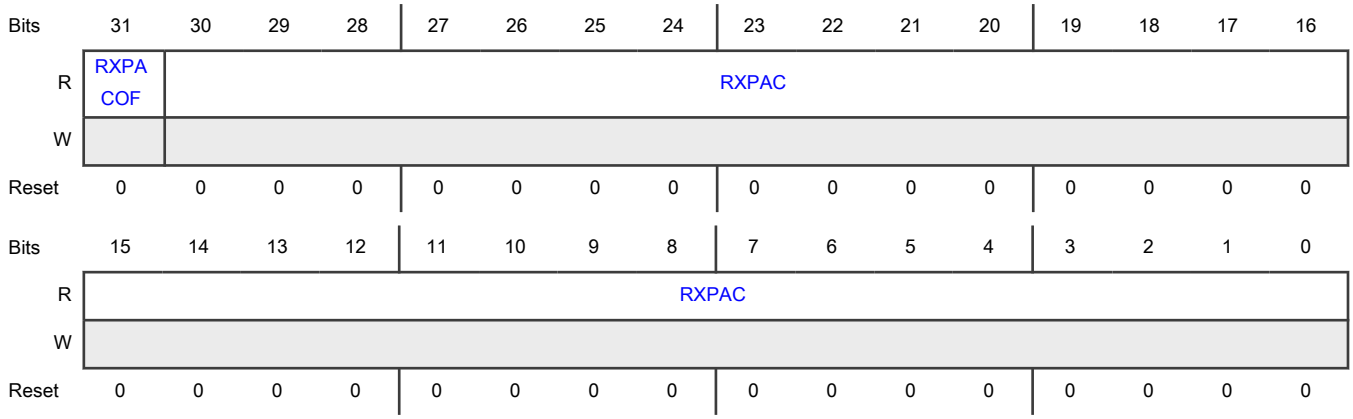
43.8.1.510 Channel 1 RXP Frames Accepted Counter (DMA_CH1_RXP_ACCEPT_CNT)

The DMA_CH1_RXP_Accept_Cnt register provides the count of the number of frames accepted by Rx Parser.

Offset

Register	Offset
DMA_CH1_RXP_ACCEPT_CNT	11E8h

Diagram



Fields

Field	Description
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Accept Counter overflow not occurred 1 - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

43.8.1.511 DMA Channel 2 Control (DMA_CH2_CONTROL)

The DMA Channel 2 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

Offset

Register	Offset
DMA_CH2_CONTROL	1200h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved							Reserved	Reserved				DSL			Reserved	PBLx8
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH2_TX_CONTROL and Bits[21:16] in DMA_CH2_RX_CONTROL is multiplied by eight times. 8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH2_TX_CONTROL and Bits[21:16] in DMA_CH2_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0 - 8xPBL mode is disabled 1 - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

43.8.1.512 DMA Channel 2 Transmit Control (DMA_CH2_TX_CONTROL)

The DMA Channel 2 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

Offset

Register	Offset
DMA_CH2_TX_CONTROL	1204h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			EDSE	Reserved				Reserved	Reserved	TxPBL					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IPBL	Reserved		Reserved	Reserved							OSF	Reserved			ST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0 - Enhanced Descriptor is disabled 1 - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in DMA_CH2_CONTROL register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ2_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0 - Ignore PBL Requirement is disabled 1 - Ignore PBL Requirement is enabled
14-13 —	Reserved
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0 - Operate on Second Packet disabled 1 - Operate on Second Packet enabled
3-1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH2_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH2_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH2_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program DMA_CH2_TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0 - Stop Transmission Command 1 - Start Transmission Command

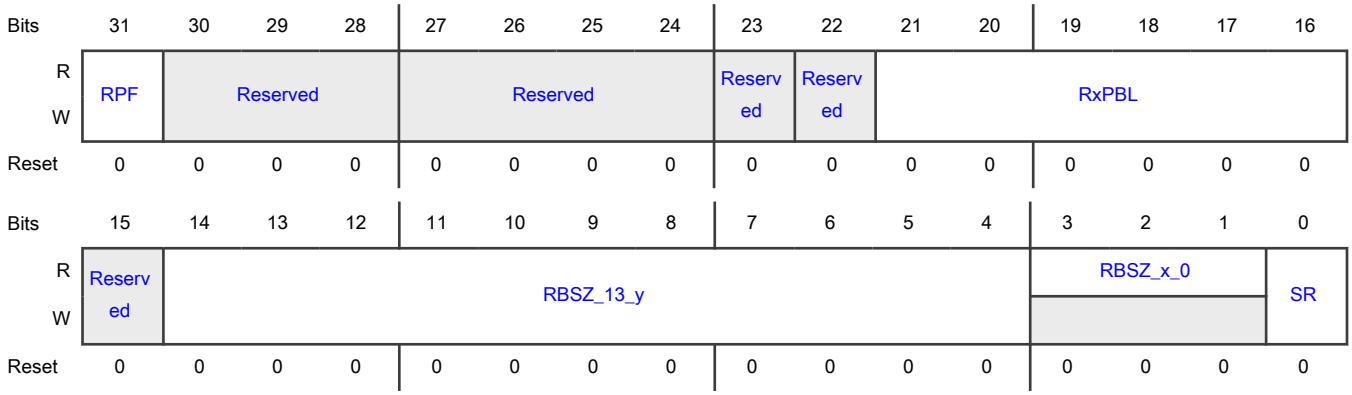
43.8.1.513 DMA Channel 2 Receive Control (DMA_CH2_RX_CONTROL)

The DMA Channel 2 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

Offset

Register	Offset
DMA_CH2_RX_CONTROL	1208h

Diagram



Fields

Field	Description
31 RPF	Rx Packet Flush. Rx Packet Flush. When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0 - Rx Packet Flush is disabled 1 - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application

Table continues on the next page...

Table continued from the previous page...

Field	Description
	bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the DMA_CH2_CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ2_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1 RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH2_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH2_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH2_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0 - Stop Receive 1 - Start Receive

43.8.1.514 Channel 2 Tx Descriptor List Address (DMA_CH2_TXDESC_LIST_ADDRESS)

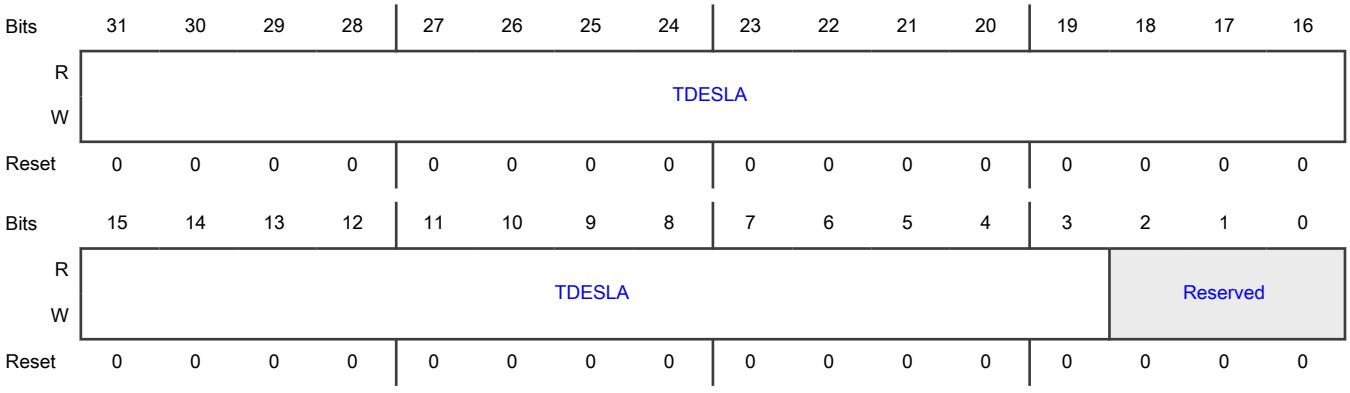
The Channel 2 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

Offset

Register	Offset
DMA_CH2_TXDESC_LIST_ADDRESS	1214h

Diagram



Fields

Field	Description
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

43.8.1.515 Channel 2 Rx Descriptor List Address (DMA_CH2_RXDESC_LIST_ADDRESS)

The Channel 2 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

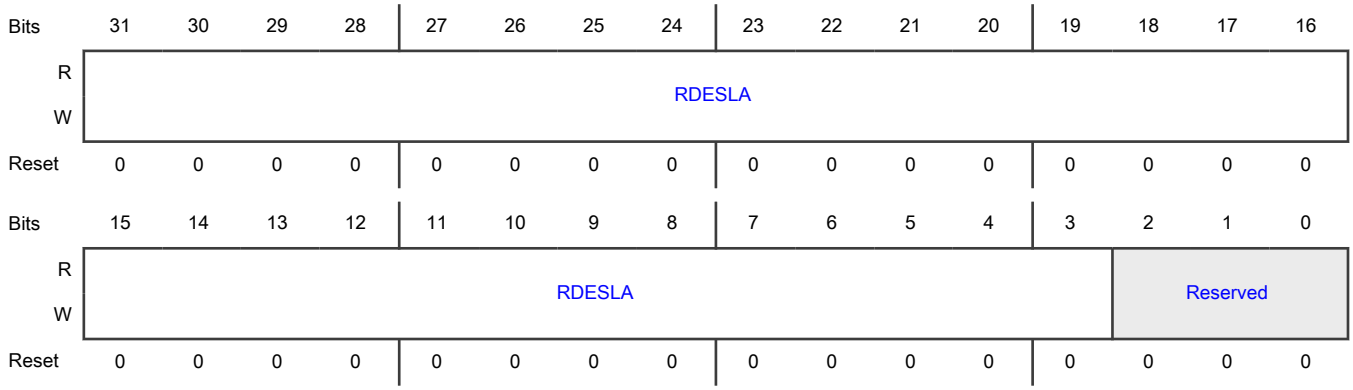
This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

Offset

Register	Offset
DMA_CH2_RXDESC_LIST_ADDRESS	121Ch

Diagram



Fields

Field	Description
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

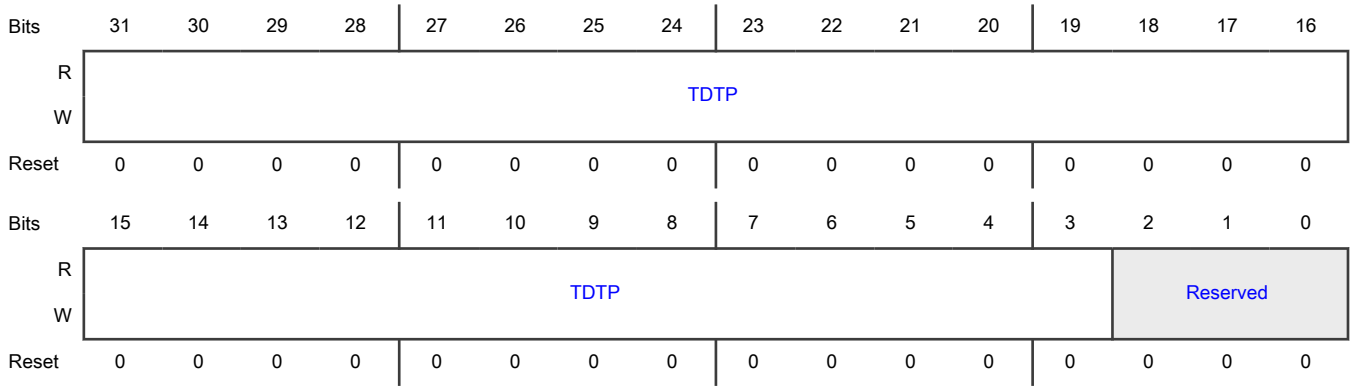
43.8.1.516 Channel 2 Tx Descriptor Tail Pointer (DMA_CH2_TXDESC_TAIL_POINTER)

The Channel 2 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

Offset

Register	Offset
DMA_CH2_TXDESC_TAIL_POINTER	1220h

Diagram



Fields

Field	Description
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

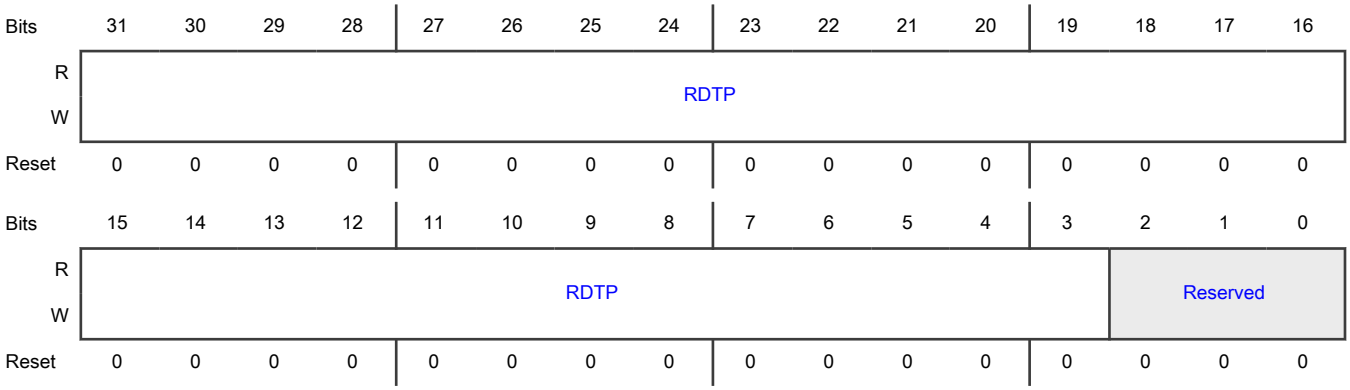
43.8.1.517 Channel 2 Rx Descriptor Tail Pointer (DMA_CH2_RXDESC_TAIL_POINTER)

The Channel 2 Rx Descriptor Tail Pointer Points to an offset from the base and indicates the location of the last valid descriptor.

Offset

Register	Offset
DMA_CH2_RXDESC_TAIL_POINTER	1228h

Diagram



Fields

Field	Description
31-3 RDTP	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

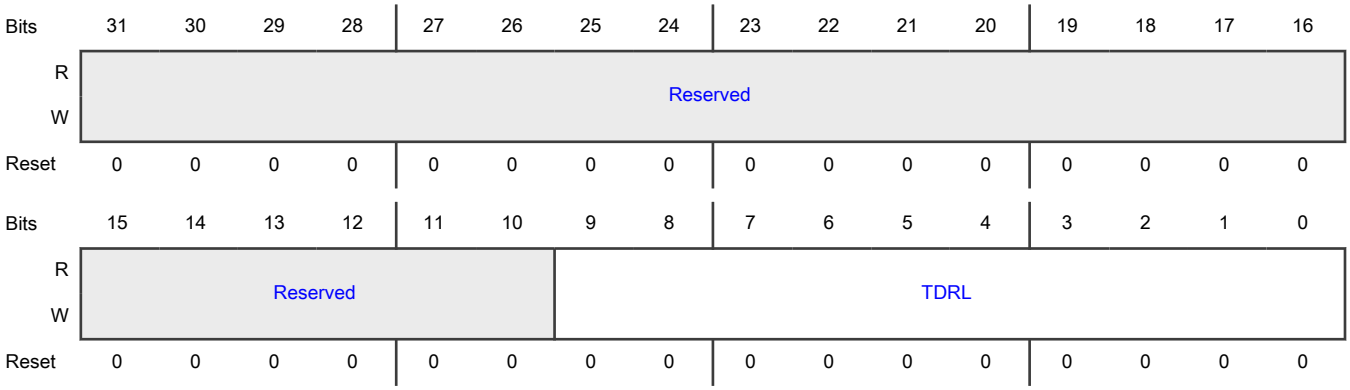
43.8.1.518 Channel 2 Tx Descriptor Ring Length (DMA_CH2_TXDESC_RING_LENGTH)

The Channel 2 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

Offset

Register	Offset
DMA_CH2_TXDESC_RING_LENGTH	122Ch

Diagram



Fields

Field	Description
31-10	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. We recommend a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

43.8.1.519 Channel 2 Receive Control 2 register (DMA_CH2_RX_CONTROL2)

The Channel 2 Receive Control register controls Rx Descriptor Ring Length and Alternate Rx Buffer Size.

Offset

Register	Offset
DMA_CH2_RX_CONTROL2	1230h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								ARBS							Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						RDRL									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23-17 ARBS	Alternate Receive Buffer Size Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header

Table continues on the next page...

Table continued from the previous page...

Field	Description
	data. The maximum alternate buffer is limited to 1020, 1016 or 1008-bytes depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). When ARBS=0, Rx Buffer1 and Rx Buffer2 sizes are based on DMA_CH2_RX_CONTROL[RBSZ]. Width of ARBS field is 8, 7 or 6-bits depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively).
16-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

43.8.1.520 Channel 2 Interrupt Enable (DMA_CH2_INTERRUPT_ENABLE)

The Channel 2 Interrupt Enable register enables the interrupts reported by the Status register.

Offset

Register	Offset
DMA_CH2_INTERRUPT_ENABLE	1234h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	Reserved			TBUE	TXSE	TIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0 - Normal Interrupt Summary is disabled 1 - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled. 0 - Abnormal Interrupt Summary is disabled 1 - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0 - Context Descriptor Error is disabled 1 - Context Descriptor Error is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0 - Fatal Bus Error is disabled 1 - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0 - Early Receive Interrupt is disabled 1 - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0 - Early Transmit Interrupt is disabled 1 - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0 - Receive Watchdog Timeout is disabled 1 - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0 - Receive Stopped is disabled 1 - Receive Stopped is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0 - Receive Buffer Unavailable is disabled 1 - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0 - Receive Interrupt is disabled 1 - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0 - Transmit Buffer Unavailable is disabled 1 - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0 - Transmit Stopped is disabled 1 - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0 - Transmit Interrupt is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Transmit Interrupt is enabled

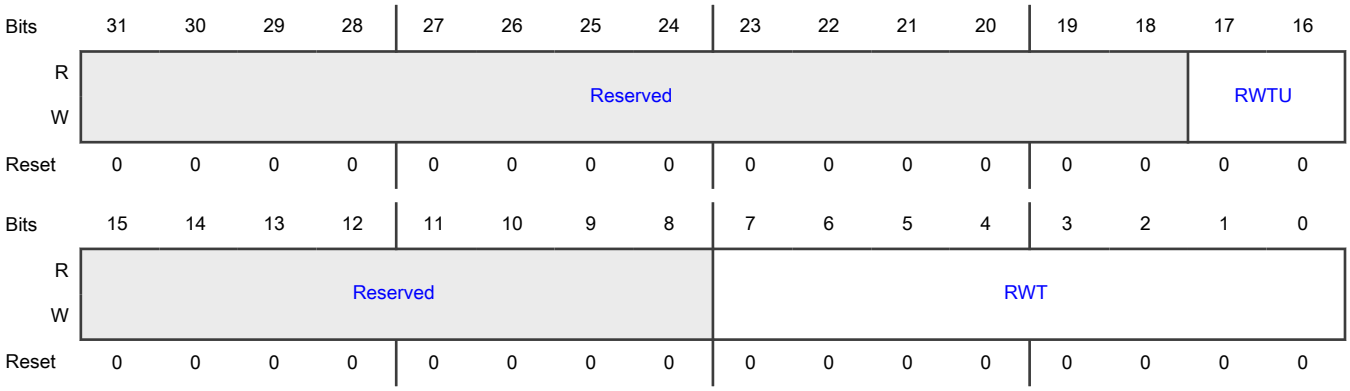
43.8.1.521 Channel 2 Receive Interrupt Watchdog Timer (DMA_CH2_RX_INTERRUPT_WATCHDOG_TIMER)

The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA_CH2_Status register.

Offset

Register	Offset
DMA_CH2_RX_INTERRUPT_WATCHDOG_TIMER	1238h

Diagram



Fields

Field	Description
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8 —	Reserved
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

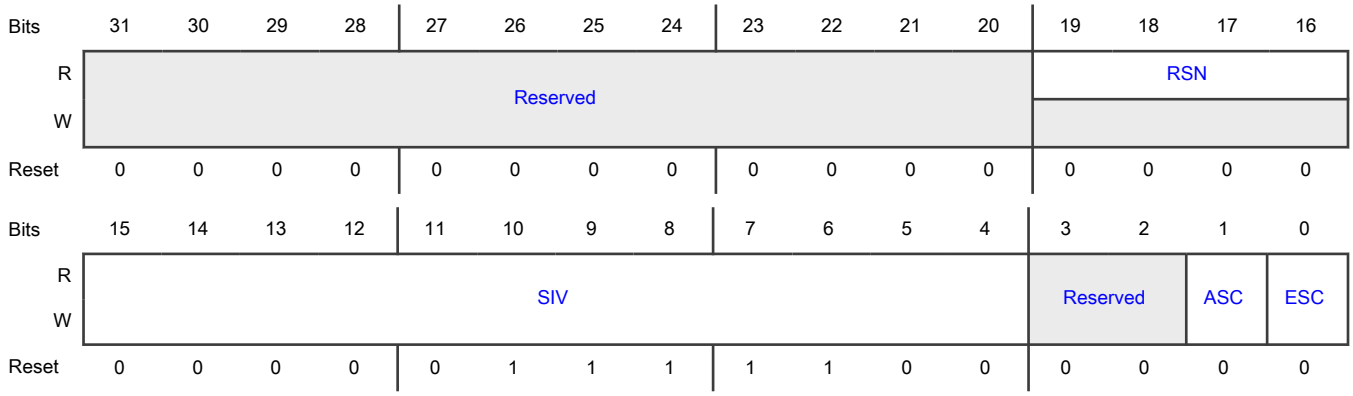
43.8.1.522 Channel 2 Slot Function Control and Status (DMA_CH2_SLOT_FUNCTION_CONTROL_STATUS)

The Channel 2 Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

Offset

Register	Offset
DMA_CH2_SLOT_FUNCTION_CONTROL_STATUS	123Ch

Diagram



Fields

Field	Description
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0 - Advance Slot Check is disabled 1 - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or - ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0 - Slot Comparison is disabled 1 - Slot Comparison is enabled

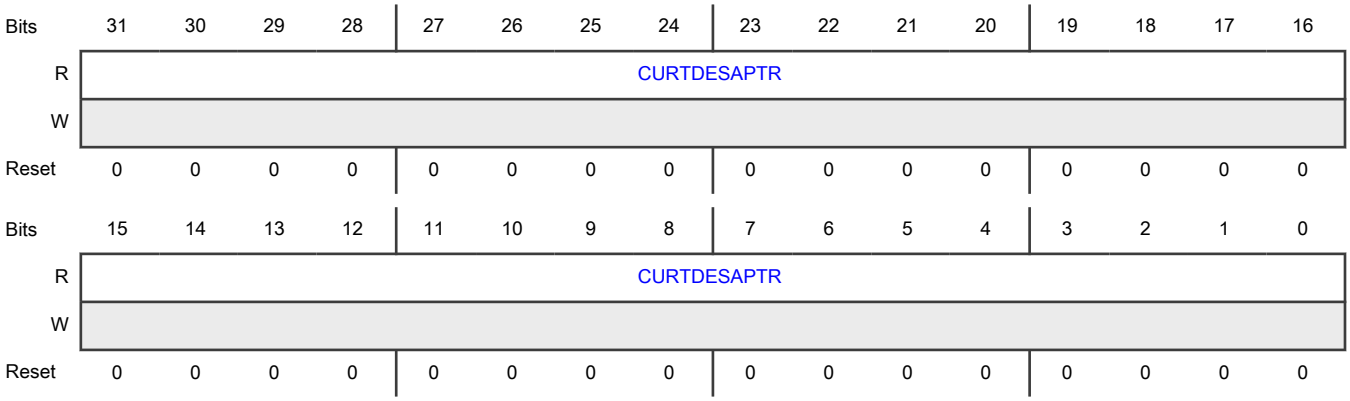
43.8.1.523 Channel 2 Current Application Transmit Descriptor (DMA_CH2_CURRENT_APP_TXDESC)

The Channel 2 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

Offset

Register	Offset
DMA_CH2_CURRENT_APP_TXDESC	1244h

Diagram



Fields

Field	Description
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

43.8.1.524 Channel 2 Current Application Receive Descriptor (DMA_CH2_CURRENT_APP_RXDESC)

The Channel 2 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

Offset

Register	Offset
DMA_CH2_CURRENT_APP_RXDESC	124Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CURRDESAPTR	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

43.8.1.525 Channel 2 Current Application Transmit Buffer Address (DMA_CH2_CURRENT_APP_TXBUFFER)

The Channel 2 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

Offset

Register	Offset
DMA_CH2_CURRENT_APP_TXBUFFER	1254h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

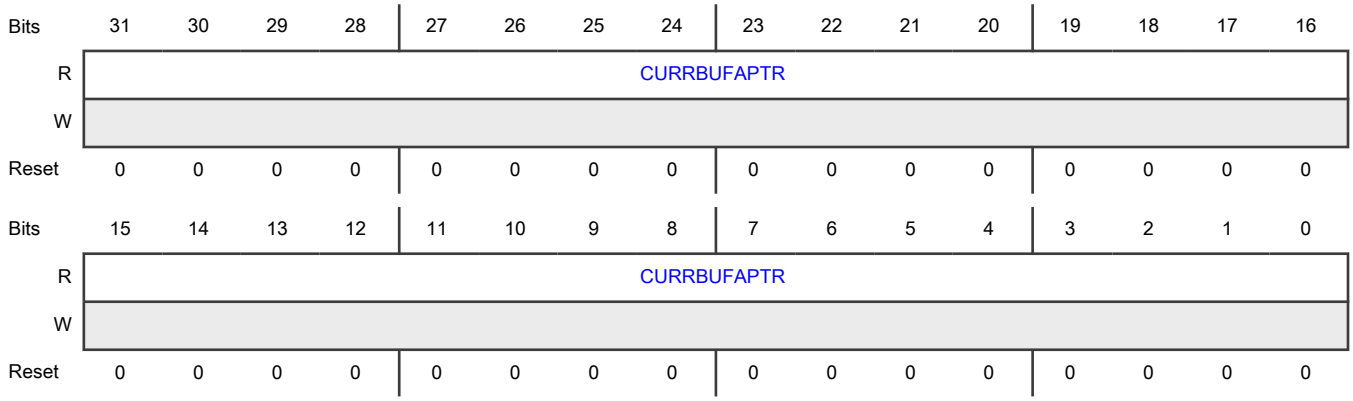
43.8.1.526 Channel 2 Current Application Receive Buffer Address (DMA_CH2_CURRENT_APP_RXBUFFER)

The Channel 2 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

Offset

Register	Offset
DMA_CH2_CURRENT_APP_RXBUFFER	125Ch

Diagram



Fields

Field	Description
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

43.8.1.527 DMA Channel 2 Status (DMA_CH2_STATUS)

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

Offset

Register	Offset
DMA_CH2_STATUS	1260h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB		TEB			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH2_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH2_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Normal Interrupt Summary status not detected 1 - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH2_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH2_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Abnormal Interrupt Summary status not detected 1 - Abnormal Interrupt Summary status detected
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Context Descriptor Error status not detected 1 - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Fatal Bus Error status not detected 1 - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Receive Interrupt status not detected 1 - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Transmit Interrupt status not detected 1 - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0 - Receive Watchdog Timeout status not detected 1 - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Process Stopped status not detected 1 - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode,

Table continues on the next page...

Table continued from the previous page...

Field	Description
	the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Buffer Unavailable status not detected 1 - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Interrupt status not detected 1 - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Buffer Unavailable status not detected 1 - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Process Stopped status not detected 1 - Transmit Process Stopped status detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Interrupt status not detected 1 - Transmit Interrupt status detected

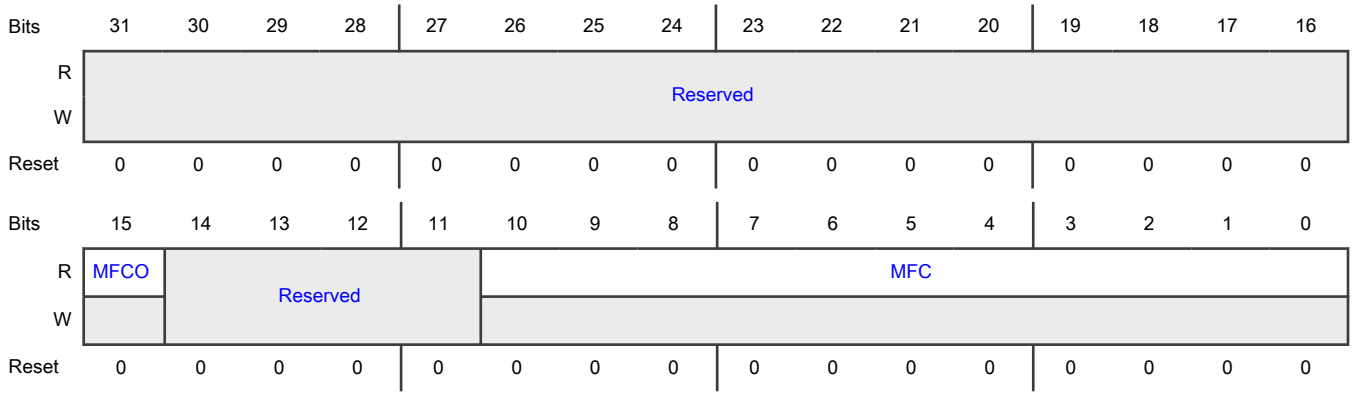
43.8.1.528 Channel 2 Missed Frame Counter (DMA_CH2_MISS_FRAME_CNT)

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA_CH2_RX_CONTROL register.

Offset

Register	Offset
DMA_CH2_MISS_FRAME_CNT	1264h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Miss Frame Counter overflow not occurred 1 - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH2_RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH2_RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

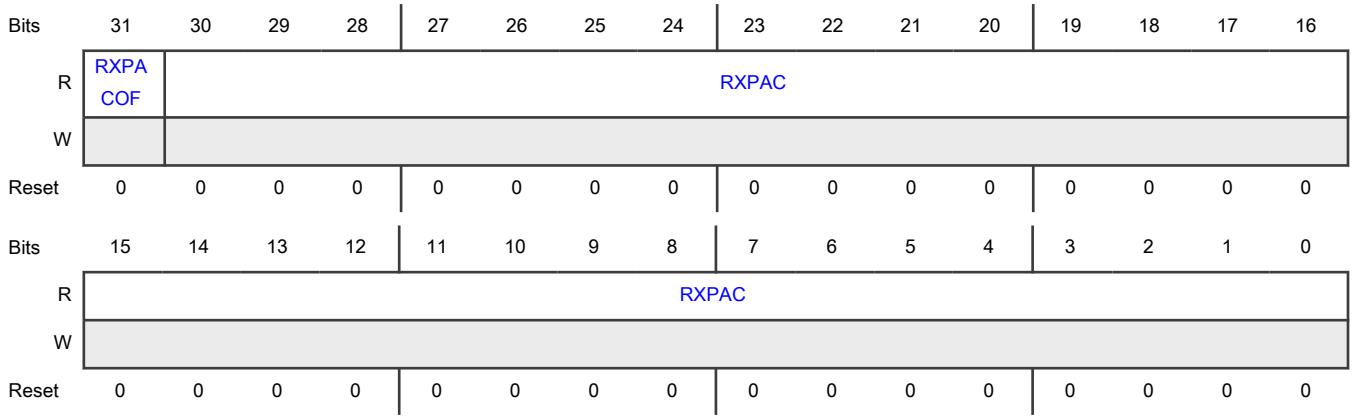
43.8.1.529 Channel 2 RXP Frames Accepted Counter (DMA_CH2_RXP_ACCEPT_CNT)

The DMA_CH2_RXP_Accept_Cnt register provides the count of the number of frames accepted by Rx Parser.

Offset

Register	Offset
DMA_CH2_RXP_ACCEPT_CNT	1268h

Diagram



Fields

Field	Description
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Accept Counter overflow not occurred 1 - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

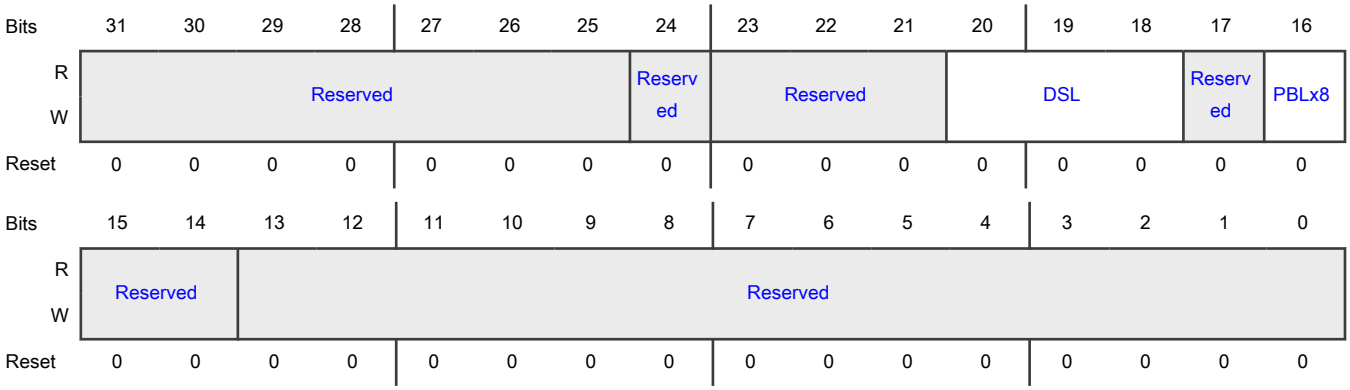
43.8.1.530 DMA Channel 3 Control (DMA_CH3_CONTROL)

The DMA Channel 3 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

Offset

Register	Offset
DMA_CH3_CONTROL	1280h

Diagram



Fields

Field	Description
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH3_TX_CONTROL and Bits[21:16] in DMA_CH3_RX_CONTROL is multiplied by eight times. 8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH3_TX_CONTROL and Bits[21:16] in DMA_CH3_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0 - 8xPBL mode is disabled 1 - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

43.8.1.531 DMA Channel 3 Transmit Control (DMA_CH3_TX_CONTROL)

The DMA Channel 3 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

Offset

Register	Offset
DMA_CH3_TX_CONTROL	1284h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			EDSE	Reserved				Reserved	Reserved	TxPBL					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IPBL	Reserved		Reserved	Reserved							OSF	Reserved			ST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0 - Enhanced Descriptor is disabled 1 - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in DMA_CH3_CONTROL register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ3_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0 - Ignore PBL Requirement is disabled 1 - Ignore PBL Requirement is enabled
14-13 —	Reserved
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0 - Operate on Second Packet disabled 1 - Operate on Second Packet enabled
3-1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH3_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH3_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH3_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program DMA_CH3_TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0 - Stop Transmission Command 1 - Start Transmission Command

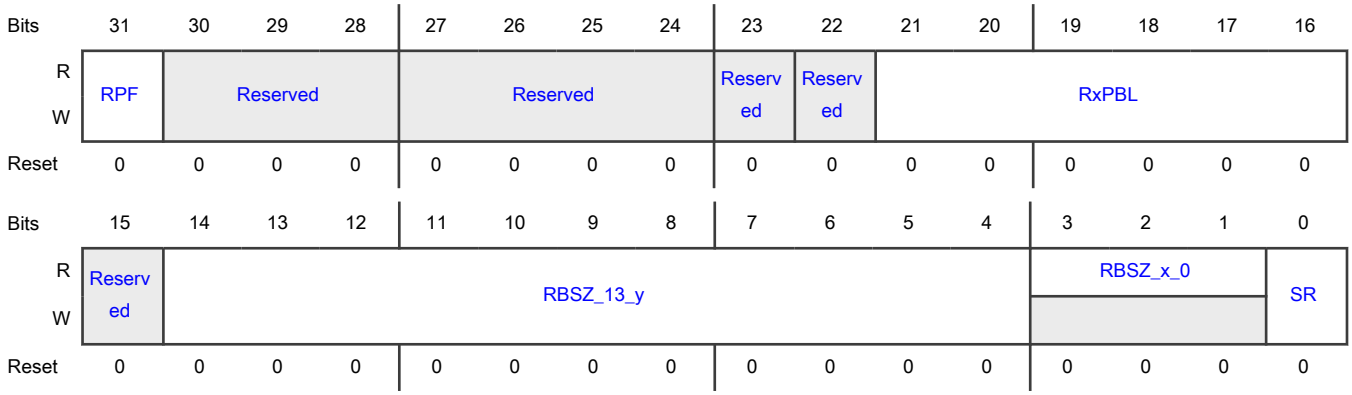
43.8.1.532 DMA Channel 3 Receive Control (DMA_CH3_RX_CONTROL)

The DMA Channel 3 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

Offset

Register	Offset
DMA_CH3_RX_CONTROL	1288h

Diagram



Fields

Field	Description
31 RPF	Rx Packet Flush. Rx Packet Flush. When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0 - Rx Packet Flush is disabled 1 - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application

Table continues on the next page...

Table continued from the previous page...

Field	Description
	bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ3_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1 RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH3_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH3_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH3_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0 - Stop Receive 1 - Start Receive

43.8.1.533 Channel 3 Tx Descriptor List Address (DMA_CH3_TXDESC_LIST_ADDRESS)

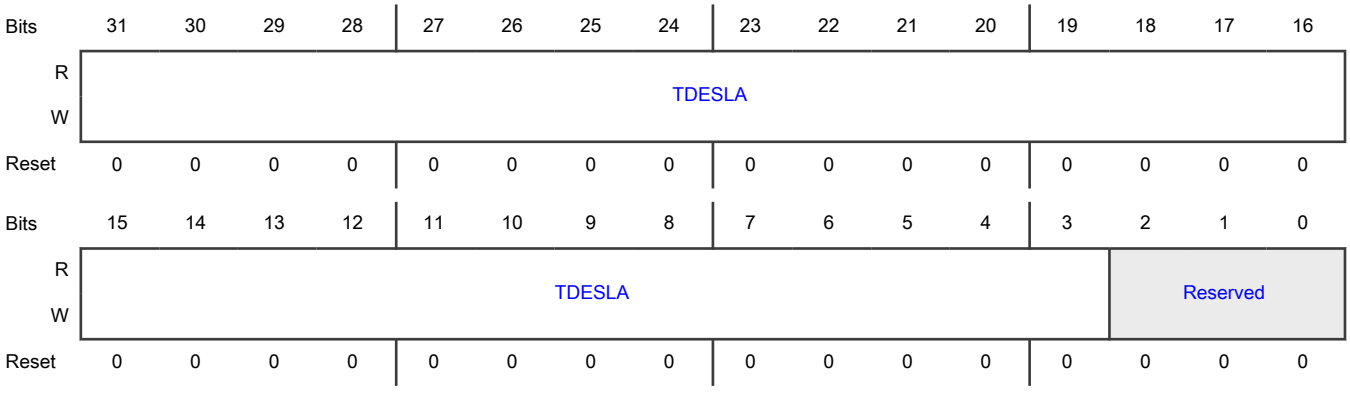
The Channel 3 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

Offset

Register	Offset
DMA_CH3_TXDESC_LIST_ADDRESS	1294h

Diagram



Fields

Field	Description
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

43.8.1.534 Channel 3 Rx Descriptor List Address (DMA_CH3_RXDESC_LIST_ADDRESS)

The Channel 3 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

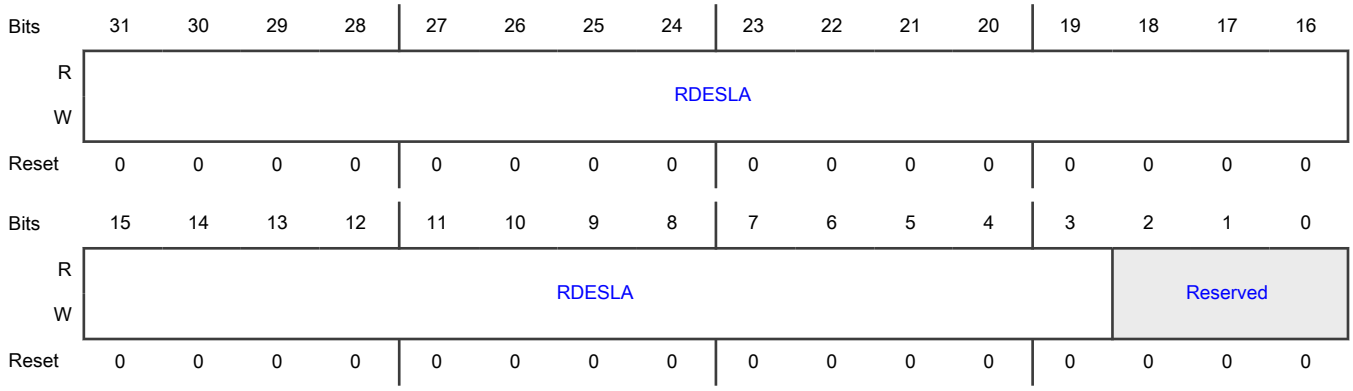
This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

Offset

Register	Offset
DMA_CH3_RXDESC_LIST_ADDRESS	129Ch

Diagram



Fields

Field	Description
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

43.8.1.535 Channel 3 Tx Descriptor Tail Pointer (DMA_CH3_TXDESC_TAIL_POINTER)

The Channel 3 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

Offset

Register	Offset
DMA_CH3_TXDESC_TAIL_POINTER	12A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDTP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDTP													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

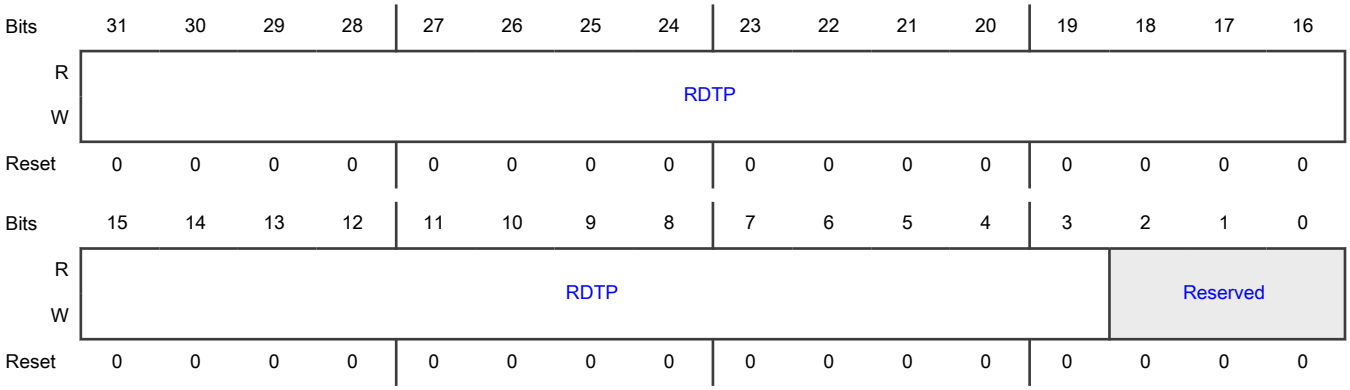
43.8.1.536 Channel 3 Rx Descriptor Tail Pointer (DMA_CH3_RXDESC_TAIL_POINTER)

The Channel 3 Rx Descriptor Tail Pointer points to an offset from the base and indicates the location of the last valid descriptor.

Offset

Register	Offset
DMA_CH3_RXDESC_TAIL_POINTER	12A8h

Diagram



Fields

Field	Description
31-3 RDTP	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

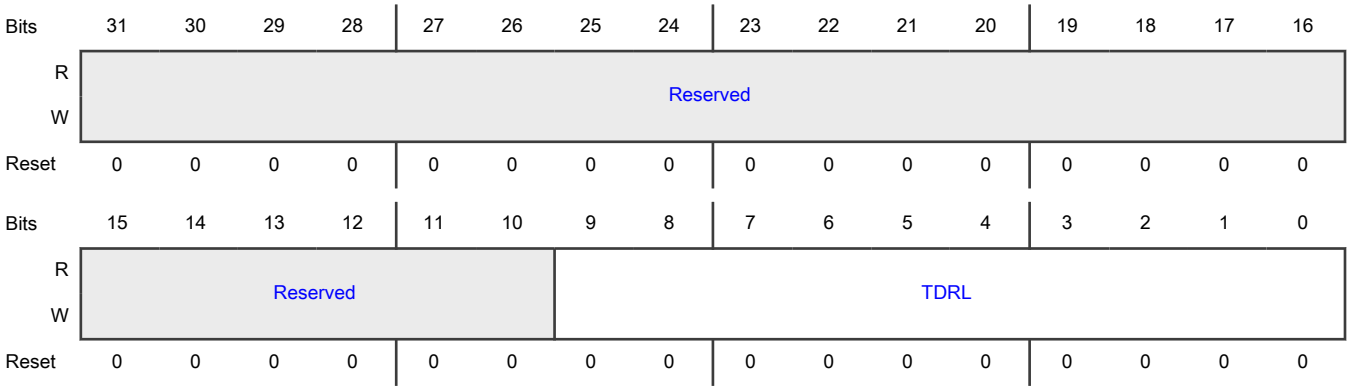
43.8.1.537 Channel 3 Tx Descriptor Ring Length (DMA_CH3_TXDESC_RING_LENGTH)

The Channel 3 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

Offset

Register	Offset
DMA_CH3_TXDESC_RING_LENGTH	12ACh

Diagram



Fields

Field	Description
31-10	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. We recommend a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

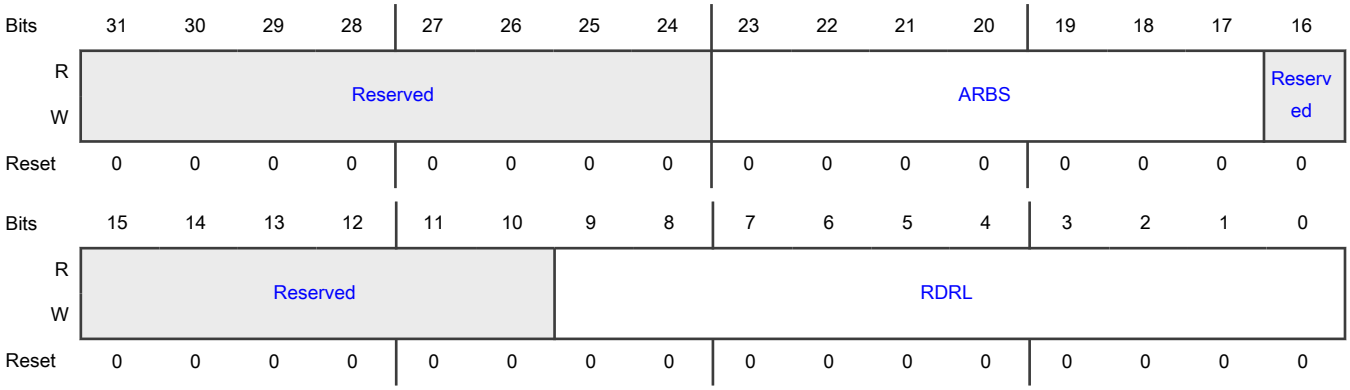
43.8.1.538 Channel 3 Receive Control 2 register (DMA_CH3_RX_CONTROL2)

The Channel 3 Receive Control register controls Rx Descriptor Ring Length and Alternate Rx Buffer Size.

Offset

Register	Offset
DMA_CH3_RX_CONTROL2	12B0h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-17 ARBS	Alternate Receive Buffer Size Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header

Table continues on the next page...

Table continued from the previous page...

Field	Description
	data. The maximum alternate buffer is limited to 1020, 1016 or 1008-bytes depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). When ARBS=0, Rx Buffer1 and Rx Buffer2 sizes are based on DMA_CH2_RX_CONTROL[RBSZ]. Width of ARBS field is 8, 7 or 6-bits depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively).
16-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

43.8.1.539 Channel 3 Interrupt Enable (DMA_CH3_INTERRUPT_ENABLE)

The Channel 3 Interrupt Enable register enables the interrupts reported by the Status register.

Offset

Register	Offset
DMA_CH3_INTERRUPT_ENABLE	12B4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	Reserved			TBUE	TXSE	TIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the DMA_CH3_STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0 - Normal Interrupt Summary is disabled 1 - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled. 0 - Abnormal Interrupt Summary is disabled 1 - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0 - Context Descriptor Error is disabled 1 - Context Descriptor Error is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0 - Fatal Bus Error is disabled 1 - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0 - Early Receive Interrupt is disabled 1 - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0 - Early Transmit Interrupt is disabled 1 - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0 - Receive Watchdog Timeout is disabled 1 - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0 - Receive Stopped is disabled 1 - Receive Stopped is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0 - Receive Buffer Unavailable is disabled 1 - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0 - Receive Interrupt is disabled 1 - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0 - Transmit Buffer Unavailable is disabled 1 - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0 - Transmit Stopped is disabled 1 - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0 - Transmit Interrupt is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Transmit Interrupt is enabled

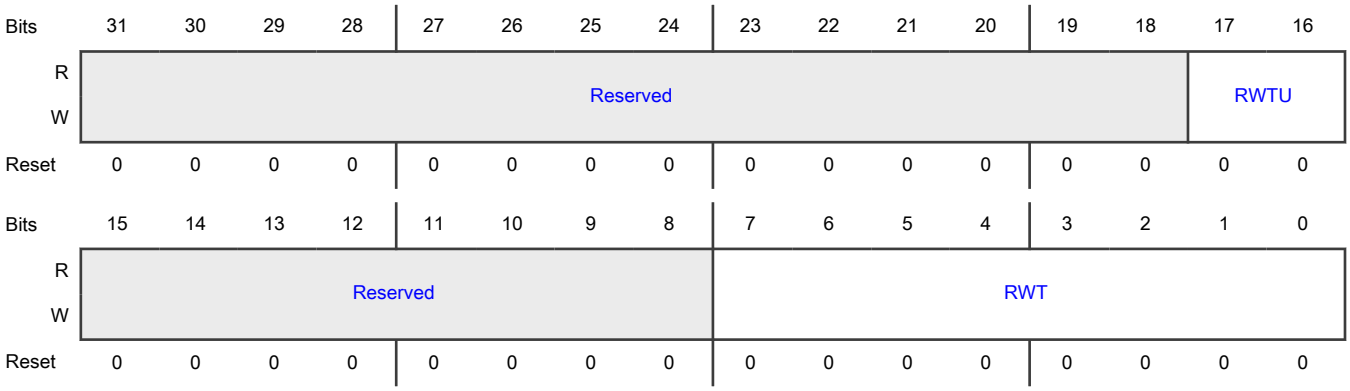
43.8.1.540 Channel 3 Receive Interrupt Watchdog Time (DMA_CH3_RX_INTERRUPT_WATCHDOG_TIMER)

The Channel 3 Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA_CH3_Status register.

Offset

Register	Offset
DMA_CH3_RX_INTERRUPT_WATCHDOG_TIMER	12B8h

Diagram



Fields

Field	Description
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8 —	Reserved
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

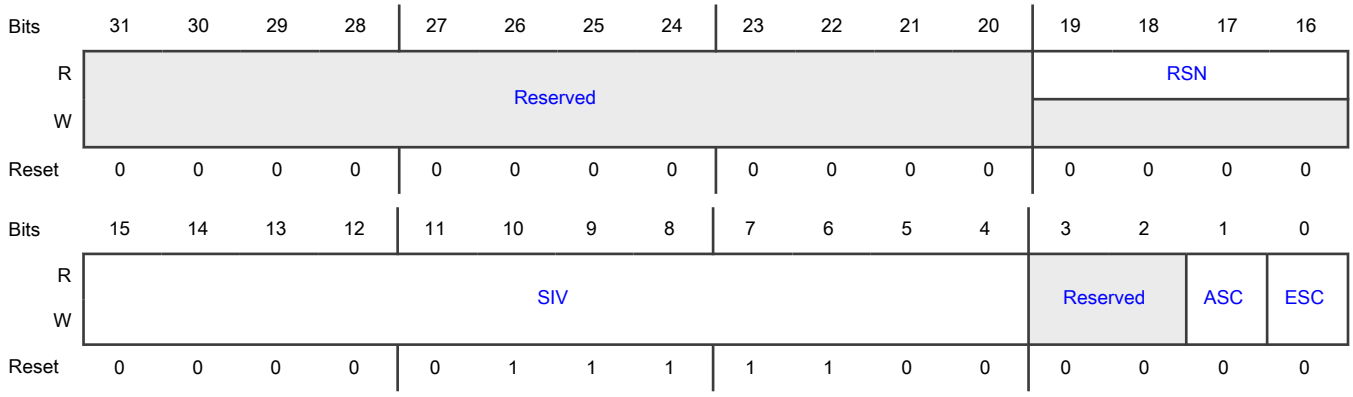
43.8.1.541 Channel 3 Slot Function Control and Status (DMA_CH3_SLOT_FUNCTION_CONTROL_STATUS)

The Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

Offset

Register	Offset
DMA_CH3_SLOT_FUNCTION_CONTROL_STATUS	12BCh

Diagram



Fields

Field	Description
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0 - Advance Slot Check is disabled 1 - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or - ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0 - Slot Comparison is disabled 1 - Slot Comparison is enabled

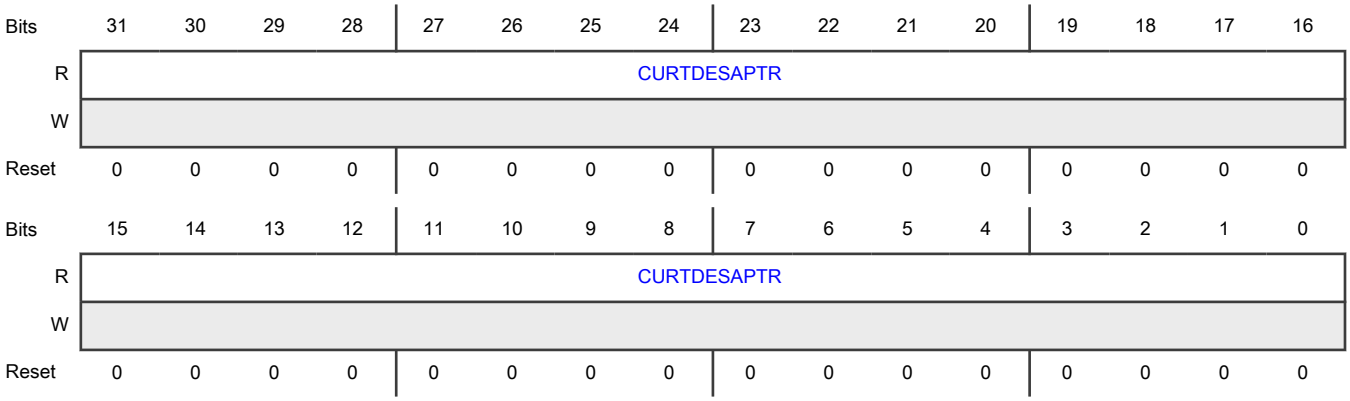
43.8.1.542 Channel 3 Current Application Transmit Descriptor (DMA_CH3_CURRENT_APP_TXDESC)

The Channel 3 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

Offset

Register	Offset
DMA_CH3_CURRENT_APP_TXDESC	12C4h

Diagram



Fields

Field	Description
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

43.8.1.543 Channel 3 Current Application Receive Descriptor (DMA_CH3_CURRENT_APP_RXDESC)

The Channel 3 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

Offset

Register	Offset
DMA_CH3_CURRENT_APP_RXDESC	12CCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CURRDESAPTR	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

43.8.1.544 Channel 3 Current Application Transmit Buffer Address (DMA_CH3_CURRENT_APP_TXBUFFER)

The Channel 3 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

Offset

Register	Offset
DMA_CH3_CURRENT_APP_TXBUFFER	12D4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

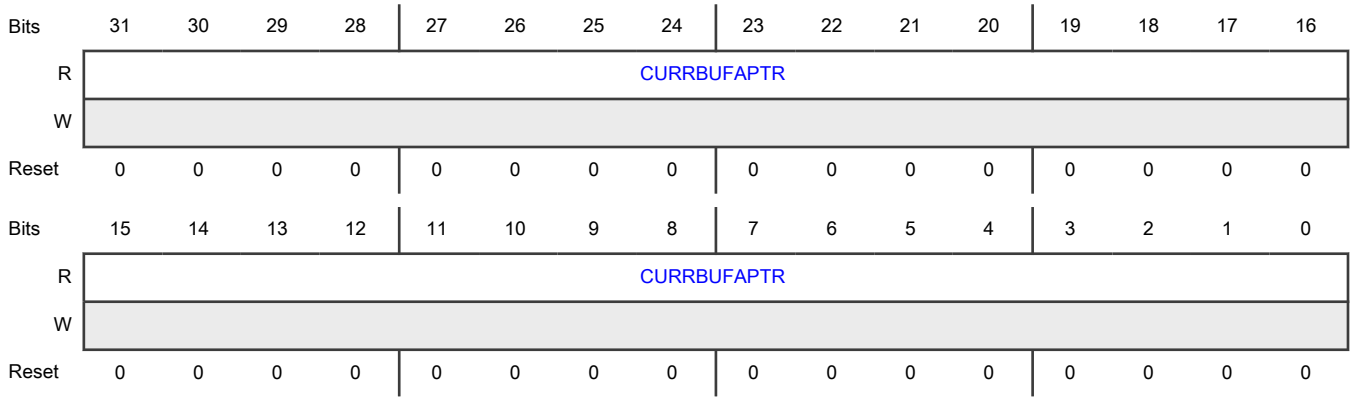
43.8.1.545 Channel 3 Current Application Receive Buffer Address (DMA_CH3_CURRENT_APP_RXBUFFER)

The Channel 3 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

Offset

Register	Offset
DMA_CH3_CURRENT_APP_RXBUFFER	12DCh

Diagram



Fields

Field	Description
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

43.8.1.546 DMA Channel 3 Status (DMA_CH3_STATUS)

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

Offset

Register	Offset
DMA_CH3_STATUS	12E0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB		TEB			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH3_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH3_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH3_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH3_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Normal Interrupt Summary status not detected 1 - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH3_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH3_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Abnormal Interrupt Summary status not detected 1 - Abnormal Interrupt Summary status detected
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Context Descriptor Error status not detected 1 - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Fatal Bus Error status not detected 1 - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Receive Interrupt status not detected 1 - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Transmit Interrupt status not detected 1 - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0 - Receive Watchdog Timeout status not detected 1 - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Process Stopped status not detected 1 - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode,

Table continues on the next page...

Table continued from the previous page...

Field	Description
	the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Buffer Unavailable status not detected 1 - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Interrupt status not detected 1 - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Buffer Unavailable status not detected 1 - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Process Stopped status not detected 1 - Transmit Process Stopped status detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Interrupt status not detected 1 - Transmit Interrupt status detected

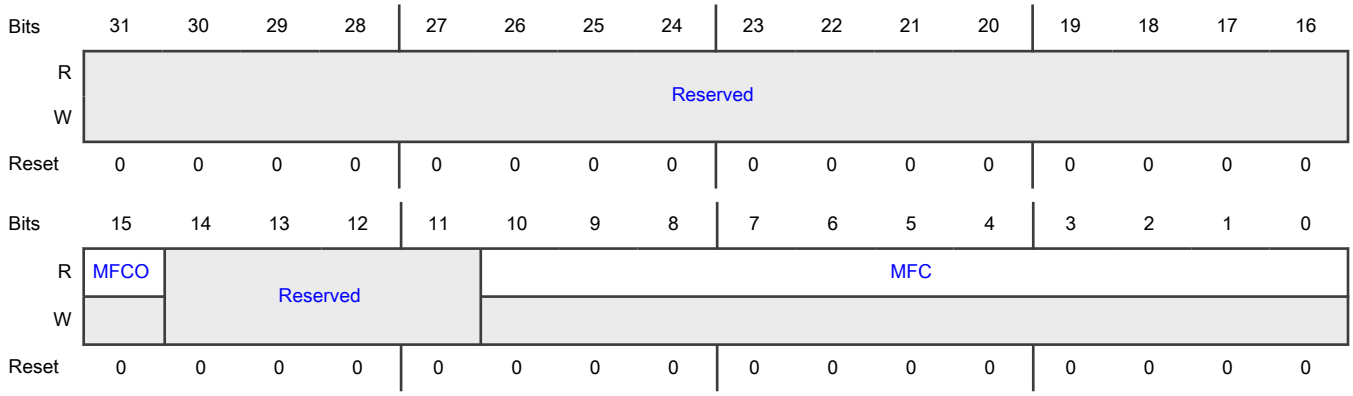
43.8.1.547 Channel 3 Missed Frame Counter (DMA_CH3_MISS_FRAME_CNT)

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA_CH3_RX_CONTROL register.

Offset

Register	Offset
DMA_CH3_MISS_FRAME_CNT	12E4h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Miss Frame Counter overflow not occurred 1 - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH3_RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

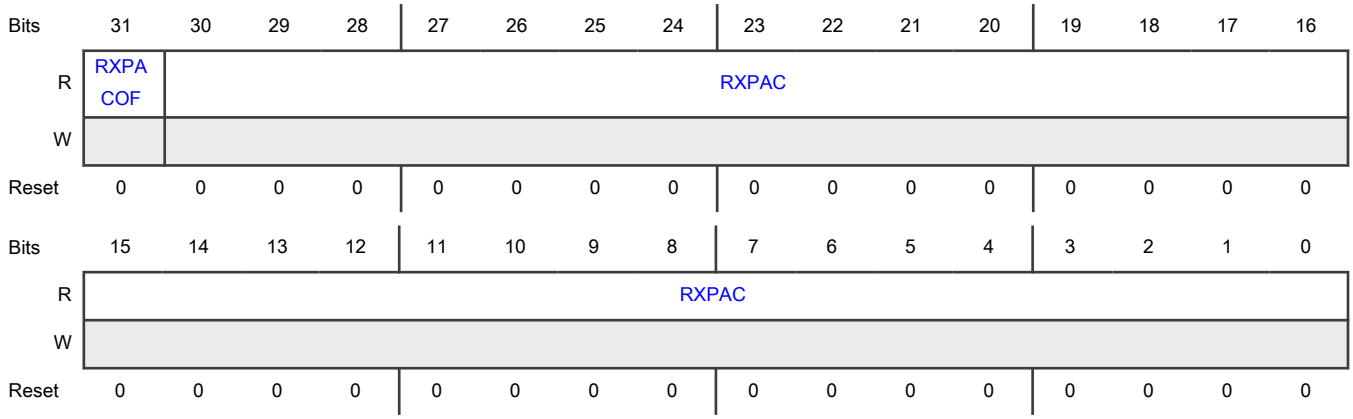
43.8.1.548 Channel 3 RXP Frames Accepted Counter (DMA_CH3_RXP_ACCEPT_CNT)

The DMA_CH3_RXP_Accept_Cnt register provides the count of the number of frames accepted by Rx Parser.

Offset

Register	Offset
DMA_CH3_RXP_ACCEPT_CNT	12E8h

Diagram



Fields

Field	Description
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Accept Counter overflow not occurred 1 - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

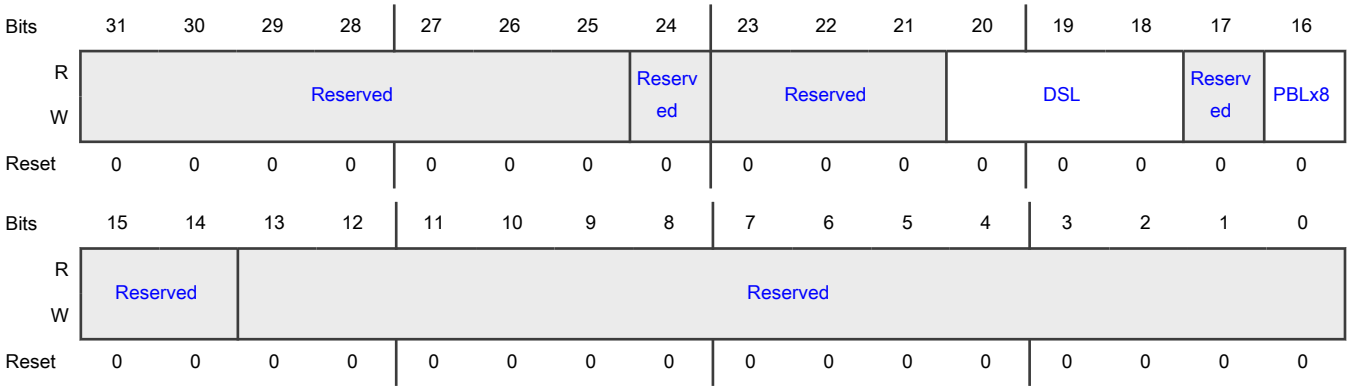
43.8.1.549 Channel 4 Control (DMA_CH4_CONTROL)

The DMA Channel 4 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

Offset

Register	Offset
DMA_CH4_CONTROL	1300h

Diagram



Fields

Field	Description
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH4_TX_CONTROL and Bits[21:16] in DMA_CH4_RX_CONTROL is multiplied by eight times. 8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH4_TX_CONTROL and Bits[21:16] in DMA_CH4_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0 - 8xPBL mode is disabled 1 - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

43.8.1.550 DMA Channel 4 Transmit Control (DMA_CH4_TX_CONTROL)

The DMA Channel 4 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

Offset

Register	Offset
DMA_CH4_TX_CONTROL	1304h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			EDSE	Reserved				Reserved	Reserved	TxPBL					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IPBL	Reserved		Reserved	Reserved							OSF	Reserved			ST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0 - Enhanced Descriptor is disabled 1 - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in CONTROL register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ4_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0 - Ignore PBL Requirement is disabled 1 - Ignore PBL Requirement is enabled
14-13 —	Reserved
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0 - Operate on Second Packet disabled 1 - Operate on Second Packet enabled
3-1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH4_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH4_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH4_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program DMA_CH4_TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0 - Stop Transmission Command 1 - Start Transmission Command

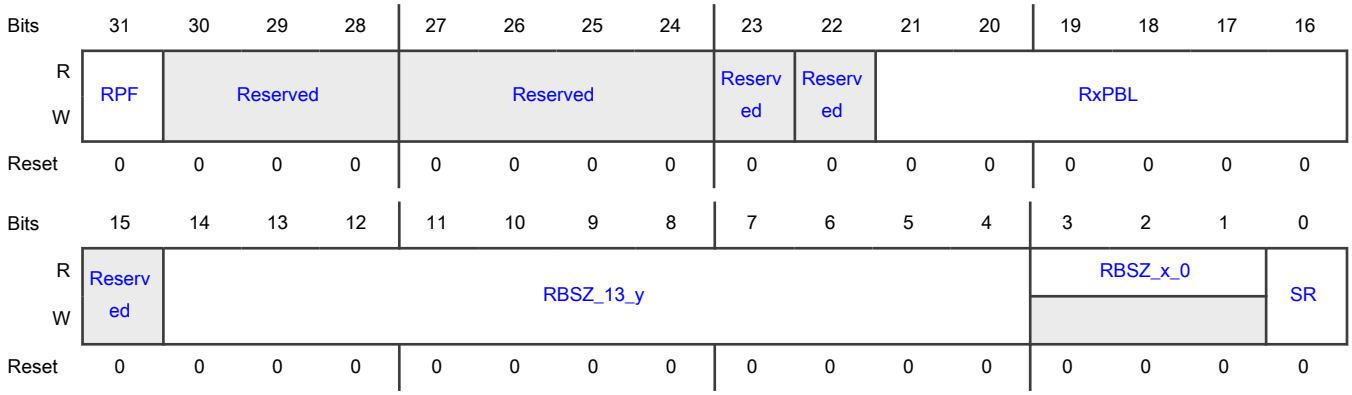
43.8.1.551 DMA Channel 4 Receive Control (DMA_CH4_RX_CONTROL)

The DMA Channel 4 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

Offset

Register	Offset
DMA_CH4_RX_CONTROL	1308h

Diagram



Fields

Field	Description
31 RPF	Rx Packet Flush. Rx Packet Flush. When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0 - Rx Packet Flush is disabled 1 - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application

Table continues on the next page...

Table continued from the previous page...

Field	Description
	bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the DMA_CH4_CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ4_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1 RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH4_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH4_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH4_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0 - Stop Receive 1 - Start Receive

43.8.1.552 Channel 4 Tx Descriptor List Address (DMA_CH4_TXDESC_LIST_ADDRESS)

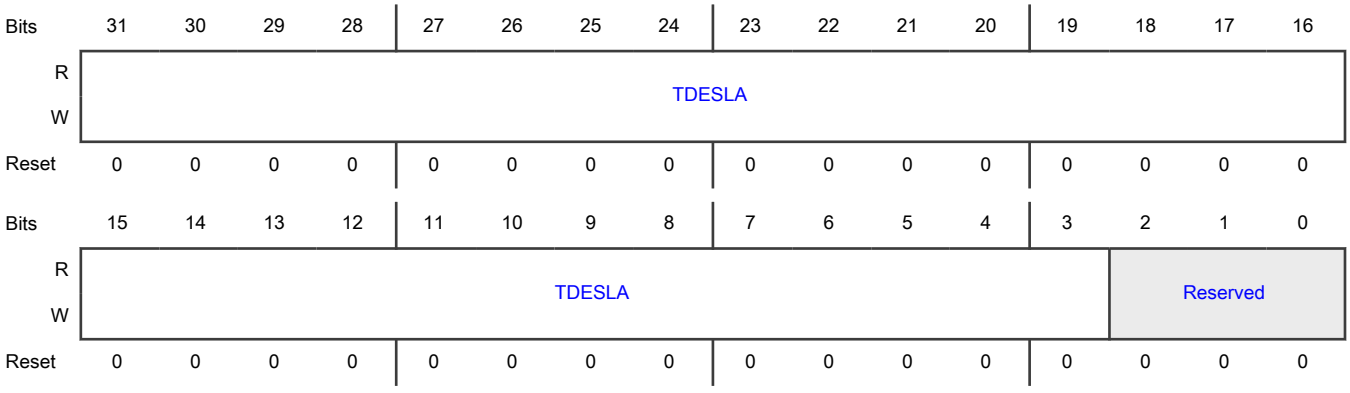
The Channel 4 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

Offset

Register	Offset
DMA_CH4_TXDESC_LIST_ADDRESS	1314h

Diagram



Fields

Field	Description
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

43.8.1.553 Channel 4 Rx Descriptor List Address (DMA_CH4_RXDESC_LIST_ADDRESS)

The Channel 4 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

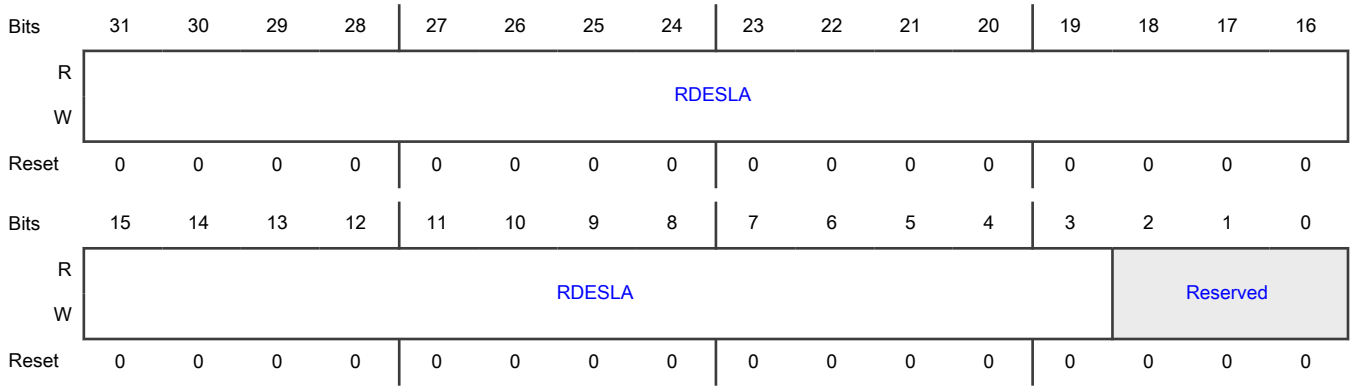
This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

Offset

Register	Offset
DMA_CH4_RXDESC_LIST_ADDRESS	131Ch

Diagram



Fields

Field	Description
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

43.8.1.554 Channel 4 Tx Descriptor Tail Pointer (DMA_CH4_TXDESC_TAIL_POINTER)

The Channel 4 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

Offset

Register	Offset
DMA_CH4_TXDESC_TAIL_POINTER	1320h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDTP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDTP													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

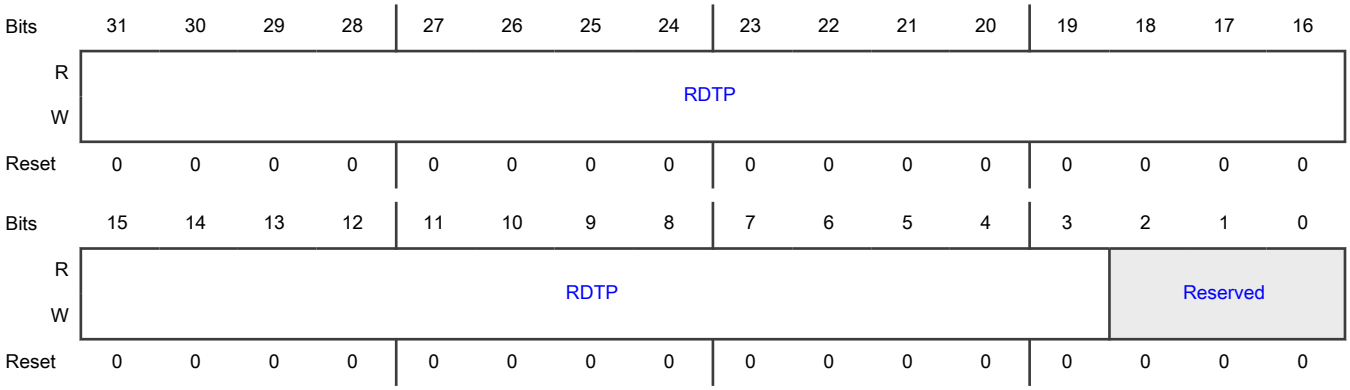
43.8.1.555 Channel 4 Rx Descriptor Tail Pointer (DMA_CH4_RXDESC_TAIL_POINTER)

The Channel 4 Rx Descriptor Tail Pointer points to an offset from the base and indicates the location of the last valid descriptor.

Offset

Register	Offset
DMA_CH4_RXDESC_TAIL_POINTER	1328h

Diagram



Fields

Field	Description
31-3 RDTP	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

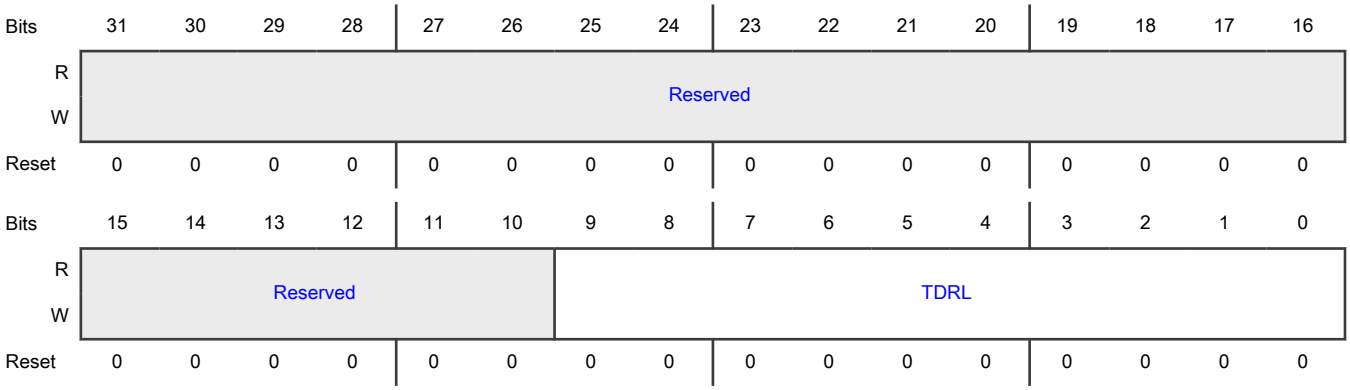
43.8.1.556 Channel 4 Tx Descriptor Ring Length (DMA_CH4_TXDESC_RING_LENGTH)

The Channel 4 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

Offset

Register	Offset
DMA_CH4_TXDESC_RING_LENGTH	132Ch

Diagram



Fields

Field	Description
31-10	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. We recommend a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

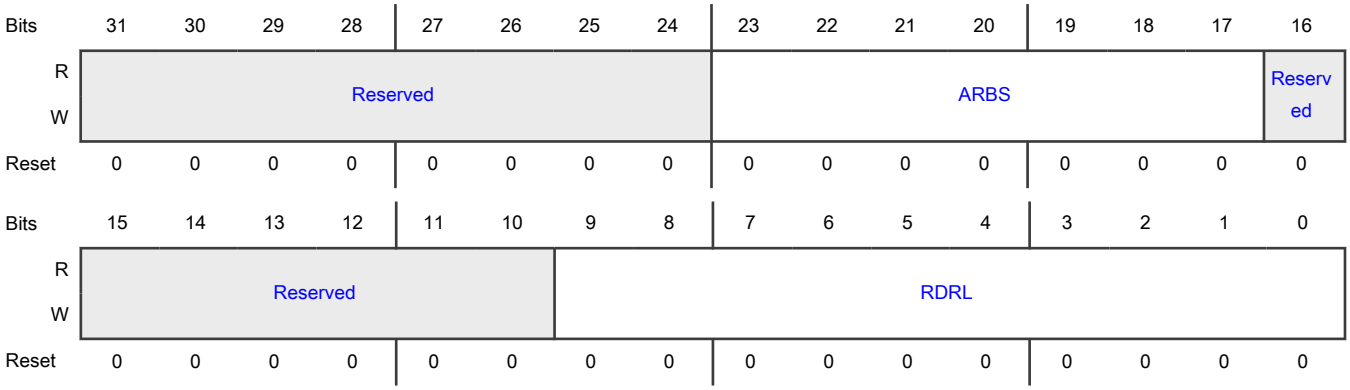
43.8.1.557 DMA Channel 4 Receive Control 2 register (DMA_CH4_RX_CONTROL2)

The Channel 4 Receive Control register controls Rx Descriptor Ring Length and Alternate Rx Buffer Size.

Offset

Register	Offset
DMA_CH4_RX_CONTROL2	1330h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-17 ARBS	Alternate Receive Buffer Size Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header

Table continues on the next page...

Table continued from the previous page...

Field	Description
	data. The maximum alternate buffer is limited to 1020, 1016 or 1008-bytes depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). When ARBS=0, Rx Buffer1 and Rx Buffer2 sizes are based on DMA_CH2_RX_CONTROL[RBSZ]. Width of ARBS field is 8, 7 or 6-bits depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively).
16-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

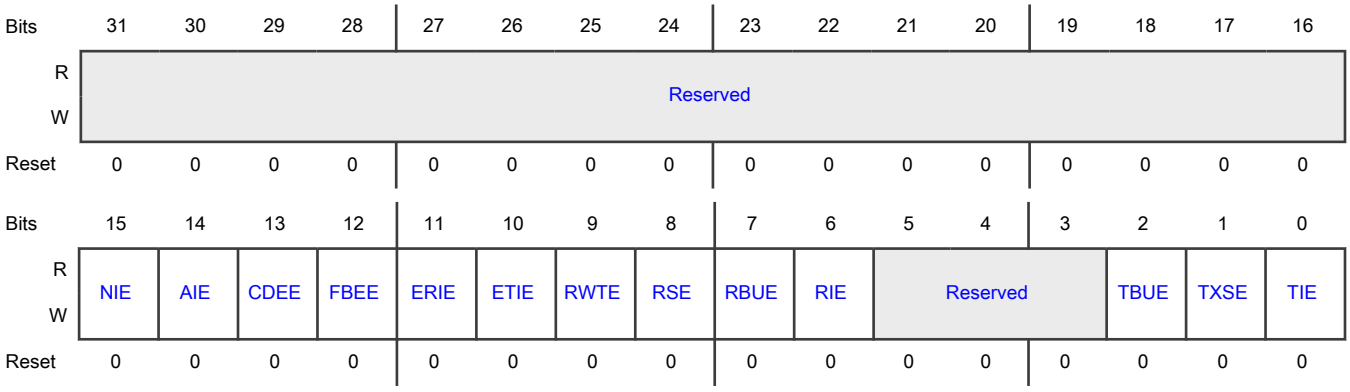
43.8.1.558 Channel 4 Interrupt Enable (DMA_CH4_INTERRUPT_ENABLE)

The Channel 4 Interrupt Enable register enables the interrupts reported by the Status register.

Offset

Register	Offset
DMA_CH4_INTERRUPT_ENABLE	1334h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0 - Normal Interrupt Summary is disabled 1 - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled. 0 - Abnormal Interrupt Summary is disabled 1 - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0 - Context Descriptor Error is disabled 1 - Context Descriptor Error is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0 - Fatal Bus Error is disabled 1 - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0 - Early Receive Interrupt is disabled 1 - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0 - Early Transmit Interrupt is disabled 1 - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0 - Receive Watchdog Timeout is disabled 1 - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0 - Receive Stopped is disabled 1 - Receive Stopped is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0 - Receive Buffer Unavailable is disabled 1 - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0 - Receive Interrupt is disabled 1 - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0 - Transmit Buffer Unavailable is disabled 1 - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0 - Transmit Stopped is disabled 1 - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0 - Transmit Interrupt is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Transmit Interrupt is enabled

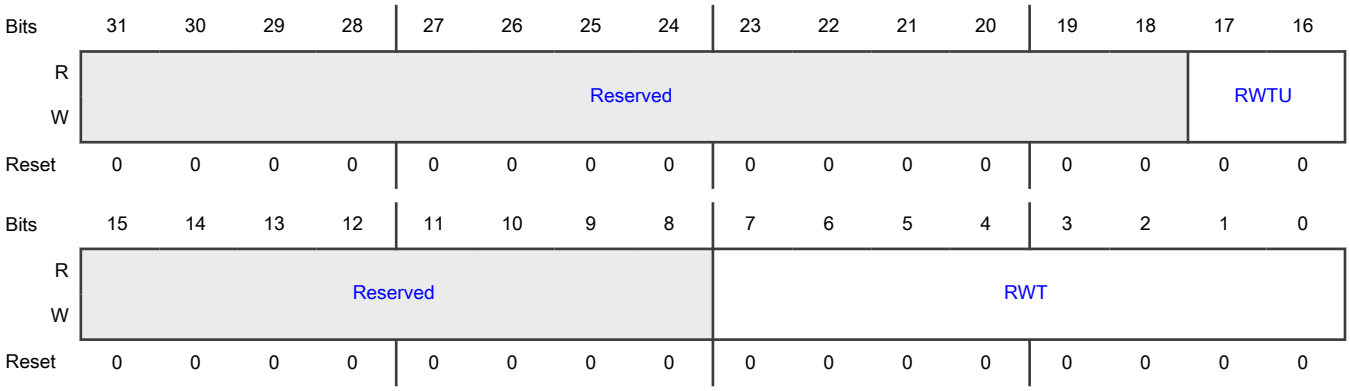
43.8.1.559 Channel 4 Receive Interrupt Watchdog Timer (DMA_CH4_RX_INTERRUPT_WATCHDOG_TIMER)

The Channel 4 Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA_CH4_Status register.

Offset

Register	Offset
DMA_CH4_RX_INTERRUPT_WATCHDOG_TIMER	1338h

Diagram



Fields

Field	Description
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024

Table continues on the next page...

Table continued from the previous page...

Field	Description
	- 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8 —	Reserved
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the DMA_CH4_STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

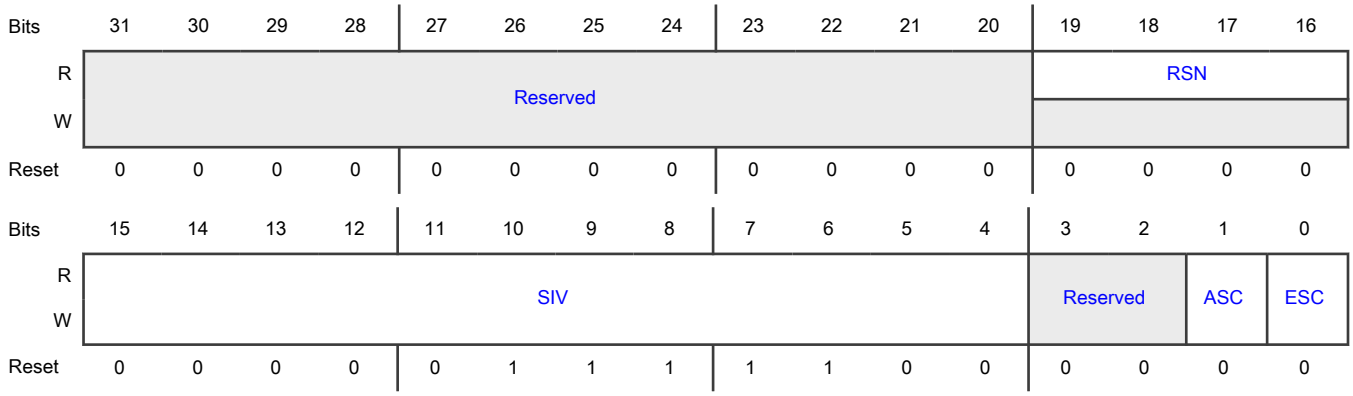
43.8.1.560 Channel 4 Slot Function Control and Status (DMA_CH4_SLOT_FUNCTION_CONTROL_STATUS)

The Channel 4 Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

Offset

Register	Offset
DMA_CH4_SLOT_FUNCTION_CONTROL_STATUS	133Ch

Diagram



Fields

Field	Description
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0 - Advance Slot Check is disabled 1 - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or - ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0 - Slot Comparison is disabled 1 - Slot Comparison is enabled

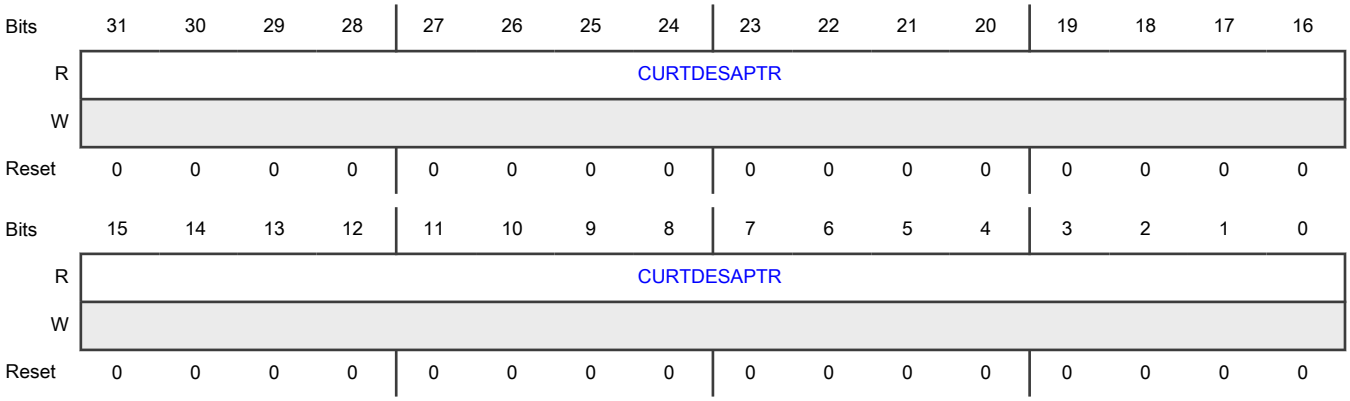
43.8.1.561 Channel 4 Current Application Transmit Descriptor (DMA_CH4_CURRENT_APP_TXDESC)

The Channel 4 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

Offset

Register	Offset
DMA_CH4_CURRENT_APP_TXDESC	1344h

Diagram



Fields

Field	Description
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

43.8.1.562 Channel 4 Current Application Receive Descriptor (DMA_CH4_CURRENT_APP_RXDESC)

The Channel 4 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

Offset

Register	Offset
DMA_CH4_CURRENT_APP_RXDESC	134Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CURRDESAPTR	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

43.8.1.563 Channel 4 Current Application Transmit Buffer Address (DMA_CH4_CURRENT_APP_TXBUFFER)

The Channel 4 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

Offset

Register	Offset
DMA_CH4_CURRENT_APP_TXBUFFER	1354h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

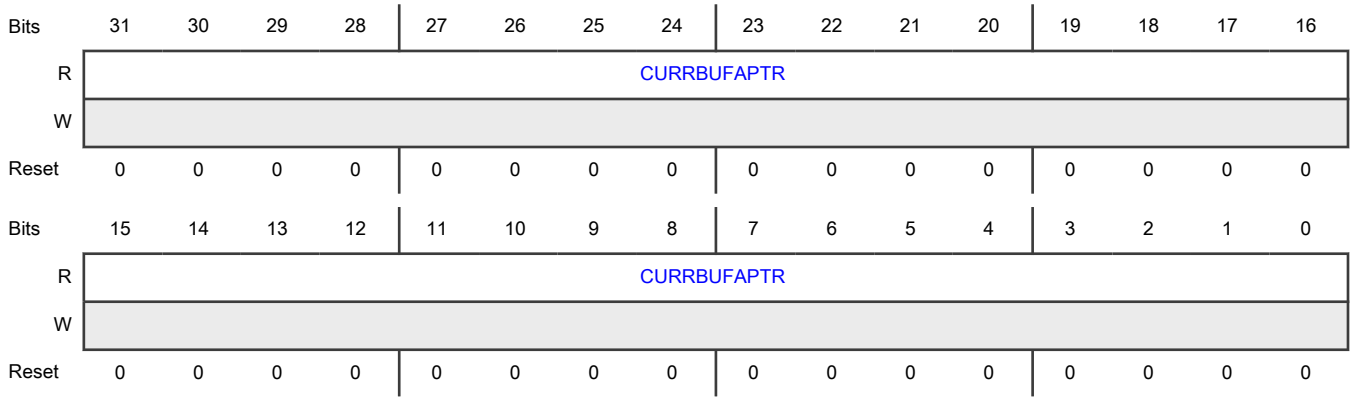
43.8.1.564 Channel 4 Current Application Receive Buffer Address (DMA_CH4_CURRENT_APP_RXBUFFER)

The Channel 4 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

Offset

Register	Offset
DMA_CH4_CURRENT_APP_RXBUFFER	135Ch

Diagram



Fields

Field	Description
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

43.8.1.565 DMA Channel 4 Status (DMA_CH4_STATUS)

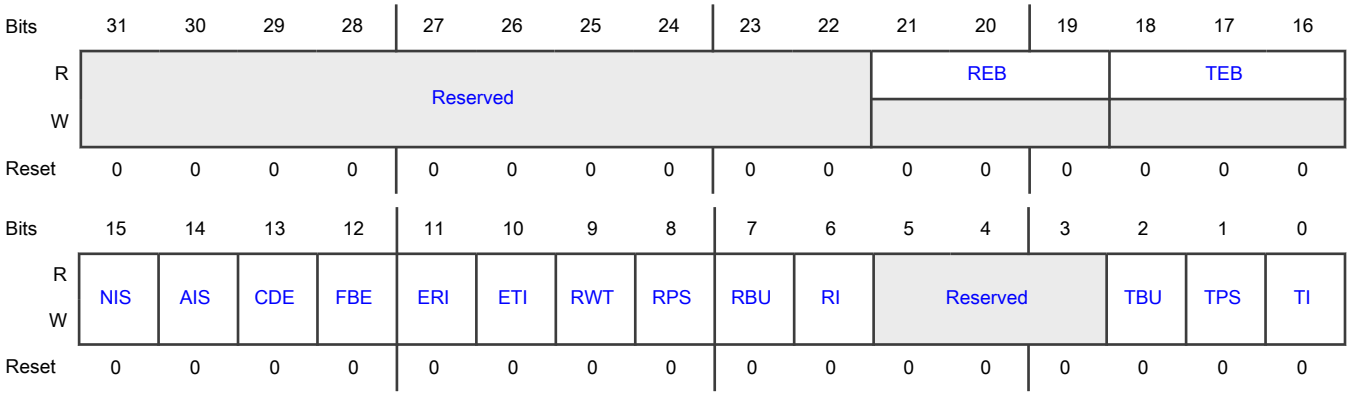
The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

Offset

Register	Offset
DMA_CH4_STATUS	1360h

Diagram



Fields

Field	Description
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Normal Interrupt Summary status not detected 1 - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Abnormal Interrupt Summary status not detected 1 - Abnormal Interrupt Summary status detected
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Context Descriptor Error status not detected 1 - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Fatal Bus Error status not detected 1 - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Receive Interrupt status not detected 1 - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Transmit Interrupt status not detected 1 - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0 - Receive Watchdog Timeout status not detected 1 - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Process Stopped status not detected 1 - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode,

Table continues on the next page...

Table continued from the previous page...

Field	Description
	the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Buffer Unavailable status not detected 1 - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Interrupt status not detected 1 - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DMA_CH4_DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Buffer Unavailable status not detected 1 - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Process Stopped status not detected 1 - Transmit Process Stopped status detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Interrupt status not detected 1 - Transmit Interrupt status detected

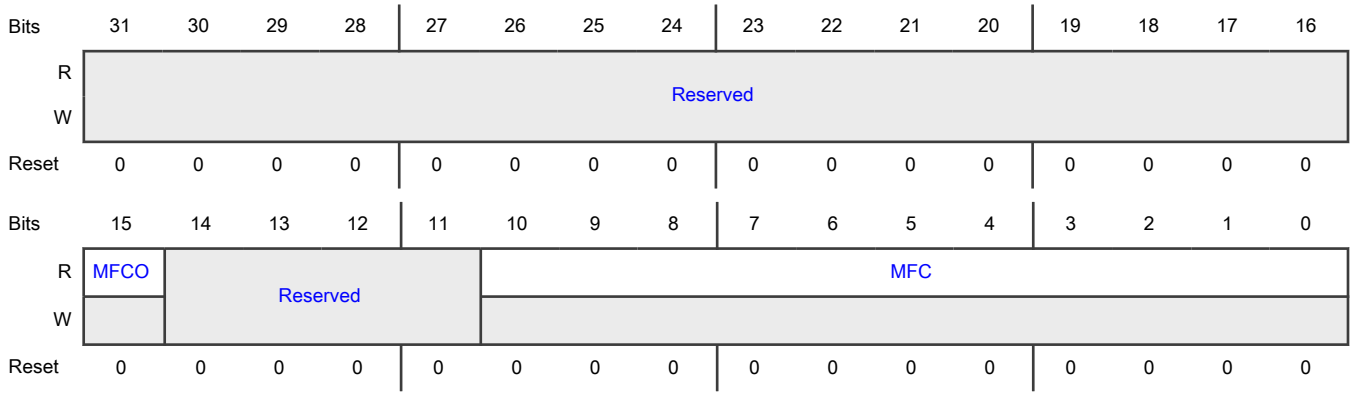
43.8.1.566 Channel 4 Missed Frame Counter (DMA_CH4_MISS_FRAME_CNT)

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA_CH4_RX_CONTROL register.

Offset

Register	Offset
DMA_CH4_MISS_FRAME_CNT	1364h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Miss Frame Counter overflow not occurred 1 - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

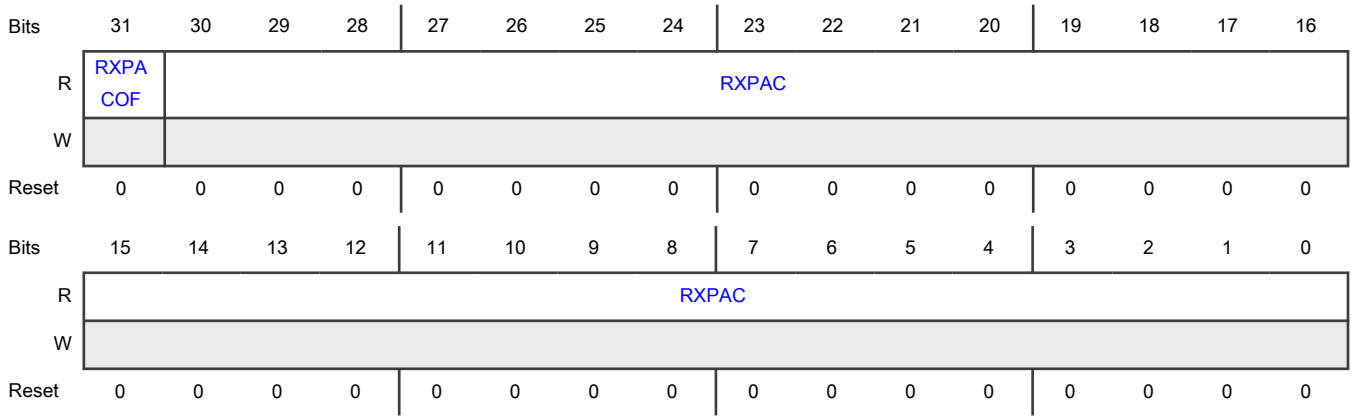
43.8.1.567 Channel 4 RXP Frames Accepted Counter (DMA_CH4_RXP_ACCEPT_CNT)

The DMA_CH4_RXP_Accept_Cnt register provides the count of the number of frames accepted by Rx Parser.

Offset

Register	Offset
DMA_CH4_RXP_ACCEPT_CNT	1368h

Diagram



Fields

Field	Description
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Accept Counter overflow not occurred 1 - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

Chapter 44
Ethernet MAC (ENET)

44.1 Overview

The core implements a triple-speed 10/100/1000-Mbit/s ENET compliant with the IEEE802.3-2002 standard. The MAC layer provides compatibility with half- or full-duplex 10/100-Mbit/s and full-duplex gigabit Ethernet LANs.

The MAC operation is fully programmable and can be used in Network Interface Card (NIC), bridging, or switching applications. The core implements the remote network monitoring (RMON) counters according to IETF RFC 2819.

The core also implements a hardware acceleration block to optimize the performance of network controllers providing TCP/IP, UDP, and ICMP protocol services. The acceleration block performs critical functions in hardware, which are typically implemented with large software overhead.

The core implements programmable embedded FIFOs that can provide buffering on the receive path for lossless flow control.

Advanced power management features are available with magic packet detection and programmable power-down modes.

A unified DMA (uDMA), internal to the ENET module, optimizes data transfer between the ENET core and the SoC, and supports an enhanced buffer descriptor programming model to support IEEE 1588 functionality.

The programmable ENET with IEEE 1588 integrates a standard IEEE 802.3 Ethernet MAC with a time-stamping module. The IEEE 1588 standard provides accurate clock synchronization for distributed control nodes for industrial automation applications.

NOTE

The terminology in this chapter has been updated to align with Annex K (informative) Optional alternative terminology, IEEE 802.3, as shown in the table below.

Table 429. Updated terms

Updated term	Deprecated term
Leader	Master
Follower	Slave

44.1.1 Block diagram

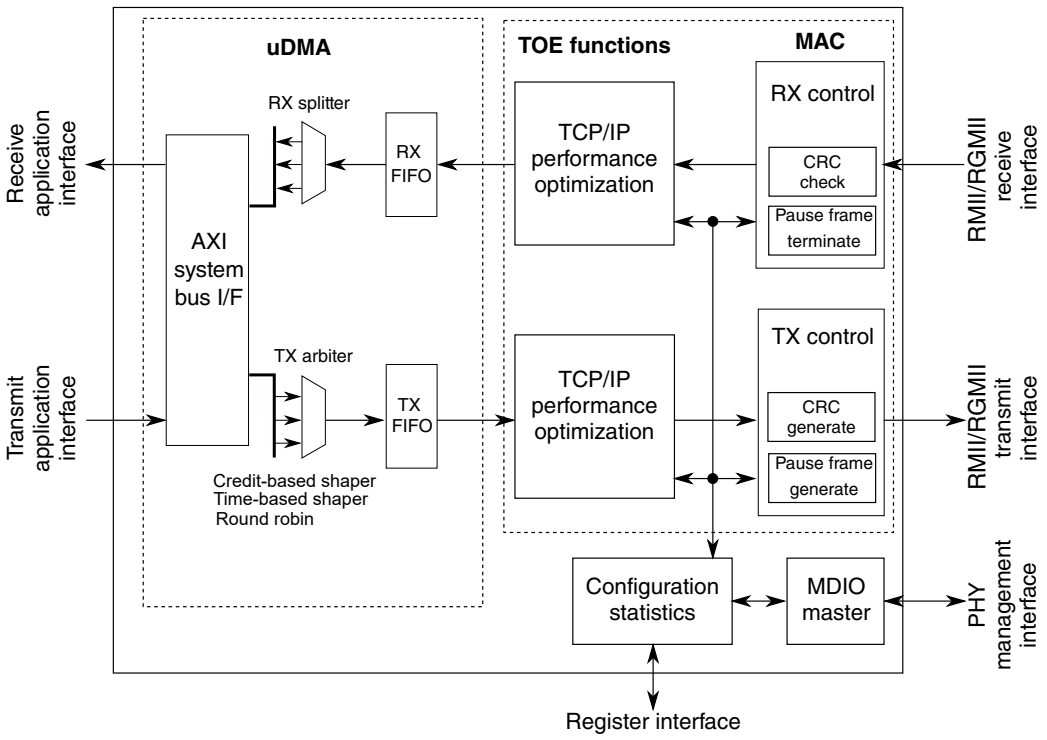


Figure 206. Ethernet MAC-NET core block diagram

44.1.2 Features

The MAC-NET core includes the following features.

44.1.2.1 Ethernet MAC features

- Implements the full 802.3 specification with preamble/SFD generation, frame padding generation, CRC generation and checking
- Supports zero-length preamble
- Dynamically configurable to support 10/100-Mbit/s and gigabit operation
- Supports 10/100 Mbit/s full-duplex and configurable half-duplex operation
- Supports gigabit full-duplex operation
- Compliant with the AMD magic packet detection with interrupt for node remote power management
- Seamless interface to commercial Ethernet PHY devices via one of the following:
  - a 2-bit Reduced MII (RMII) operating at 50 MHz.
  - a (double data rate) 4-bit Reduced GMII (RGMII) operating at 125 MHz.
- CRC-32 checking at full speed with optional forwarding of the frame check sequence (FCS) field to the client
- CRC-32 generation and append on transmit or forwarding of user application provided FCS selectable on a per-frame basis
- In full-duplex mode:
  - Supports Ethernet pause frame (802.3 Annex 31A) providing fully automated flow control without any user application overhead

- Pause quanta used to form pause frames, dynamically programmable
- Pause frame generation additionally controllable by user application offering flexible traffic flow control
- Optional forwarding of received pause frames to the user application
- Implements standard flow-control mechanism

- In half-duplex mode: provides full collision support, including jamming, backoff, and automatic retransmission
- Supports VLAN-tagged frames according to IEEE 802.1Q
- Programmable MAC address: Insertion on transmit; discards frames with mismatching destination address on receive (except broadcast and pause frames)
- Programmable promiscuous mode support to omit MAC destination address checking on receive
- Multicast and unicast address filtering on receive based on 64-entry hash table, reducing higher layer processing load
- Programmable frame maximum length providing support for any standard or proprietary frame length
- Statistics indicators for frame traffic and errors (alignment, CRC, length) and pause frames providing for IEEE 802.3 basic and mandatory management information database (MIB) package and remote network monitoring (RFC 2819)
- Simple handshake user application FIFO interface with fully programmable depth and threshold levels
- Provides separate status word for each received frame on the user interface providing information such as frame length, frame type, VLAN tag, and error information
- Supports internal loopback option
- Programmable Clause 22 and Clause 45 MDIO Leader interface for PHY device configuration and management
- Supports legacy FEC buffer descriptors
- Supports interrupt coalescing
- Supports credit-based shaping and round robin QoS scheme for traffic shaping bandwidth distribution. Either policy can be combined with time-based shaping.
- AVB (Audio Video Bridging, IEEE 802.1Qav) features:
  - Credit-based bandwidth distribution policy can be combined with time-based shaping
  - AVB endpoint talker and listener support
  - Support for arbitration between different priority traffic (for example, AVB class A, AVB class B, and non-AVB)

44.1.2.2 IP protocol performance optimization features

- Operates on TCP/IP and UDP/IP and ICMP/IP protocol data or IP header only
- Enables wire-speed processing
- Supports IPv4 and IPv6
- Transparent passing of frames of other types and protocols
- Supports VLAN tagged frames according to IEEE 802.1q with transparent forwarding of VLAN tag and control field
- Automatic IP-header and payload (protocol specific) checksum calculation and verification on receive
- Automatic IP-header and payload (protocol specific) checksum generation and insertion on transmit. Configurable on a per-frame basis
- Supports IP and TCP, UDP, ICMP data for checksum generation and checking
- Supports full header options for IPv4 and TCP protocol headers
- Provides IPv6 support to datagrams with base header only — datagrams with extension headers are passed transparently unmodifed/unchecked

- Provides statistics information for received IP and protocol errors
- Configurable automatic discard of erroneous frames
- Configurable automatic host-to-network (RX) and network-to-host (TX) byte order conversion for IP and TCP/UDP/ICMP headers within the frame
- Configurable padding remove for short IP datagrams on receive
- Configurable Ethernet payload alignment to allow for 32-bit word-aligned header and payload processing
- Programmable store-and-forward operation with clock and rate decoupling FIFOs

44.1.2.3 IEEE 1588 features

- Supports all IEEE 1588 frames.
- Allows reference clock to be chosen independently of network speed.
- Software-programmable precise time-stamping of ingress and egress frames
- Timer monitoring capabilities for system calibration and timing accuracy management
- Precise time-stamping of external events with programmable interrupt generation
- Programmable event and interrupt generation for external system control
- Supports hardware- and software-controllable timer synchronization.
- Provides a 4-channel IEEE 1588 timer. Each channel supports input capture and output compare using the 1588 counter.

44.2 Functional description

This section provides a complete functional description of the MAC-NET core.

44.2.1 Ethernet MAC frame formats

- Minimum length of 64 bytes
- Maximum length of 1518 bytes excluding the preamble and the start frame delimiter (SFD) bytes

An Ethernet frame consists of the following fields:

- Seven bytes preamble
- Start frame delimiter (SFD)
- Two address fields
- Length or type field
- Data field
- Frame check sequence (CRC value)
- Extension field is defined only for Gigabit Ethernet half-duplex implementations and is not supported by the MAC core

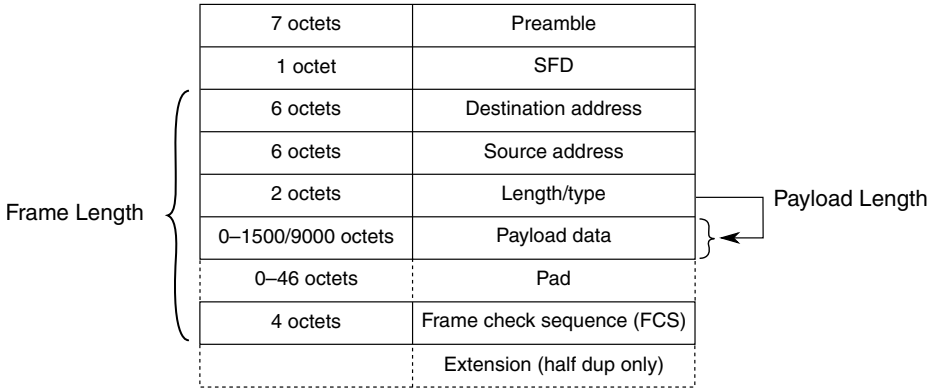


Figure 207. MAC frame format overview

Optionally, MAC frames can be VLAN-tagged with an additional four-byte field inserted between the MAC source address and the type/length field. VLAN tagging is defined by the IEEE P802.1q specification. VLAN-tagged frames have a maximum length of 1522 bytes, excluding the preamble and the SFD bytes.

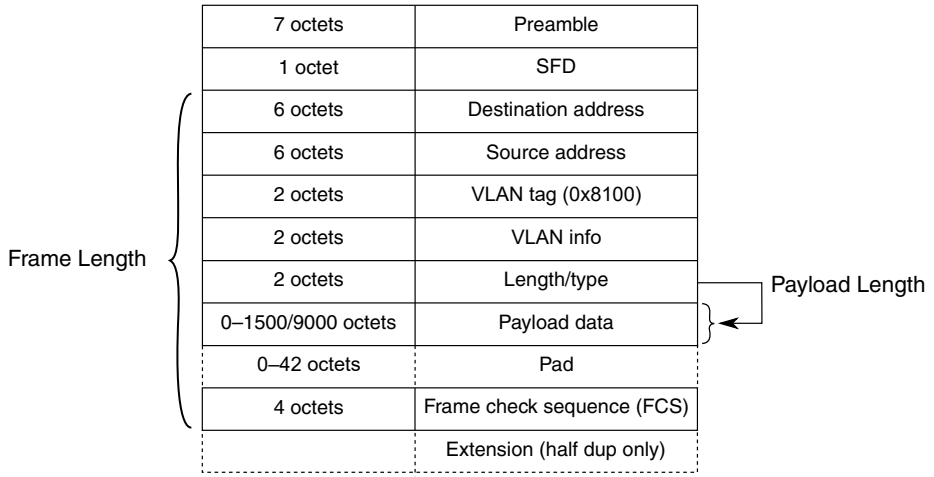


Figure 208. VLAN-tagged MAC frame format overview

Table 430. MAC frame definition

Term	Description
Frame length	Defines the length, in octets, of the complete frame without preamble and SFD. A frame has a valid length if it contains at least 64 octets and does not exceed the programmed maximum length.
Payload length	The length/type field indicates the length of the frame's payload section. The most significant byte is sent/received first. • If the length/type field is set to a value less than 46, the payload is padded so that the minimum frame length requirement (64 bytes) is met. For VLAN-tagged frames, a value less than 42 indicates a padded frame. • If the length/type field is set to a value larger than the programmed frame maximum length (e.g. 1518) it is interpreted as a type field.
Destination and source address	48-bit MAC addresses. The least significant byte is sent/received first and the first two least significant bits of the MAC address distinguish MAC frames, as detailed in MAC address check.

NOTE

Although the IEEE specification defines a maximum frame length, the MAC core provides the flexibility to program any value for the frame maximum length.

44.2.1.1 Pause frames

The receiving device generates a pause frame to indicate a congestion to the emitting device, which should stop sending data.

Pause frames are indicated by the length/type set to 0x8808. The two first bytes of a pause frame following the type, defines a 16-bit opcode field set to 0x0001 always. A 16-bit pause quanta is defined in the frame payload bytes 2 (P1) and 3 (P2) as defined in the following table. The P1 pause quanta byte is the most significant.

Table 431. Pause Frame Format (Values in Hex)

1	2	3	4	5	6	7	8	9	10	11	12	13	14
55	55	55	55	55	55	55	D5	01	80	C2	00	00	01
Preamble							SFD	Multicast Destination Address					
15	16	17	18	19	20	21	22	23	24	25	26	27 –68	
00	00	00	00	00	00	88	08	00	01	hi	lo	00	
Source Address						Type		Opcode		P1	P2	pad (42)	
69	70	71	72										
26	6B	AE	0A										
CRC-32													

There is no payload length field found within a pause frame and a pause frame is always padded with 42 bytes (0x00).

If a pause frame with a pause value greater than zero (XOFF condition) is received, the MAC stops transmitting data as soon the current frame transfer is completed. The MAC stops transmitting data for the value defined in pause quanta. One pause quanta fraction refers to 512 bit times.

If a pause frame with a pause value of zero (XON condition) is received, the transmitter is allowed to send data immediately (see Full-duplex flow control operation for details).

44.2.1.2 Magic packets

A magic packet is a unicast, multicast, or broadcast packet, which carries a defined sequence in the payload section. Magic packets are received and inspected only under specific conditions as described in Magic packet detection.

The defined sequence to decode a magic packet is formed with a synchronization stream which consists of six consecutive 0xFF bytes, and is followed by sequence of sixteen consecutive unicast MAC addresses of the node to be awakened.

This sequence can be located anywhere in the magic packet payload. The magic packet is formed with a standard Ethernet header, optional padding, and CRC.

44.2.2 IP and higher layers frame format

The following sections use the term datagram to describe the protocol specific data unit that is found within the payload section of its container entity.

For example, an IP datagram specifies the payload section of an Ethernet frame. A TCP datagram specifies the payload section within an IP datagram.

44.2.2.1 Ethernet types

IP datagrams are carried in the payload section of an Ethernet frame. The Ethernet frame type/length field discriminates several datagram types. The following table lists the types of interest:

Table 432. Ethernet type value examples

Type	Description
0x8100	VLAN-tagged frame. The actual type is found 4 octets later in the frame.
0x0800	IPv4
0x0806	ARP
0x86DD	IPv6

44.2.2.2 IPv4 datagram format

The following figure shows the IP Version 4 (IPv4) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words. The first byte sent/received is the leftmost byte of the first word (in other words, version/IHL field).

The IP header can contain further options, which are always padded if necessary to guarantee the payload following the header is aligned to a 32-bit boundary.

The IP header is immediately followed by the payload, which can contain further protocol headers (for example, TCP or UDP, as indicated by the protocol field value). The complete IP datagram is transported in the payload section of an Ethernet frame.

Table 433. IPv4 header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Version	IHL	TOS		Length			
Fragment ID				Flags	Fragment offset		
TTL		Protocol		Header checksum			
Source address							
Destination address							
Options							

Table 434. IPv4 header fields

Field name	Description
Version	4-bit IP version information. 0x4 for IPv4 frames.
IHL	4-bit Internet header length information. Determines number of 32-bit words found within the IP header. If no options are present, the default value is 0x5.
TOS	Type of service/DiffServ field.

Table continues on the next page...

Table 434. IPv4 header fields (continued)

Field name	Description
Length	Total length of the datagram in bytes, including all octets of header and payload.
Fragment ID, flags, fragment offset	Fields used for IP fragmentation.
TTL	Time-to-live. In effect, is decremented at each router arrival. If zero, datagram must be discarded.
Protocol	Identifier of protocol that follows in the datagram.
Header checksum	Checksum of IP header. For computational purposes, this field's value is zero.
Source address	Source IP address.
Destination address	Destination IP address.

44.2.2.3 IPv6 datagram format

The following figure shows the IP version 6 (IPv6) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words and has a fixed length of ten words (40 bytes). The next header field identifies the type of the header that follows the IPv6 header. It is defined similar to the protocol identifier within IPv4, with new definitions for identifying extension headers. These headers can be inserted between the IPv6 header and the protocol header, which will shift the protocol header accordingly. The accelerator currently only supports IPv6 without extension headers (in other words, the next header specifies TCP, UDP, or IMCP).

The first byte sent/received is the leftmost byte of the first word (in other words, version/traffic class fields).

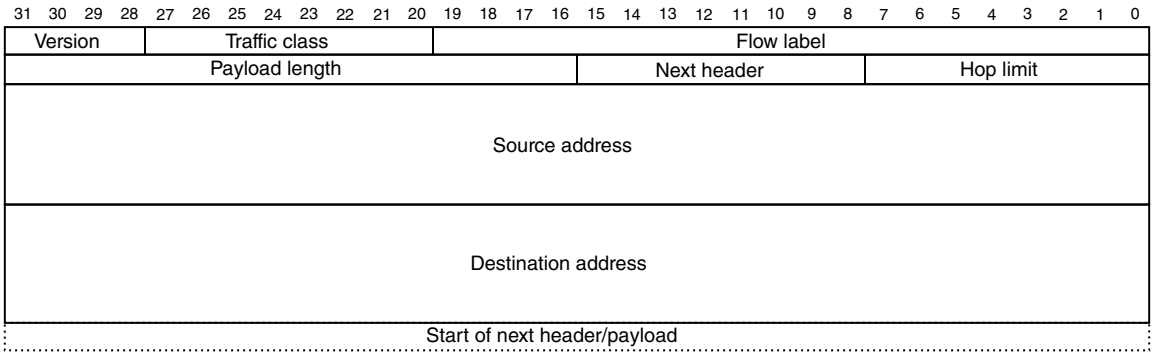


Figure 209. IPv6 header format

Table 435. IPv6 header fields

Field name	Description
Version	4-bit IP version information. 0x6 for all IPv6 frames.
Traffic class	8-bit field defining the traffic class.
Flow label	20-bit flow label identifying frames of the same flow.

Table continues on the next page...

Table 435. IPv6 header fields (continued)

Field name	Description
Payload length	16-bit length of the datagram payload in bytes. It includes all octets following the IPv6 header.
Next header	Identifies the header that follows the IPv6 header. This can be the protocol header or any IPv6 defined extension header.
Hop limit	Hop counter, decremented by one by each station that forwards the frame. If hop limit is 0 the frame must be discarded.
Source address	128-bit IPv6 source address.
Destination address	128-bit IPv6 destination address.

44.2.2.4 Internet Control Message Protocol (ICMP) datagram format

An internet control message protocol (ICMP) is found following the IP header, if the protocol identifier is 1. The ICMP datagram has a four-octet header followed by additional message data.

Table 436. ICMP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Type		Code		Checksum			
ICMP message data							

Table 437. IP header fields

Field name	Description
Type	8-bit type information
Code	8-bit code that is related to the message type
Checksum	16-bit one's complement checksum over the complete ICMP datagram

44.2.2.5 User Datagram Protocol (UDP) datagram format

A user datagram protocol header is found after the IP header, when the protocol identifier is 17.

The payload of the datagram is after the UDP header. The header byte order follows the conventions given for the IP header above.

Table 438. UDP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source port				Destination port			
Length				Checksum			

Table 439. UDP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Length	Length of user data which immediately follows the header, including the UDP header (that is, minimum value is 8)
Checksum	Checksum over the complete datagram and some IP header information

44.2.2.6 TCP datagram format

A TCP header is found following the IP header, when the protocol identifier has a value of 6. The TCP payload immediately follows the TCP header.

Table 440. TCP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10	9 8	7 6 5 4	3 2 1 0
Source port				Destination port				
Sequence number								
Acknowledgement number								
Offset	Reserved		Flags	Window				
Checksum				Urgent pointer				
Options								

Table 441. TCP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Sequence number	Transmit sequence number
Ack. number	Receive sequence number
Offset	Data offset, which is number of 32-bit words within TCP header — if no options selected, defaults to value of 5
Flags	URG, ACK, PSH, RST, SYN, FIN flags
Window	TCP receive window size information

Table continues on the next page...

Table 441. TCP header fields (continued)

Field name	Description
Checksum	Checksum over the complete datagram (TCP header and data) and IP header information
Options	Additional 32-bit words for protocol options

44.2.3 IEEE 1588 message formats

The following sections describe the IEEE 1588 message formats.

44.2.3.1 Transport encapsulation

The precision time protocol (PTP) datagrams are encapsulated in Ethernet frames using the UDP/IP transport mechanism, or optionally, with the newer 1588v2 directly in Ethernet frames (layer 2). Typically, multicast addresses are used to allow efficient distribution of the synchronization messages.

44.2.3.1.1 UDP/IP

The 1588 messages (v1 and v2) can be transported using UDP/IP multicast messages. The table below shows IP multicast groups defined for PTP. The table also shows their respective MAC layer multicast address mapping according to RFC 1112 (last three octets of IP follow the fixed value of 01-00-5E).

Table 442. UDP/IP multicast domains

Name	IP Address	MAC Address mapping
DefaultPTPdomain	224.0.1.129	01-00-5E-00-01-81
AlternatePTPdomain1	224.0.1.130	01-00-5E-00-01-82
AlternatePTPdomain2	224.0.1.131	01-00-5E-00-01-83
AlternatePTPdomain3	224.0.1.132	01-00-5E-00-01-84

Table 443. UDP port numbers

Message type	UDP port	Note
Event	319	Used for SYNC and DELAY_REQUEST messages
General	320	All other messages (for example, follow-up, delay-response)

44.2.3.1.2 Native Ethernet (PTPv2)

In addition to using UDP/IP frames, IEEE 1588v2 defines a native Ethernet frame format that uses ethertype = 0x88F7. The payload of the Ethernet frame immediately contains the PTP datagram, starting with the PTPv2 header.

Besides others, version 2 adds a peer delay mechanism to allow delay measurements between individual point-to-point links along a path over multiple nodes. The following multicast domains are also defined in PTPv2.

Table 444. PTPv2 multicast domains

Name	MAC address
Normal messages	01-1B-19-00-00-00
Peer delay messages	01-80-C2-00-00-0E

44.2.3.2 PTP header

All PTP frames contain a common header that determines the protocol version and the type of message, which defines the remaining content of the message. All multi-octet fields are transmitted in big-endian order (the most significant byte is transmitted/received first).

The last four bits of versionPTP are at the same position (second byte) for PTPv1 and PTPv2 headers. This allows accurate identification by inspecting the first two bytes of the message.

44.2.3.2.1 PTPv1 header

Table 445. Common PTPv1 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	2	versionPTP = 0x0001							
2	2	versionNetwork							
4	16	subdomain							
20	1	messageType							
21	1	sourceCommunicationTechnology							
22	6	sourceUuid							
28	2	sourcePortId							
30	2	sequenceId							
32	1	control							
33	1	0x00							
34	2	flags							
36	4	reserved							

The type of message is encoded in the messageType and control fields as shown in Table 446:

Table 446. PTPv1 message type identification

messageType	control	Message Name	Message
0x01	0x0	SYNC	Event message
0x01	0x1	DELAY_REQ	Event message
0x02	0x2	FOLLOW_UP	General message
0x02	0x3	DELAY_RESP	General message
0x02	0x4	MANAGEMENT	General message
other	other	—	Reserved

The field sequenceId is used to non-ambiguously identify a message.

44.2.3.2.2 PTPv2 header

Table 447. Common PTPv2 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	1	transportSpecific				messageId			
1	1	reserved				versionPTP = 0x2			
2	2	messageLength							
4	1	domainNumber							
5	1	reserved							
6	2	flags							
8	8	correctionField							
16	4	reserved							
20	10	sourcePortIdentity							
30	2	sequenceId							
32	1	control							
33	1	logMeanMessageInterval							

The type of message is encoded in the field messageId as follows:

Table 448. PTPv2 message type identification

messageId	Message name	Message
0x0	SYNC	Event message
0x1	DELAY_REQ	Event message
0x2	PATH_DELAY_REQ	Event message
0x3	PATH_DELAY_RESP	Event message
0x4–0x7	—	Reserved
0x8	FOLLOW_UP	General message
0x9	DELAY_RESP	General message
0xa	PATH_DELAY_FOLLOW_UP	General message
0xb	ANNOUNCE	General message
0xc	SIGNALING	General message
0xd	MANAGEMENT	General message

The PTPv2 flags field contains further details on the type of message, especially if one-step or two-step implementations are used. The one- or two-step implementation is controlled by the TWO_STEP bit in the first octet of the flags field as shown below. Reserved bits are cleared.

Table 449. PTPv2 message flags field definitions

Bit	Name	Description
0	ALTERNATE_MASTER	See IEEE 1588 Clause 17.4
1	TWO_STEP	1 Two-step clock 0 One-step clock
2	UNICAST	1 Transport layer address uses a unicast destination address 0 Multicast is used
3	—	Reserved
4	—	Reserved
5	Profile specific	
6	Profile specific	
7	—	Reserved

44.2.4 MAC receive

The MAC receive engine performs the following tasks:

- Check frame framing
- Remove frame preamble and frame SFD field
- Discard frame based on frame destination address field
- Terminate pause frames
- Check frame length
- Remove payload padding if it exists
- Calculate and verify CRC-32
- Write received frames in the core receive FIFO

If the MAC is programmed to operate in half-duplex mode, it will also check if the frame is received with a collision.

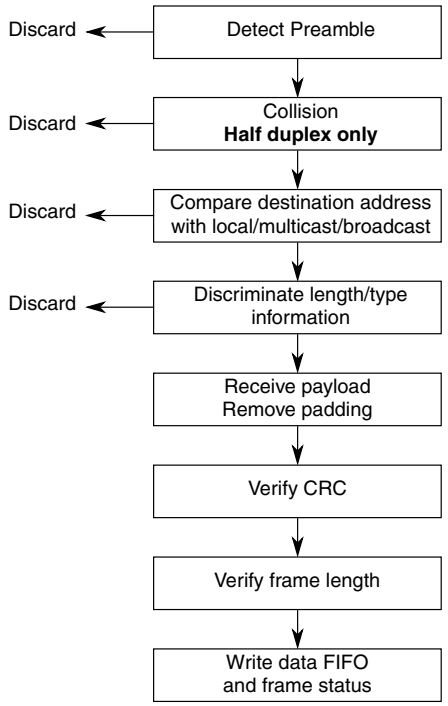


Figure 210. MAC receive flow

44.2.4.1 Collision detection in half-duplex mode

If the packet is received with a collision detected during reception of the first 64 bytes, the packet is discarded (if frame size was less than ~14 octets) or transmitted to the user application with an error and RxBD[CE] set.

44.2.4.2 Preamble processing

The IEEE 802.3 standard allows a maximum size of 56 bits (seven bytes) for the preamble, while the MAC core allows any preamble length, including zero length preamble. The MAC core checks for the start frame delimiter (SFD) byte. If the next byte of the preamble, which is different from 0x55, is not 0xD5, the frame is discarded.

Although the IEEE specification dictates that the inner-packet gap should be at least 96 bits, the MAC core is designed to accept frames separated by only 64 10/100-Mbit/s operation (MII) bits.

The MAC core removes the preamble and SFD bytes.

44.2.4.3 MAC address check

The destination address bit 0 differentiates between multicast and unicast addresses.

- If bit 0 is 0, the MAC address is an individual (unicast) address.
- If bit 0 is 1, the MAC address defines a group (multicast) address.
- If all 48 bits of the MAC address are set, it indicates a broadcast address.

44.2.4.3.1 Unicast address check

If a unicast address is received, the destination MAC address is compared to the node MAC address programmed by the host in the PADDR1/2 registers. If the destination address matches any of the programmed MAC addresses, the frame is accepted.

If Promiscuous mode is enabled (RCR[PROM] = 1) no address checking is performed and all unicast frames are accepted.

44.2.4.3.2 Multicast and unicast address resolution

The hash table algorithm used in the group and individual hash filtering operates as follows.

- The 48-bit destination address is mapped into one of 64 bits, represented by 64 bits in ENET*n*_GAUR/GALR (group address hash match) or ENET*n*_IAUR/IALR (individual address hash match).
- This mapping is performed by passing the 48-bit address through the on-chip 32-bit CRC generator and selecting the six most significant bits of the CRC-encoded result to generate a number between 0 and 63.
- The msb of the CRC result selects ENET*n*_GAUR (msb = 1) or ENET*n*_GALR (msb = 0).
- The five least significant bits of the hash result select the bit within the selected register.
- If the CRC generator selects a bit set in the hash table, the frame is accepted; else, it is rejected.

For example, if eight group addresses are stored in the hash table and random group addresses are received, the hash table prevents roughly 56/64 (or 87.5%) of the group address frames from reaching memory. Those that do reach memory must be further filtered by the processor to determine if they truly contain one of the eight desired addresses.

The effectiveness of the hash table declines as the number of addresses increases.

The user must initialize the hash table registers. Use this CRC32 polynomial to compute the hash:

- FCS(x) = x^{32}+ x^{26}+ x^{23}+ x^{22}+ x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^{8}+ x^{7}+ x^{5}+ x^{4}+ x^{2}+ x^{1}+ 1

If Promiscuous mode is enabled (ENET*n*_RCR[PROM] = 1) all unicast and multicast frames are accepted regardless of ENET*n*_GAUR/GALR and ENET*n*_IAUR/IALR settings.

44.2.4.3.3 Broadcast address reject

All broadcast frames are accepted if BC_REJ is cleared or ENET*n*_RCR[PROM] is set. If PROM is cleared when ENET*n*_RCR[BC_REJ] is set, all broadcast frames are rejected.

Table 450. Broadcast address reject programming

PROM	BC_REJ	Broadcast frames
0	0	Accepted
0	1	Rejected
1	0	Accepted
1	1	Accepted

44.2.4.3.4 Miss-bit implementation

For higher layer filtering purposes, RxBD[M] indicates an address miss when the MAC operates in promiscuous mode and accepts a frame that would otherwise be rejected.

If a group/individual hash or exact match does not occur and Promiscuous mode is enabled (RCR[PROM] = 1), the frame is accepted and the M bit is set in the buffer descriptor; otherwise, the frame is rejected.

This means the status bit is set in any of the following conditions during Promiscuous mode:

- A broadcast frame is received when BC_REJ is set
- A unicast is received that does not match either:
  - Node address (PALR[PADDR1] and PAUR[PADDR2])
  - Hash table for unicast (IAUR[IADDR1] and IALR[IADDR2])
- A multicast is received that does not match the GAUR[GADDR1] and GALR[GADDR2] hash table entries

44.2.4.4 Frame length/type verification: payload length check

If the length/type is less than 0x600 and NLC is set, the MAC checks the payload length and reports any error in the frame status word RCR[NLC] and interrupt bit EIR[PLR].

If the length/type is greater than or equal to 0x600, the MAC interprets the field as a type and no payload length check is performed.

The length check is performed on VLAN and stacked VLAN frames. If a padded frame is received, no length check can be performed due to the extended frame payload because padded frames can never have a payload length error.

44.2.4.5 Frame length/type verification: frame length check

When the receive frame length exceeds MAX_FL bytes, the BABR interrupt is generated and the RxBD[LG] bit is set.

The frame is not truncated unless the frame length exceeds the value programmed in ENET*n*_FTRL[TRUNC_FL]. If the frame is truncated, RxBD[TR] is set. In addition, a truncated frame always has the CRC error indication set (RxBD[CR]).

44.2.4.6 VLAN frames processing

VLAN frames have a length/type field set to 0x8100 immediately followed by a 16-Bit VLAN control information field. VLAN-tagged frames are received as normal frames because the VLAN tag is not interpreted by the MAC function, and are pushed complete with the VLAN tag to the user application. If the length/type field of the VLAN-tagged frame, which is found four octets later in the frame, is less than 42, the padding is removed. In addition, the frame status word (RxBD[VLAN]) indicates that the current frame is VLAN tagged.

44.2.4.7 Pause frame termination

The receive engine terminates pause frames and does not transfer them to the receive FIFO. The quanta is extracted and sent to the MAC transmit path via a small internal clock rate decoupling asynchronous FIFO.

The quanta is written only if a correct CRC and frame length are detected by the control state machine. If not, the quanta is discarded and the MAC transmit path is not paused.

Valid pause frames are ignored if ENET*n*_RCR[FCE] is cleared and are forwarded to the client interface when ENET*n*_RCR[PAUFWD] is set.

44.2.4.8 CRC check

The CRC-32 field is checked and forwarded to the core FIFO interface if ENET*n*_RCR[CRCFWD] is cleared and ENET*n*_RCR[PADEN] is cleared. When CRCFWD is set (regardless of PADEN), the CRC-32 field is checked and terminated (not transmitted to the FIFO).

The CRC polynomial, as specified in the 802.3 standard, is:

- FCS(x) = x^{32}+ x^{26}+ x^{23}+ x^{22}+ x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^{8}+ x^{7}+ x^{5}+ x^{4}+ x^{2}+ x^{1}+ 1

The 32 bits of the CRC value are placed in the frame check sequence (FCS) field with the x^{31} term as right-most bit of the first octet. The CRC bits are thus received in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

If a CRC error is detected, the frame is marked invalid and RxBD[CR] is set.

44.2.4.9 Frame padding removal

When a frame is received with a payload length field set to less than 46 (42 for VLAN-tagged frames and 38 for frames with stacked VLANs), the zero padding can be removed before the frame is written into the data FIFO depending on the setting of ENET*n*_RCR[PADEN].

NOTE

If a frame is received with excess padding (in other words, the length field is set as mentioned above, but the frame has more than 64 octets) and padding removal is enabled, then the padding is removed as normal and no error is reported if the frame is otherwise correct (for example: good CRC, less than maximum length, and no other error).

44.2.4.10 Frame classification (AVB)

To support protocols such as Audio Video Bridging (AVB, IEEE 802.1Qav), normal traffic is separated from time-sensitive traffic immediately at the application interface to allow storing them in different queues for further processing. For every frame received, a classification based on VLAN priority can be performed. When a frame is received, and it contains a VLAN tag, the priority is compared against the values set in the classification match registers (see RCMR register and the following figure.

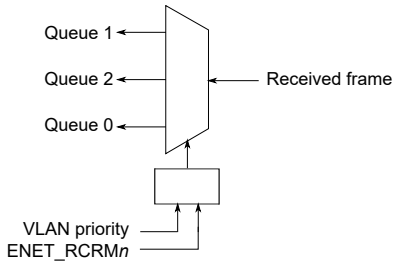


Figure 211. AVB frame classification

44.2.4.11 Receive flushing

RX flushing prevents frames in the RX FIFO from being blocked. Blocking can occur if the frame at the head of the RX FIFO cannot be forwarded because the ring it is associated with cannot accept it. This situation occurs when either the ring's RxBD[EMPTY] is not set or ENET_RDAR*n* is not set. When RX flushing is enabled, via QOS[RX_FLUSHn], the blocking frame will be flushed (discarded).

44.2.5 MAC transmit

Frame transmission starts when the transmit FIFO holds enough data.After a transfer starts, the MAC transmit function performs the following tasks:

- Generates preamble and SFD field before frame transmission
- Generates XOFF pause frames if the receive FIFO reports a congestion or if ENET*n*_TCR[TFC_PAUSE] is set with ENET*n*_OPD[PAUSE_DUR] set to a non-zero value
- Generates XON pause frames if the receive FIFO congestion condition is cleared or if TFC_PAUSE is set with PAUSE_DUR cleared
- Suspends Ethernet frame transfer (XOFF) if a non-zero pause quanta is received from the MAC receive path
- Adds padding to the frame if required

- Calculates and appends CRC-32 to the transmitted frame
- Sends the frame with correct inter-packet gap (IPG) (deferring)

When the MAC is configured to operate in half-duplex mode, the following additional tasks are performed:

- Collision detection
- Frame retransmit after back-off timer expires

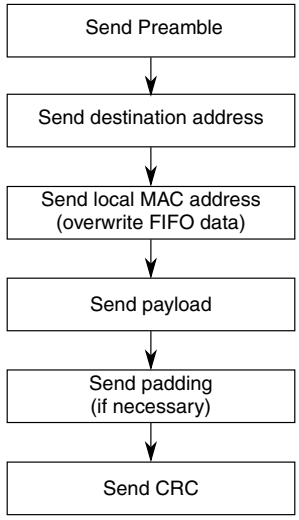


Figure 212. Frame transmit overview

44.2.5.1 Frame payload padding

The IEEE specification defines a minimum frame length of 64 bytes. If the frame sent to the MAC from the user application has a size smaller than 60 bytes, the MAC automatically adds padding bytes (0x00) to comply with the Ethernet minimum frame length specification. Transmit padding is always performed and cannot be disabled.

If the MAC is not allowed to append a CRC (TxBD[TC] = 1), the user application is responsible for providing frames with a minimum length of 64 octets.

44.2.5.2 MAC address insertion

On each frame received from the core transmit FIFO interface, the source MAC address is either:

- Replaced by the address programmed in the PADDR1/2 fields (ENET*n*_TCR[ADDINS] = 1)
- Transparently forwarded to the Ethernet line (ENET*n*_TCR[ADDINS] = 0)

44.2.5.3 CRC-32 generation

The CRC-32 field is optionally generated and appended at the end of a frame. The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the FCS field so that the x^{31} term is the right-most bit of the first octet. The CRC bits are thus transmitted in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

44.2.5.4 Inter-packet gap (IPG)

In full-duplex mode, after frame transmission and before transmission of a new frame, an IPG (programmed in ENET*n*_TIPG) is maintained. The minimum IPG can be programmed between 8 and 26 byte-times (64 and 208 bit-times).

In half-duplex mode, the core constantly monitors the line. Actual transmission of the data onto the network occurs only if it has been idle for a 96-bit time period, and any back-off time requirements have been satisfied. In accordance with the standard, the core begins to measure the IPG from CRS/GMII_CRS de-assertion.

44.2.5.5 Collision detection and handling — half-duplex operation only

A collision occurs on a half-duplex network when concurrent transmissions from two or more nodes take place. During transmission, the core monitors the line condition and detects a collision when the PHY device asserts COL.

When the core detects a collision while transmitting, it stops transmission of the data and transmits a 32-bit jam pattern. If the collision is detected during the preamble or the SFD transmission, the jam pattern is transmitted after completing the SFD, which results in a minimum 96-bit fragment. The jam pattern is a fixed pattern that is not compared to the actual frame CRC, and has a very low probability (0.532) of having a jam pattern identical to the CRC.

If a collision occurs before transmission of 64 bytes (including preamble and SFD), the MAC core waits for the backoff period and retransmits the packet data (stored in a 64-byte re-transmit buffer) that has already been sent on the line. The backoff period is generated from a pseudo-random process (truncated binary exponential backoff).

If a collision occurs after transmission of 64 bytes (including preamble and SFD), the MAC discards the remainder of the frame, optionally sets the LC interrupt bit, and sets TxBD[LCE].

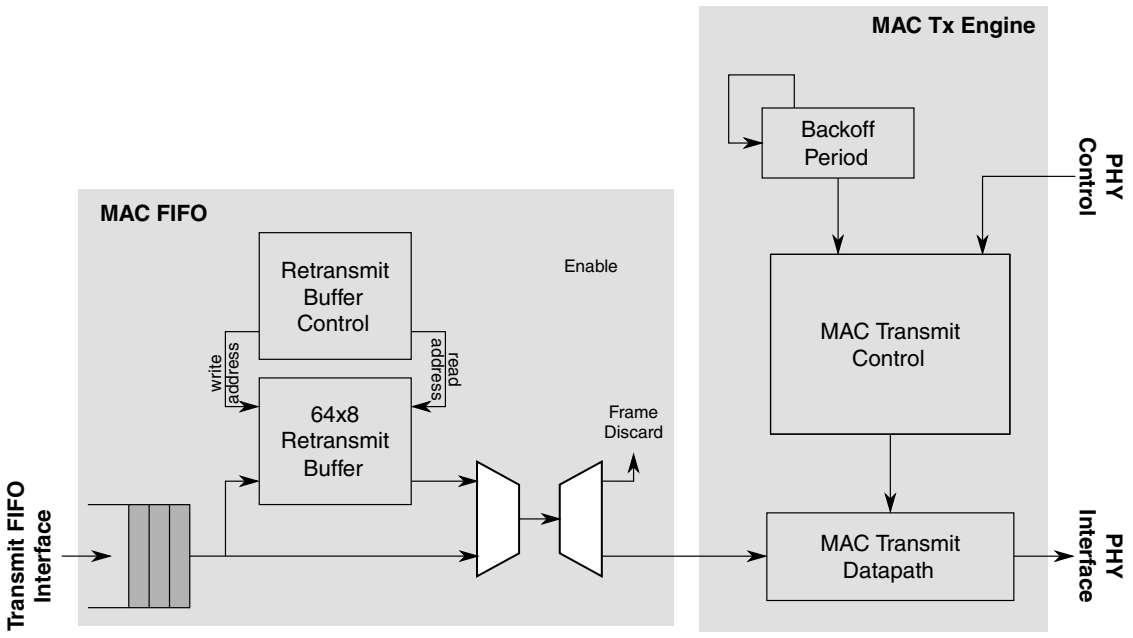


Figure 213. Packet re-transmit overview

The backoff time is represented by an integer multiple of slot times. One slot is equal to a 512-bit time period. The number of the delay slot times, before the n^{th} re-transmission attempt, is chosen as a uniformly-distributed random integer in the range:

- $0 < r < 2^k$
- k = min(*n*, N); where *n* is the number of retransmissions and N = 10

For example, after the first collision, the backoff period is 0 or 1 slot time. If a collision occurs on the first retransmission, the backoff period is 0, 1, 2, or 3, and so on.

The maximum backoff time (in 512-bit time slots) is limited by N = 10 as specified in the IEEE 802.3 standard.

If a collision occurs after 16 consecutive retransmissions, the core reports an excessive collision condition (ENET*n*_EIR[RL] interrupt field and TxBD[EE]) and discards the current packet from the FIFO.

In networks violating the standard requirements, a collision may occur after transmission of the first 64 bytes. In this case, the core stops the current packet transmission and discards the rest of the packet from the transmit FIFO. The core resumes transmission with the next packet available in the core transmit FIFO.

warning

Ethernet PHYs that support the SQE Test, or "heartbeat," feature must disable this feature. When this feature is enabled, the PHY asserts the collision signal after a frame is transmitted to indicate to the ENET that the PHY's collision logic is working. This may cause data corruption in the next frame from the ENET. This corrupted frame contains up to 21 zero bytes which start somewhere within the MAC destination address field. The ENET, however, will still generate a good FCS (CRC-32) but with corrupted data.

44.2.5.6 Rate limiting / traffic shaping support

The MAC-NET supports two methods to optimize frame traffic for either time-sensitive AVB frames (Class A and Class B) or best-effort non-AVB frames:

- Round-robin scheme
- Credit-based traffic shaping

To ensure that sufficient bandwidth is allocated for the AVB frames, the credit-based shaper must be used. Either method can be combined with a time-based shaper to ensure that a frame is always transmitted in its correct time slot.

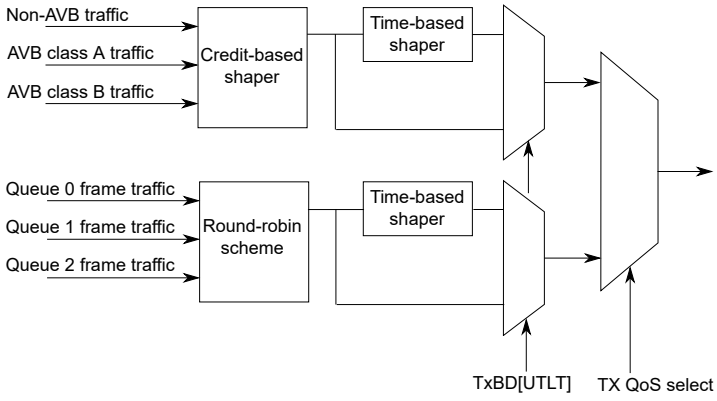


Figure 214. Transmit arbitration schemes

44.2.5.6.1 Round-robin policy

In the round-robin QoS scheme, each queue is given an equal opportunity to transmit one frame. For example, if queue *n* has a frame to transmit, the queue transmits its frame. After queue *n* has transmitted its frame, or if queue *n* does not have a frame to transmit, queue *n*+1 is then allowed to transmit its frame, and so on.

44.2.5.6.2 Credit-based shaper

The AVB credit based shaper acts independently, per class, to control the bandwidth distribution between normal traffic and time-sensitive traffic with respect to the total link bandwidth available. As per the IEEE 802.1Q, the maximum bandwidth distribution that can be allocated for the time-sensitive frames is 75%. The following example uses 70% for AVB frames and 30% for non-AVB frames:

Table 451. Bandwidth allocation example

AVB Class A	50%
AVB Class B	20%

Table continues on the next page...

Table 451. Bandwidth allocation example (continued)

non-AVB frames	30%
Total	100%

See DMACFG register for information on how to allocate bandwidth. The following figure shows how frame traffic is controlled within the MAC-NET core.

44.2.5.6.3 Time-based shaper

The time-based shaper enables the user to specify when a frame can be transmitted. It is always used in combination with either the round-robin scheme or the credit-based shaper. Use of the time-based shaper can ensure that frames are always transmitted in the correct time slot.

44.2.5.6.3.1 Time-based shaper example

Time-based shaping involves the following procedure:

1. Read the current value of the timer. See ATVR register and ATCR register.
2. Calculate the frame launch time and write this value to the TLT field of the frame's TxBD. See Enhanced transmit buffer descriptor.
3. Instruct the ENET to use the TLT by setting TxBD[UTLT].

The frame will be fetched and transmitted only if TxBD[TLT] is less than the current value of the timer. In other words, the timer must be past, that is, be greater than, the transmit launch time before the frame will be fetched and transmitted.

The transmit launch time must not be greater than the current value of ENET_ATVR + (0.5 × ENET_ATPER). This means the application cannot prepare frames with launch times beyond ENET_ATVR + (0.5 × ENET_ATPER). Because ENET_ATVR wraps, calculate TLT using the equation:

(ENET_ATVR + (0.5 × ENET_ATPER)) mod ENET_ATPER

For example, if ENET_ATPER is set to the recommended value of 1,000,000,000 ns (one second), then the user must not prepare future frames with launch times greater than (500,000,000 + ENET_ATVR).

As a specific example, if the current value of ENET_ATVR is 100,000, then you can prepare the following frames:

1. Frame 1 TxBD[TLT] = 100,000ns + 125,000ns
2. Frame 2 TxBD[TLT] = 100,000ns + 250,000ns
3. And so on.

44.2.6 Full-duplex flow control operation

Three conditions are handled by the core's flow control engine:

- Remote device congestion — The remote device connected to the same Ethernet segment as the core reports congestion and requests that the core stop sending data.
- Core FIFO congestion — When the core's receive FIFO reaches a user-programmable threshold (RX section empty), the core sends a pause frame back to the remote device requesting the data transfer to stop.
- Local device congestion — Any device connected to the core can request (typically, via the host processor) the remote device to stop transmitting data.

44.2.6.1 Remote device congestion

When the MAC transmit control gets a valid pause quanta from the receive path and if ENET*n*_RCR[FCE] is set, the MAC transmit logic:

- Completes the transfer of the current frame.

- Stops sending data for the amount of time specified by the pause quanta in 512 bit time increments.
- Sets ENET*n*_TCR[RFC_PAUSE].

Frame transfer resumes when the time specified by the quanta expires and if no new quanta value is received, or if a new pause frame with a quanta value set to 0x0000 is received. The MAC also resets RFC_PAUSE to zero.

If ENET*n*_RCR[FCE] cleared, the MAC ignores received pause frames.

Optionally and independent of ENET*n*_RCR[FCE], pause frames are forwarded to the client interface if PAUFWD is set.

44.2.6.2 Local device/FIFO congestion

The MAC transmit engine generates pause frames when the local receive FIFO is not able to receive more than a pre-defined number of words (FIFO programmable threshold) or when pause frame generation is requested by the local host processor.

- To generate a pause frame, the host processor sets ENET*n*_TCR[TFC_PAUSE]. A single pause frame is generated when the current frame transfer is completed and TFC_PAUSE is automatically cleared. Optionally, an interrupt is generated.
- An XOFF pause frame is generated when the receive FIFO asserts its section empty flag (internal). An XOFF pause frame is generated automatically, when the current frame transfer completes.
- An XON pause frame is generated when the receive FIFO deasserts its section empty flag (internal). An XON pause frame is generated automatically, when the current frame transfer completes.

When an XOFF pause frame is generated, the pause quanta (payload byte P1 and P2) is filled with the value programmed in ENET*n*_OPD[PAUSE_DUR].

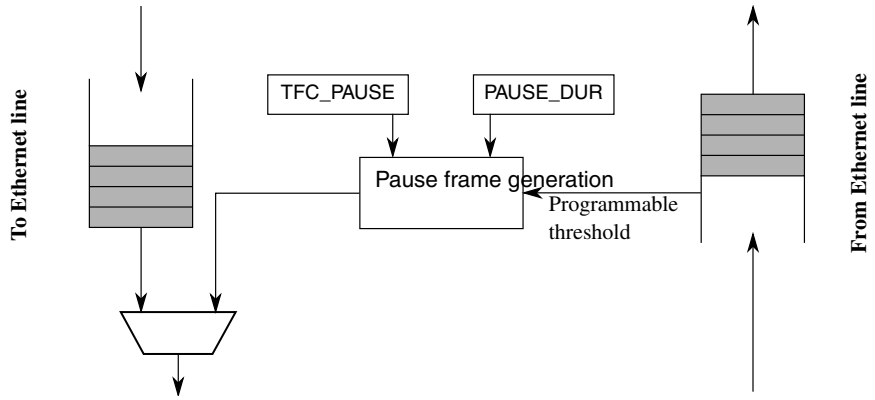


Figure 215. Pause frame generation overview

NOTE

Although the flow control mechanism should prevent any FIFO overflow on the MAC core receive path, the core receive FIFO is protected. When an overflow is detected on the receive FIFO, the current frame is truncated with an error indication set in the frame status word. The frame should subsequently be discarded by the user application.

44.2.7 IP accelerator functions

The following sections describe the IP accelerator functions.

44.2.7.1 Checksum calculation

The IP and ICMP, TCP, UDP checksums are calculated with one's complement arithmetic summing up 16-bit values.

- For ICMP, the checksum is calculated over the complete ICMP datagram, in other words without IP header.
- For TCP and UDP, the checksums contain the header and data sections and values from the IP header, which can be seen as a pseudo-header that is not actually present in the data stream.

Table 452. IPv4 pseudo-header for checksum calculation

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10	9 8	7 6 5 4	3 2 1 0
Source address								
Destination address								
Zero		Protocol		TCP/UDP length				

Table 453. IPv6 pseudo-header for checksum calculation

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10	9 8	7 6 5 4	3 2 1 0
Source address								
Destination address								
TCP/UDP length								
Zero							Next header	

The TCP/UDP length value is the length of the TCP or UDP datagram, which is equal to the payload of an IP datagram. It is derived by subtracting the IP header length from the complete IP datagram length that is given in the IP header (IPv4), or directly taken from the IP header (IPv6). The protocol field is the corresponding value from the IP header. The Zero fields are all zeroes.

For IPv6, the complete 128-bit addresses are considered. The next header value identifies the upper layer protocol as either TCP or UDP. It may differ from the next header value of the IPv6 header if extension headers are inserted before the protocol header.

The checksum calculation uses 16-bit words in network byte order: The first byte sent/received is the MSB, and the second byte sent/received is the LSB of the 16-bit value to add to the checksum. If the frame ends on an odd number of bytes, a zero byte is appended for checksum calculation only, and is not actually transmitted.

44.2.7.2 Additional padding processing

According to IEEE 802.3, any Ethernet frame must have a minimum length of 64 octets. The MAC usually removes padding on receive when a frame with length information is received. Because IP frames have a type value instead of length, the MAC does not remove padding for short IP frames, as it is not aware of the frame contents.

The IP accelerator function can be configured to remove the Ethernet padding bytes that might follow the IP datagram.

On transmit, the MAC automatically adds padding as necessary to fill any frame to a 64-byte length.

44.2.7.3 32-bit Ethernet payload alignment

The data FIFOs allow inserting two additional arbitrary bytes in front of a frame. This extends the 14-byte Ethernet header to a 16-byte header, which leads to alignment of the Ethernet payload, following the Ethernet header, on a 32-bit boundary.

This function can be enabled for transmit and receive independently with the corresponding SHIFT16 bits in the ENET*n*_TACC and ENET*n*_RACC registers.

When enabled, the valid frame data is arranged as shown in Table 454.

Table 454. 64-bit interface data structure with SHIFT16 enabled

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0	Any value	Any value
Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8	Byte 7	Byte 6
...							

44.2.7.3.1 Receive processing

When ENET*n*_RACC[SHIFT16] is set, each frame is received with two additional bytes in front of the frame.The user application must ignore these first two bytes and find the first byte of the frame in bits 23–16 of the first word from the RX FIFO.

NOTE

SHIFT16 must be set during initialization and kept set during the complete operation, because it influences the FIFO write behavior.

44.2.7.3.2 Transmit processing

When ENET*n*_TACC[SHIFT16] is set, the first two bytes of the first word written (bits 15–0) are discarded immediately by the FIFO write logic. The SHIFT16 bit can be enabled/disabled for each frame individually if required, but can be changed only between frames.

44.2.7.4 Received frame discard

Because the receive FIFO must be operated in store and forward mode (ENET*n*_RSFL cleared), received frames can be discarded based on the following errors:

- The MAC function receives the frame with an error:
  - The frame has an invalid payload length
  - Frame length is greater than MAX_FL
  - Frame received with a CRC-32 error
  - Frame truncated due to receive FIFO overflow
  - Frame is corrupted as PHY signaled an error (RX_ERR asserted during reception)
- An IP frame is detected and the IP header checksum is wrong
- An IP frame with a valid IP header and a valid IP header checksum is detected, the protocol is known but the protocol-specific checksum is wrong

If one of the errors occurs and the IP accelerator function is configured to discard frames (ENET*n*_RACC), the frame is automatically discarded. Statistics are maintained normally and are not affected by this discard function.

44.2.7.5 IPv4 fragments

When an IPv4 IP fragment frame is received, only the IP header is inspected and its checksum verified. 32-bit alignment operates the same way on fragments as it does on normal IP frames, as specified above.

The IP fragment frame payload is not inspected for any protocol headers. As such, a protocol header would only exist in the very first fragment. To assist in protocol-specific checksum verification, the one's-complement sum is calculated on the IP payload (all bytes following the IP header) and provided with the frame status word.

The frame fragment status field (RxBD[FRAG]) is set to indicate a fragment reception, and the one's-complement sum of the IP payload is available in RxBD[Payload checksum].

NOTE

After all fragments have been received and reassembled, the application software can take advantage of the payload checksum delivered with the frame's status word to calculate the protocol-specific checksum of the datagram.

For example, if a TCP payload is delivered by multiple IP fragments, the application software can calculate the pseudo-header checksum value from the first fragment, and add the payload checksums delivered with the status for all fragments to verify the TCP datagram checksum.

44.2.7.6 IPv6 support

The following sections describe the IPv6 support.

44.2.7.6.1 Receive processing

An Ethernet frame of type 0x86DD identifies an IP Version 6 frame (IPv6) frame. If an IPv6 frame is received, the first IP header is inspected (first ten words), which is available in every IPv6 frame.

If the receive SHIFT16 function is enabled, the IP header is aligned on a 32-bit boundary allowing more efficient processing (see 32-bit Ethernet payload alignment).

For TCP and UDP datagrams, the pseudo-header checksum calculation is performed and verified.

To assist in protocol-specific checksum verification, the one's-complement sum is always calculated on the IP payload (all bytes following the IP header) and provided with the frame status word. For example, if extension headers were present, their sums can be subtracted in software from the checksum to isolate the TCP/UDP datagram checksum, if required.

44.2.7.6.2 Transmit processing

For IPv6 transmission, the SHIFT16 function is supported to process 32-bit aligned datagrams.

IPv6 has no IP header checksum; therefore, the IP checksum insertion configuration is ignored.

The protocol checksum is inserted only if the next header of the IP header is a known protocol (TCP, UDP, or ICMP). If a known protocol is detected, the checksum over all bytes following the IP header is calculated and inserted in the correct position.

The pseudo-header checksum calculation is performed for TCP and UDP datagrams accordingly.

44.2.8 IEEE 1588 functions

To allow for IEEE 1588 or similar time synchronization protocol implementations, the MAC is combined with a time-stamping module to support precise time-stamping of incoming and outgoing frames. Set ENET*n*_ECR[EN1588] to enable 1588 support.

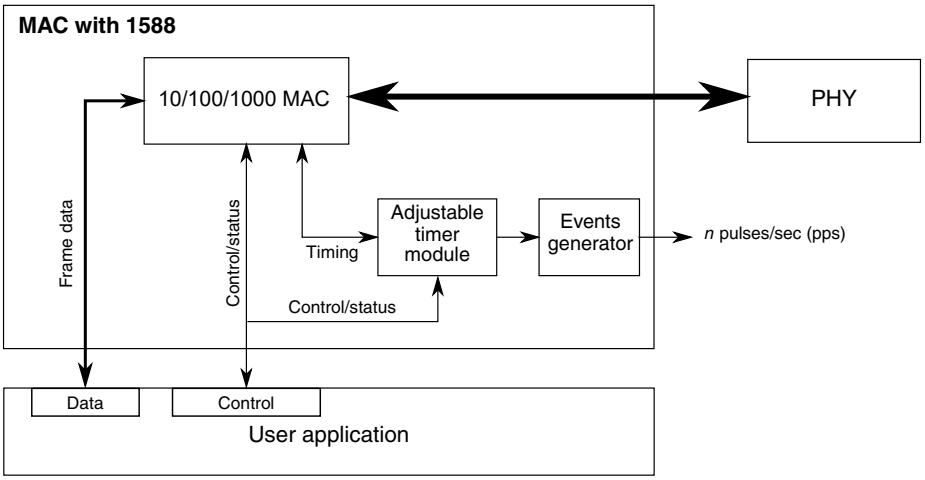


Figure 216. IEEE 1588 functions overview

44.2.8.1 Adjustable timer module

The adjustable timer module (TSM) implements the free-running counter (FRC), which generates the timestamps. The FRC operates with the time-stamping clock, which can be set to any value depending on your system requirements.

Through dedicated correction logic, the timer can be adjusted to allow synchronization to a remote leader and provide a synchronized timing reference to the local system. The timer can be configured to cause an interrupt after a fixed time period, to allow synchronization of software timers or perform other synchronized system functions.

The timer is typically used to implement a period of one second; hence, its value ranges from 0 to (1×10^9)-1. The period event can trigger an interrupt, and software can maintain the seconds and hours time values as necessary.

44.2.8.1.1 Adjustable timer implementation

The adjustable timer consists of a programmable counter/accumulator and a correction counter. The periods of both counters and their increment rates are freely configurable, allowing very fine tuning of the timer.

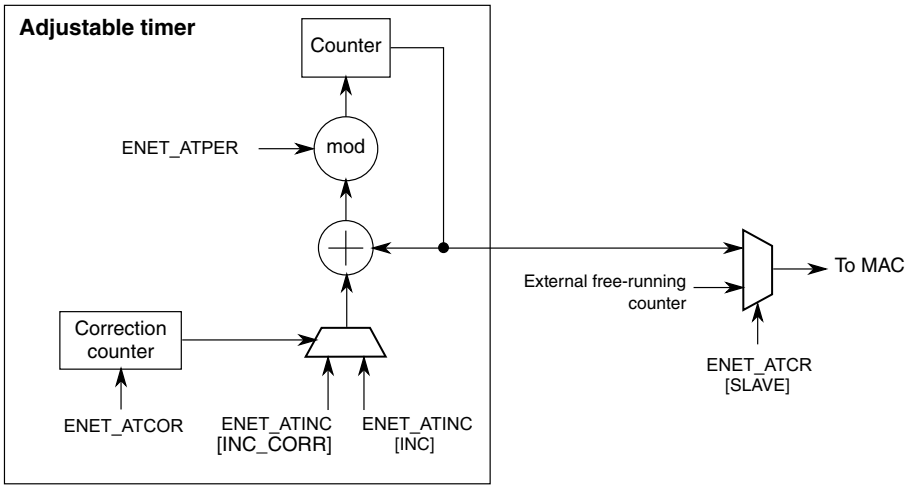


Figure 217. Adjustable timer implementation detail

The counter produces the current time. During each time-stamping clock cycle, a constant value is added to the current time as programmed in ENET*n*_ATINC. The value depends on the chosen time-stamping clock frequency. For example, if it operates at 125 MHz, setting the increment to eight represents 8 ns.

The period, configured in ENET*n*_ATPER, defines the modulo when the counter wraps. In a typical implementation, the period is set to 1×10^9 so that the counter wraps every second, and hence all timestamps represent the absolute nanoseconds within the one second period. When the period is reached, the counter wraps to start again respecting the period modulo. This means it does not necessarily start from zero, but instead the counter is loaded with the value (Current + Inc $-(1 \times 10^9)$), assuming the period is set to 1×10^9.

The correction counter operates fully independently, and increments by one with each time-stamping clock cycle. When it reaches the value configured in ENET*n*_ATCOR, it restarts and instructs the timer once to increment by the correction value, instead of the normal value.

The normal and correction increments are configured in ENET*n*_ATINC. To speed up the timer, set the correction increment more than the normal increment value. To slow down the timer, set the correction increment less than the normal increment value.

The correction counter only defines the distance of the corrective actions, not the amount. This allows very fine corrections and low jitter (in the range of 1 ns) independent of the chosen clock frequency.

By enabling follower mode (ENET*n*_ATCR[SLAVE] = 1), the timer is ignored and the current time is externally provided from one of the external modules. See the Chip Configuration details for which clock source is used. This is useful if multiple modules within the system must operate from a single timer. When follower mode is enabled, you still must set ENET*n*_ATINC[INC] to the value of the leader, since it is used for internal comparisons.

44.2.8.2 Transmit timestamping

Only 1588 event frames need to be time-stamped on transmit. The client application (for example, the MAC driver) should detect 1588 event frames and set TxBD[TS] together with the frame.

If TxBD[TS] is set, the MAC records the timestamp for the frame in ENET*n*_ATSTMP. ENET*n*_EIR[TS_AVAIL] is set to indicate that a new timestamp is available.

Software implements a handshaking procedure by setting TxBD[TS] when it transmits the frame for which a timestamp is needed, and then waits for ENET*n*_EIR[TS_AVAIL] to determine when the timestamp is available. The timestamp is then read from ENET*n*_ATSTMP. This is done for all event frames. Other frames do not use TxBD[TS] and, therefore, do not interfere with the timestamp capture.

44.2.8.3 Receive timestamping

When a frame is received, the MAC latches the value of the timer when the frame's start of frame delimiter (SFD) field is detected, and provides the captured timestamp on RxBD[1588 timestamp]. This is done for all received frames.

44.2.8.4 Time synchronization

The adjustable timer module is available to synchronize the local clock of a node to a remote leader. It implements a free running 32-bit counter, and also contains an additional correction counter.The correction counter increases or decreases the rate of the free running counter, enabling very fine granular changes of the timer for synchronization, yet adding only very low jitter when performing corrections.

The application software implements, in a follower scenario, the required control algorithm, setting the correction to compensate for local oscillator drifts and locking the timer to the remote leader clock on the network.

The timer and all timestamp-related information should be configured to show the true nanoseconds value of a second (in other words, the timer is configured to have a period of one second). Hence, the values range from 0 to (1×10^9)–1. In this application, the seconds counter is implemented in software using an interrupt function that is executed when the nanoseconds counter wraps at 1×10^9.

44.2.8.5 Input capture and output compare

The input capture and output compare block can be used to provide precise hardware timing for input and output events.

44.2.8.5.1 Input capture

The TCCR*n* capture registers latch the time value when the corresponding external event occurs. An event can be a rising-, falling-, or either-edge of one of the 1588_TMR*n* signals. An event will cause the corresponding TCSR*n*[TF] timer flag to be set, indicating that an input capture has occurred. If the corresponding interrupt is enabled with the TCSR*n*[TIE] field, an interrupt can be generated.

44.2.8.5.2 Output compare

The TCCR*n* compare registers are loaded with the time at which the corresponding event should occur. When the ENET free-running counter value matches the output compare reference value in the TCCR*n* register, the corresponding flag, TCSR*n*[TF], is set, indicating that an output compare has occurred. The corresponding interrupt, if enabled by TCSR*n*[TIE], will be generated.The corresponding 1588_TMR*n* output signal will be asserted according to TCSR*n*[TMODE].

NOTE

If TCSR*n*[TMODE] is set to 10X1b or 1010b then the timer output pin toggle-on-overflow will occur only when PINPER, PEREN, and EN bits of the ATCR register are one.

44.2.8.5.3 DMA requests

A DMA request can be enabled by setting TCSR*n*[TDRE]. The corresponding DMA request is generated when the TCSR*n*[TF] timer flag is set. When the DMA has completed, the corresponding TCSRn[TF] flag is cleared.

44.2.9 FIFO thresholds

The core FIFO thresholds are fully programmable to dynamically change the FIFO operation.

For example, store and forward transfer can be enabled by a simple change in the FIFO threshold registers.

The thresholds are defined in 64-bit words.

The receive and transmit FIFOs both have a depth of 1024 words.

44.2.9.1 Receive FIFO

Four programmable thresholds are available, which can be set to any value to control the core operation as follows.

Table 455. Receive FIFO thresholds definition

Register	Description
ENET*n*_RSFL [RX_SECTION_FULL]	When the FIFO level reaches the ENET*n*_RSFL value, the MAC status signal is asserted to indicate that data is available in the receive FIFO (cut-through operation). Once asserted, if the FIFO empties below the threshold set with ENET*n*_RAEM and if the end-of-frame is not yet stored in the FIFO, the status signal is deasserted again. If a frame has a size smaller than the threshold (in other words, an end-of-frame is available for the frame), the status is also asserted. To enable store and forward on the receive path, clear ENET*n*_RSFL. The MAC status signal is asserted only when a complete frame is stored in the receive FIFO. When programming a non-zero value to ENET*n*_RSFL (cut-through operation) it should be greater than ENET*n*_RAEM.
ENET*n*_RAEM [RX_ALMOST_EMPTY]	When the FIFO level reaches the ENET*n*_RAEM value and the end-of-frame has not been received, the core receive read control stops the FIFO read (and subsequently stops transferring data to the MAC client application). It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. Set ENET*n*_RAEM to a minimum of four.
ENET*n*_RAFL [RX_ALMOST_FULL]	When the FIFO level approaches the maximum and there is no more space remaining for at least ENETn_RAFL number of words, the MAC control logic stops writing data in the FIFO and truncates the receive frame to avoid FIFO overflow. The corresponding error status is set when the frame is delivered to the application. Set ENET*n*_RAFL to a minimum of 4.
ENET*n*_RSEM [RX_SECTION_EMPTY]	When the FIFO level reaches the ENET*n*_RSEM value, an indication is sent to the MAC transmit logic, which generates an XOFF pause frame. This indicates FIFO congestion to the remote Ethernet client. When the FIFO level goes below the value programmed in ENET*n*_RSEM, an indication is sent to the MAC transmit logic, which generates an XON pause frame. This indicates the FIFO congestion is cleared to the remote Ethernet client. Clearing ENET*n*_RSEM disables any pause frame generation.

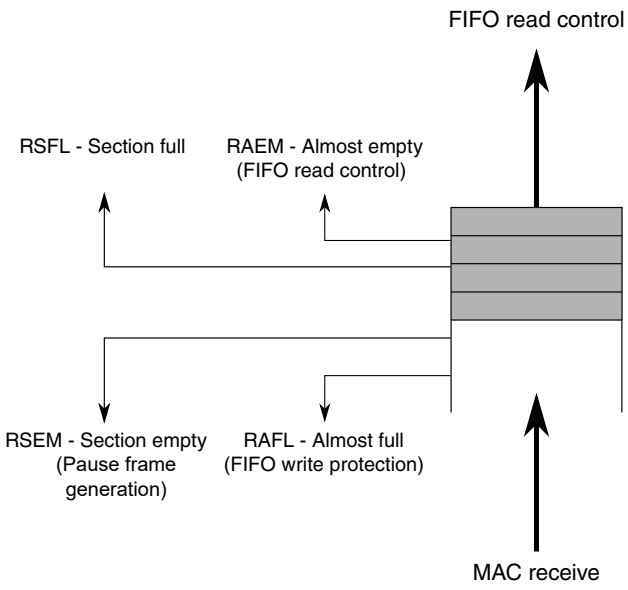


Figure 218. Receive FIFO overview

44.2.9.2 Transmit FIFO

Four programmable thresholds are available which control the core operation as described below.

Table 456. Transmit FIFO thresholds definition

Register	Description
ENET*n*_TAEM [TX_ALMOST _EMPTY]	When the FIFO level reaches the ENET*n*_TAEM value and no end-of-frame is available for the frame, the MAC transmit logic avoids a FIFO underflow by stopping FIFO reads and transmitting the Ethernet frame with an MII error indication. Set ENET*n*_TAEM to a minimum of 4.
ENET*n*_TAFL [TX_ALMOST _FULL]	When the FIFO level approaches the maximum, so that there is no more space for at least ENET*n*_TAFL number of words, the MAC deasserts its control signal to the application. If the application does not react on this signal, the FIFO write control logic avoids FIFO overflow by truncating the current frame and setting the error status. As a result, the frame is transmitted with an GMII/MII error indication. Set ENET*n*_TAFL to a minimum of 4. Larger values allow more latency for the application to react on the MAC control signal deassertion, before the frame is truncated. A typical setting is 8, which offers 3–4 clock cycles of latency to the application to react on the MAC control signal deassertion.
ENET*n*_TSEM [TX_SECTION _EMPTY]	When the FIFO level reaches the ENET*n*_TSEM value, a MAC status signal is deasserted to indicate that the transmit FIFO is getting full. This gives the ENET module an indication to slow or stop its write transaction to avoid a buffer overflow. This is a pure indication function to the application. It has no effect within the MAC. When ENET*n*_TSEM is 0, the signal is never deasserted.
ENET*n*_TFWR	When the FIFO level reaches the ENET*n*_TFWR value and when STRFWD is cleared, the MAC transmit control logic starts frame transmission before the end-of-frame is available in the FIFO (cut-through operation).

Table continues on the next page...

Table 456. Transmit FIFO thresholds definition (continued)

Register	Description
	If a complete frame has a size smaller than the ENET*n*_TFWR threshold, the MAC also transmits the frame to the line. To enable store and forward on the transmit path, set STRFWD. In this case, the MAC starts to transmit data only when a complete frame is stored in the transmit FIFO.

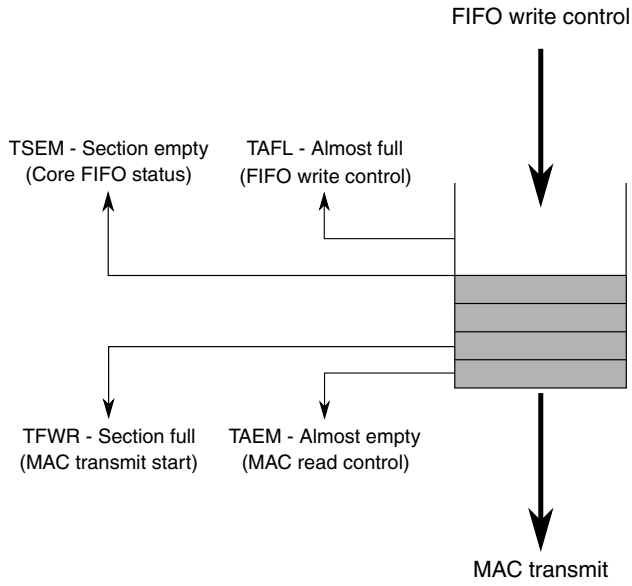


Figure 219. Transmit FIFO overview

44.2.10 Internal loopback option

The core implements internal loopback option. To use internal loopback mode, the RX clock must be provided externally at the RGMII_RCK pin of the PHY interface. The loopback mode is controlled by the following ENET*n*_RCR register field:

Table 457. Loopback option

Register field	Description
LOOP	Internal MII loopback. The MAC transmit is returned to the MAC receive. No data is transmitted to the external interfaces.

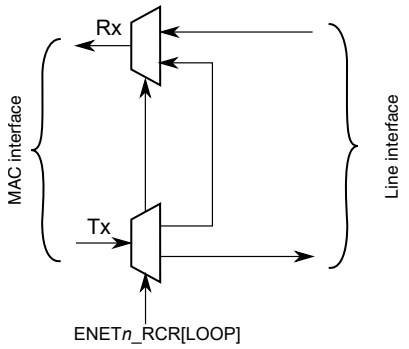


Figure 220. Loopback option

44.2.11 Legacy buffer descriptors

To support the Ethernet controller on previous chips, legacy FEC buffer descriptors are available. To enable legacy support, write 0 to ENET*n*_ECR[1588EN].

NOTE

- Legacy buffer descriptors are used in single-ring mode, that is, when DMA*n*CFG[DMA_CLASS_EN] are zero. In multi-ring mode, legacy TxBDs are used only with the round-robin scheme.
- The legacy buffer descriptor tables show the byte order for little-endian chips. ECR[DBSWP] must be set to 1 after reset to enable little-endian mode.

44.2.11.1 Legacy receive buffer descriptor

The following table shows the legacy FEC receive buffer descriptor. Table 461 contains the descriptions for each field.

Table 458. Legacy FEC receive buffer descriptor (RxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 4	Rx data buffer pointer — low halfword															
Offset + 6	Rx data buffer pointer — high halfword															

44.2.11.2 Legacy transmit buffer descriptor

The following table shows the legacy FEC transmit buffer descriptor. Table 463 contains the descriptions for each field.

Table 459. Legacy FEC transmit buffer descriptor (TxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table continues on the next page...

Table 459. Legacy FEC transmit buffer descriptor (TxBD) (continued)

Offset + 0	Data Length															
Offset + 2	R	TO1	W	TO2	L	TC	ABC [1]	—	—	—	—	—	—	—	—	—
Offset + 4	Tx Data Buffer Pointer — low halfword															
Offset + 6	Tx Data Buffer Pointer — high halfword															

1. This field is not supported by the uDMA.

44.2.12 Enhanced buffer descriptors

This section provides a description of the enhanced operation of the driver/uDMA via the buffer descriptors. It is followed by a detailed description of the receive and transmit descriptor fields. To enable the enhanced features, set ENET*n*_ECR[1588EN].

NOTE

The enhanced buffer descriptor tables show the byte order for little-endian chips. ECR[DBSWP] must be set to 1 after reset to enable little-endian mode.

44.2.12.1 Enhanced receive buffer descriptor

The following table shows the enhanced uDMA receive buffer descriptor. Table 461 contains the descriptions for each field.

Table 460. Enhanced uDMA receive buffer descriptor (RxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 4	Rx data buffer pointer – low halfword															
Offset + 6	Rx data buffer pointer – high halfword															
Offset + 8	VPCP			—	—	—	—	—	—	—	ICE	PCR	—	VLAN	IPV6	FRAG
Offset + A	ME	—	—	—	—	PE	CE	UC	INT	—	—	—	—	—	—	—
Offset + C	Payload checksum															
Offset + E	Header length					—	—	—	Protocol type							
Offset + 10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – low halfword															

Table continues on the next page...

Table 460. Enhanced uDMA receive buffer descriptor (RxBD) (continued)

Offset + 16	1588 timestamp – high halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 461. Receive buffer descriptor field definitions

Word	Field	Description
Offset + 0	15–0 Data Length	Data length. Written by the MAC. Data length is the number of octets written by the MAC into this BD's data buffer if L is cleared (the value is equal to EMRBR), or the length of the frame including CRC if L is set. It is written by the MAC once as the BD is closed.
Offset + 2	15 E	Empty. Written by the MAC (= 0) and user (= 1). 0 The data buffer associated with this BD is filled with received data, or data reception has aborted due to an error condition. The status and length fields have been updated as required.
		1 The data buffer associated with this BD is empty, or reception is currently in progress.
Offset + 2	14 RO1	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 2	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location.
		1 The next buffer descriptor is found at the location defined in ENET*n*_RDSR.
Offset + 2	12 RO2	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 2	11 L	Last in frame. Written by the uDMA. 0 The buffer is not the last in a frame.
		1 The buffer is the last in a frame.
Offset + 2	10–9	Reserved, must be cleared.
Offset + 2	8 M	Miss. Written by the MAC. This field is set by the MAC for frames accepted in promiscuous mode, but flagged as a miss by the internal address recognition. Therefore, while in promiscuous mode, you can use the this field to quickly determine whether the frame was destined to this station. This field is valid only if the L and PROM bits are set.

Table continues on the next page...

Table 461. Receive buffer descriptor field definitions (continued)

Word	Field	Description
		0 The frame was received because of an address recognition hit. 1 The frame was received because of promiscuous mode.
		The information needed for this field comes from the promiscuous_miss(ff_rx_err_stat[26]) sideband signal.
Offset + 2	7 BC	Set if the DA is broadcast (FFFF_FFFF_FFFF).
Offset + 2	6 MC	Set if the DA is multicast and not BC.
Offset + 2	5 LG	Receive frame length violation. Written by the MAC. A frame length greater than RCR[MAX_FL] was recognized. This field is valid only if the L field is set. The receive data is not altered in any way unless the length exceeds TRUNC_FL bytes.
Offset + 2	4 NO	Receive non-octet aligned frame. Written by the MAC. A frame that contained a number of bits not divisible by 8 was received, and the CRC check that occurred at the preceding byte boundary generated an error or a PHY error occurred. This field is valid only if the L field is set. If this field is set, the CR field is not set.
Offset + 2	3	Reserved, must be cleared.
Offset + 2	2 CR	Receive CRC or frame error. Written by the MAC. This frame contains a PHY or CRC error and is an integral number of octets in length. This field is valid only if the L field is set.
Offset + 2	1 OV	Overrun. Written by the MAC. A receive FIFO overrun occurred during frame reception. If this field is set, the other status fields, M, LG, NO, and CR, lose their normal meaning and are zero. This field is valid only if the L field is set.
Offset + 2	0 TR	Set if the receive frame is truncated (frame length >TRUNC_FL). If the TR field is set, the frame must be discarded and the other error fields must be ignored because they may be incorrect.
Offset + 4	15–0 Data buffer pointer low	Receive data buffer pointer, low halfword
0ffset + 6	15–0 Data buffer pointer high	Receive data buffer pointer, high halfword[1]
Offset + 8	15–13 VPCP	VLAN priority code point. This field is written by the uDMA to indicate the frame priority level. Valid values are from 0 (best effort) to 7 (highest). This value can be used to prioritize different classes of traffic (e.g., voice, video, data). This field is only valid if the L field is set.

Table continues on the next page...

Table 461. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 8	12–6	Reserved, must be cleared.
Offset + 8	5 ICE	IP header checksum error. This is an accelerator option. This field is written by the uDMA. Set when either a non-IP frame is received or the IP header checksum was invalid. An IP frame with less than 3 bytes of payload is considered to be an invalid IP frame. This field is only valid if the L field is set.
Offset + 8	4 PCR	Protocol checksum error. This is an accelerator option. This field is written by the uDMA. Set when the checksum of the protocol is invalid or an unknown protocol is found and checksumming could not be performed. This field is only valid if the L field is set.
Offset + 8	3	Reserved, must be cleared.
Offset + 8	2 VLAN	VLAN. This is an accelerator option. This field is written by the uDMA. It means that the frame has a VLAN tag. This field is valid only if the L field is set.
Offset + 8	1 IPV6	IPV6 Frame. This field is written by the uDMA. This field indicates that the frame has an IPv6 frame type. If this field is not set it means that an IPv4 or other protocol frame was received. This field is valid only if the L field is set.
Offset + 8	0 FRAG	IPv4 Fragment. This is an accelerator option. This field is written by the uDMA. It indicates that the frame is an IPv4 fragment frame. This field is only valid when the L field is set.
Offset + A	15 ME	MAC error. This field is written by the uDMA. This field means that the frame stored in the system memory was received with an error (typically, a receive FIFO overflow). This field is only valid when the L field is set.
Offset + A	14–11	Reserved, must be cleared.
Offset + A	10 PE	PHY Error. This field is written by the uDMA. Set to "1" when the frame was received with an Error character on the PHY interface. The frame is invalid. This field is valid only when the L field is set.
Offset + A	9 CE	Collision. This field is written by the uDMA. Set when the frame was received with a collision detected during reception. The frame is invalid and sent to the user application. This field is valid only when the L field is set.
Offset + A	8 UC	Unicast. This field is written by the uDMA, and means that the frame is unicast. This field is valid regardless of whether the L field is set.
Offset + A	7 INT	Generate RXB/RXF interrupt. This field is set by the user to indicate that the uDMA is to generate an interrupt on the *dma_int_rxb* / *dma_int_rxfevent*.
Offset + A	6–0	Reserved, must be cleared.
Offset + C	15–0	Internet payload checksum. This is an accelerator option. It is the one's complement sum of the payload section of the IP frame. The sum is calculated over all data following the IP header until the end of the IP payload. This field is valid only when the L field is set.

Table continues on the next page...

Table 461. Receive buffer descriptor field definitions (continued)

Word	Field	Description
	Payload checksum	
Offset + E	15–11 Header length	Header length. This is an accelerator option. This field is written by the uDMA. This field is the sum of 32-bit words found within the IP and its following protocol headers. If an IP datagram with an unknown protocol is found, then the value is the length of the IP header. If no IP frame or an erroneous IP header is found, the value is 0. The following values are minimum values if no header options exist in the respective headers: • ICMP/IP: 6 (5 IP header, 1 ICMP header) • UDP/IP: 7 (5 IP header, 2 UDP header) • TCP/IP: 10 (5 IP header, 5 TCP header) This field is only valid if the L field is set.
Offset + E	10–8	Reserved, must be cleared.
Offset + E	7–0 Protocol type	Protocol type. This is an accelerator option. The 8-bit protocol field found within the IP header of the frame. It is valid only when ICE is cleared. This field is valid only when the L field is set.
Offset + 10	15–0	Reserved, must be cleared.
Offset + 12	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 12	14–0	Reserved, must be cleared.
Offset + 14	15–0 1588 timestamp	This value is written by the uDMA. It is only valid if the L field is set.
Offset + 16		
Offset + 18 – Offset + 1E	15–0	Reserved, must be cleared.

1. The receive buffer pointer, containing the address of the associated data buffer, must always be evenly divisible by 64. The buffer must reside in memory external to the MAC. The Ethernet controller never modifies this value.

44.2.12.2 Enhanced transmit buffer descriptor

Table 462. Enhanced transmit buffer descriptor (TxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															

Table continues on the next page...

Table 462. Enhanced transmit buffer descriptor (TxBD) (continued)

Offset + 2	R	TO1	W	TO2	L	TC	—	—	—	—	—	—	—	—	—	—
Offset + 4	Tx Data Buffer Pointer – low halfword															
Offset + 6	Tx Data Buffer Pointer – high halfword															
Offset + 8	TXE	—	UE	EE	FE	LCE	OE	TSE	—	—	—	—	—	—	—	—
Offset + A	—	INT	TS	PINS	IINS	—	—	UTLT	FTYPE				—	—	—	—
Offset + C	TLT – low halfword															
Offset + E	TLT – high halfword															
Offset + 10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – low halfword															
Offset + 16	1588 timestamp – high halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 463. Enhanced transmit buffer descriptor field definitions

Word	Field	Description
Offset + 0	15–0 Data Length	Data length, written by user. Data length is the number of octets the MAC should transmit from this BD's data buffer. It is never modified by the MAC.
Offset + 2	15 R	Ready. Written by the MAC and you. 0 The data buffer associated with this BD is not ready for transmission. You are free to manipulate this BD or its associated data buffer. The MAC clears this field after the buffer has been transmitted or after an error condition is encountered. 1 The data buffer, prepared for transmission by you, has not been transmitted or currently transmits. You may write no fields of this BD after this field is set.
Offset + 2	14 TO1	Transmit software ownership. This field is reserved for software use. This read/write field is not modified by hardware and its value does not affect hardware.

Table continues on the next page...

Table 463. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 2	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location
		1 The next buffer descriptor is found at the location defined in ETDSR.
Offset + 2	12 TO2	Transmit software ownership. This field is reserved for use by software. This read/write field is not modified by hardware and its value does not affect hardware.
Offset + 2	11 L	Last in frame. Written by user. 0 The buffer is not the last in the transmit frame
		1 The buffer is the last in the transmit frame
Offset + 2	10 TC	Transmit CRC. Written by user, and valid only when L is set. 0 End transmission immediately after the last data byte 1 Transmit the CRC sequence after the last data byte
		This field is valid only when the L field is set.
Offset + 2	9 ABC	Append bad CRC. **NOTE** This field is not supported by the uDMA and is ignored.
Offset + 2	8–0	Reserved, must be cleared.
Offset + 4	15–0 Data buffer pointer low	Tx data buffer pointer, low halfword
Offset + 6	15–0 Data buffer pointer high	Tx data buffer pointer, high halfword. The buffer must reside in memory external to the MAC. This value is never modified by the Ethernet controller. **NOTE** For optimal performance, make the transmit buffer pointer evenly divisible by 64.
Offset + 8	15 TXE	Transmit error occurred. This field is written by the uDMA. This field indicates that there was a transmit error of some sort reported with the frame. Effectively this field is an OR of the other error fields including UE, EE, FE, LCE, OE, and TSE. This field is valid only when the L field is set.
Offset + 8	14	Reserved, must be cleared.

Table continues on the next page...

Table 463. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 8	13 UE	Underflow error. This field is written by the uDMA. This field indicates that the MAC reported an underflow error on transmit. This field is valid only when the L field is set.
Offset + 8	12 EE	Excess Collision error. This field is written by the uDMA. This field indicates that the MAC reported an excess collision error on transmit. This field is valid only when the L field is set.
Offset + 8	11 FE	Frame with error. This field is written by the uDMA. This field indicates that the MAC reported that the uDMA reported an error when providing the packet. This field is valid only when the L field is set.
Offset + 8	10 LCE	Late collision error. This field is written by the uDMA. This field indicates that the MAC reported that there was a Late Collision on transmit. This field is valid only when the L field is set.
Offset + 8	9 OE	Overflow error. This field is written by the uDMA. This field indicates that the MAC reported that there was a FIFO overflow condition on transmit. This field is only valid when the L field is set.
Offset + 8	8 TSE	Timestamp error. This field is written by the uDMA. This field indicates that the MAC reported a different frame type then a timestamp frame. This field is valid only when the L field is set.
Offset + 8	7–0	Reserved, must be cleared.
Offset + A	15	Reserved, must be cleared.
Offset + A	14 INT	Generate interrupt flags. This field is written by the user. This field is valid regardless of the L field and must be the same for all EBD for a given frame. The uDMA does not update this value.
Offset + A	13 TS	Timestamp. This field is written by the user. This indicates that the uDMA is to generate a timestamp frame to the MAC. This field is valid regardless of the L field and must be the same for all EBD for the given frame. The uDMA does not update this value.
Offset + A	12 PINS	Insert protocol specific checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the protocol checksum and overwrites the corresponding checksum field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + A	11 IINS	Insert IP header checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the IP header checksum and overwrites the corresponding header field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.

Table continues on the next page...

Table 463. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + A	10–9	Reserved, must be cleared.
Offset + A	8 UTLT	Use transmit launch time. If set to 1, TLT high and low values are used to determine if frame can be transmitted. This field must only be set in the *first* BD of a frame. TLT is always used in combination with either the round-robin scheme or the credit-based shaper. TLT can be used with a single BD queue. However, at least one DMA class, DMA*n*CFG[DMA_CLASS_ EN] must be enabled. For example, to use TLT with only queue 0, enable one of the DMA*n*CFG[DMA_CLASS_EN] fields and clear both TDAR1 and TDAR2. Although this enables multi-queue, only the single queue 0 is used.
Offset + A	7–4 FTYPE	Type of frame to be transmitted. If a credit-based scheme is used, this field must match the BD ring queue. 0x0 – Non-AVB. Corresponds to ENET_TDSR. 0x1 – AVB Class A. Corresponds to ENET_TDSR1. 0x2 – AVB Class B. Corresponds to ENET_TDSR2. All other values are reserved.
Offset + A	3–0	Reserved, must be cleared.
Offset + C	15–0 TLT low	Transmit launch time low. Low two bytes of time value that specifies when frame can be transmitted.
Offset + E	15–0 TLT high	Transmit launch time high. High two bytes of time value that specifies when frame can be transmitted.
Offset + 10	15–0	Reserved, must be cleared.
Offset + 12	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 12	14–0	Reserved, must be cleared.
Offset + 14	15–0 1588 timestamp	This value is written by the uDMA . It is valid only when the L field is set.
Offset + 16		
Offset + 18–Offset + 1E	15–0	Reserved, must be cleared.

44.2.13 Client FIFO application interface

The FIFO interface is completely asynchronous from the Ethernet line, and the transmit and receive interface can operate at a different clock rate.

All transfers to/from the user application are handled independently of the core operation, and the core provides a simple interface to user applications based on a two-signal handshake.

44.2.13.1 Data structure description

The data structure defined in the following tables for the FIFO interface must be respected to ensure proper data transmission on the Ethernet line. Byte 0 is sent to and received from the line first.

Table 464. FIFO interface data structure

	63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Word 0	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Word 1	Byte 15	Byte 14	Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8
...	...							

The size of a frame on the FIFO interface may not be a modulo of 64-bit.

The user application may not care about the Ethernet frame formats in full detail. It needs to provide and receive an Ethernet frame with the following structure:

- Ethernet MAC destination address
- Ethernet MAC source address
- Optional 802.1q VLAN tag (VLAN type and info field)
- Ethernet length/type field
- Payload

Frames on the FIFO interface do not contain preamble and SFD fields, which are inserted and discarded by the MAC on transmit and receive, respectively.

- On receive, CRC and frame padding can be stripped or passed through transparently.
- On transmit, padding and CRC can be provided by the user application, or appended automatically by the MAC independently for each frame. No size restrictions apply.

NOTE

On transmit, if ENET*n*_TCR[ADDINS] is set, bytes 6–11 of each frame can be set to any value, since the MAC overwrites the bytes with the MAC address programmed in the ENET*n*_PAUR and ENET*n*_PALR registers.

Table 465. FIFO interface frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address
12–13	Length/type field
14–N	Payload data

VLAN-tagged frames are supported on both transmit and receive, and implement additional information (VLAN type and info).

Table 466. FIFO interface VLAN frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address
12–15	VLAN tag and info
16–17	Length/type field
18–N	Payload data

NOTE

The standard defines that the LSB of the MAC address is sent/received first, while for all the other header fields — in other words, length/type, VLAN tag, VLAN info, and pause quanta — the MSB is sent/received first.

44.2.13.2 Data structure examples

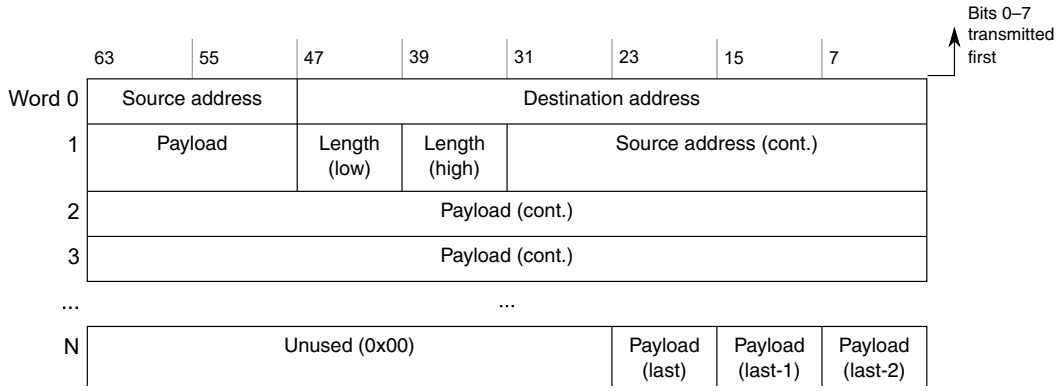


Figure 221. Normal Ethernet frame 64-bit mapping example

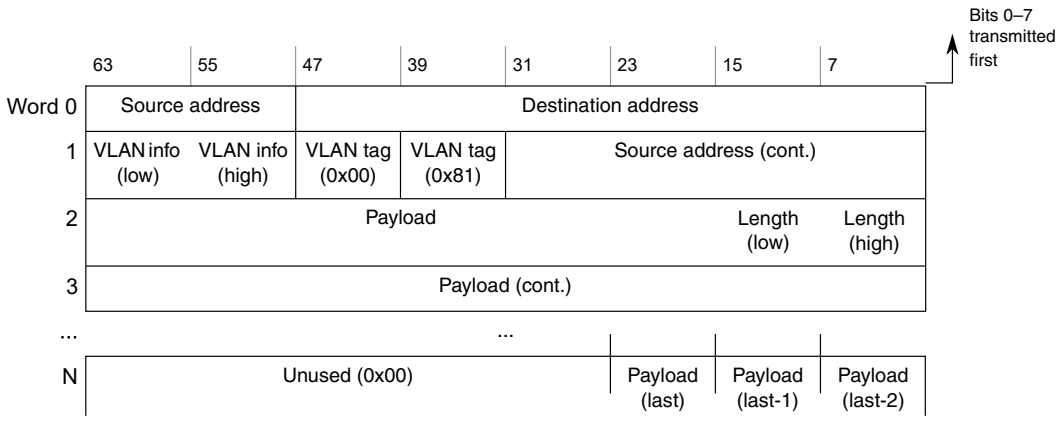


Figure 222. VLAN-tagged frame 64-bit mapping example

If CRC forwarding is enabled (CRCFWD = 0), the last four valid octets of the frame contain the FCS field. The non-significant bytes of the last word can have any value.

44.2.13.3 Frame status

A MAC layer status word and an accelerator status word is available in the receive buffer descriptor. See Enhanced buffer descriptors for details. The status is available with each frame with the last data of the frame.

If the frame status contains a MAC layer error (for example, CRC or length error), RxBD[ME] is also set with the last data of the frame.

44.2.14 FIFO protection

The following sections describe the FIFO protection mechanisms.

44.2.14.1 Transmit FIFO underflow

During a frame transfer, when the transmit FIFO reaches the almost empty threshold with no end-of-frame indication stored in the FIFO, the MAC logic:

- Stops reading data from the FIFO
- Asserts the MII error signal (MII_TXER) (1 in Figure 223) to indicate that the fragment already transferred is not valid
- Deasserts the MII transmit enable signal (MII_TXEN) to terminate the frame transfer (2)

After an underflow, when the application completes the frame transfer (3), the MAC transmit logic discards any new data available in the FIFO until the end of packet is reached (4) and sets the enhanced TxBD[UE] field.

The MAC starts to transfer data on the MII interface when the application sends a new frame with a start of frame indication (5).

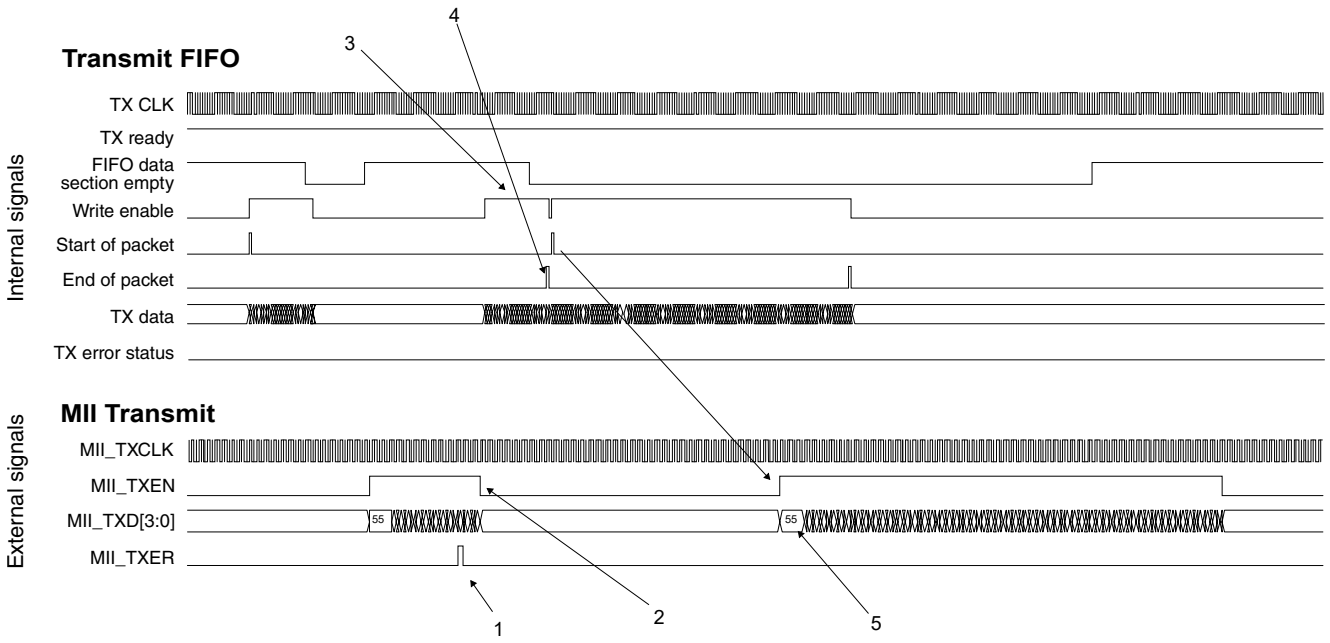


Figure 223. Transmit FIFO underflow protection

44.2.14.2 Transmit FIFO overflow

On the transmit path, when the FIFO reaches the programmable almost full threshold, the internal MAC ready signal is deasserted. The application should stop sending new data. However, if the application keeps sending data, the transmit FIFO overflows, corrupting contents that were previously stored. The core logic sets the enhanced TxBD[OE] field for the next frame transmitted to indicate this overflow occurence.

NOTE

Overflow is a fatal error and must be addressed by resetting the core or clearing ENET*n*_ECR[ETHER_EN], to clear the FIFOs and prepare for normal operation again.

44.2.14.3 Receive FIFO overflow

During a frame reception, if the client application is not able to receive data (1), the MAC receive control truncates the incoming frame when the FIFO reaches the programmable almost-full threshold to avoid an overflow. The frame is subsequently received on the FIFO interface with an error indication (enhanced RxBD[ME] field set together with receive end-of-packet) (2) with the truncation error status field set (3).

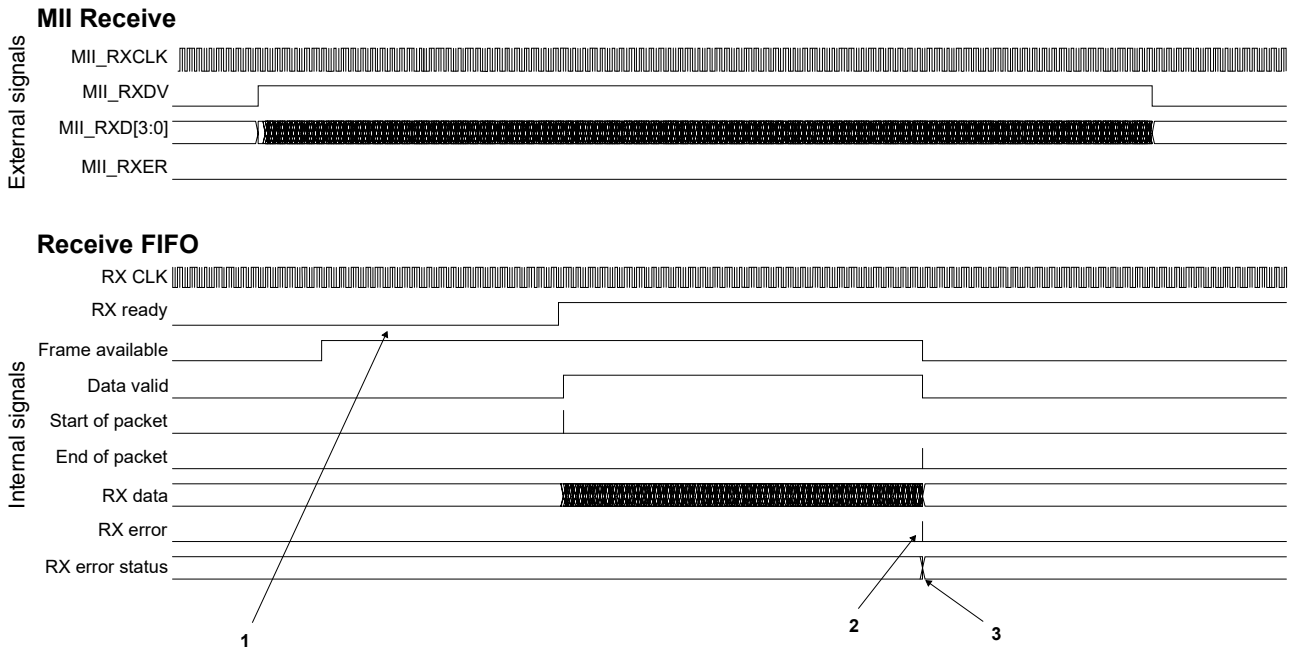


Figure 224. Receive FIFO overflow protection

44.2.15 PHY management interface

The MDIO interface is a two-wire management interface. The MDIO management interface implements a standardized method to access the PHY device management registers. The core implements a leader MDIO interface, which can be connected to up to 32 PHY devices.

44.2.15.1 MDIO clause 22 frame format

The core MDIO leader controller communicates with the follower (PHY device) using frames that are defined in the following table.

A complete frame has a length of 64 bits made up of an optional 32-bit preamble, 14-bit command, 2-bit bus direction change, and 16-bit data. Each bit is transferred on the rising edge of the MDIO clock (MDC signal). The MDIO data signal is tri-stated between frames.

The core PHY management interface supports the standard MDIO specification (IEEE 802.3 Clause 22).

Table 467. MDIO clause 22 frame structure

ST	OP	PHYADR	REGADR	TA	DATA

Table 468. MDIO frame field descriptions

Field	Description
ST (2 bits)	Start indication field, programmed with ENET*n*_MMFR[ST] and equal to 01 for Standard MDIO (Clause 22).
OP (2 bits)	Opcode defines type of operation. Programmed with ENET*n*_MMFR[OP]. 01 Write operation 10 Read operation
PHYADR (5 bits)	Five-bit PHY device address, programmed with ENET*n*_MMFR[PA]. Up to 32 devices can be addressed.
REGADR (5 bits)	Five-bit register address, programmed with ENET*n*_MMFR[RA]. Each PHY can implement up to 32 registers.
TA (2 bits)	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.
Data (16 bits)	Data, set by ENET*n*_MMFR[DATA]. Written to or read from the PHY

44.2.15.2 MDIO clause 45 frame format

The extended MDIO frame structure defined in IEEE 802.3 Clause 45 introduces indirect addressing. First, a write transaction to an address register is done, followed by a write or read transaction which will put the 16-bit data in the register or retrieve the register contents respectively. A preamble of 32 bits of logical ones is sent prior to every transaction. The MDIO data signal is tri-stated between frames.

The extended MDIO defines four transactions, which are determined by the two-bit opcode field.

Table 469. MDIO clause 45 frame structure

ST	OP	PRTAD	DEVAD	TA	ADDR/DATA

All bits are transmitted from left to right (Preamble bits first) and all fields have their Most-Significant bit sent first (leftmost in above table). The complete frame has a length of 64 bits (32-bit preamble, 14-bit command, 2-bit bus direction change, 16-bit data). Each bit is transferred with the rising edge of the MDIO clock (MDC).

The fields and transactions are summarized in the following tables.

Table 470. MDIO clause 45 frame field descriptions

Field	Description
ST	Start indication. Indicates the end of the preamble and start of the frame. This value is 00 for extended MDIO (Clause 45) frames.
OP	Opcode defines if a read or write operation is performed and is programmed with ENET*n*_MMFR[OP]. See Table 471 for more information.

Table continues on the next page...

Table 470. MDIO clause 45 frame field descriptions (continued)

Field	Description
	00 Address write 01 Write operation 10 Read inc. operation 11 Read operation
PRTAD	The port address specifies a MDIO port. Each Port can have up to 32 devices which each can have a separate set of registers.
DEVAD	Device address. Up to 32 devices can be addressed (within a port).
TA	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.
ADDR/DATA	16-bit address (for address write) or data, set by ENET*n*_MMFR[DATA], written to or read from the PHY.

Table 471. MDIO Clause 45 Transactions

Transaction Type	Description
Address	A write transaction to the internal address register of the device/port. The data section of the frame contains the value to be stored in the device's internal address "pointer" register for further transactions.
Write	Data write to a register. The 16 bit data will be written to the register identified by the device-internal address.
Read	Data is read from the register identified by the device-internal address.
Read inc.	Read with address postincrement. The register identified by the device-internal address is read. After this, the device-internal address is incremented. If the address register is all '1' (0xFFFF) no increment is done (i.e. increment does not wrap around).

44.2.15.3 MDIO clock generation

The MDC clock is generated from the internal bus clock (i.e., IPS bus clock) divided by the value programmed in ENET*n*_MSCR[MII_SPEED].

44.2.15.4 MDIO operation

To perform an MDIO access, set the MDIO command register (ENET*n*_MMFR) according to the description provided in MII Management Frame Register (ENETn_MMFR). To check when the programmed access completes, read the ENET*n*_EIR[MII] field.

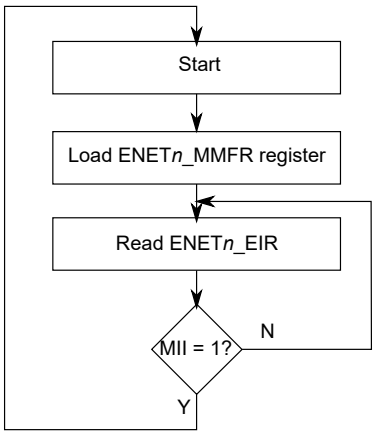


Figure 225. MDIO access overview

44.2.16 Ethernet interfaces

RMII 10/100 is implemented using interface converters/gaskets.

The following table shows how to configure ENET registers to select each interface.

Mode	ECR[SPEED]	RCR[RMII_10T]	RCR[RMII_MODE]	RCR[RGMII_EN]
RMII - 10 Mbit/s	0	1	1	0
RMII - 100 Mbit/s	0	0	1	0
RGMII - 10 Mbit/s	0	1	0	1
RGMII - 100 Mbit/s	0	0	0	1
RGMII - 1000 Mbit/s	1	—	0	1

44.2.16.1 RMII interface

In RMII receive mode, for normal reception following assertion of CRS_DV, RXD[1:0] is 00 until the receiver determines that the receive event has a proper start-of-stream delimiter (SSD). The preamble appears (RXD[1:0]=01) and the MACs begin capturing data following detection of SFD.

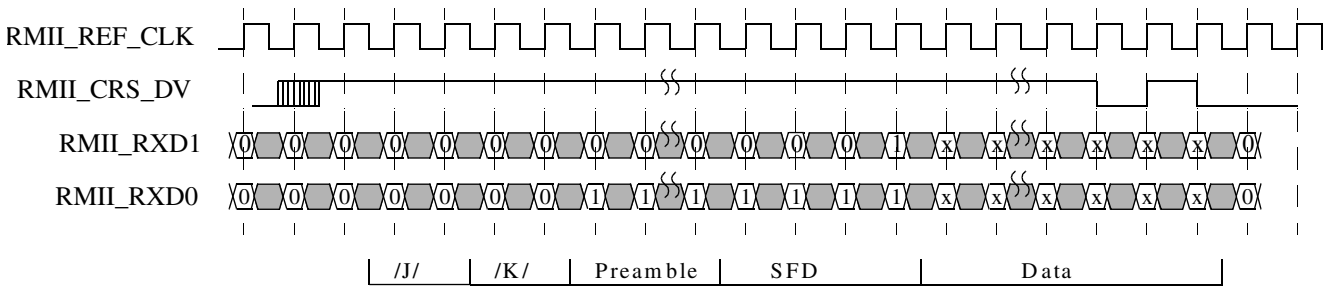


Figure 226. RMII receive operation

If a false carrier is detected (bad SSD), then RXD[1:0] is 10 until the end of the receive event. This is a unique pattern since a false carrier can only occur at the beginning of a packet where the preamble is decoded (RXD[1:0] = 01).

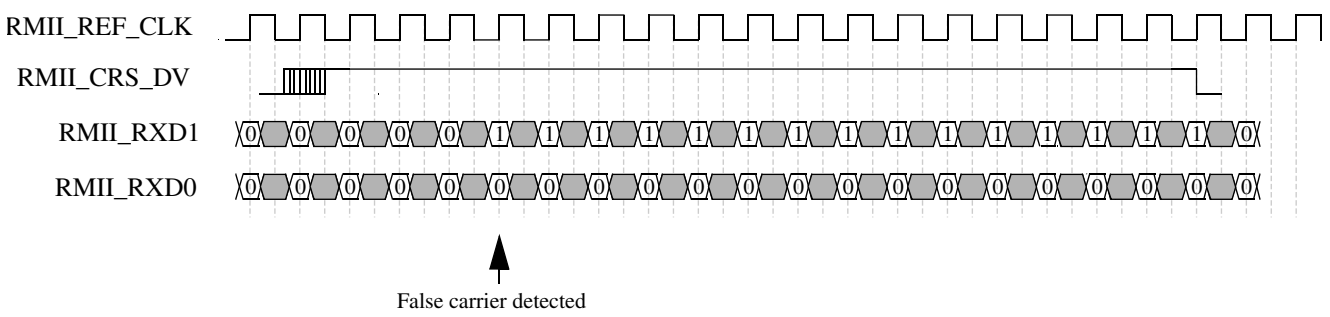


Figure 227. RMII receive operation with false carrier

In RMII transmit mode, TXD[1:0] provides valid data for each REF_CLK period while TXEN is asserted.

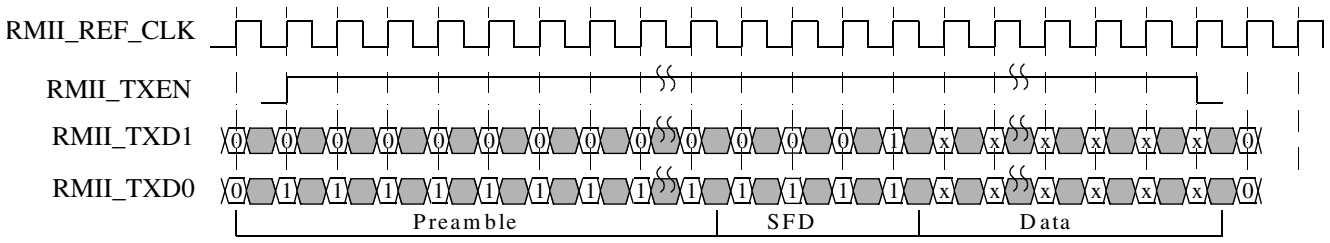


Figure 228. RMII transmit operation

44.2.16.2 RGMII interface

In RGMII modes, the data and control information is multiplexed by taking advantage of both edges of the reference clocks.

The data signals contain the lower four data bits on the rising edge and the upper four bits on the falling edge. The control signals are multiplexed into a single clock cycle using the same technique.

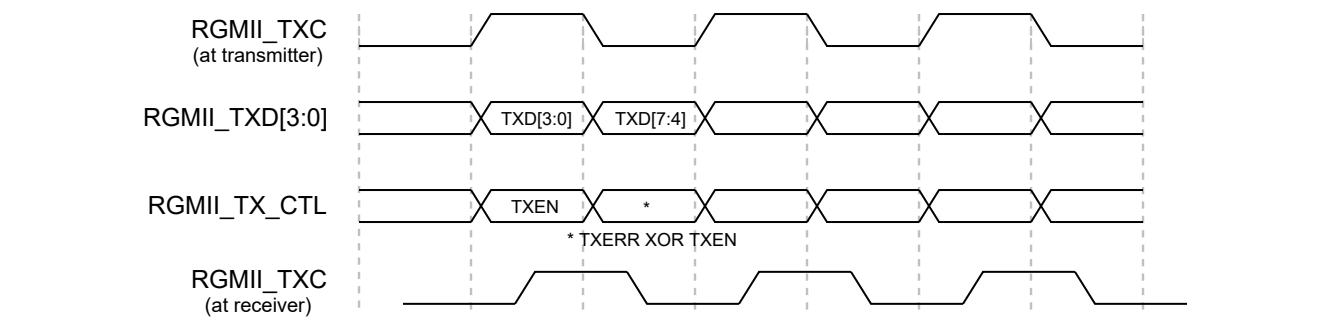


Figure 229. RGMII transmit operation

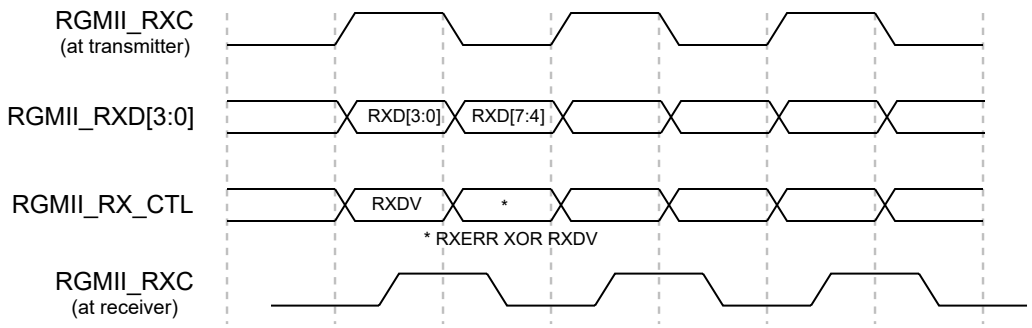


Figure 230. RGMII receive operation

44.2.17 AVB configuration

The following steps give an example of how to initialize the ENET module for AVB.

1. Set up QOS register:
   - Set TX_SCHEME to 000b, credit-based scheme.
   - Set RX_FLUSH0 to 1, enable RX flush for ring 0.
2. Set up TX BD ring and RX BD ring for each queue, 0-2.
   - Program ENET_MRBR, ENET_TDSR, and ENET_RDSR.
   - Program ENET_MRBR1, ENET_TDSR1, and ENET_RDSR1.
   - Program ENET_MRBR2, ENET_TDSR2, and ENET_RDSR2.
   - Program each TX and RX BD ring queue for all classes used in memory.

     NOTE

     If using credit-based scheme, ensure that enhanced transmit buffer descriptor FTYPE field matches BD ring queue, for example:

     — FTYPE = 0h corresponds to ENET_TDSR

     — FTYPE = 1h corresponds to ENET_TDSR1

     — FTYPE = 2h corresponds to ENET_TDSR2
3. Program ENET_DMA1CFG and ENET_DMA2CFG for class 1 and class 2 corresponding to BD ring queue 1 and 2. DMA_CLASS_EN must be set to one for that class to be enabled. See DMACFG register for information on how to program IDLE_SLOPE.
4. Program ENET_RCMR1 and ENET_RCMR2 for class 1 and class 2 matching for receive.

   NOTE

   Even if a match occurs, if ENET_DMA*n*CFG[DMA_CLASS_EN] is zero for the corresponding class, the RX frame will be automatically forwarded to BD ring queue 0.
5. Program the other ENET registers according to application requirements.
6. Program ENET_RDAR, ENET_RDAR1, ENET_RDAR2, ENET_TDAR, ENET_TDAR1, and ENET_TDAR2 to 0100_0000h according to the classes used.
7. Set ENET_ECR[ETHEREN] to one.

44.2.18 Magic packet detection

Magic packet detection wakes a node that is put in power-down mode by the node management agent. Magic packet detection is supported only if the MAC is configured in sleep mode.

44.2.18.1 Sleep mode

To put the MAC in Sleep mode, set ENET*n*_ECR[SLEEP]. At the same time ENET*n*_ECR[MAGICEN] should also be set to enable magic packet detection.

In addition, if ENET is enabled, write 1 to ENET*n*_ECR[SLEEP] before entering into low power mode.

When the MAC is in Sleep mode:

- The transmit logic is disabled.
- The FIFO receive/transmit functions are disabled.
- The receive logic is kept in Normal mode, but it ignores all traffic from the line except magic packets. They are detected so that a remote agent can wake the node.

44.2.18.2 Magic packet detection

The core is designed to detect magic packets (See "Magic Packets" topics) with the destination address set to:

- Any multicast address
- The broadcast address
- The unicast address programmed in PADDR1/2

When a magic packet is detected, EIR[WAKEUP] is set and none of the statistic registers are incremented.

44.2.18.3 Wakeup

When a magic packet is detected, indicated by ENET*n*_EIR[WAKEUP], ENET*n*_ECR[SLEEP] should be cleared to resume normal operation of the MAC. Clearing the SLEEP bit automatically masks ENET*n*_ECR[MAGICEN], disabling magic packet detection.

44.2.19 Clocking

The table found here describes the clock sources for ENET. See chip-specific clocking section for clock setting, configuration and gating information.

Table 472. ENET clocks

Clock name	Description
ipg_clk	Module clock
ipg_clk_rmii	Module clock for RMII
ipg_clk_s	Peripheral access clock
ipg_clk_time (ts_clk)	Peripheral clock
txc_sampling_clk	Transmit sampling clock

44.2.19.1 Reference clock

The input clocks to the ENET module must meet the specifications in the following table.

Ethernet speed mode	Ethernet bus clock	Minimum ENET system clock
10 Mbit/s	2.5 MHz	5 MHz
100 Mbit/s	25.0 MHz	50 MHz
1000 Mbit/s	125.0 MHz	133 MHz

44.2.20 Reset

The following sections describe the resets and stop controls.

44.2.20.1 Hardware reset

To reset the Ethernet module, set ENET*n*_ECR[RESET].

44.2.20.2 Soft reset

When ENET*n*_ECR[ETHER_EN] is cleared during operation, the following occurs:

- uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers.
- A currently ongoing transmit is terminated by asserting GMII/TXER to the PHY.
- A currently ongoing transmit FIFO write from the application is terminated by stopping the write to the FIFO, and all further data from the application is ignored. All subsequent writes are ignored until re-enabled.
- A currently ongoing receive FIFO read is terminated. The RxBD has arbitrary values in this case.

44.2.20.3 Debug mode

When the processor enters debug mode and ECR[DBGEN] is set, the MAC enters a freeze state where it stops all transmit and receive activities gracefully. The following happens when the MAC enters hardware freeze:

- A currently ongoing receive transaction on the receive application interface is completed as normal. No further frames are read from the FIFO.
- A currently ongoing transmit transaction on the transmit application interface is completed as normal (in other words, until writing end-of-packet (EOP)).
- A currently ongoing frame receive is completed normally, after which no further frames are accepted from the RMII/GMII.
- A currently ongoing frame transmit is completed normally, after which no further frames are transmitted.

44.2.20.4 Graceful stop

During a graceful stop, any currently ongoing transactions are completed normally and no further frames are accepted. The MAC can resume from a graceful stop without the need for a reset (for example, clearing ETHER_EN is not required). The following conditions lead to a graceful stop of the MAC transmit or receive datapaths.

44.2.20.4.1 Graceful transmit stop (GTS)

When gracefully stopped, the MAC is no longer reading frame data from the transmit FIFO and has completed any ongoing transmission. In any of the following conditions, the transmit datapath stops after an ongoing frame transmission has been completed normally.

- ENET*n*_TCR[GTS] is set by software.
- ENET*n*_TCR[TFC_PAUSE] is set by software requesting a pause frame transmission. The status (and register bit) is cleared after the pause frame has been sent.
- A pause frame was received stopping the transmitter. The stopped situation is terminated when the pause timer expires or a pause frame with zero quanta is received.

- MAC is placed in Sleep mode by software or the processor entering Stop mode (see Sleep mode).
- The MAC is in Hardware Freeze mode.

When the transmitter has reached its stopped state, the following events occur:

- The GRA interrupt is asserted, when transitioned into stopped state.
- In Hardware Freeze mode, the GRA interrupt does not wait for the application write completion and asserts when the transmit state machine (in other words, line side of TX FIFO) reaches its stopped state.

44.2.20.4.2 Graceful receive stop (GRS)

When gracefully stopped, the MAC is no longer writing frames into the receive FIFO. The receive datapath stops after any ongoing frame reception has been completed normally, if any of the following conditions occur:

- MAC is placed in Sleep mode either by the software or the processor is in Stop mode). The MAC continues to receive frames and search for magic packets if enabled (see Magic packet detection). However, no frames are written into the receive FIFO, and therefore are not forwarded to the application.
- The MAC is in Hardware Freeze mode. The MAC does not accept any frames from the RMII/GMII.

When the receive datapath is stopped, the following events occur:

- If the RX is in the stopped state, RCR[GRS] is set
- The GRA interrupt is asserted when the transmitter and receiver are stopped
- Any ongoing receive transaction to the application (RX FIFO read) continues normally until the frame end of package (EOP) is reached. After this, the following occurs:
  - When Sleep mode is active, all further frames are discarded, flushing the RX FIFO
  - In Hardware Freeze mode, no further frames are delivered to the application and they stay in the receive FIFO.

NOTE

The assertion of GRS does not wait for an ongoing FIFO read transaction on the application side of the FIFO (FIFO read).

44.2.20.4.3 Graceful stop interrupt (GRA)

The graceful stop interrupt (GRA) is asserted for the following conditions:

- In Sleep mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- In Hardware Freeze mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- The MAC transmit datapath is stopped for any other condition (GTS, TFC_PAUSE, pause received).

The GRA interrupt is triggered only once when the stopped state is entered. If the interrupt is cleared while the stop condition persists, no further interrupt is triggered.

44.2.21 Interrupts

The purpose of the interrupt coalescing is to reduce the number of interrupts generated by the MAC so as to reduce the CPU loading.

To facilitate this interrupt coalescing for each queue, these registers are available with the same control and configuration fields.

- Transmit Interrupt Coalescing Register (TXICn) where n=0,1,2 for queue/class 0,1,2.
- Receive Interrupt Coalescing Register (RXICn) where n=0,1,2 for queue/class 0,1,2.

When coalescing is enabled by asserting the corresponding ICEN field and such interrupt is also enabled by the corresponding interrupt mask of the EIMR register, the MAC generates an interrupt when the threshold number of frames is reached (defined by ICFT) or when the threshold timer expires (defined by ICTT).

When coalescing is disabled by de-asserting ICEN, but interrupt is enabled by the corresponding interrupt mask of the EIMR register, the MAC generates an interrupt as they are received without using coalescing. Interrupt coalescing is done for each transmit and receive queue/class independently.

44.2.21.1 Interrupt coalescence setup

Interrupt coalescence supports both legacy and enhanced BDs. The following guidelines are recommended when setting up interrupt coalescence.

- When the MAC is configured for enhanced (IEEE 1588) mode, that is, enhanced BDs:
  - Set the INT bit in the enhanced received buffer descriptor to one.
  - Set the INT bit in the enhanced transmit buffer descriptor(s) to one.
- Clear the RXB, RXB1, and RXB2 fields in the EIMR register.
- Clear the TXB, TXB1, and TXB2 fields in the EIMR register.

44.2.21.2 Updating the frame count threshold on-the-fly

To update the ICFT field in the RXIC*n* and TXIC*n* registers:

1. Disable interrupt coalescence by clearing the appropriate ICEN field. This will allow the internal interrupt coalescence counter to reset to zero.

   NOTE

   When disabling interrupt coalescence, if an interrupt event is pending, that is, the interrupt counter is not zero, then an interrupt will occur.

2. Write the new threshold value to the ICFT field.
3. Set ICEN to one.

   NOTE

   The ICFT field can be updated on-the-fly without disabling the ICEN field. The hardware interrupt will continue and there is a possibility that an interrupt will occur depending on the state of the hardware counter and the previous ICFT value.

44.2.21.3 Updating the timer threshold on-the-fly

To update the ICTT field in the RXIC*n* and TXIC*n* registers:

1. Disable interrupt coalescence by clearing the appropriate ICEN field. This will allow the internal interrupt coalescence counter to reset to zero.

   NOTE

   When disabling interrupt coalescence, if an interrupt event is pending, that is, the interrupt counter is not zero, then an interrupt will occur.

2. Write the new timer value to the ICTT field.
3. Set ICEN to one.

44.3 External signals

The table found here describes the external signals of ENET.

Table 473. ENET external signals

Signal	Description	Mode	Direction
1588_EVENT_IN/ 1588_EVENT_OUT	Capture/compare block input/output event bus signal. • When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. • When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in the register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.	RMII	IO
ENET_1588_CLKIN	Alternate IEEE 1588 Ethernet clock input; Clock period should be an integer number in nanoseconds.	RMII	I
MDC	Output clock provides a timing reference to the PHY for data transfers on the MDIO signal.	RGMII/RMII	O
MDIO	Transfers control information between the external PHY and the media access controller. Data is synchronous to MDC. This signal is an input after reset.	RGMII/RMII	IO
RXER	When asserted with RXDV, indicates an error detection from the PHY in the current frame.	RMII	I
RX_EN	Asserting this input indicates the PHY has valid nibbles present on the MII. RX_EN must remain asserted from the first recovered nibble of the frame through to the last nibble. Asserting RX_EN must start no later than the SFD and exclude any EOF. **NOTE** In RMII mode, this pin also generates the CRS signal.	RMII	I
TXEN	Indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is de-asserted before the first TXCLK following the final nibble of the frame. **NOTE** In RGMII mode, it is present as RGMII_TX_CTL, and contains TXEN on the rising edge of RGMII_TXC, and a logical derivative of TXEN and TXER (TXEN XOR TXER) on the falling edge of RGMII_TXC.	RMII	O
RXD[3:0]	Contains the Ethernet input data transferred from the PHY to the media-access controller when RXEN is asserted.	RMII/RGMII	I

Table continues on the next page...

Table 473. ENET external signals (continued)

Signal	Description	Mode	Direction
	NOTE In RMII mode, bits [1:0] provide the Ethernet input data.		
TXD[3:0]	Serial output Ethernet data. Only valid during TX_EN assertion. **NOTE** In RMII mode, bits [1:0] provide the Serial output Ethernet data.	RMII/RGMII	O
RGMII_RX_CTL	Asserting this input indicates the PHY has valid nibbles present on the MII. RXDV must remain asserted from the first recovered nibble of the frame through to the last nibble. Asserting RXDV must start no later than the SFD and exclude any EOF. **NOTE** • In RMII mode, this signal is used as RMII_CRS_DV. • In RGMII mode, contains RXDV on the rising edge of RGMII_RXC, and a logical derivative of RXDV and RXER (RXDV XOR RXER) on the falling edge of RGMII_RXC.	RGMII	I
RGMII_RXC	Timing reference for RGMII_RXD[3:0] and RGMII_RX_CTL	RGMII	I
RGMII_TX_CTL	Indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is deasserted before the first TXCLK following the final nibble of the frame. **NOTE** In RGMII mode, contains TXEN on the rising edge of RGMII_TXC, and a logical derivative of TXEN and TXER (TXEN XOR TXER) on the falling edge of RGMII_TXC.	RGMII	O
RGMII_TXC	Timing reference for RGMII_TXD[3:0] and RGMII_TX_CTL.	RGMII	O
RMII_REF_CLK	In RMII mode, this signal is the reference clock for receive, transmit, and the control interface.	RMII	IO
RGMII_REF_CLK	In RGMII mode, this signal provides 125Mhz external reference clock input.	RGMII	I

44.4 Initialization

Initialization is not required for this module.

44.5 Memory map/register definition

ENET registers must be read or written with 32-bit accesses. Non-32 bit accesses will terminate with an error.

Reserved bits should be written with 0 and ignored on read. Unused registers read zero and a write has no effect.

This table shows Ethernet registers organization.

Table 474. Register map summary

Offset Address	Section	Description
0x0000 – 0x01FF	Configuration	Core control and status registers
0x0200 – 0x03FF	Statistics counters	MIB and Remote Network Monitoring (RFC 2819) registers
0x0400 – 0x0430	1588 control	1588 adjustable timer (TSM) and 1588 frame control
0x0600 – 0x07FC	Capture/Compare block	Registers for the Capture/Compare block

44.5.1 ENET register descriptions

44.5.1.1 ENET memory map

ENET base address: 4289_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	Interrupt Event (EIR)	32	RW	0000_0000
8	Interrupt Mask (EIMR)	32	RW	0000_0000
10	Receive Descriptor Active Register - Ring 0 (RDAR)	32	RW	0000_0000
14	Transmit Descriptor Active Register - Ring 0 (TDAR)	32	RW	0000_0000
24	Ethernet Control (ECR)	32	RW	7000_0000
40	MII Management Frame (MMFR)	32	RW	0000_0000
44	MII Speed Control (MSCR)	32	RW	0000_0000
64	MIB Control (MIBC)	32	RW	C000_0000
84	Receive Control (RCR)	32	RW	05EE_0001
C4	Transmit Control (TCR)	32	RW	0000_0000
E4	Physical Address Lower (PALR)	32	RW	0000_0000
E8	Physical Address Upper (PAUR)	32	RW	0000_8808
EC	Opcode/Pause Duration (OPD)	32	RW	0001_0000
F0 - F8	Transmit Interrupt Coalescing (TXIC0 - TXIC2)	32	RW	0000_0000
100 - 108	Receive Interrupt Coalescing (RXIC0 - RXIC2)	32	RW	0000_0000
118	Descriptor Individual Upper Address (IAUR)	32	RW	0000_0000
11C	Descriptor Individual Lower Address (IALR)	32	RW	0000_0000
120	Descriptor Group Upper Address (GAUR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
124	Descriptor Group Lower Address (GALR)	32	RW	0000_0000
144	Transmit FIFO Watermark (TFWR)	32	RW	0000_0000
160	Receive Descriptor Ring 1 Start (RDSR1)	32	RW	0000_0000
164	Transmit Buffer Descriptor Ring 1 Start (TDSR1)	32	RW	0000_0000
168	Maximum Receive Buffer Size Register - Ring 1 (MRBR1)	32	RW	0000_0000
16C	Receive Descriptor Ring 2 Start (RDSR2)	32	RW	0000_0000
170	Transmit Buffer Descriptor Ring 2 Start (TDSR2)	32	RW	0000_0000
174	Maximum Receive Buffer Size Register - Ring 2 (MRBR2)	32	RW	0000_0000
180	Receive Descriptor Ring 0 Start (RDSR)	32	RW	0000_0000
184	Transmit Buffer Descriptor Ring 0 Start (TDSR)	32	RW	0000_0000
188	Maximum Receive Buffer Size Register - Ring 0 (MRBR)	32	RW	0000_0000
190	Receive FIFO Section Full Threshold (RSFL)	32	RW	0000_0000
194	Receive FIFO Section Empty Threshold (RSEM)	32	RW	0000_0000
198	Receive FIFO Almost Empty Threshold (RAEM)	32	RW	0000_0004
19C	Receive FIFO Almost Full Threshold (RAFL)	32	RW	0000_0004
1A0	Transmit FIFO Section Empty Threshold (TSEM)	32	RW	0000_0000
1A4	Transmit FIFO Almost Empty Threshold (TAEM)	32	RW	0000_0004
1A8	Transmit FIFO Almost Full Threshold (TAFL)	32	RW	0000_0008
1AC	Transmit Inter-Packet Gap (TIPG)	32	RW	0000_000C
1B0	Frame Truncation Length (FTRL)	32	RW	0000_07FF
1C0	Transmit Accelerator Function Configuration (TACC)	32	RW	0000_0000
1C4	Receive Accelerator Function Configuration (RACC)	32	RW	0000_0000
1C8 - 1CC	Receive Classification Match Register for Class n (RCMR1 - RCMR2)	32	RW	0000_0000
1D8 - 1DC	DMA Class Based Configuration (DMA1CFG - DMA2CFG)	32	RW	0000_0000
1E0	Receive Descriptor Active Register - Ring 1 (RDAR1)	32	RW	0000_0000
1E4	Transmit Descriptor Active Register - Ring 1 (TDAR1)	32	RW	0000_0000
1E8	Receive Descriptor Active Register - Ring 2 (RDAR2)	32	RW	0000_0000
1EC	Transmit Descriptor Active Register - Ring 2 (TDAR2)	32	RW	0000_0000
1F0	QOS Scheme (QOS)	32	RW	0000_0000
204	Tx Packet Count Statistic (RMON_T_PACKETS)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
208	Tx Broadcast Packets Statistic (RMON_T_BC_PKT)	32	R	0000_0000
20C	Tx Multicast Packets Statistic (RMON_T_MC_PKT)	32	R	0000_0000
210	Tx Packets with CRC/Align Error Statistic (RMON_T_CRC_ALIGN)	32	R	0000_0000
214	Tx Packets Less Than Bytes and Good CRC Statistic (RMON_T_UNDERSIZE)	32	R	0000_0000
218	Tx Packets GT MAX_FL bytes and Good CRC Statistic (RMON_T_OVERSIZE)	32	R	0000_0000
21C	Tx Packets Less Than 64 Bytes and Bad CRC Statistic (RMON_T_FRAG)	32	R	0000_0000
220	Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic (RMON_T_JAB)	32	R	0000_0000
224	Tx Collision Count Statistic (RMON_T_COL)	32	R	0000_0000
228	Tx 64-Byte Packets Statistic (RMON_T_P64)	32	R	0000_0000
22C	Tx 65- to 127-byte Packets Statistic (RMON_T_P65TO127)	32	R	0000_0000
230	Tx 128- to 255-byte Packets Statistic (RMON_T_P128TO255)	32	R	0000_0000
234	Tx 256- to 511-byte Packets Statistic (RMON_T_P256TO511)	32	R	0000_0000
238	Tx 512- to 1023-byte Packets Statistic (RMON_T_P512TO1023)	32	R	0000_0000
23C	Tx 1024- to 2047-byte Packets Statistic (RMON_T_P1024TO2047)	32	R	0000_0000
240	Tx Packets Greater Than 2048 Bytes Statistic (RMON_T_P_GTE2048)	32	R	0000_0000
244	Tx Octets Statistic (RMON_T_OCTETS)	32	R	0000_0000
24C	Frames Transmitted OK Statistic (IEEE_T_FRAME_OK)	32	R	0000_0000
250	Frames Transmitted with Single Collision Statistic (IEEE_T_1COL)	32	R	0000_0000
254	Frames Transmitted with Multiple Collisions Statistic (IEEE_T_MCOL)	32	R	0000_0000
258	Frames Transmitted after Deferral Delay Statistic (IEEE_T_DEF)	32	R	0000_0000
25C	Frames Transmitted with Late Collision Statistic (IEEE_T_LCOL)	32	R	0000_0000
260	Frames Transmitted with Excessive Collisions Statistic (IEEE_T_EXCOL)	32	R	0000_0000
264	Frames Transmitted with Tx FIFO Underrun Statistic (IEEE_T_MACERR)	32	R	0000_0000
268	Frames Transmitted with Carrier Sense Error Statistic (IEEE_T_CSERR)	32	R	0000_0000
26C	Reserved Statistic (IEEE_T_SQE)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
270	Flow Control Pause Frames Transmitted Statistic (IEEE_T_FDXFC)	32	R	0000_0000
274	Octet Count for Frames Transmitted w/o Error Statistic (IEEE_T_OCTETS_OK)	32	R	0000_0000
284	Rx Packet Count Statistic (RMON_R_PACKETS)	32	R	0000_0000
288	Rx Broadcast Packets Statistic (RMON_R_BC_PKT)	32	R	0000_0000
28C	Rx Multicast Packets Statistic (RMON_R_MC_PKT)	32	R	0000_0000
290	Rx Packets with CRC/Align Error Statistic (RMON_R_CRC_ALIGN)	32	R	0000_0000
294	Rx Packets with Less Than 64 Bytes and Good CRC Statistic (RMON_R_UNDERSIZE)	32	R	0000_0000
298	Rx Packets Greater Than MAX_FL and Good CRC Statistic (RMON_R_OVERSIZE)	32	R	0000_0000
29C	Rx Packets Less Than 64 Bytes and Bad CRC Statistic (RMON_R_FRAG)	32	R	0000_0000
2A0	Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic (RMON_R_JAB)	32	R	0000_0000
2A8	Rx 64-Byte Packets Statistic (RMON_R_P64)	32	R	0000_0000
2AC	Rx 65- to 127-Byte Packets Statistic (RMON_R_P65TO127)	32	R	0000_0000
2B0	Rx 128- to 255-Byte Packets Statistic (RMON_R_P128TO255)	32	R	0000_0000
2B4	Rx 256- to 511-Byte Packets Statistic (RMON_R_P256TO511)	32	R	0000_0000
2B8	Rx 512- to 1023-Byte Packets Statistic (RMON_R_P512TO1023)	32	R	0000_0000
2BC	Rx 1024- to 2047-Byte Packets Statistic (RMON_R_P1024TO2047)	32	R	0000_0000
2C0	Rx Packets Greater than 2048 Bytes Statistic (RMON_R_P_GTE2048)	32	R	0000_0000
2C4	Rx Octets Statistic (RMON_R_OCTETS)	32	R	0000_0000
2C8	Frames not Counted Correctly Statistic (IEEE_R_DROP)	32	R	0000_0000
2CC	Frames Received OK Statistic (IEEE_R_FRAME_OK)	32	R	0000_0000
2D0	Frames Received with CRC Error Statistic (IEEE_R_CRC)	32	R	0000_0000
2D4	Frames Received with Alignment Error Statistic (IEEE_R_ALIGN)	32	R	0000_0000
2D8	Receive FIFO Overflow Count Statistic (IEEE_R_MACERR)	32	R	0000_0000
2DC	Flow Control Pause Frames Received Statistic (IEEE_R_FDXFC)	32	R	0000_0000
2E0	Octet Count for Frames Received without Error Statistic (IEEE_R_OCTETS_OK)	32	R	0000_0000
400	Adjustable Timer Control (ATCR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
404	Timer Value (ATVR)	32	RW	0000_0000
408	Timer Offset (ATOFF)	32	RW	0000_0000
40C	Timer Period (ATPER)	32	RW	3B9A_CA00
410	Timer Correction (ATCOR)	32	RW	0000_0000
414	Time-Stamping Clock Period (ATINC)	32	RW	0000_0000
418	Timestamp of Last Transmitted Frame (ATSTMP)	32	R	0000_0000
604	Timer Global Status (TGSR)	32	RW	0000_0000
608	Timer Control Status (TCSR0)	32	RW	0000_0000
60C	Timer Compare Capture (TCCR0)	32	RW	0000_0000
610	Timer Control Status (TCSR1)	32	RW	0000_0000
614	Timer Compare Capture (TCCR1)	32	RW	0000_0000
618	Timer Control Status (TCSR2)	32	RW	0000_0000
61C	Timer Compare Capture (TCCR2)	32	RW	0000_0000
620	Timer Control Status (TCSR3)	32	RW	0000_0000
624	Timer Compare Capture (TCCR3)	32	RW	0000_0000

44.5.1.2 Interrupt Event (EIR)

When an event occurs that sets a bit in EIR, an interrupt occurs if the corresponding bit in the interrupt mask register (EIMR) is also set. Writing a 1 to an EIR bit clears it; writing 0 has no effect. This register is cleared upon hardware reset.

NOTE

TxBD[INT] and RxBD[INT] must be set to 1 to allow setting the corresponding EIR register flags in enhanced mode, ENET_ECR[EN1588] = 1. Legacy mode does not require these flags to be enabled.

Offset

Register	Offset
EIR	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EBERR	LC	RL	UN	PLR	WAKEUP	TS_AVAIL
W		W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TS_TIMER	RXFLUSH...	RXFLUSH...	RXFLUSH...					TXF2	TXB2	RXF2	RXB2	TXF1	TXB1	RXF1	RXB1
W	W1C	W1C	W1C	W1C	0			0	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30 BABR	Babbling Receive Error Indicates a frame was received with length in excess of RCR[MAX_FL] bytes.
29 BABT	Babbling Transmit Error Indicates the transmitted frame length exceeds RCR[MAX_FL] bytes. Usually this condition is caused when a frame that is too long is placed into the transmit data buffer(s). Truncation does not occur.
28 GRA	Graceful Stop Complete This interrupt is asserted after the transmitter is put into a pause state after completion of the frame currently being transmitted. See Graceful Transmit Stop (GTS) for conditions that lead to graceful stop. **NOTE** The GRA interrupt is asserted only when the TX transitions into the stopped state. If this bit is cleared by writing 1 and the TX is still stopped, the bit is not set again.
27 TXF	Transmit Frame Interrupt Indicates a frame has been transmitted and the last corresponding buffer descriptor has been updated.
26 TXB	Transmit Buffer Interrupt Indicates a transmit buffer descriptor has been updated.
25 RXF	Receive Frame Interrupt Indicates a frame has been received and the last corresponding buffer descriptor has been updated.
24 RXB	Receive Buffer Interrupt Indicates a receive buffer descriptor is not the last in the frame has been updated.

Table continues on the next page...

Table continued from the previous page...

Field	Description
23 MII	MII Interrupt. Indicates that the MII has completed the data transfer requested.
22 EBERR	Ethernet Bus Error Indicates a system bus error occurred when a uDMA transaction is underway. When this bit is set, ECR[ETHEREN] is cleared, halting frame processing by the MAC. When this occurs, software must ensure proper actions, possibly resetting the system, to resume normal operation.
21 LC	Late Collision Indicates a collision occurred beyond the collision window (slot time) in half-duplex mode. The frame truncates with a bad CRC and the remainder of the frame is discarded.
20 RL	Collision Retry Limit Indicates a collision occurred on each of 16 successive attempts to transmit the frame. The frame is discarded without being transmitted and transmission of the next frame commences. This error can only occur in half-duplex mode.
19 UN	Transmit FIFO Underrun Indicates the transmit FIFO became empty before the complete frame was transmitted. **NOTE** In situations where the device has various leaders generating high traffic, a FIFO underrun can occur on the transmit FIFO. To avoid transmit FIFO underrun, store and forward can be enabled in ENET_TFWR[STRFWD]. See Transmit FIFO Watermark (TFWR). Also, a higher priority can be set for ENET traffic using available means on the central bus fabric connecting the ENET module.
18 PLR	Payload Receive Error Indicates a frame was received with a payload length error. See Frame Length/Type Verification: Payload Length Check for more information.
17 WAKEUP	Node Wakeup Request Indication Read-only status bit to indicate that a magic packet has been detected. Will act only if ECR[MAGICEN] is set.
16 TS_AVAIL	Transmit Timestamp Available Indicates that the timestamp of the last transmitted timing frame is available in the ATSTMP register.
15 TS_TIMER	Timestamp Timer The adjustable timer reached the period event. A period event interrupt can be generated if ATCR[PEREN] is set and the timer wraps according to the periodic setting in the ATPER register. Set the timer period value before setting ATCR[PEREN].
14 RXFLUSH_2	RX DMA Ring 2 flush indication. This ring's RX frame has been flushed due to either RDAR2[RDAR] or RxBD[E] being clear and only if QOS[RXFLUSH_2] is enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Description
13 RXFLUSH_1	RX DMA Ring 1 flush indication. This ring's RX frame has been flushed due to either RDAR1[RDAR] or RxBD[E] being clear and only if QOS[RXFLUSH_1] is enabled.
12 RXFLUSH_0	RX DMA Ring 0 flush indication. . This ring's RX frame has been flushed due to either RDAR[RDAR] or RxBD[E] being clear and only if QOS[RXFLUSH_0] is enabled.
11-9 —	This write-only field is reserved. It must always be written with the value 0.
8 —	This write-only field is reserved. It must always be written with the value 0.
7 TXF2	Transmit frame interrupt, class 2 This bit indicates that a frame has been transmitted and the last corresponding buffer descriptor has been updated (ring/class 2).
6 TXB2	Transmit buffer interrupt, class 2 This field indicates that a transmit buffer descriptor has been updated (ring/class 2).
5 RXF2	Receive frame interrupt, class 2 This field indicates that a frame has been received and the last corresponding buffer descriptor has been updated (ring/class 2).
4 RXB2	Receive buffer interrupt, class 2 This field indicates that a receive buffer descriptor, that not the last in the frame, has been updated (ring/class 2).
3 TXF1	Transmit frame interrupt, class 1 This bit indicates that a frame has been transmitted and the last corresponding buffer descriptor has been updated (ring/class 1).
2 TXB1	Transmit buffer interrupt, class 1 This field indicates that a transmit buffer descriptor has been updated (ring/class 1).
1 RXF1	Receive frame interrupt, class 1 This field indicates that a frame has been received and the last corresponding buffer descriptor has been updated (ring/class 1).
0 RXB1	Receive buffer interrupt, class 1 This field indicates that a receive buffer descriptor, that not the last in the frame, has been updated (ring/class 1).

44.5.1.3 Interrupt Mask (EIMR)

EIMR controls which interrupt events are allowed to generate actual interrupts. A hardware reset clears this register. If the corresponding bits in the EIR and EIMR registers are set, an interrupt is generated. The interrupt signal remains asserted until a 1 is written to the EIR field (write 1 to clear) or a 0 is written to the EIMR field.

Offset

Register	Offset
EIMR	8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R		BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EBERR	LC	RL	UN	PLR	WAKEUP	TS_AVAIL
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TS_TIMER	RXFLUSH...	RXFLUSH...	RXFLUSH...					TXF2	TXB2	RXF2	RXB2	TXF1	TXB1	RXF1	RXB1
W					0			0								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	This write-only field is reserved. It must always be written with the value 0.
30 BABR	BABR Interrupt Mask Corresponds to interrupt source EIR[BABR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
29 BABT	BABT Interrupt Mask Corresponds to interrupt source EIR[BABT] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABT field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.

Table continues on the next page...

Table continued from the previous page...

Field	Description
28 GRA	GRA Interrupt Mask Corresponds to interrupt source EIR[GRA] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR GRA field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
27 TXF	TXF Interrupt Mask Corresponds to interrupt source EIR[TXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
26 TXB	TXB Interrupt Mask Corresponds to interrupt source EIR[TXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
25 RXF	RXF Interrupt Mask Corresponds to interrupt source EIR[RXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
24 RXB	RXB Interrupt Mask Corresponds to interrupt source EIR[RXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXB field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
23	MII Interrupt Mask

Table continues on the next page...

Table continued from the previous page...

Field	Description
MII	Corresponds to interrupt source EIR[MII] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR MII field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
22 EBERR	EBERR Interrupt Mask Corresponds to interrupt source EIR[EBERR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR EBERR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
21 LC	LC Interrupt Mask Corresponds to interrupt source EIR[LC] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR LC field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
20 RL	RL Interrupt Mask Corresponds to interrupt source EIR[RL] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
19 UN	UN Interrupt Mask Corresponds to interrupt source EIR[UN] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR UN field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
18 PLR	PLR Interrupt Mask Corresponds to interrupt source EIR[PLR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	The corresponding EIR PLR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
17 WAKEUP	WAKEUP Interrupt Mask Corresponds to interrupt source EIR[WAKEUP] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR WAKEUP field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
16 TS_AVAIL	TS_AVAIL Interrupt Mask Corresponds to interrupt source EIR[TS_AVAIL] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_AVAIL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
15 TS_TIMER	TS_TIMER Interrupt Mask Corresponds to interrupt source EIR[TS_TIMER] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_TIMER field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
14 RXFLUSH_2	Corresponds to interrupt source EIR[RXFLUSH_2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXFLUSH_2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
13 RXFLUSH_1	Corresponds to interrupt source EIR[RXFLUSH_1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXFLUSH_1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 RXFLUSH_0	Corresponds to interrupt source EIR[RXFLUSH_0] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXFLUSH_0] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
11-9 —	This write-only field is reserved. It must always be written with the value 0.
8 —	This write-only field is reserved. It must always be written with the value 0.
7 TXF2	Transmit frame interrupt, class 2 Corresponds to interrupt source EIR[TXF2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXF2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
6 TXB2	Transmit buffer interrupt, class 2 Corresponds to interrupt source EIR[TXB2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXB2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
5 RXF2	Receive frame interrupt, class 2 Corresponds to interrupt source EIR[RXF2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXF2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
4 RXB2	Receive buffer interrupt, class 2 Corresponds to interrupt source EIR[RXB2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXB2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
3	Transmit frame interrupt, class 1

Table continues on the next page...

Table continued from the previous page...

Field	Description
TXF1	Corresponds to interrupt source EIR[TXF1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXF1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
2 TXB1	Transmit buffer interrupt, class 1 Corresponds to interrupt source EIR[TXB1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXB1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
1 RXF1	Receive frame interrupt, class 1 Corresponds to interrupt source EIR[RXF1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXF1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
0 RXB1	Receive buffer interrupt, class 1 Corresponds to interrupt source EIR[RXB1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXB1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.

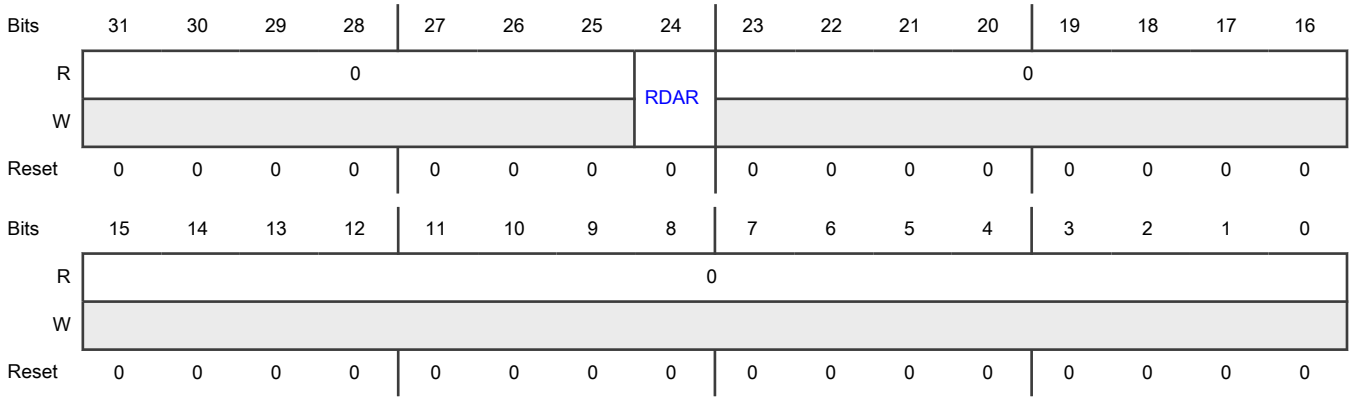
44.5.1.4 Receive Descriptor Active Register - Ring 0 (RDAR)

RDAR is a command register, written by the user, to indicate that the receive descriptor ring 0 has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

Offset

Register	Offset
RDAR	10h

Diagram



Fields

Field	Description
31-25 —	Reserved
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

44.5.1.5 Transmit Descriptor Active Register - Ring 0 (TDAR)

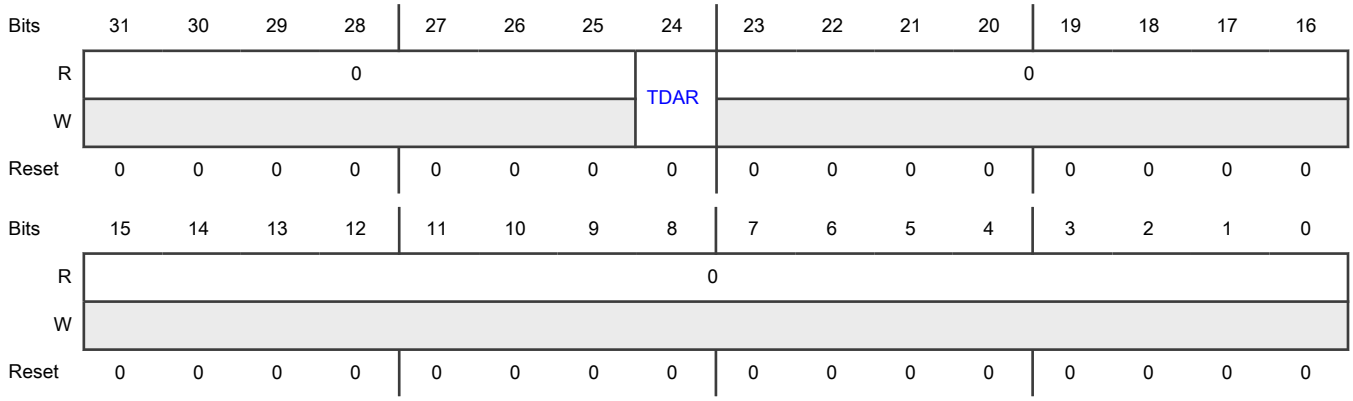
The TDAR is a command register that the user writes to indicate that the transmit descriptor ring 0 has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

The TDAR register is cleared at reset, when ECR[ETHEREN] transitions from set to cleared, or when ECR[RESET] is set.

Offset

Register	Offset
TDAR	14h

Diagram



Fields

Field	Description
31-25 —	Reserved
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

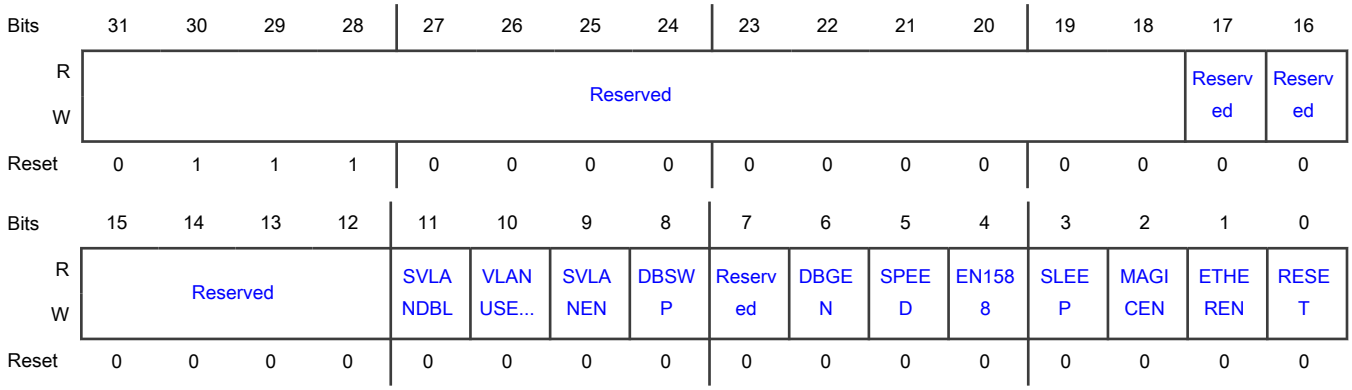
44.5.1.6 Ethernet Control (ECR)

ECR is a read/write user register, though hardware may also alter fields in this register. It controls many of the high level features of the Ethernet MAC, including legacy FEC support through the EN1588 field.

Offset

Register	Offset
ECR	24h

Diagram



Fields

Field	Description
31-18 —	Always write 01110000000000b to this field.
17 —	Always write 0 to this field.
16 —	Always write 0 to this field.
15-12 —	Always write 0 to this field.
11 SVLANDBL	S-VLAN Double Tag If enabled, S-VLAN detection requires a double-tagged frame to define a frame as being a VLAN frame. The following rules apply: • If the first tag is the S-VLAN type, it must be followed by a second tag with the C-VLAN type to declare the frame as VLAN frame. • If the first tag is the S-VLAN type but no 2nd tag follows, it is not considered a VLAN frame but instead treated as an untagged frame. • If the first tag is the C-VLAN type, it is considered a VLAN frame as normal. **NOTE** VLANUSE2ND can be used to determine from which tag the data should be extracted. This applies only if SVLAN_EN = 1, ignored otherwise. 0 - Disable S-VLAN double tag 1 - Enable S-VLAN double tag
10 VLANUSE2ND	VLAN Use Second Tag 0 - Always extract data from the first VLAN tag if it exists.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - When a double-tagged frame is detected, the data of the second tag is extracted for further processing. A double-tagged frame is defined as: The first tag can be a C-VLAN or a S-VLAN (if SVLAN_ENA = 1) The second tag must be a C-VLAN
9 SVLANEN	S-VLAN Enable Enable additional detection of S-VLAN tag according to IEEE802.1Q. 0 - Only the EtherType 0x8100 will be considered for VLAN detection. 1 - The EtherType 0x88a8 will be considered in addition to 0x8100 (C-VLAN) to identify a VLAN frame in receive. When a VLAN frame is identified, the two bytes following the VLAN type are extracted and used by the classification match comparators, RCMRn.
8 DBSWP	Descriptor Byte Swapping Enable Swaps the byte locations of the buffer descriptors. **NOTE** This field must be written to 1 after reset. 0 - The buffer descriptor bytes are not swapped to support big-endian devices. 1 - The buffer descriptor bytes are swapped to support little-endian devices.
7 —	Always write 0 to this field.
6 DBGEN	Debug Enable Enables the MAC to enter hardware freeze mode when the device enters debug mode. 0 - MAC continues operation in debug mode. 1 - MAC enters hardware freeze mode when the processor is in debug mode.
5 SPEED	Selects between 10/100-Mbit/s and 1000-Mbit/s modes of operation. 0 - 10/100-Mbit/s mode 1 - 1000-Mbit/s mode
4 EN1588	EN1588 Enable Enables enhanced functionality of the MAC. 0 - Legacy FEC buffer descriptors and functions enabled. 1 - Enhanced frame time-stamping functions enabled. Has no effect within the MAC besides controlling the DMA control bit ena_1588.
3 SLEEP	Sleep Mode Enable 0 - Normal operating mode. 1 - Sleep mode.
2	Magic Packet Detection Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
MAGICEN	Enables/disables magic packet detection. **NOTE** MAGICEN is relevant only if the SLEEP field is set. If MAGICEN is set, changing the SLEEP field enables/disables sleep mode and magic packet detection. **NOTE** EIMR[WAKEUP] must be written to one if Magic packet wakeup is programed to wake up the chip from low power mode. 0 - Magic detection logic disabled. 1 - The MAC core detects magic packets and asserts EIR[WAKEUP] when a frame is detected.
1 ETHEREN	Ethernet Enable Enables/disables the Ethernet MAC. When the MAC is disabled, the buffer descriptors for an aborted transmit frame are not updated. The uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers. Hardware clears this field under the following conditions: • RESET is set by software • An error condition causes the EBERR field to set. **NOTE** • ETHEREN must be set at the very last step during ENET configuration/setup/ initialization, only *after* all other ENET-related registers have been configured. • If ETHEREN is cleared to 0 by software then next time ETHEREN is set, the EIR interrupts must cleared to 0 due to previous pending interrupts. 0 - Reception immediately stops and transmission stops after a bad CRC is appended to any currently transmitted frame. 1 - MAC is enabled, and reception and transmission are possible.
0 RESET	Ethernet MAC Reset When this field is set, it clears the ETHEREN field.

44.5.1.7 MII Management Frame (MMFR)

Writing to MMFR triggers a management frame transaction to the PHY device unless MSCR is programmed to zero.

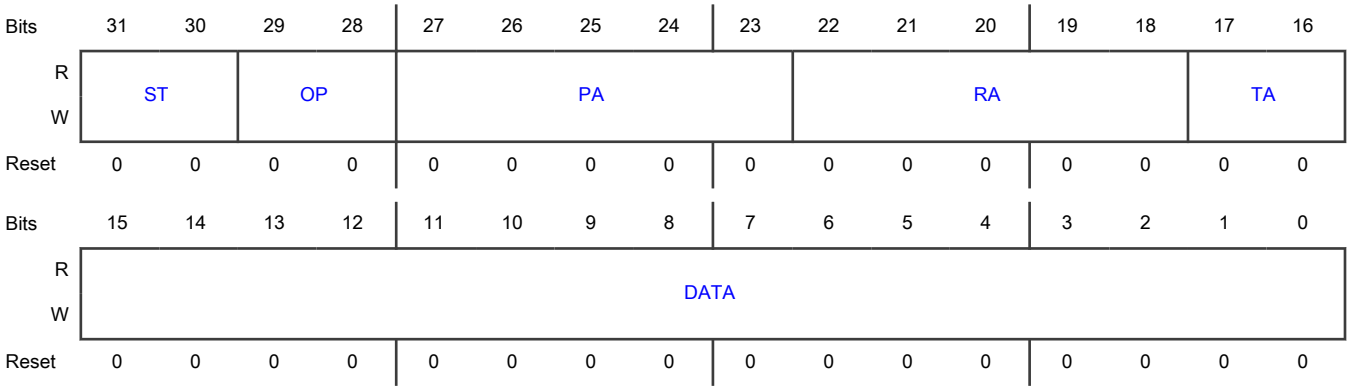
If MSCR is changed from zero to non-zero during a write to MMFR, an MII frame is generated with the data previously written to the MMFR. This allows MMFR and MSCR to be programmed in either order if MSCR is currently zero.

If the MMFR register is written while frame generation is in progress, the frame contents are altered. Software must use the EIR[MII] interrupt indication to avoid writing to the MMFR register while frame generation is in progress.

Offset

Register	Offset
MMFR	40h

Diagram



Fields

Field	Description
31-30 ST	Start Of Frame Delimiter See Table 467 (Clause 22) or Table 469 (Clause 45) for correct value.
29-28 OP	Operation Code See Table 467 (Clause 22) or Table 469 (Clause 45) for correct value.
27-23 PA	PHY Address See Table 467 (Clause 22) or Table 469 (Clause 45) for correct value.
22-18 RA	Register Address See Table 467 (Clause 22) or Table 469 (Clause 45) for correct value.
17-16 TA	Turn Around This field must be programmed to 10 to generate a valid MII management frame.
15-0 DATA	Management Frame Data This is the field for data to be written to or read from the PHY register.

44.5.1.8 MII Speed Control (MSCR)

MSCR provides control of the MII clock (MDC pin) frequency and allows a preamble drop on the MII management frame.

The MII_SPEED field must be programmed with a value to provide an MDC frequency of less than or equal to 2.5 MHz to be compliant with the IEEE 802.3 MII specification. The MII_SPEED must be set to a non-zero value to source a read or write management frame. After the management frame is complete, the MSCR register may optionally be cleared to turn off MDC. The MDC signal generated has a 50% duty cycle except when MII_SPEED changes during operation. This change takes effect following a rising or falling edge of MDC.

For example, if the internal module clock (that is, peripheral bus clock) is 25 MHz, programming MII_SPEED to 0x4 results in an MDC as given in the following equation:

MII clock frequency = 25 MHz / ((4 + 1) x 2) = 2.5 MHz

The following table shows the optimum values for MII_SPEED as a function of IPS bus clock frequency.

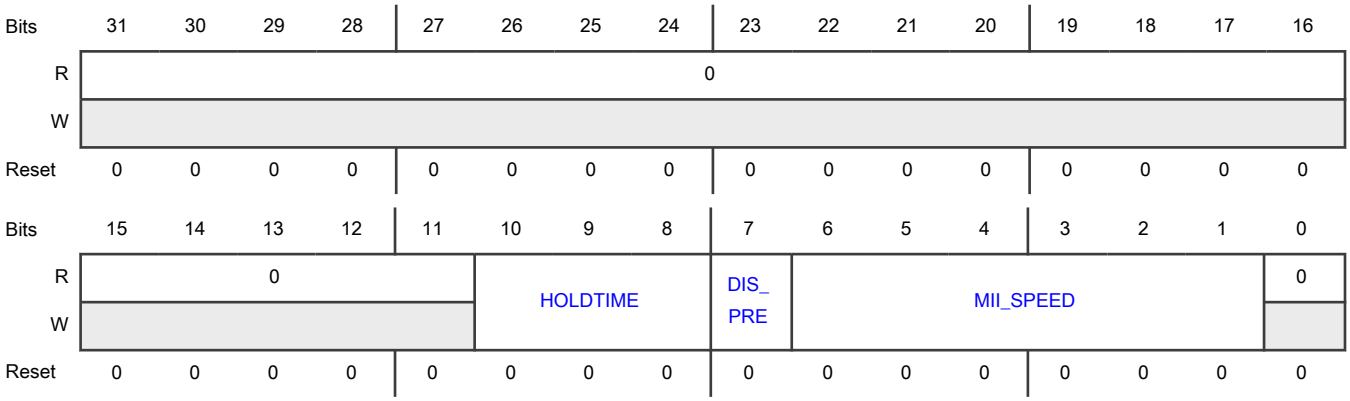
Table 475. Programming Examples for MSCR

Internal module clock frequency	MSCR [MII_SPEED]	MDC frequency
25 MHz	0x4	2.50 MHz
33 MHz	0x6	2.36 MHz
40 MHz	0x7	2.50 MHz
50 MHz	0x9	2.50 MHz
66 MHz	0xD	2.36 MHz

Offset

Register	Offset
MSCR	44h

Diagram



Fields

Field	Description
31-11 —	Reserved
10-8 HOLDTIME	Hold time On MDIO Output IEEE802.3 clause 22 defines a minimum of 10 ns for the hold time on the MDIO output. Depending on the host bus frequency, the setting may need to be increased. 000 - 1 internal module clock cycle 001 - 2 internal module clock cycles

Table continues on the next page...

Table continued from the previous page...

Field	Description
	010 - 3 internal module clock cycles 111 - 8 internal module clock cycles
7 DIS_PRE	Disable Preamble Enables/disables prepending a preamble to the MII management frame. The MII standard allows the preamble to be dropped if the attached PHY devices do not require it. 0 - Preamble enabled. 1 - Preamble (32 ones) is not prepended to the MII management frame.
6-1 MII_SPEED	MII Speed Controls the frequency of the MII management interface clock (MDC) relative to the internal module clock. A value of 0 in this field turns off MDC and leaves it in low voltage state. Any non-zero value results in the MDC frequency of: 1/((MII_SPEED + 1) x 2) of the internal module clock frequency
0 —	Reserved

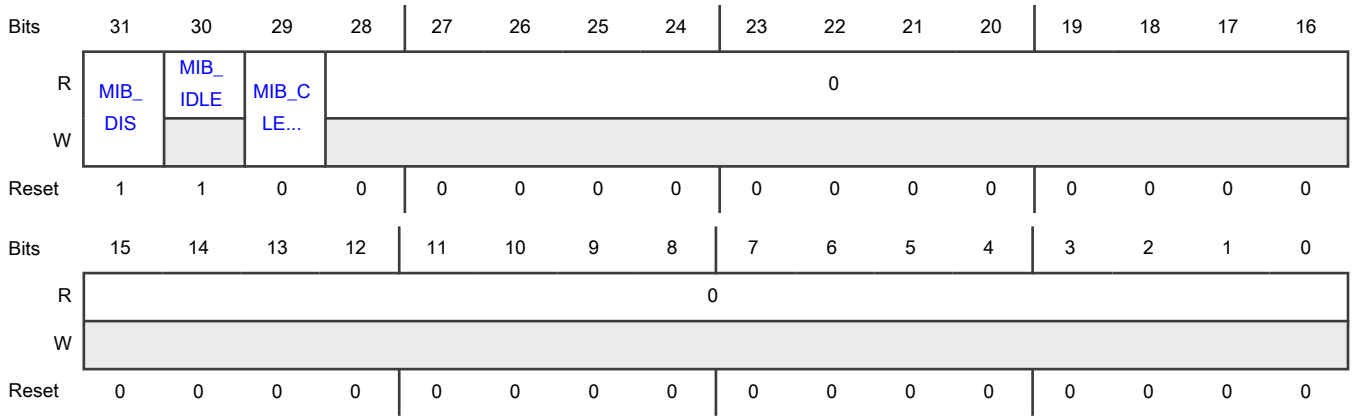
44.5.1.9 MIB Control (MIBC)

MIBC is a read/write register controlling and observing the state of the MIB block. Access this register to disable the MIB block operation or clear the MIB counters. The MIB_DIS field resets to 1.

Offset

Register	Offset
MIBC	64h

Diagram



Fields

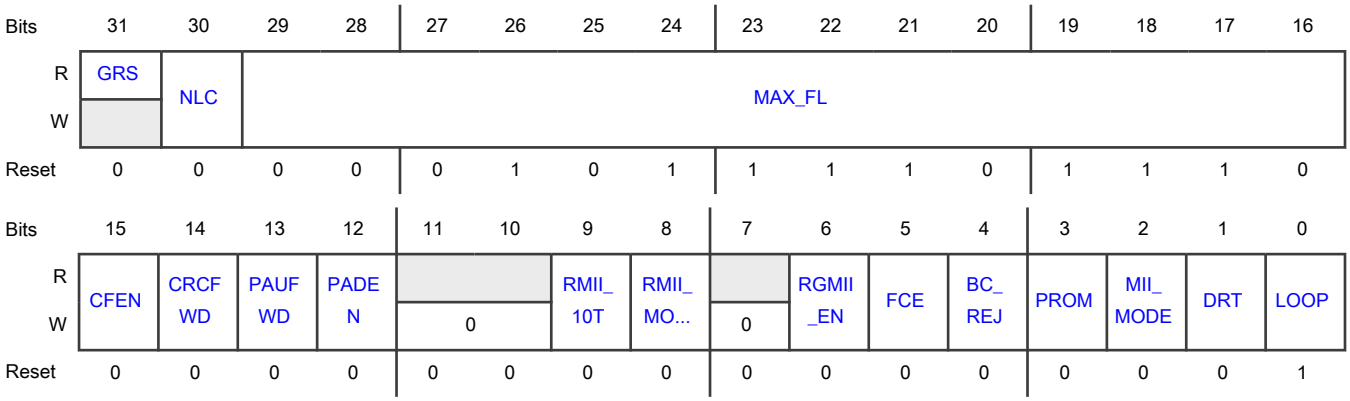
Field	Description
31 MIB_DIS	Disable MIB Logic If this control field is set, 0 - MIB logic is enabled. 1 - MIB logic is disabled. The MIB logic halts and does not update any MIB counters.
30 MIB_IDLE	MIB Idle 0 - The MIB block is updating MIB counters. 1 - The MIB block is not currently updating any MIB counters.
29 MIB_CLEAR	MIB Clear **NOTE** This field is not self-clearing. To clear the MIB counters set and then clear this field. 0 - See note above. 1 - All statistics counters are reset to 0.
28-0 —	Reserved

44.5.1.10 Receive Control (RCR)

Offset

Register	Offset
RCR	84h

Diagram



Fields

Field	Description
31 GRS	Graceful Receive Stopped Read-only status indicating that the MAC receive datapath is stopped. 0 - Receive not stopped 1 - Receive stopped
30 NLC	Payload Length Check Disable Enables/disables a payload length check. 0 - The payload length check is disabled. 1 - The core checks the frame's payload length with the frame length/type field. Errors are indicated in the EIR[PLR] field.
29-16 MAX_FL	Maximum Frame Length Resets to decimal 1518. Length is measured starting at DA and includes the CRC at the end of the frame. Transmit frames longer than MAX_FL cause the BABT interrupt to occur. Receive frames longer than MAX_FL cause the BABR interrupt to occur and set the LG field in the end of frame receive buffer descriptor. The recommended default value to be programmed is 1518 or 1522 if VLAN tags are supported.
15 CFEN	MAC Control Frame Enable Enables/disables the MAC control frame. 0 - MAC control frames with any opcode other than 0x0001 (pause frame) are accepted and forwarded to the client interface. 1 - MAC control frames with any opcode other than 0x0001 (pause frame) are silently discarded.
14 CRCFWD	Terminate/Forward Received CRC Specifies whether the CRC field of received frames is transmitted or stripped. **NOTE** If padding function is enabled (PADEN = 1), CRCFWD is ignored and the CRC field is checked and always terminated and removed. 0 - The CRC field of received frames is transmitted to the user application. 1 - The CRC field is stripped from the frame.
13 PAUFWD	Terminate/Forward Pause Frames Specifies whether pause frames are terminated or forwarded. 0 - Pause frames are terminated and discarded in the MAC. 1 - Pause frames are forwarded to the user application.
12 PADEN	Enable Frame Padding Remove On Receive Specifies whether the MAC removes padding from received frames. 0 - No padding is removed on receive by the MAC.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Padding is removed from received frames.
11-10 —	This write-only field is reserved. It must always be written with the value 0.
9 RMII_10T	Enables 10-Mbit/s mode of the RMII or RGMII . 0 - 100-Mbit/s or 1-Gbit/s operation. 1 - 10-Mbit/s operation.
8 RMII_MODE	RMII Mode Enable Specifies whether the MAC is configured for MII mode or RMII operation , when ECR[SPEED] is cleared . **NOTE** Do not set both RCR[RGMII_EN] and RCR[RMII_MODE]. 0 - MAC configured for MII mode. 1 - MAC configured for RMII operation.
7 —	This write-only field is reserved. It must always be written with the value 0.
6 RGMII_EN	RGMII Mode Enable **NOTE** Do not set both RCR[RGMII_EN] and RCR[RMII_MODE]. 0 - MAC configured for non-RGMII operation 1 - MAC configured for RGMII operation. ECR[SPEED] should be configured before enable RGMII. If ECR[SPEED] is set, the MAC is in RGMII 1000-Mbit/s mode. If ECR[SPEED] is cleared, the MAC is in RGMII 10/100-Mbit/s mode.
5 FCE	Flow Control Enable If set, the receiver detects PAUSE frames. Upon PAUSE frame detection, the transmitter stops transmitting data frames for a given duration. 0 - Disable flow control 1 - Enable flow control
4 BC_REJ	Broadcast Frame Reject If set, frames with destination address (DA) equal to 0xFFFF_FFFF_FFFF are rejected unless the PROM field is set. If BC_REJ and PROM are set, frames with broadcast DA are accepted and the MISS (M) is set in the receive buffer descriptor. 0 - Will not reject frames as described above 1 - Will reject frames as described above

Table continues on the next page...

Table continued from the previous page...

Field	Description
3 PROM	Promiscuous Mode All frames are accepted regardless of address matching. 0 - Disabled. 1 - Enabled.
2 MII_MODE	Media Independent Interface Mode This field must always be set. 0 - Reserved. 1 - MII or RMII mode, as indicated by the RMII_MODE field.
1 DRT	Disable Receive On Transmit 0 - Receive path operates independently of transmit (i.e., full-duplex mode). Can also be used to monitor transmit activity in half-duplex mode. 1 - Disable reception of frames while transmitting. (Normally used for half-duplex mode.)
0 LOOP	Internal Loopback This is an MII internal loopback, therefore MII_MODE must be written to 1 and RMII_MODE must be written to 0. 0 - Loopback disabled. 1 - Transmitted frames are looped back internal to the device and transmit MII output signals are not asserted. DRT must be cleared.

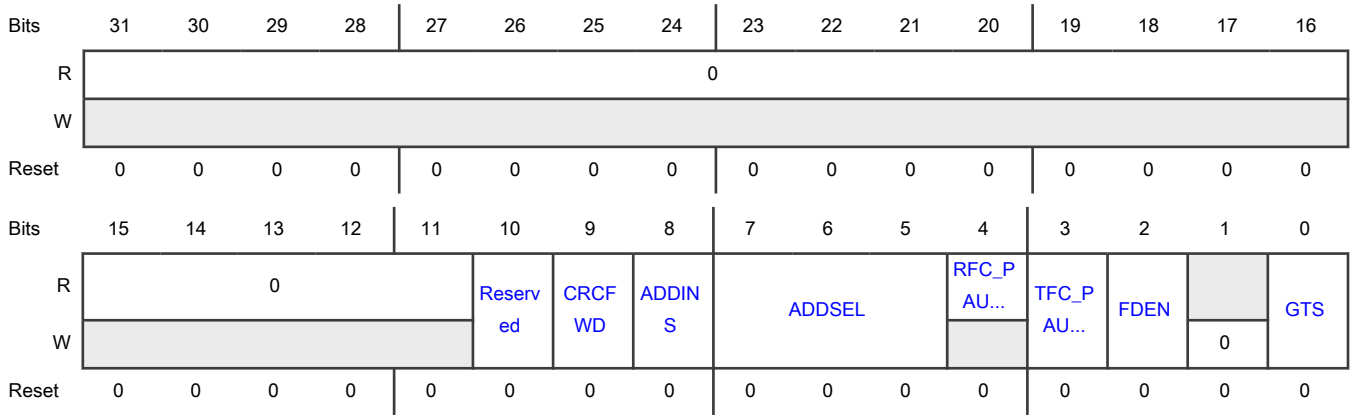
44.5.1.11 Transmit Control (TCR)

TCR is read/write and configures the transmit block. This register is cleared at system reset. FDEN can only be modified when ECR[ETHEREN] is cleared.

Offset

Register	Offset
TCR	C4h

Diagram



Fields

Field	Description
31-11 —	Reserved
10 —	This field is read/write and must be set to 0.
9 CRCFWD	Forward Frame From Application With CRC 0 - TxBD[TC] controls whether the frame has a CRC from the application. 1 - The transmitter does not append any CRC to transmitted frames, as it is expecting a frame with CRC from the application.
8 ADDINS	Set MAC Address On Transmit 0 - The source MAC address is not modified by the MAC. 1 - The MAC overwrites the source MAC address with the programmed MAC address according to ADDSEL.
7-5 ADDSEL	Source MAC Address Select On Transmit If ADDINS is set, indicates the MAC address that overwrites the source MAC address. 000 - Node MAC address programmed on PADDR1/2 registers. 100 - Reserved. 101 - Reserved. 110 - Reserved.
4 RFC_PAUSE	Receive Frame Control Pause This status field is set when a full-duplex flow control pause frame is received and the transmitter pauses for the duration defined in this pause frame. This field automatically clears when the pause duration is complete.
3	Transmit Frame Control Pause

Table continues on the next page...

Table continued from the previous page...

Field	Description
TFC_PAUSE	Pauses frame transmission. When this field is set, EIR[GRA] is set. With transmission of data frames stopped, the MAC transmits a MAC control PAUSE frame. Next, the MAC clears TFC_PAUSE and resumes transmitting data frames. If the transmitter pauses due to user assertion of GTS or reception of a PAUSE frame, the MAC may continue transmitting a MAC control PAUSE frame. 0 - No PAUSE frame transmitted. 1 - The MAC stops transmission of data frames after the current transmission is complete.
2 FDEN	Full-Duplex Enable If this field is set, frames transmit independent of carrier sense and collision inputs. Only modify this bit when ECR[ETHEREN] is cleared. 0 - Disable full-duplex 1 - Enable full-duplex
1 —	This write-only field is reserved. It must always be written with the value 0.
0 GTS	Graceful Transmit Stop When this field is set, MAC stops transmission after any frame currently transmitted is complete and EIR[GRA] is set. If frame transmission is not currently underway, the GRA interrupt is asserted immediately. After transmission finishes, clear GTS to restart. The next frame in the transmit FIFO is then transmitted. If an early collision occurs during transmission when GTS is set, transmission stops after the collision. The frame is transmitted again after GTS is cleared. There may be old frames in the transmit FIFO that transmit when GTS is reasserted. To avoid this, clear ECR[ETHEREN] following the GRA interrupt. 0 - Disable graceful transmit stop 1 - Enable graceful transmit stop

44.5.1.12 Physical Address Lower (PALR)

PALR contains the lower 32 bits (bytes 0, 1, 2, 3) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 0 through 3 of the six-byte source address field when transmitting PAUSE frames.

Offset

Register	Offset
PALR	E4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PADDR1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PADDR1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 PADDR1	Pause Address Bytes 0 (bits 31:24), 1 (bits 23:16), 2 (bits 15:8), and 3 (bits 7:0) of the 6-byte individual address are used for exact match and the source address field in PAUSE frames.

44.5.1.13 Physical Address Upper (PAUR)

PAUR contains the upper 16 bits (bytes 4 and 5) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 4 and 5 of the six-byte source address field when transmitting PAUSE frames. Bits 15:0 of PAUR contain a constant type field (0x8808) for transmission of PAUSE frames.

Offset

Register	Offset
PAUR	E8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PADDR2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TYPE															
W																
Reset	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

Fields

Field	Description
31-16 PADDR2	Bytes 4 (bits 31:24) and 5 (bits 23:16) of the 6-byte individual address used for exact match, and the source address field in PAUSE frames.
15-0 TYPE	Type Field In PAUSE Frames These fields have a constant value of 0x8808.

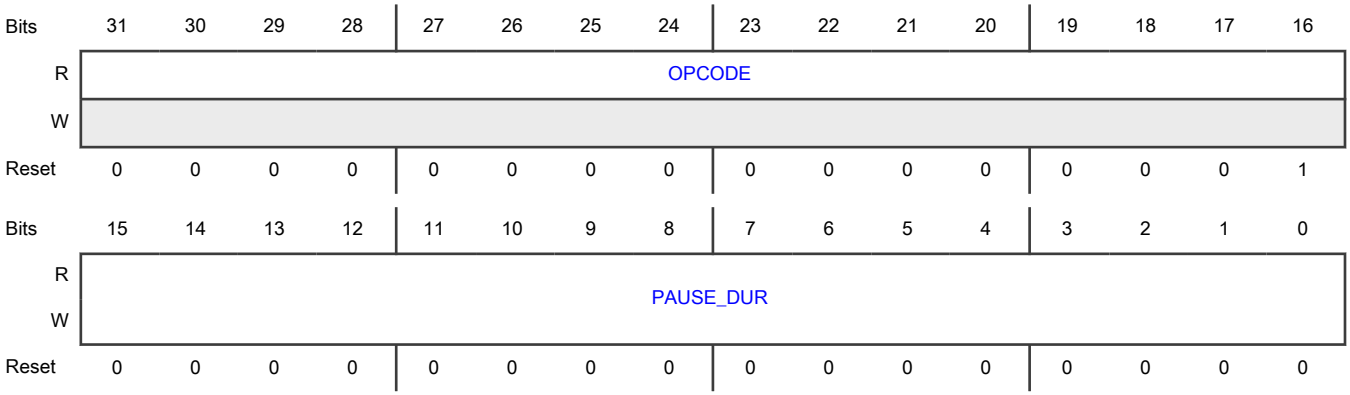
44.5.1.14 Opcode/Pause Duration (OPD)

OPD is read/write accessible. This register contains the 16-bit opcode and 16-bit pause duration fields used in transmission of a PAUSE frame. The opcode field is a constant value, 0x0001. When another node detects a PAUSE frame, that node pauses transmission for the duration specified in the pause duration field. The lower 16 bits of this register are not reset and you must initialize it.

Offset

Register	Offset
OPD	ECh

Diagram



Fields

Field	Description
31-16 OPCODE	Opcode Field In PAUSE Frames These fields have a constant value of 0x0001.
15-0 PAUSE_DUR	Pause Duration Pause duration field used in PAUSE frames.

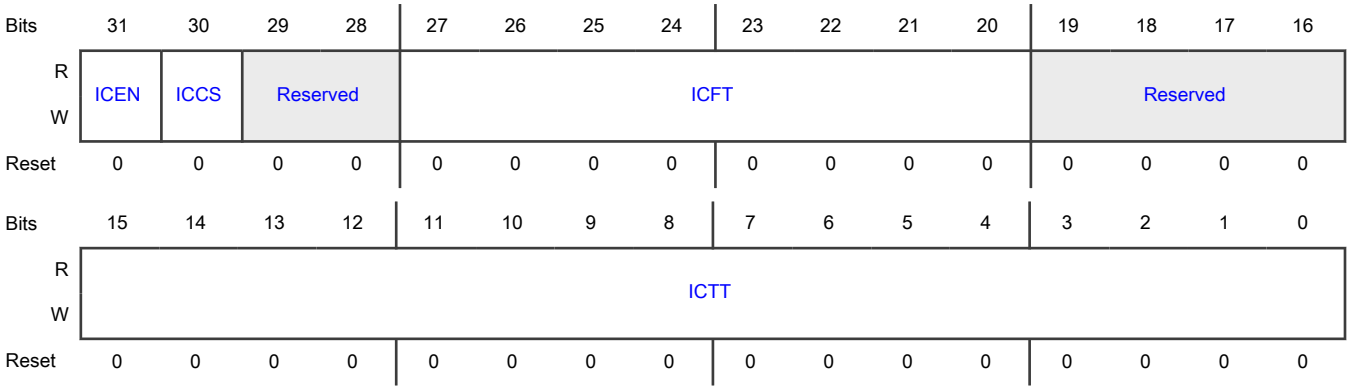
44.5.1.15 Transmit Interrupt Coalescing (TXIC0 - TXIC2)

See Interrupts for more information.

Offset

Register	Offset
TXIC0	F0h
TXIC1	F4h
TXIC2	F8h

Diagram



Fields

Field	Description
31 ICEN	Interrupt Coalescing Enable 0 - Disable Interrupt coalescing. 1 - Enable Interrupt coalescing.
30 ICCS	Interrupt Coalescing Timer Clock Source Select 0 - Use MII/GMII TX clocks. 1 - Use ENET system clock.
29-28 —	This field must be set to 0.
27-20 ICFT	Interrupt coalescing frame count threshold This value determines the number of frames needed to be transmitted for raising an interrupt. Frame counter restarts after reaching this threshold value or after the expiring of the coalescing timer. Must be greater than zero to avoid unpredictable behavior.
19-16 —	This field must be set to 0.
15-0 ICTT	Interrupt coalescing timer threshold

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Interrupt coalescing timer threshold in units of 64 TX clock periods. This value determines the maximum amount of time after transmitting a frame before raising an interrupt. The threshold timer is disabled after expiring or number of frame transmission defined by ICFT and starts again upon transmission of the next first frame. Must be greater than zero to avoid unpredictable behavior.

44.5.1.16 Receive Interrupt Coalescing (RXIC0 - RXIC2)

See Interrupts for more information.

Offset

Register	Offset
RXIC0	100h
RXIC1	104h
RXIC2	108h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ICEN	ICCS	Reserved		ICFT								Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ICTT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 ICEN	Interrupt Coalescing Enable 0 - Disable Interrupt coalescing. 1 - Enable Interrupt coalescing.
30 ICCS	Interrupt Coalescing Timer Clock Source Select 0 - Use MII/GMII TX clocks. 1 - Use ENET system clock.
29-28	This field must be set to 0.

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
27-20 ICFT	Interrupt coalescing frame count threshold This value determines the number of frames needed to be received for raising an interrupt. Frame counter restarts after reaching this threshold value or after the expiring of the coalescing timer. Must be greater than zero to avoid unpredictable behavior.
19-16 —	This field must be set to 0.
15-0 ICTT	Interrupt coalescing timer threshold Interrupt coalescing timer threshold in units of 64 RX clock periods. This value determines the maximum amount of time after receiving a frame before raising an interrupt. The threshold timer is disabled after expiring or number of frame reception defined by ICFT and starts again upon reception of the next first frame. Must be greater than zero to avoid unpredictable behavior.

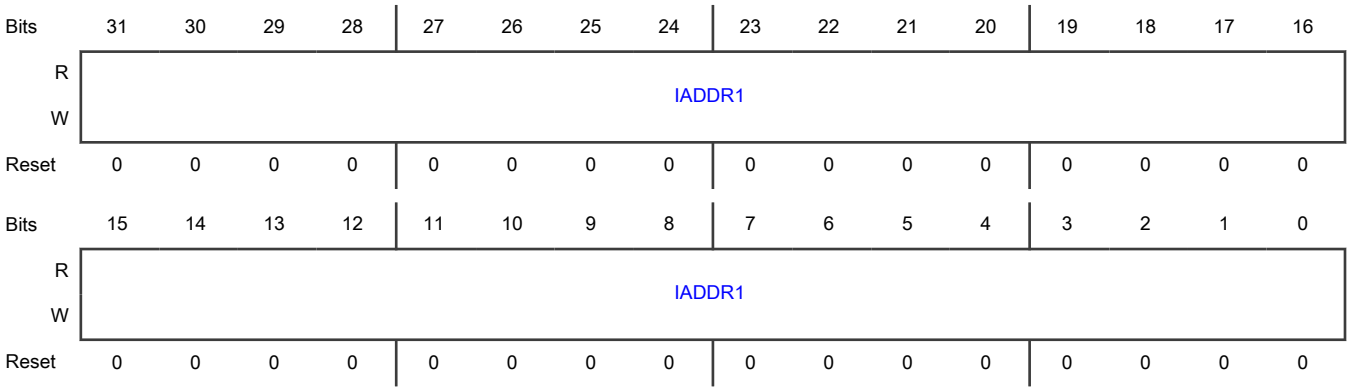
44.5.1.17 Descriptor Individual Upper Address (IAUR)

IAUR contains the upper 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the destination address (DA) field of receive frames with an individual DA. This register is not reset and you must initialize it.

Offset

Register	Offset
IAUR	118h

Diagram



Fields

Field	Description
31-0 IADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR1 contains hash index bit 63. Bit 0 of IADDR1 contains hash index bit 32.

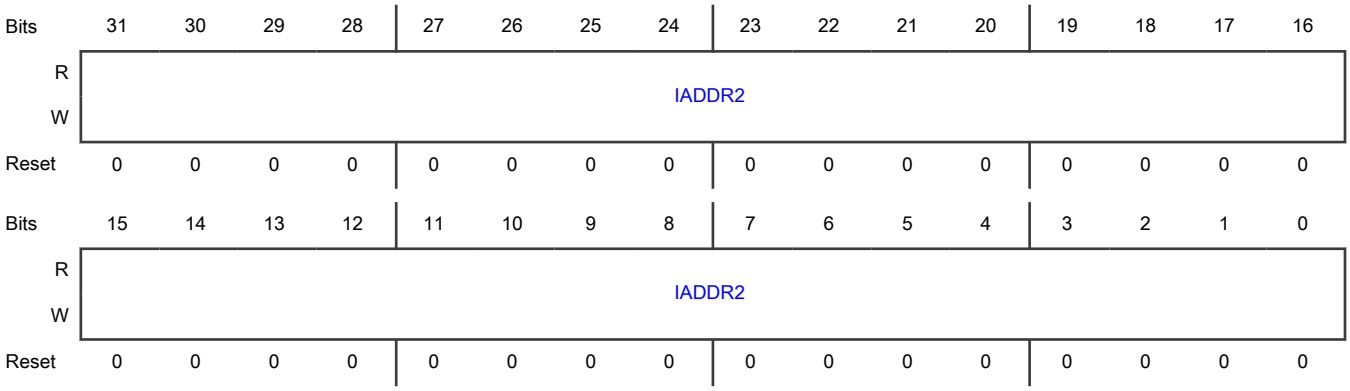
44.5.1.18 Descriptor Individual Lower Address (IALR)

IALR contains the lower 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the DA field of receive frames with an individual DA. This register is not reset and you must initialize it.

Offset

Register	Offset
IALR	11Ch

Diagram



Fields

Field	Description
31-0 IADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR2 contains hash index bit 31. Bit 0 of IADDR2 contains hash index bit 0.

44.5.1.19 Descriptor Group Upper Address (GAUR)

GAUR contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

Offset

Register	Offset
GAUR	120h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR1 contains hash index bit 63. Bit 0 of GADDR1 contains hash index bit 32.

44.5.1.20 Descriptor Group Lower Address (GALR)

GALR contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

Offset

Register	Offset
GALR	124h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 GADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR2 contains hash index bit 31. Bit 0 of GADDR2 contains hash index bit 0.

44.5.1.21 Transmit FIFO Watermark (TFWR)

If TFWR[STRFWD] is cleared, TFWR[TFWR] controls the amount of data required in the transmit FIFO before transmission of a frame can begin. This allows you to minimize transmit latency (TFWR = 00 or 01) or allow for larger bus access latency (TFWR = 11) due to contention for the system bus. Setting the watermark to a high value minimizes the risk of transmit FIFO underrun due to contention for the system bus. The byte counts associated with the TFWR field may need to be modified to match a given system requirement, for example, worst-case bus access latency by the transmit data uDMA channel.

When the FIFO level reaches the value the TFWR field and when the STR_FWD is set to '0', the MAC transmit control logic starts frame transmission even before the end-of-frame is available in the FIFO (cut-through operation).

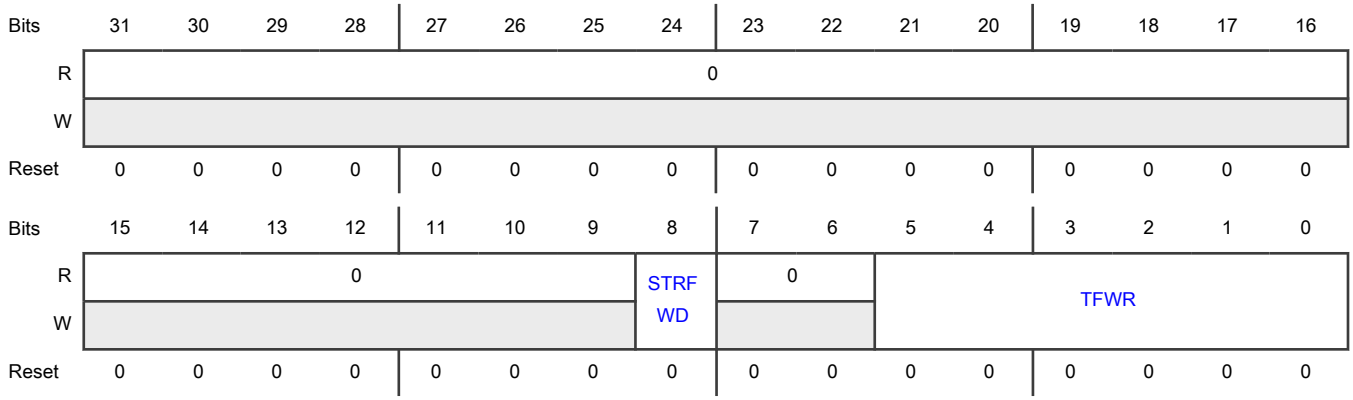
If a complete frame has a size smaller than the threshold programmed with TFWR, the MAC also transmits the Frame to the line.

To enable store and forward on the Transmit path, set STR_FWD to '1'. In this case, the MAC starts to transmit data only when a complete frame is stored in the Transmit FIFO.

Offset

Register	Offset
TFWR	144h

Diagram



Fields

Field	Description
31-9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 STRFWD	Store And Forward Enable 0 - Reset. The transmission start threshold is programmed in TFWR[TFWR]. 1 - Enabled.
7-6 —	Reserved
5-0 TFWR	Transmit FIFO Write If TFWR[STRFWD] is cleared, this field indicates the number of bytes, in steps of 64 bytes, written to the transmit FIFO before transmission of a frame begins. **NOTE** If a frame with less than the threshold is written, it is still sent independently of this threshold setting. The threshold is relevant only if the frame is larger than the threshold given. 00_0000 - 64 bytes written. 00_0001 - 64 bytes written. 00_0010 - 128 bytes written. 00_0011 - 192 bytes written. 11_1111 - 4032 bytes written.

44.5.1.22 Receive Descriptor Ring 1 Start (RDSR1)

RDSR1 points to the beginning of circular receive buffer descriptor queue 1 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

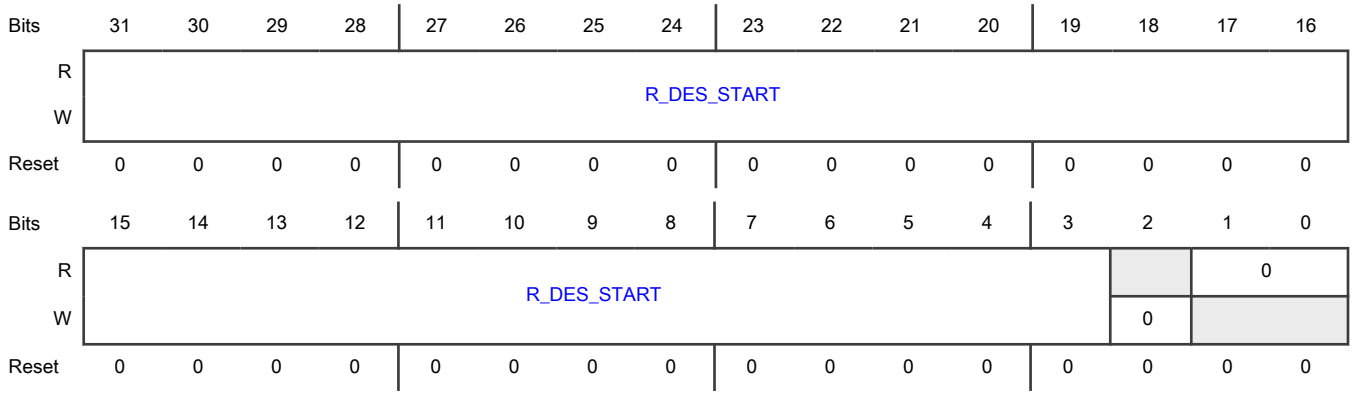
NOTE

This register must be initialized prior to operation.

Offset

Register	Offset
RDSR1	160h

Diagram



Fields

Field	Description
31-3 R_DES_START	Pointer to the beginning of the receive buffer descriptor queue 1.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

44.5.1.23 Transmit Buffer Descriptor Ring 1 Start (TDSR1)

TDSR1 provides a pointer to the beginning of the circular transmit buffer descriptor queue 1 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

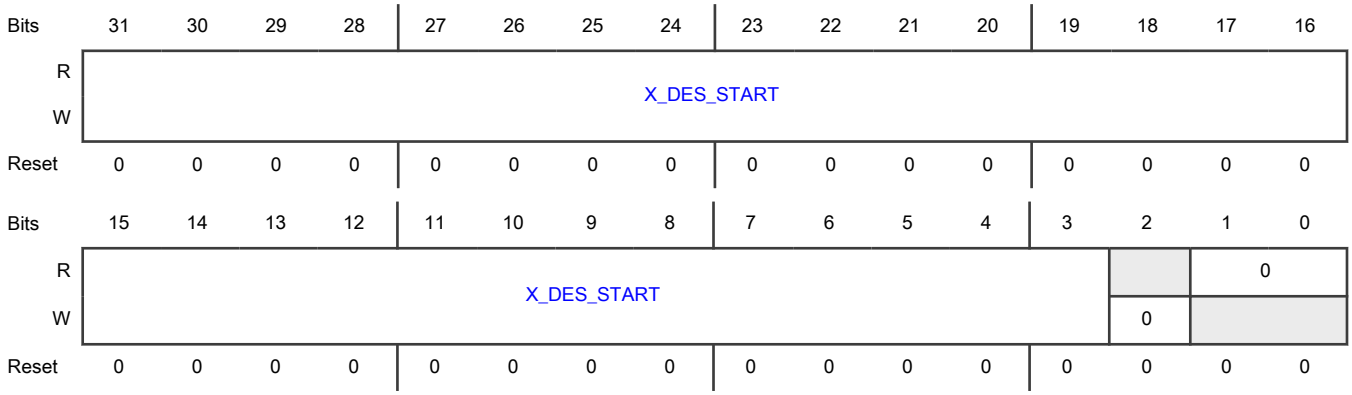
NOTE

This register must be initialized prior to operation.

Offset

Register	Offset
TDSR1	164h

Diagram



Fields

Field	Description
31-3 X_DES_START	Pointer to the beginning of transmit buffer descriptor queue 1.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

44.5.1.24 Maximum Receive Buffer Size Register - Ring 1 (MRBR1)

MRBR1 is a user-programmable register that dictates the maximum size of all ring-1 receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- To allow one maximum size frame per buffer, MRBR1 must be set to RCR[MAX_FL] or larger.
- R_BUF_SIZE is concatenated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To properly align the buffer, MRBR1 must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_ZIZE to zero. The lower four bits are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR1 greater than or equal to 256 bytes.

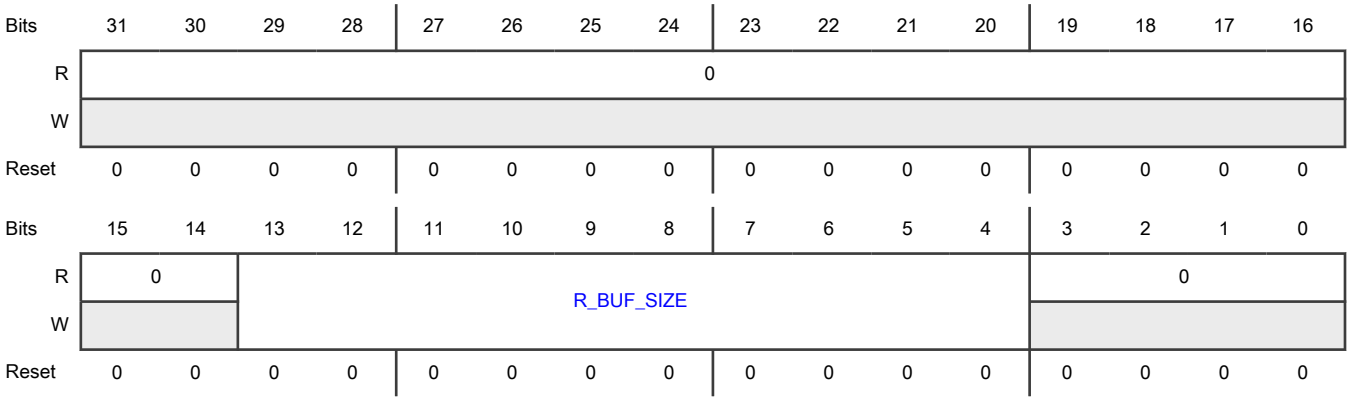
NOTE

This register must be initialized before operation.

Offset

Register	Offset
MRBR1	168h

Diagram



Fields

Field	Description
31-14 —	Reserved
13-4 R_BUF_SIZE	Receive buffer size (in bytes)
3-0 —	Reserved

44.5.1.25 Receive Descriptor Ring 2 Start (RDSR2)

RDSR points to the beginning of circular receive buffer descriptor queue 2 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

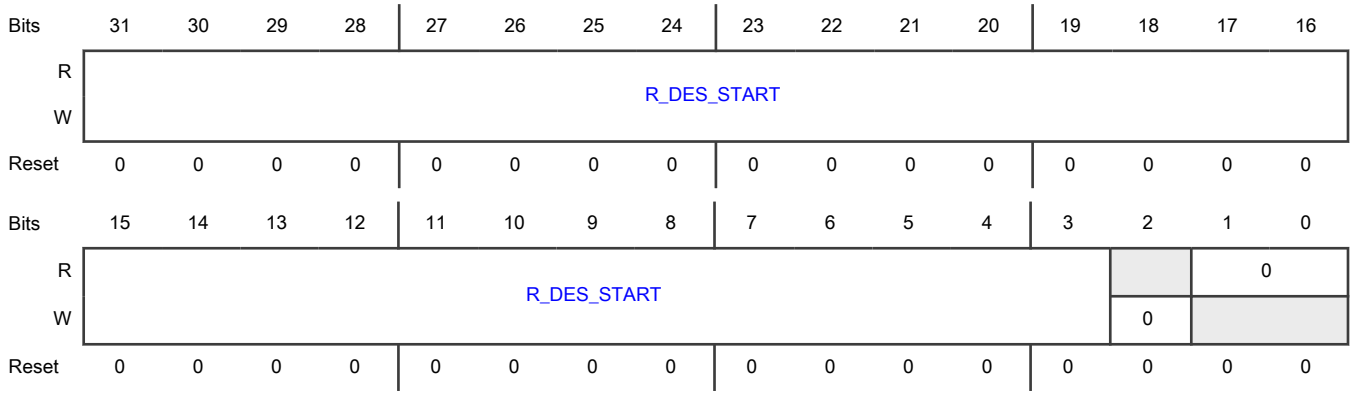
NOTE

This register must be initialized prior to operation

Offset

Register	Offset
RDSR2	16Ch

Diagram



Fields

Field	Description
31-3 R_DES_START	Pointer to the beginning of receive buffer descriptor queue 2.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

44.5.1.26 Transmit Buffer Descriptor Ring 2 Start (TDSR2)

TDSR2 provides a pointer to the beginning of circular transmit buffer descriptor queue 2 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

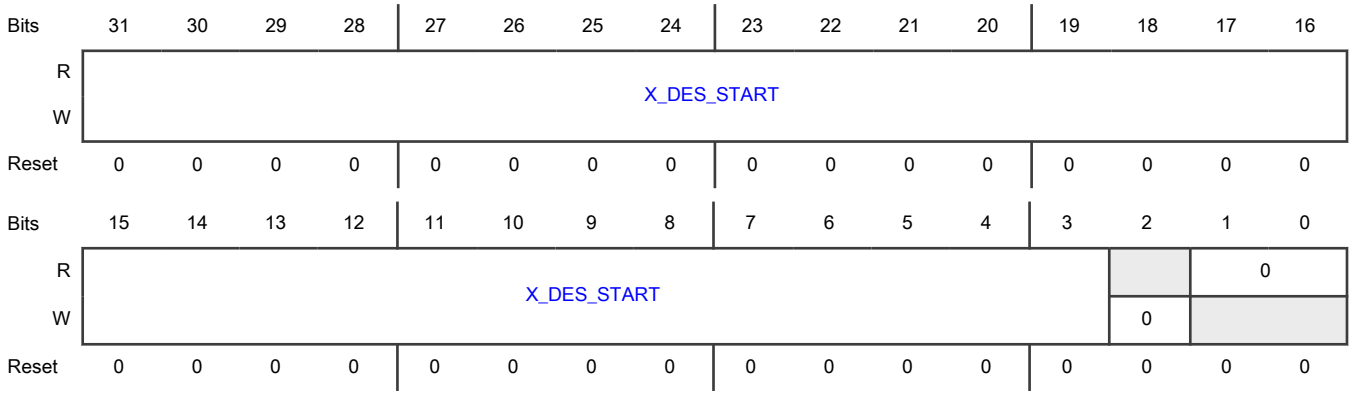
NOTE

This register must be initialized prior to operation

Offset

Register	Offset
TDSR2	170h

Diagram



Fields

Field	Description
31-3 X_DES_START	Pointer to the beginning of transmit buffer descriptor queue 2.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

44.5.1.27 Maximum Receive Buffer Size Register - Ring 2 (MRBR2)

MRBR2 is a user-programmable register that dictates the maximum size of all ring-2 receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- To allow one maximum size frame per buffer, MRBR2 must be set to RCR[MAX_FL] or larger.
- R_BUF_SIZE is concatenated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To properly align the buffer, MRBR2 must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_SIZE to zero. The lower four bits are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR2 greater than or equal to 256 bytes.

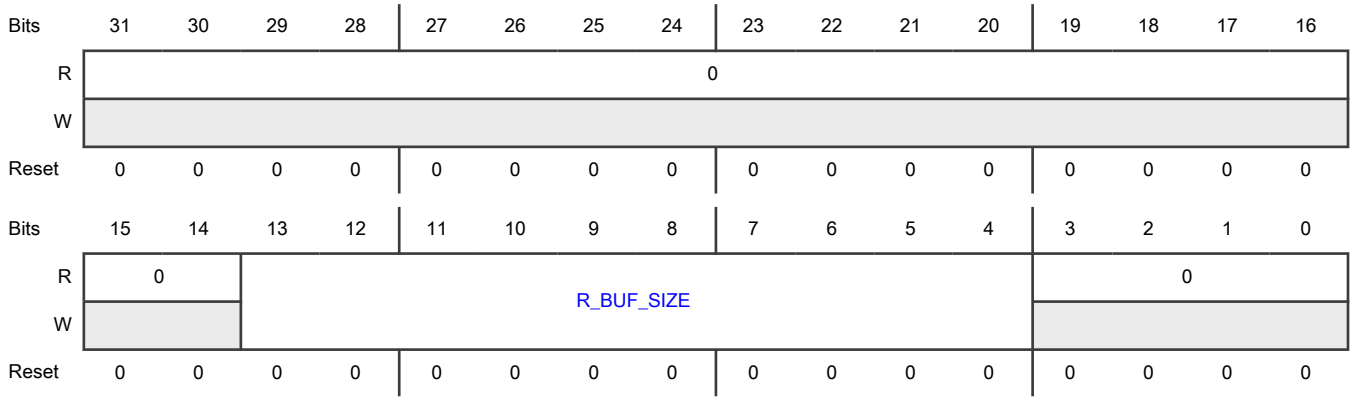
NOTE

This register must be initialized prior to operation

Offset

Register	Offset
MRBR2	174h

Diagram



Fields

Field	Description
31-14 —	Reserved
13-4 R_BUF_SIZE	Receive buffer size (in bytes)
3-0 —	Reserved

44.5.1.28 Receive Descriptor Ring 0 Start (RDSR)

RDSR points to the beginning of the circular receive buffer descriptor queue in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

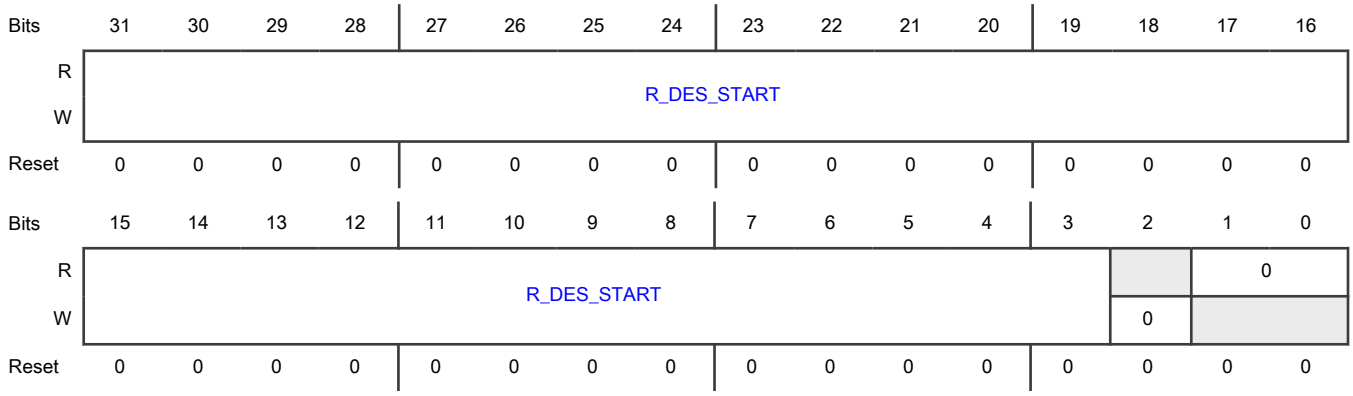
NOTE

This register must be initialized prior to operation

Offset

Register	Offset
RDSR	180h

Diagram



Fields

Field	Description
31-3 R_DES_START	Pointer to the beginning of the receive buffer descriptor queue. 0
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

44.5.1.29 Transmit Buffer Descriptor Ring 0 Start (TDSR)

TDSR provides a pointer to the beginning of the circular transmit buffer descriptor queue 0 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

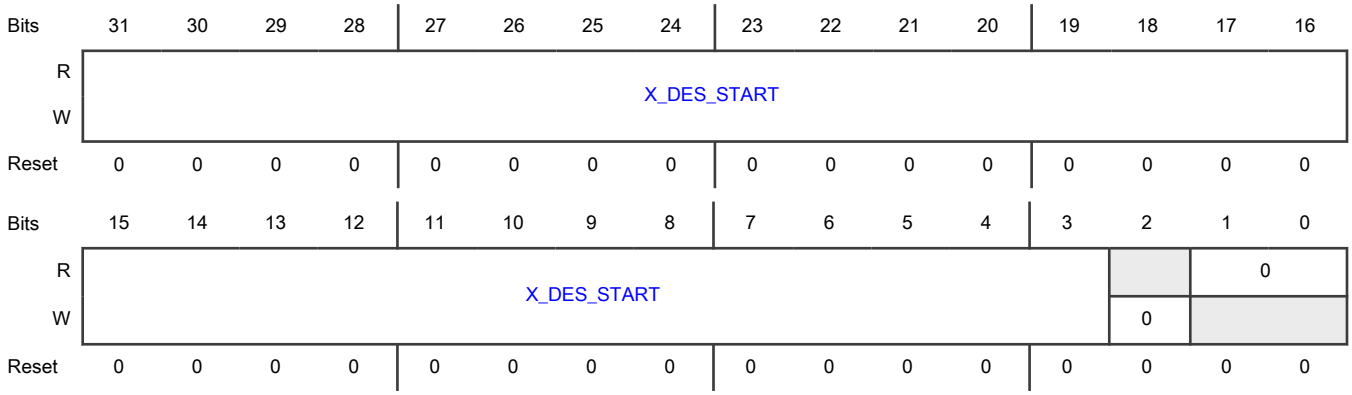
NOTE

This register must be initialized prior to operation.

Offset

Register	Offset
TDSR	184h

Diagram



Fields

Field	Description
31-3 X_DES_START	Pointer to the beginning of the transmit buffer descriptor queue.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

44.5.1.30 Maximum Receive Buffer Size Register - Ring 0 (MRBR)

The MRBR is a user-programmable register that dictates the maximum size of all ring-0 receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- R_BUF_SIZE is concatenated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To allow one maximum size frame per buffer, MRBR must be set to RCR[MAX_FL] or larger.
- To properly align the buffer, MRBR must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_SIZE to zero. The lower four bits of this register are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR greater than or equal to 256 bytes.

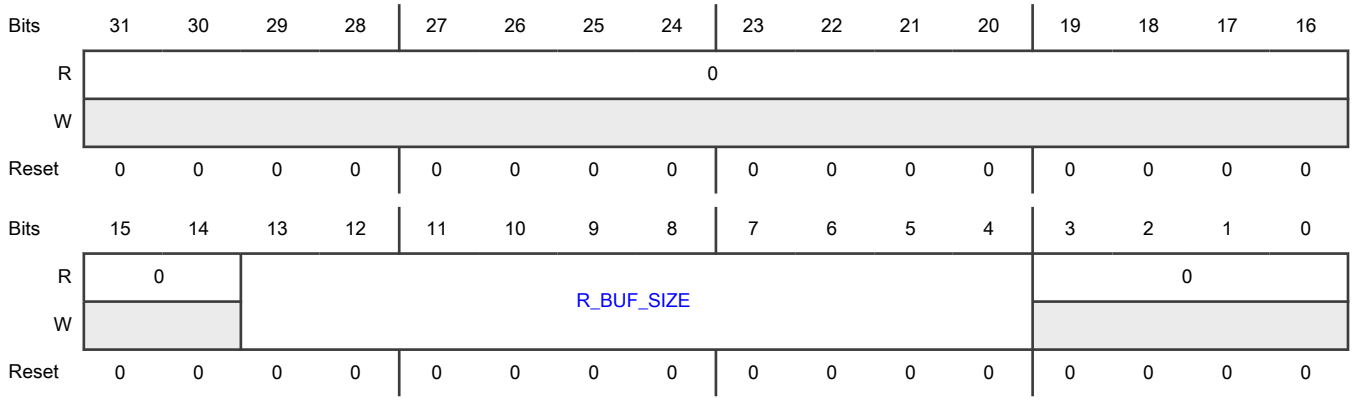
NOTE

This register must be initialized before operation.

Offset

Register	Offset
MRBR	188h

Diagram



Fields

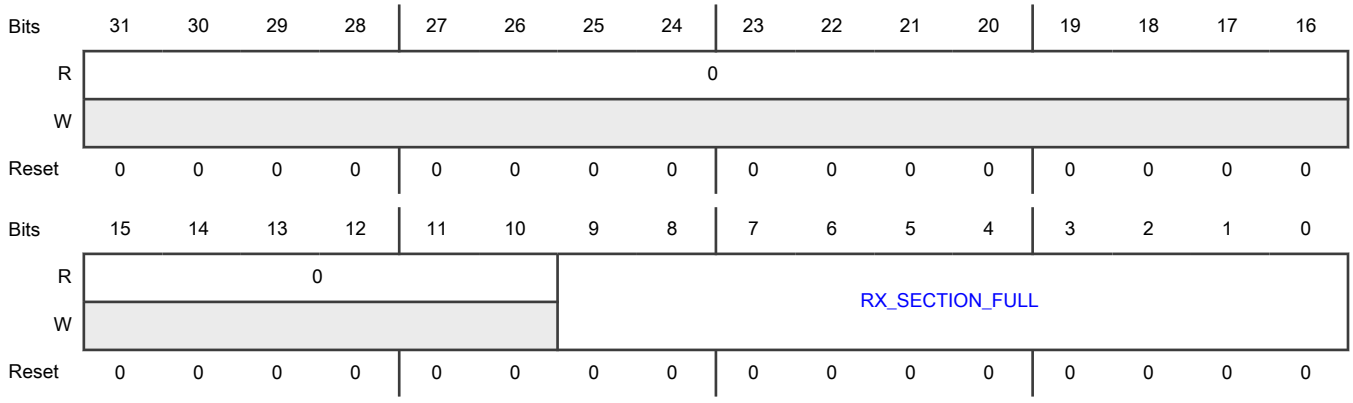
Field	Description
31-14 —	Reserved
13-4 R_BUF_SIZE	Receive buffer size in bytes. This value, concatenated with the four least-significant bits of this register (which are always zero), is the effective maximum receive buffer size.
3-0 —	This field, which is always zero, is the four least-significant bits of the maximum receive buffer size.

44.5.1.31 Receive FIFO Section Full Threshold (RSFL)

Offset

Register	Offset
RSFL	190h

Diagram



Fields

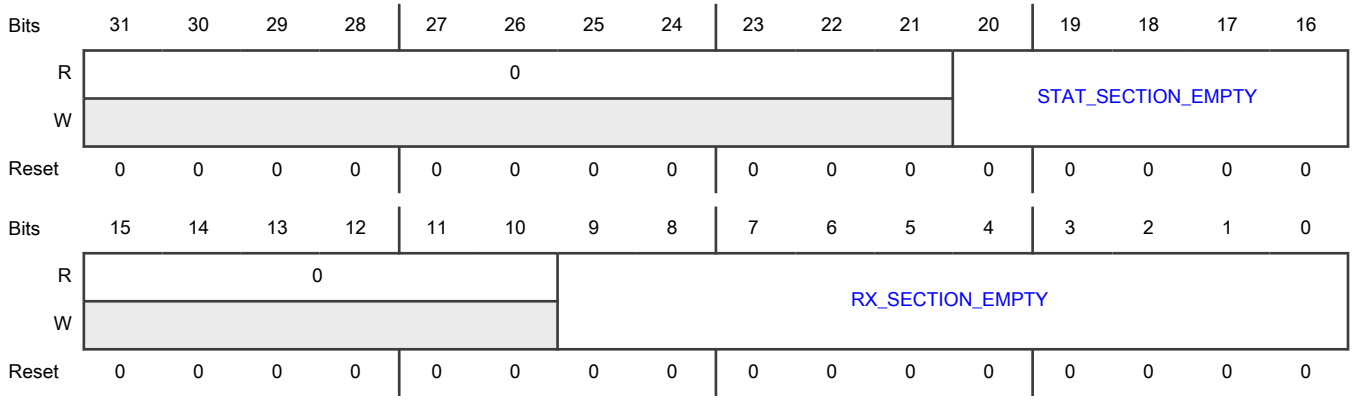
Field	Description
31-10 —	Reserved
9-0 RX_SECTION_FULL	Value Of Receive FIFO Section Full Threshold Value, in 64-bit words, of the receive FIFO section full threshold. Clear this field to enable store and forward on the RX FIFO. When programming a value greater than 0 (cut-through operation), it must be greater than RAEM[RX_ALMOST_EMPTY]. When the FIFO level reaches the value in this field, data is available in the Receive FIFO (cut-through operation).

44.5.1.32 Receive FIFO Section Empty Threshold (RSEM)

Offset

Register	Offset
RSEM	194h

Diagram



Fields

Field	Description
31-21 —	Reserved
20-16 STAT_SECTION_EMPTY	RX Status FIFO Section Empty Threshold Defines number of frames in the receive FIFO, independent of its size, that can be accepted. If the limit is reached, reception will continue normally, however a pause frame will be triggered to indicate a

Table continues on the next page...

Table continued from the previous page...

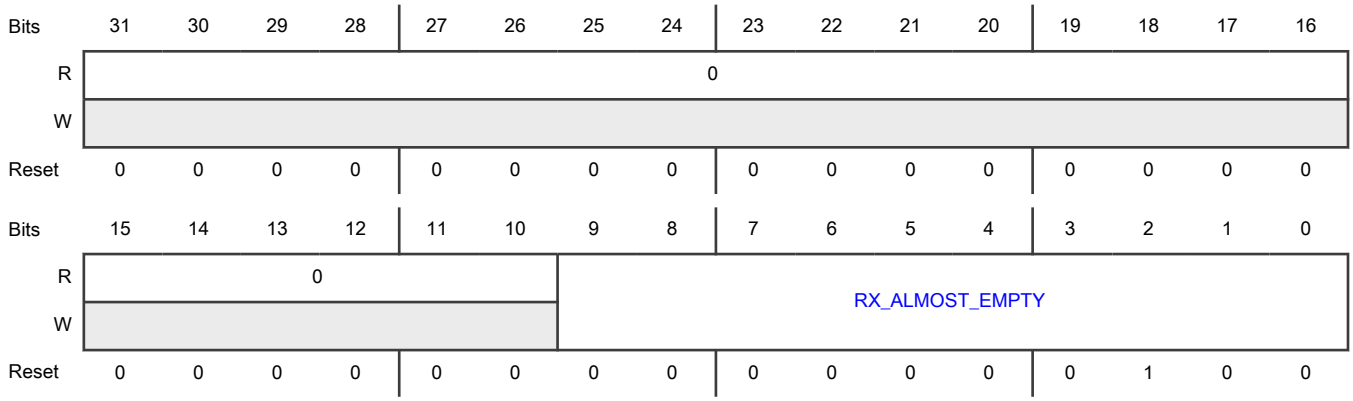
Field	Description
	possible congestion to the remote device to avoid FIFO overflow. A value of 0 disables automatic pause frame generation
15-10 —	Reserved
9-0 RX_SECTION_EMPTY	Value Of The Receive FIFO Section Empty Threshold Value, in 64-bit words, of the receive FIFO section empty threshold. When the FIFO has reached this level, a pause frame will be issued. A value of 0 disables automatic pause frame generation. When the FIFO level goes below the value programmed in this field, an XON pause frame is issued to indicate the FIFO congestion is cleared to the remote Ethernet client. **NOTE** The section-empty threshold indications from both FIFOs are OR'ed to cause XOFF pause frame generation.

44.5.1.33 Receive FIFO Almost Empty Threshold (RAEM)

Offset

Register	Offset
RAEM	198h

Diagram



Fields

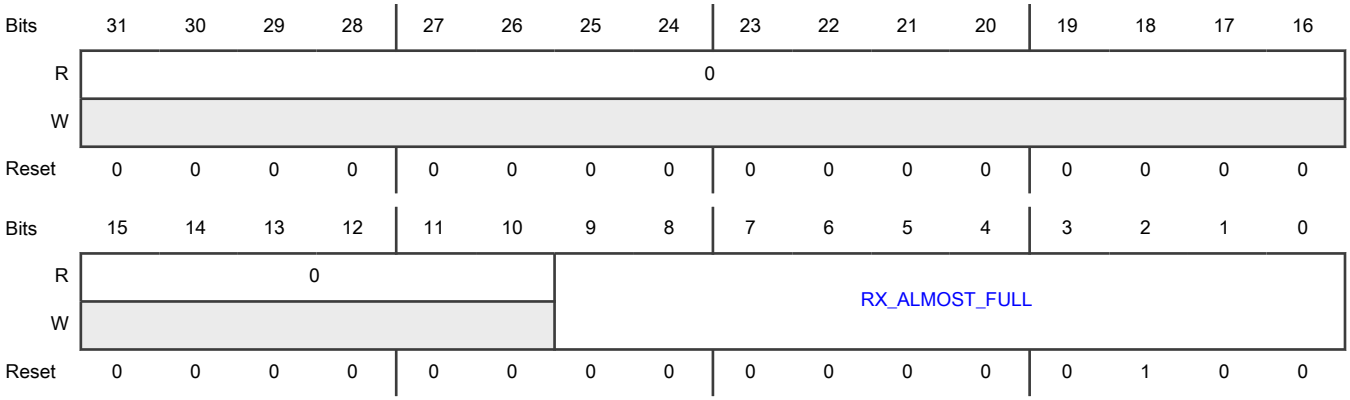
Field	Description
31-10 —	Reserved
9-0 RX_ALMOST_EMPTY	Value Of The Receive FIFO Almost Empty Threshold Value, in 64-bit words, of the receive FIFO almost empty threshold. When the FIFO level reaches the value programmed in this field and the end-of-frame has not been received for the frame yet, the core receive read control stops FIFO read (and subsequently stops transferring data to the MAC client application). It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. A minimum value of 4 should be set.

44.5.1.34 Receive FIFO Almost Full Threshold (RAFL)

Offset

Register	Offset
RAFL	19Ch

Diagram



Fields

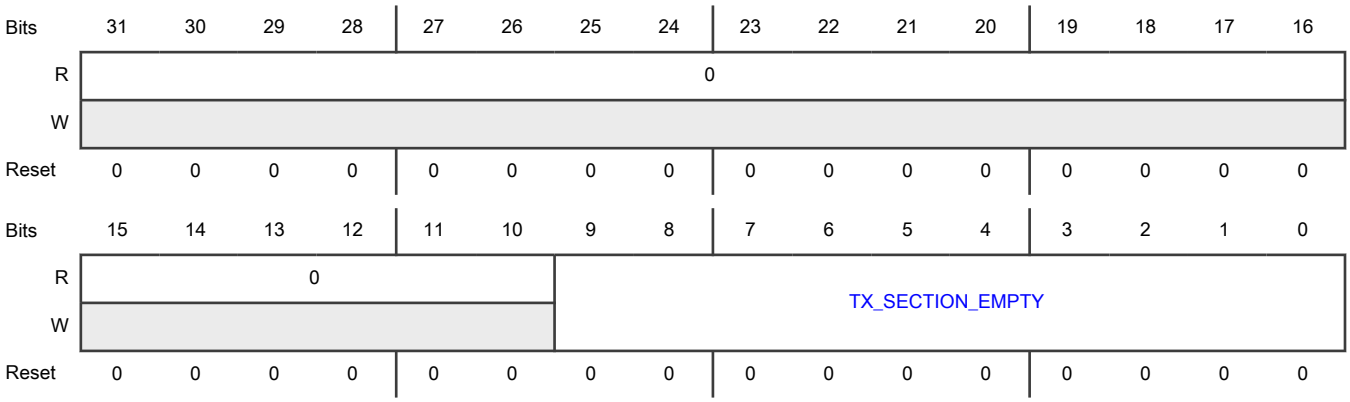
Field	Description
31-10 —	Reserved
9-0 RX_ALMOST_FULL	Value Of The Receive FIFO Almost Full Threshold Value, in 64-bit words, of the receive FIFO almost full threshold. When the FIFO level comes close to the maximum, so that there is no more space for at least RX_ALMOST_FULL number of words, the MAC stops writing data in the FIFO and truncates the received frame to avoid FIFO overflow. The corresponding error status will be set when the frame is delivered to the application. A minimum value of 4 should be set.

44.5.1.35 Transmit FIFO Section Empty Threshold (TSEM)

Offset

Register	Offset
TSEM	1A0h

Diagram



Fields

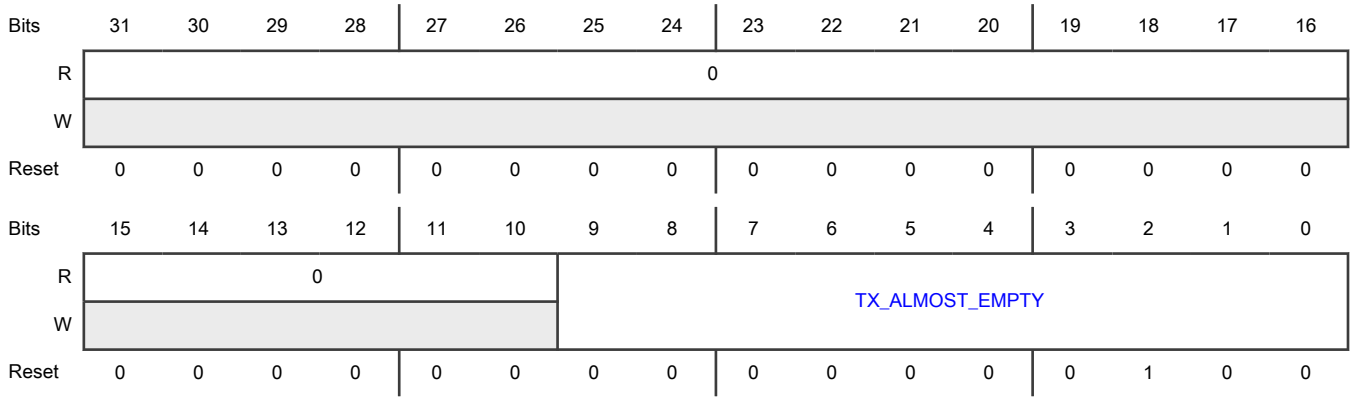
Field	Description
31-10 —	Reserved
9-0 TX_SECTION_EMPTY	Value Of The Transmit FIFO Section Empty Threshold Value, in 64-bit words, of the transmit FIFO section empty threshold. See Transmit FIFO for more information.

44.5.1.36 Transmit FIFO Almost Empty Threshold (TAEM)

Offset

Register	Offset
TAEM	1A4h

Diagram



Fields

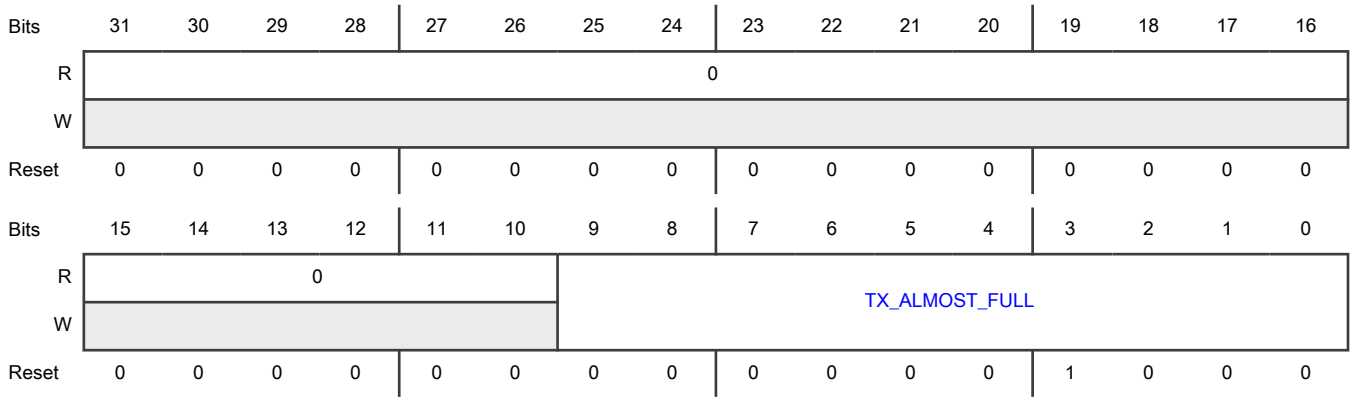
Field	Description
31-10 —	Reserved
9-0 TX_ALMOST_EMPTY	Value of Transmit FIFO Almost Empty Threshold Value, in 64-bit words, of the transmit FIFO almost empty threshold. When the FIFO level reaches the value programmed in this field, and no end-of-frame is available for the frame, the MAC transmit logic, to avoid FIFO underflow, stops reading the FIFO and transmits a frame with an MII error indication. See Transmit FIFO for more information. A minimum value of 4 should be set.

44.5.1.37 Transmit FIFO Almost Full Threshold (TAFL)

Offset

Register	Offset
TAFL	1A8h

Diagram



Fields

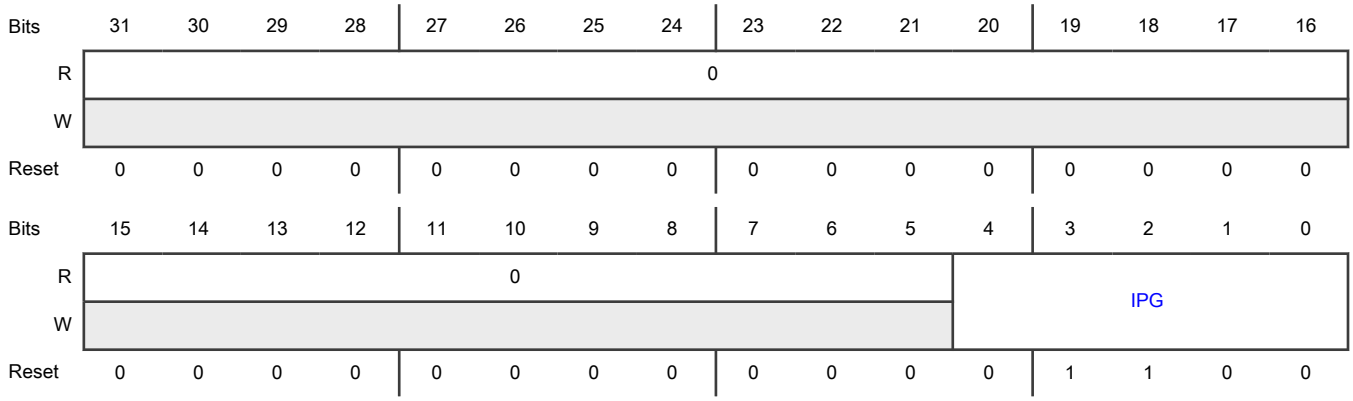
Field	Description
31-10 —	Reserved
9-0 TX_ALMOST_FULL	Value Of The Transmit FIFO Almost Full Threshold Value, in 64-bit words, of the transmit FIFO almost full threshold. A minimum value of six is required . A recommended value of at least 8 should be set allowing a latency of two clock cycles to the application. If more latency is required the value can be increased as necessary (latency = TAFL - 5). When the FIFO level comes close to the maximum, so that there is no more space for at least TX_ALMOST_FULL number of words, the pin ff_tx_rdy is deasserted. If the application does not react on this signal, the FIFO write control logic, to avoid FIFO overflow, truncates the current frame and sets the error status. As a result, the frame will be transmitted with an GMII/MII error indication. See Transmit FIFO for more information. **NOTE** A FIFO overflow is a fatal error and requires a global reset on the transmit datapath or at least deassertion of ETHEREN.

44.5.1.38 Transmit Inter-Packet Gap (TIPG)

Offset

Register	Offset
TIPG	1ACh

Diagram



Fields

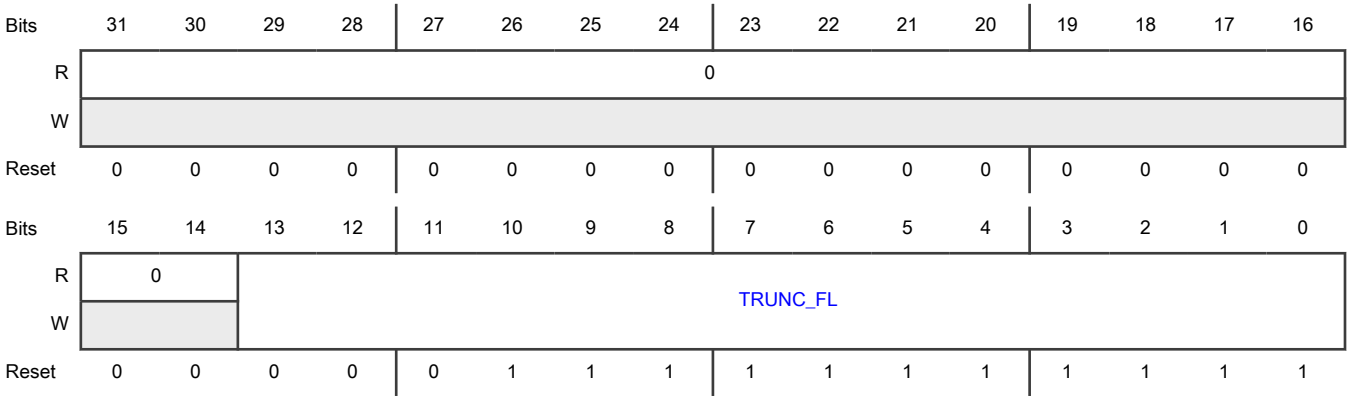
Field	Description
31-5 —	Reserved
4-0 IPG	Transmit Inter-Packet Gap Indicates the IPG, in bytes, between transmitted frames. Valid values range from 8 to 26. If the written value is less than 8 or greater than 26, the internal (effective) IPG is 12. **NOTE** The IPG value read will be the value that was written, even if it is out of range.

44.5.1.39 Frame Truncation Length (FTRL)

Offset

Register	Offset
FTRL	1B0h

Diagram



Fields

Field	Description
31-14 —	Reserved
13-0 TRUNC_FL	Frame Truncation Length Indicates the value a receive frame is truncated, if it is greater than this value. Must be greater than or equal to RCR[MAX_FL].

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE Truncation happens at TRUNC_FL. However, when truncation occurs, the application (FIFO) may receive less data, guaranteeing that it never receives more than the set limit.

44.5.1.40 Transmit Accelerator Function Configuration (TACC)

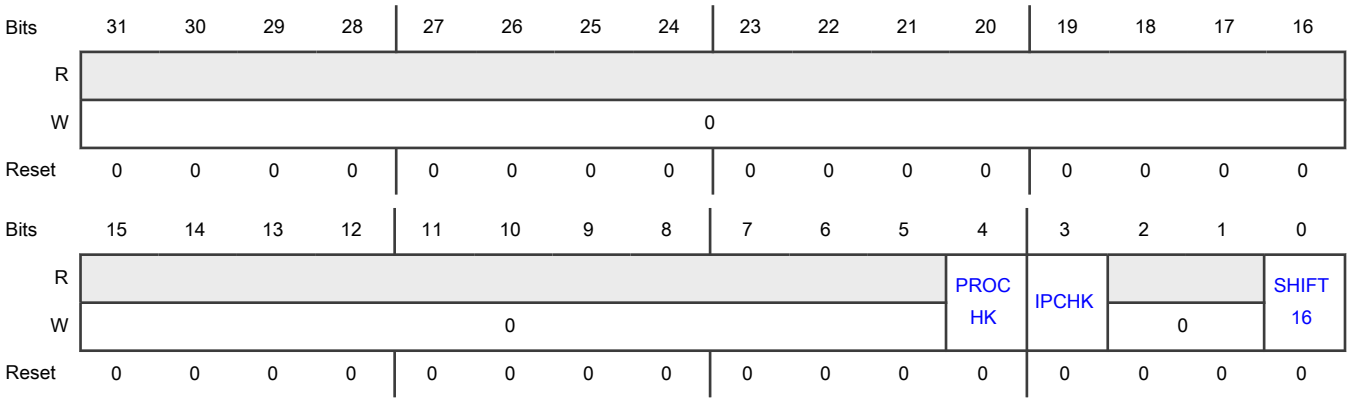
TACC controls accelerator actions when sending frames. The register can be changed before or after each frame, but it must remain unmodified during frame writes into the transmit FIFO.

The TFWR[STRFWD] field must be set to use the checksum feature.

Offset

Register	Offset
TACC	1C0h

Diagram



Fields

Field	Description
31-5 —	This write-only field is reserved. It must always be written with the value 0.
4 PROCHK	Enables insertion of protocol checksum. 0 - Checksum not inserted. 1 - If an IP frame with a known protocol is transmitted, the checksum is inserted automatically into the frame. The checksum field must be cleared. The other frames are not modified.
3	Enables insertion of IP header checksum.

Table continues on the next page...

Table continued from the previous page...

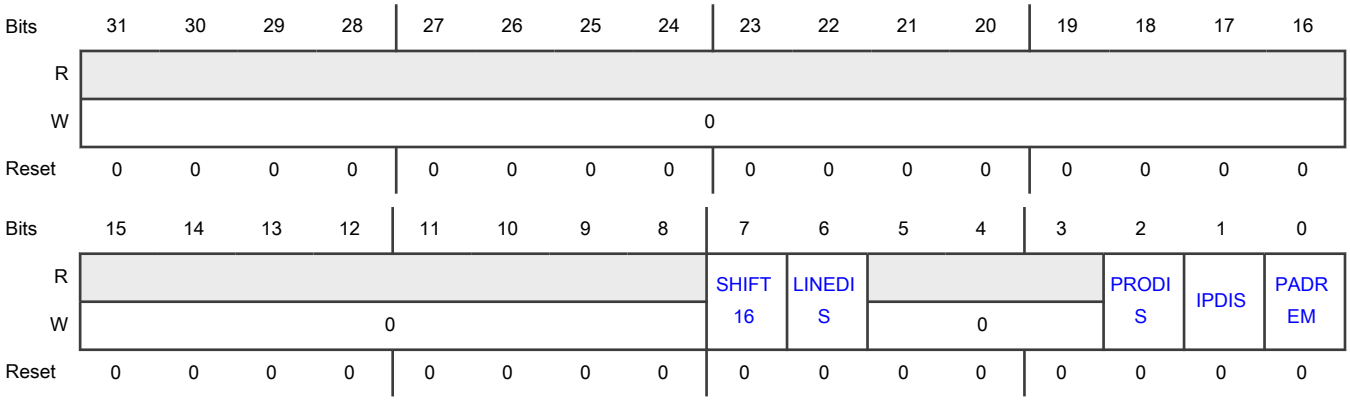
Field	Description
IPCHK	0 - Checksum is not inserted. 1 - If an IP frame is transmitted, the checksum is inserted automatically. The IP header checksum field must be cleared. If a non-IP frame is transmitted the frame is not modified.
2-1 —	This write-only field is reserved. It must always be written with the value 0.
0 SHIFT16	TX FIFO Shift-16 0 - Disabled. 1 - Indicates to the transmit data FIFO that the written frames contain two additional octets before the frame data. This means the actual frame begins at bit 16 of the first word written into the FIFO. This function allows putting the frame payload on a 32-bit boundary in memory, as the 14-byte Ethernet header is extended to a 16-byte header.

44.5.1.41 Receive Accelerator Function Configuration (RACC)

Offset

Register	Offset
RACC	1C4h

Diagram



Fields

Field	Description
31-8 —	This write-only field is reserved. It must always be written with the value 0.

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 SHIFT16	RX FIFO Shift-16 When this field is set, the actual frame data starts at bit 16 of the first word read from the RX FIFO aligning the Ethernet payload on a 32-bit boundary. **NOTE** This function only affects the FIFO storage and has no influence on the statistics, which use the actual length of the frame received. 0 - Disabled. 1 - Instructs the MAC to write two additional bytes in front of each frame received into the RX FIFO.
6 LINEDIS	Enable Discard Of Frames With MAC Layer Errors 0 - Frames with errors are not discarded. 1 - Any frame received with a CRC, length, or PHY error is automatically discarded and not forwarded to the user application interface.
5-3 —	This write-only field is reserved. It must always be written with the value 0.
2 PRODIS	Enable Discard Of Frames With Wrong Protocol Checksum 0 - Frames with wrong checksum are not discarded. 1 - If a TCP/IP, UDP/IP, or ICMP/IP frame is received that has a wrong TCP, UDP, or ICMP checksum, the frame is discarded. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
1 IPDIS	Enable Discard Of Frames With Wrong IPv4 Header Checksum 0 - Frames with wrong IPv4 header checksum are not discarded. 1 - If an IPv4 frame is received with a mismatching header checksum, the frame is discarded. IPv6 has no header checksum and is not affected by this setting. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
0 PADREM	Enable Padding Removal For Short IP Frames 0 - Padding not removed. 1 - Any bytes following the IP payload section of the frame are removed from the frame.

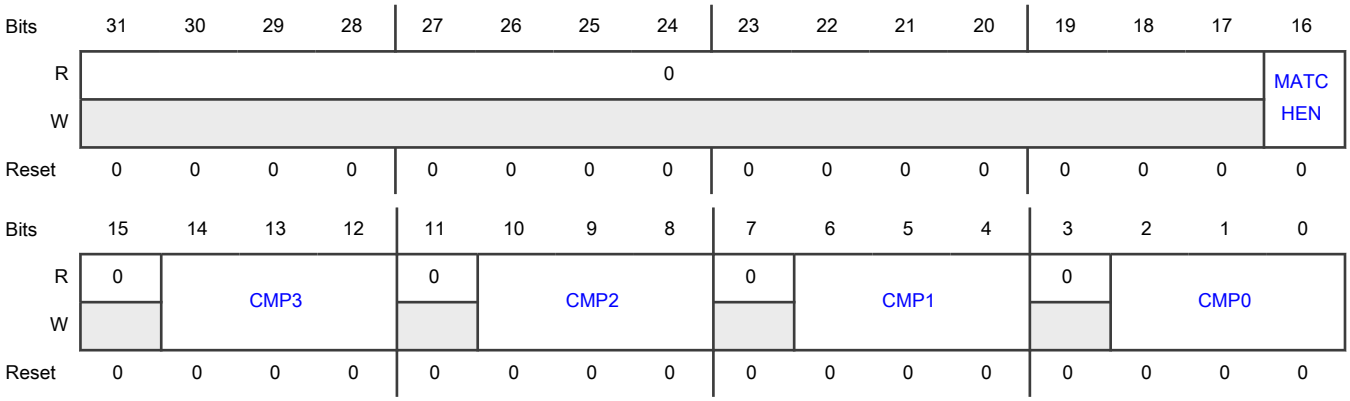
44.5.1.42 Receive Classification Match Register for Class n (RCMR1 - RCMR2)

This match register allows specifying up to four priorities, which are tested (OR'ed) simultaneously. The match detection uses the extracted VLAN field according to the rules for VLAN detection configured through the ECR register. If both match registers, RCMR1 and RCMR2, report a match at the same time, only the class 1 match is indicated as the final result.

Offset

Register	Offset
RCMR1	1C8h
RCMR2	1CCh

Diagram



Fields

Field	Description
31-17 —	This write-only field is reserved. It must always be written with the value 0.
16 MATCHEN	Match Enable **NOTE** A comparison is done only on incoming VLAN frames. If no VLAN frame is received no match will occur. If both match registers have overlapping compare values and hence can match both on the same frame, only class 1 will be indicated and the class 2 match is ignored. 0 - Disabled (default): no compares will occur and the classification indicator for this class will never assert. 1 - The register contents are valid and a comparison with all compare values is done when a VLAN frame is received.
15 —	This write-only field is reserved. It must always be written with the value 0.
14-12 CMP3	Compare 3 Fourth value to compare against. If unused it must be set to the same value as CMP0.
11 —	This write-only field is reserved. It must always be written with the value 0.

Table continues on the next page...

Table continued from the previous page...

Field	Description
10-8 CMP2	Compare 2 Third value to compare against. If unused it must be set to the same value as CMP0.
7 —	This write-only field is reserved. It must always be written with the value 0.
6-4 CMP1	Compare 1 Second value to compare against. If unused it must be set to the same value as CMP0.
3 —	This write-only field is reserved. It must always be written with the value 0.
2-0 CMP0	Compare 0 A three-bit value that will be compared with the frame's VLAN priority field (if a VLAN frame is received). All four compare values, CMP0..3, will be used in parallel. If any of the values match, a match for the class is reported (if MATCHEN is 1). **NOTE** To implement a single priority match, all four compare values must be set to the same value.

44.5.1.43 DMA Class Based Configuration (DMA1CFG - DMA2CFG)

The DMA class based configuration registers are used to configure the DMA controller interface to support the additional class 1 (buffer descriptor ring 1) and class 2 (buffer descriptor ring 2) traffic and define configuration options such as bandwidth allocation as needed.

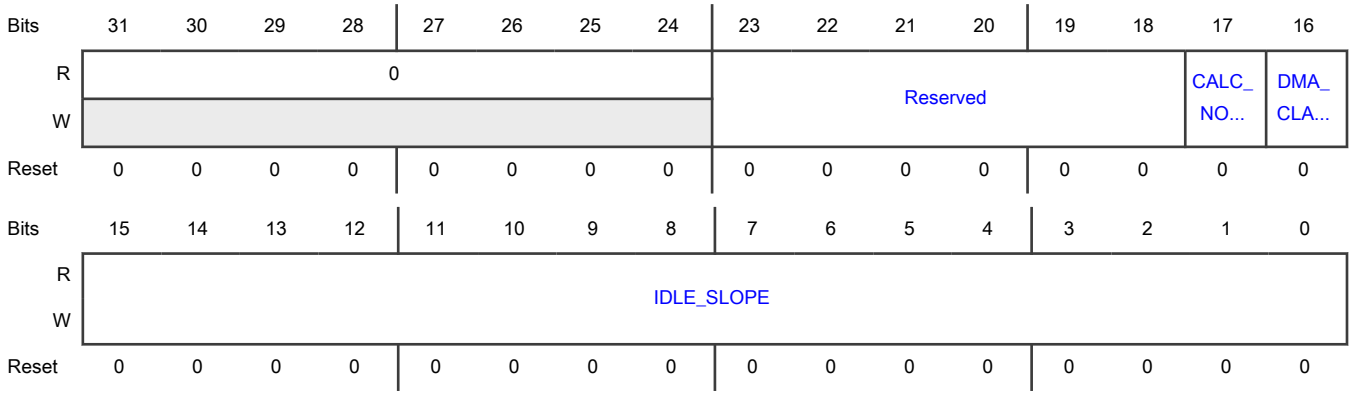
NOTE

The registers are cleared when ECR[ETHEREN] becomes 0.

Offset

Register	Offset
DMA1CFG	1D8h
DMA2CFG	1DCh

Diagram



Fields

Field	Description
31-24 —	Reserved
23-18 —	Write only zeroes to this field **NOTE** Write only zeroes to this field.
17 CALC_NOIPG	Calculate no IPG Disable inclusion of IPG bytes for bandwidth calculations. 0 - The traffic shaper function should consider 12 octets of IPG in addition to the frame data transferred for a frame when doing bandwidth calculations. This is the default. 1 - Addition of 12 bytes for the IPG should be omitted when calculating the bandwidth (for traffic shaping, when writing a frame into the transmit FIFO, the shaper will usually consider 12 bytes of IPG for every frame as part of the bandwidth allocated by the frame. This addition can be suppressed, meaning short frames will become more bandwidth than large frames due to the relation of data to IPG overhead).
16 DMA_CLASS_EN	DMA class enable 0 - The DMA controller's channel for the class is not used. Disabling the DMA controller of a class also requires disabling the class match comparator for the class (see registers RCMRn). When class 1 and class 2 queues are disabled then their frames will be placed in queue 0. 1 - Enable the DMA controller to support the corresponding descriptor ring for this class of traffic.
15-0 IDLE_SLOPE	Idle slope 16-bit value to define the per class idle slope setting used by the credit based shaper defining allocated bandwidth for the class. This value is used to calculate the BW (bandwidth) fraction, given by the equation, BW fraction = 1/(1+512/IDLE_SLOPE). Idle slope is restricted to certain values. For values less than 128, idle slope = 2^n, where n = 0, 1, 2, ...6. For values equal to or greater than 128, idle slope = 128×m, where m = 1, 2, 3, ...12. Example 1. BW fraction = 0.20 = 1/(1+(512/128)); therefore idleslope = 128.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Example 2. BW fraction = 0.33 = 1/(1+(512/256)); therefore idleslope = 256. Example 3. BW fraction = 0.75 = 1/(1+(512/1536)); therefore idleslope = 1536. **NOTE** For AVB applications, the BW fraction of class 1 and class 2 combined must not exceed .75.

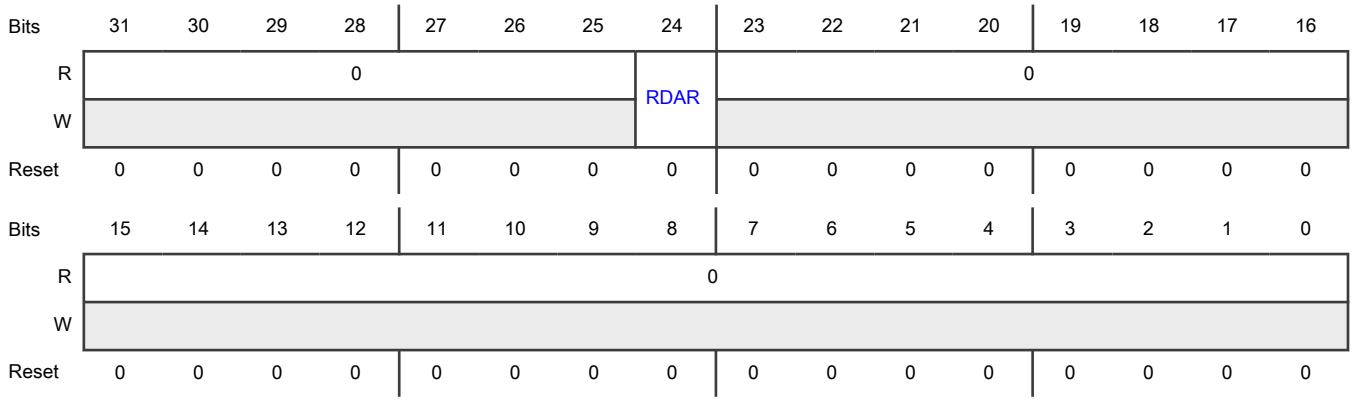
44.5.1.44 Receive Descriptor Active Register - Ring 1 (RDAR1)

RDAR1 is a command register, written by the user, to indicate that the receive descriptor ring 1 has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

Offset

Register	Offset
RDAR1	1E0h

Diagram



Fields

Field	Description
31-25 —	Reserved
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

44.5.1.45 Transmit Descriptor Active Register - Ring 1 (TDAR1)

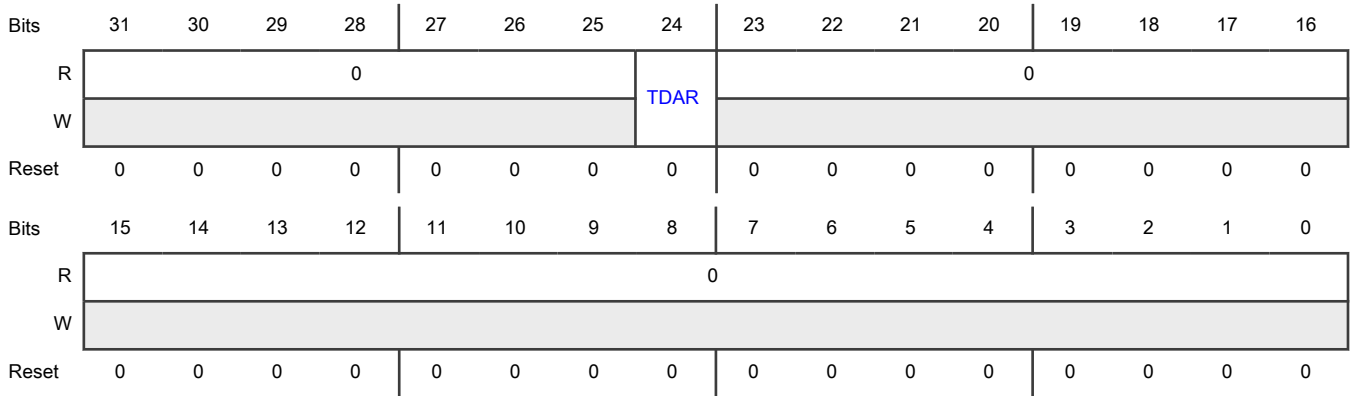
TDAR1 is a command register that the user writes to indicate that transmit descriptor ring 1 has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

The TDAR register is cleared at reset, when ECR[ETHEREN] transitions from set to cleared, or when ECR[RESET] is set.

Offset

Register	Offset
TDAR1	1E4h

Diagram



Fields

Field	Description
31-25 —	Reserved
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

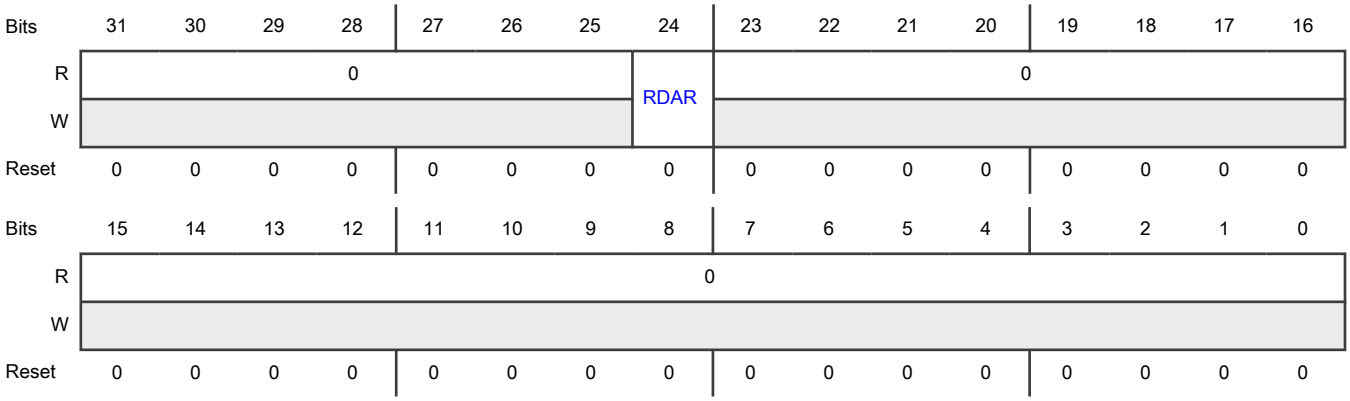
44.5.1.46 Receive Descriptor Active Register - Ring 2 (RDAR2)

RDAR2 is a command register, written by the user, to indicate that the receive descriptor ring 2 has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

Offset

Register	Offset
RDAR2	1E8h

Diagram



Fields

Field	Description
31-25 —	Reserved
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

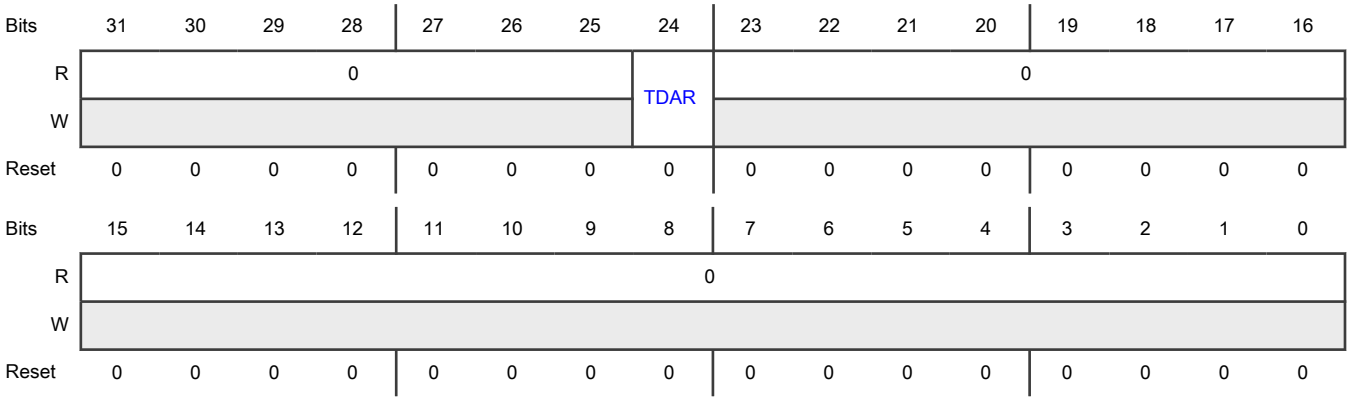
44.5.1.47 Transmit Descriptor Active Register - Ring 2 (TDAR2)

TDAR2 is a command register that the user writes to indicate that transmit descriptor ring 2 has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

Offset

Register	Offset
TDAR2	1ECh

Diagram



Fields

Field	Description
31-25 —	Reserved
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

44.5.1.48 QOS Scheme (QOS)

This register sets the QOS scheme.

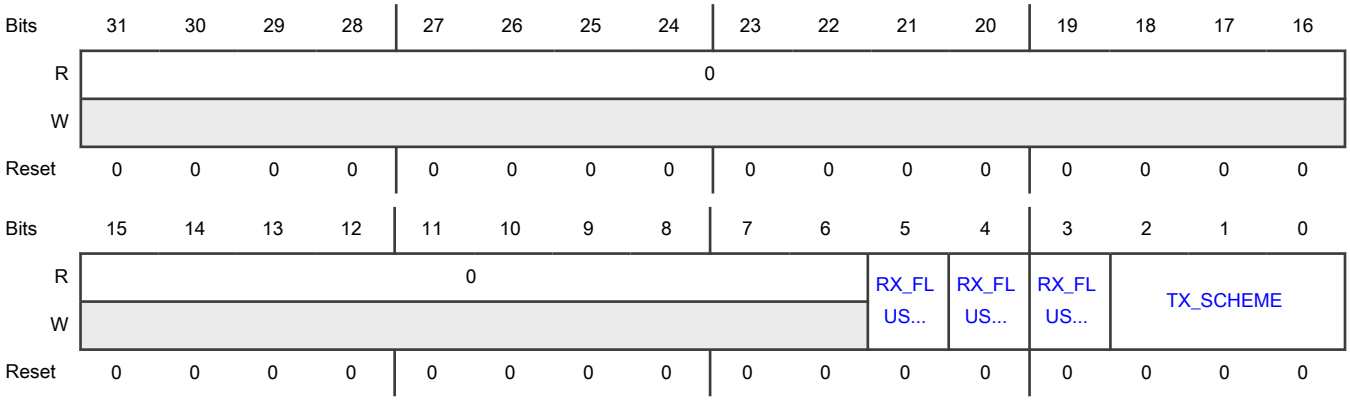
NOTE

When both class 1 and class 2 are disabled, RX flushing for these rings must also be disabled.

Offset

Register	Offset
QOS	1F0h

Diagram



Fields

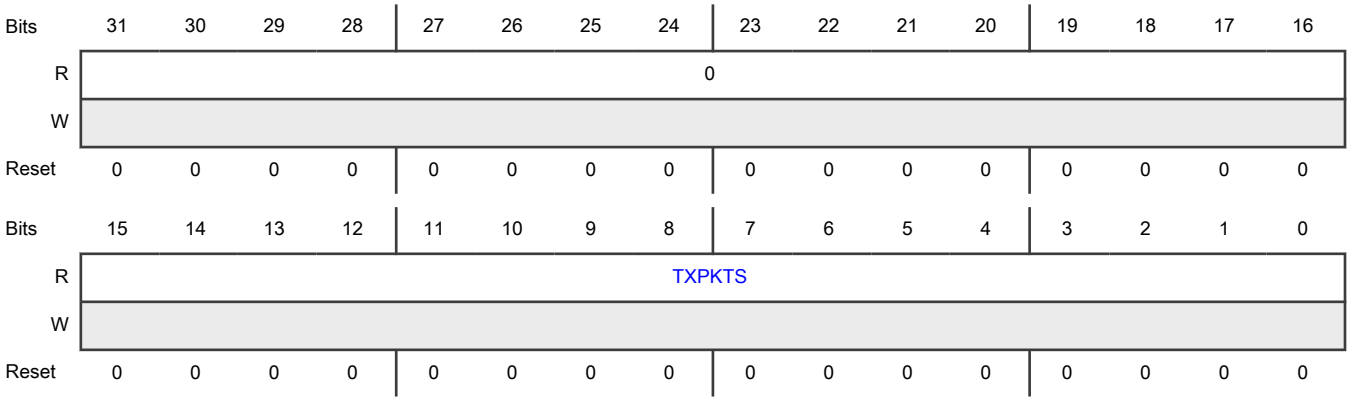
Field	Description
31-6 —	Reserved
5 RX_FLUSH2	RX Flush Ring 2 Enable or disable RX Flush for ring 2. See Receive flushing. 0 - Disable 1 - Enable
4 RX_FLUSH1	RX Flush Ring 1 Enable or disable RX Flush for ring 1. See Receive flushing. 0 - Disable 1 - Enable
3 RX_FLUSH0	RX Flush Ring 0 Enable or disable RX Flush for ring 0. See Receive flushing. 0 - Disable 1 - Enable
2-0 TX_SCHEME	TX scheme configuration Configuration information for DMA to select transmitter queue selection/arbitration scheme. 000 - Credit-based scheme 001 - Round-robin scheme 010-111 - Reserved

44.5.1.49 Tx Packet Count Statistic (RMON_T_PACKETS)

Offset

Register	Offset
RMON_T_PACKETS	204h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Packet count Transmit packet count

44.5.1.50 Tx Broadcast Packets Statistic (RMON_T_BC_PKT)

RMON Tx Broadcast Packets

Offset

Register	Offset
RMON_T_BC_PKT	208h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of broadcast packets

44.5.1.51 Tx Multicast Packets Statistic (RMON_T_MC_PKT)

Offset

Register	Offset
RMON_T_MC_PKT	20Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

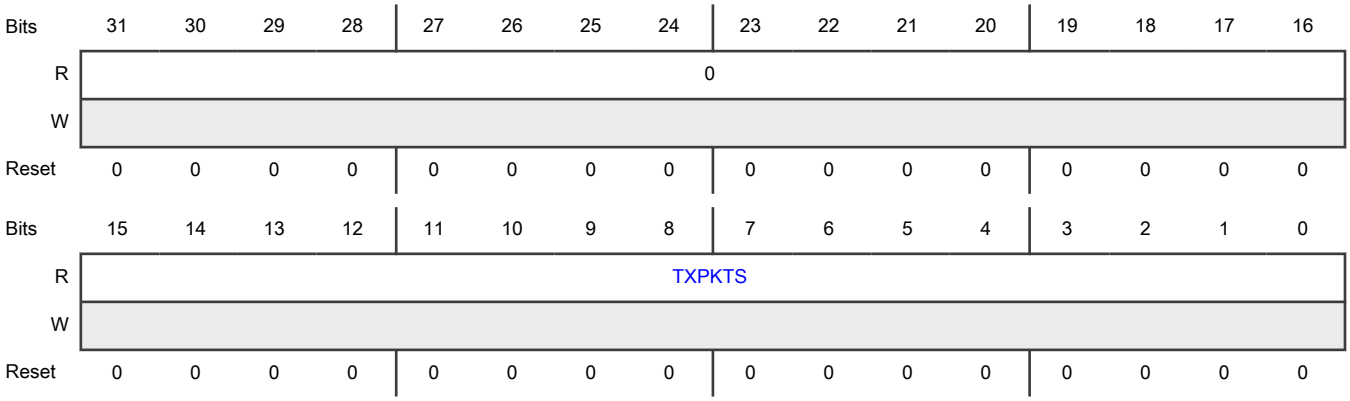
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of multicast packets

44.5.1.52 Tx Packets with CRC/Align Error Statistic (RMON_T_CRC_ALIGN)

Offset

Register	Offset
RMON_T_CRC_ALIGN	210h

Diagram



Fields

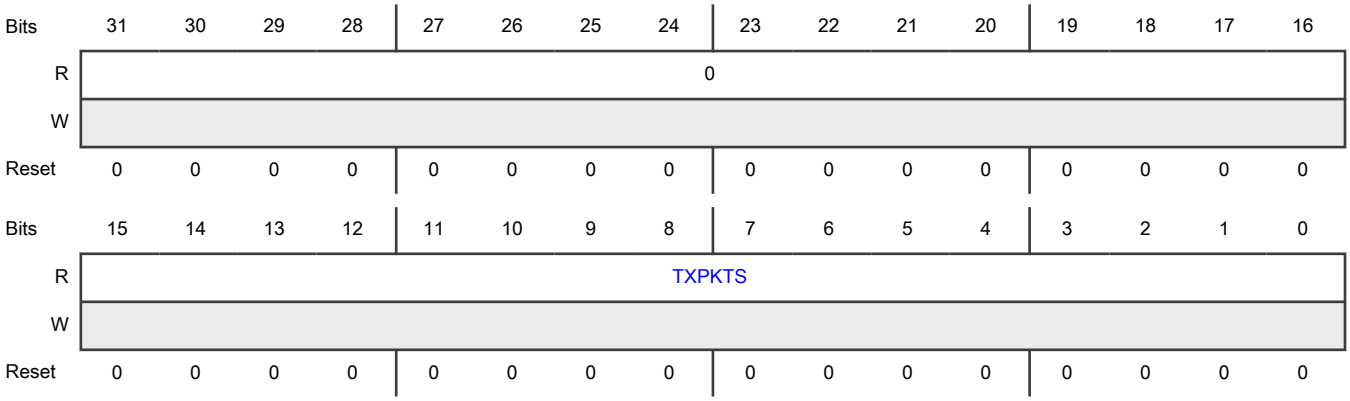
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of packets with CRC/align error

44.5.1.53 Tx Packets Less Than Bytes and Good CRC Statistic (RMON_T_UNDERSIZE)

Offset

Register	Offset
RMON_T_UNDERSIZE	214h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets less than 64 bytes with good CRC

44.5.1.54 Tx Packets GT MAX_FL bytes and Good CRC Statistic (RMON_T_OVERSIZE)

Offset

Register	Offset
RMON_T_OVERSIZE	218h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets greater than MAX_FL bytes with good CRC

44.5.1.55 Tx Packets Less Than 64 Bytes and Bad CRC Statistic (RMON_T_FRAG)

.

Offset

Register	Offset
RMON_T_FRAG	21Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

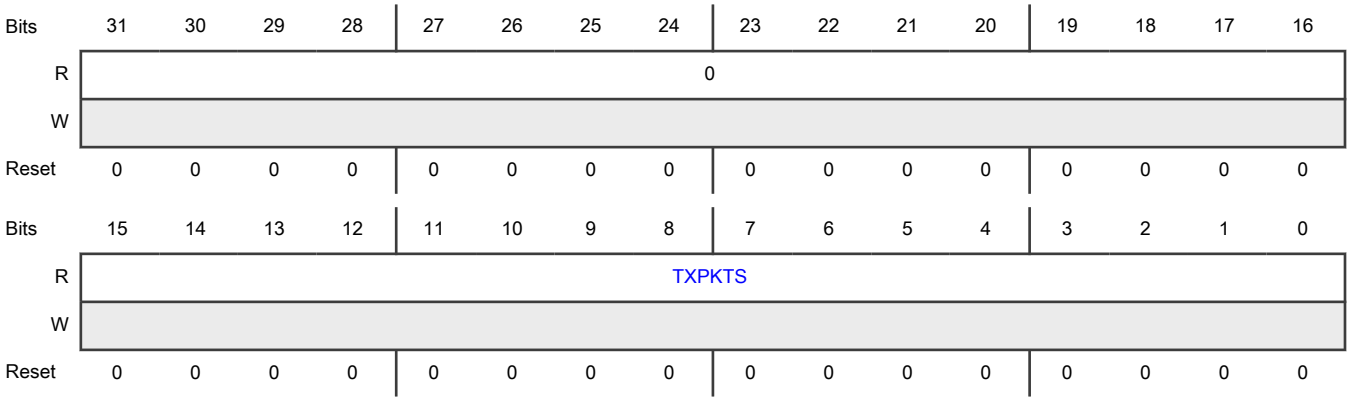
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of packets less than 64 bytes with bad CRC

44.5.1.56 Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic (RMON_T_JAB)

Offset

Register	Offset
RMON_T_JAB	220h

Diagram



Fields

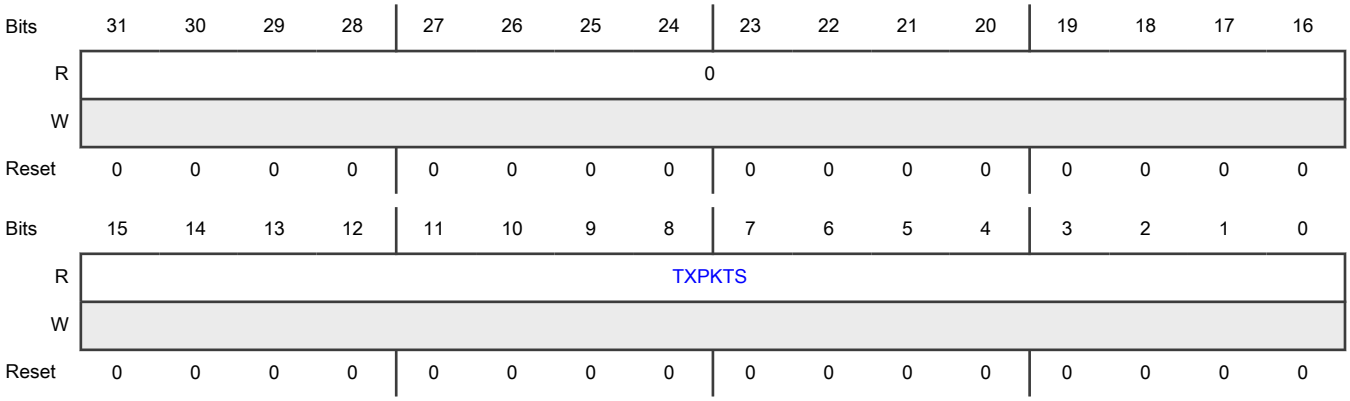
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets greater than MAX_FL bytes and bad CRC

44.5.1.57 Tx Collision Count Statistic (RMON_T_COL)

Offset

Register	Offset
RMON_T_COL	224h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of transmit collisions

44.5.1.58 Tx 64-Byte Packets Statistic (RMON_T_P64)

.

Offset

Register	Offset
RMON_T_P64	228h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of 64-byte transmit packets

44.5.1.59 Tx 65- to 127-byte Packets Statistic (RMON_T_P65TO127)

Offset

Register	Offset
RMON_T_P65TO127	22Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

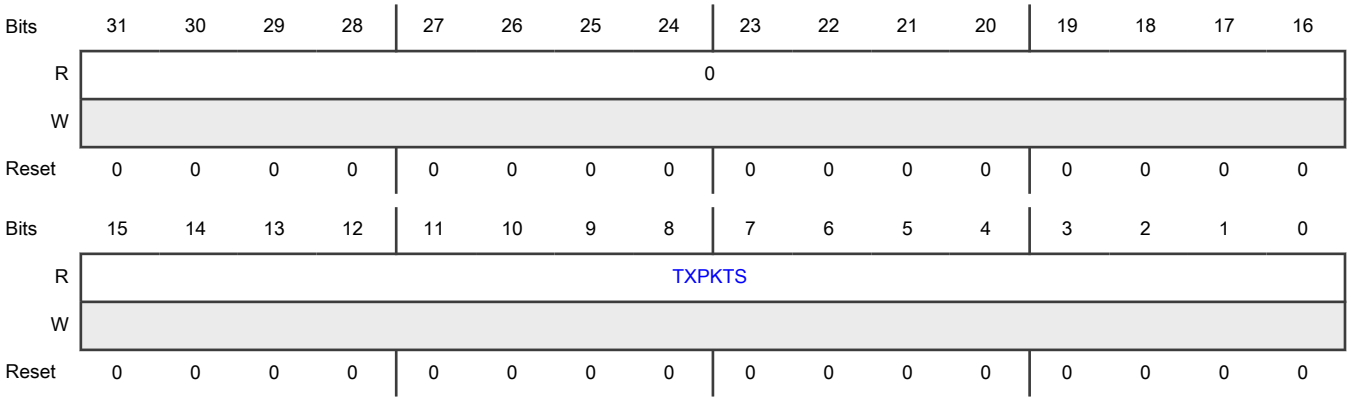
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of 65- to 127-byte transmit packets

44.5.1.60 Tx 128- to 255-byte Packets Statistic (RMON_T_P128TO255)

Offset

Register	Offset
RMON_T_P128TO255	230h

Diagram



Fields

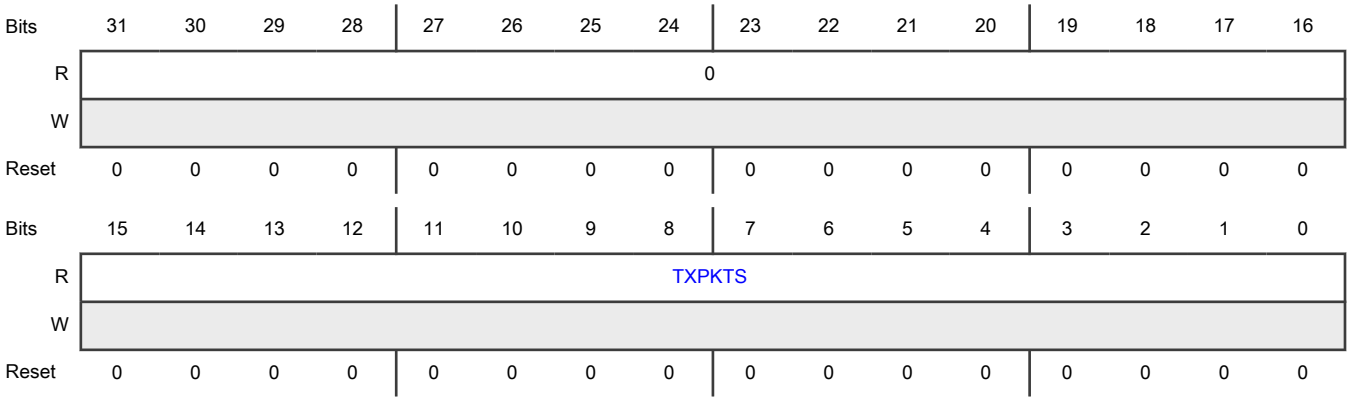
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of 128- to 255-byte transmit packets

44.5.1.61 Tx 256- to 511-byte Packets Statistic (RMON_T_P256TO511)

Offset

Register	Offset
RMON_T_P256TO511	234h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of 256- to 511-byte transmit packets

44.5.1.62 Tx 512- to 1023-byte Packets Statistic (RMON_T_P512TO1023)

.

Offset

Register	Offset
RMON_T_P512TO1023	238h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of 512- to 1023-byte transmit packets

44.5.1.63 Tx 1024- to 2047-byte Packets Statistic (RMON_T_P1024TO2047)

Offset

Register	Offset
RMON_T_P1024TO2047	23Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

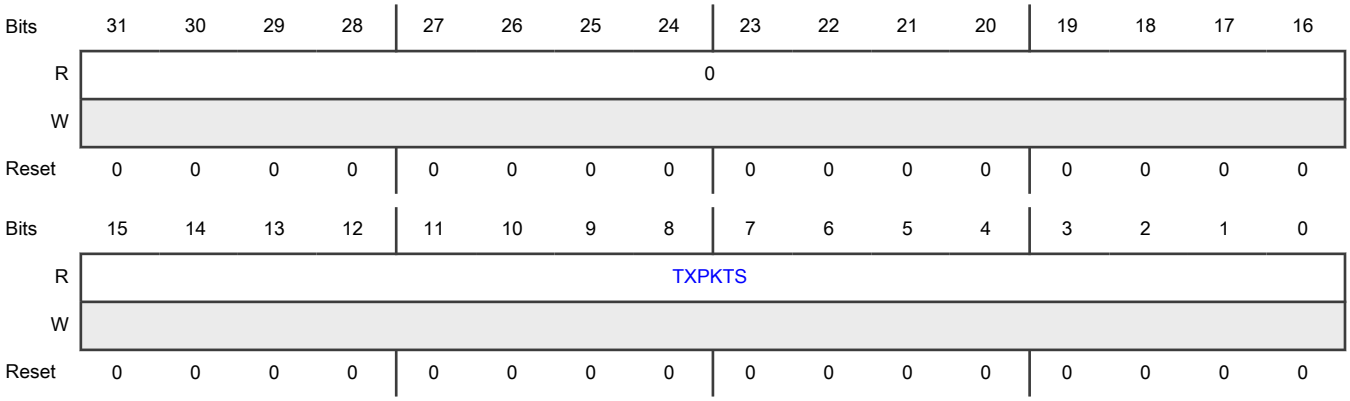
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of 1024- to 2047-byte transmit packets

44.5.1.64 Tx Packets Greater Than 2048 Bytes Statistic (RMON_T_P_GTE2048)

Offset

Register	Offset
RMON_T_P_GTE2048	240h

Diagram



Fields

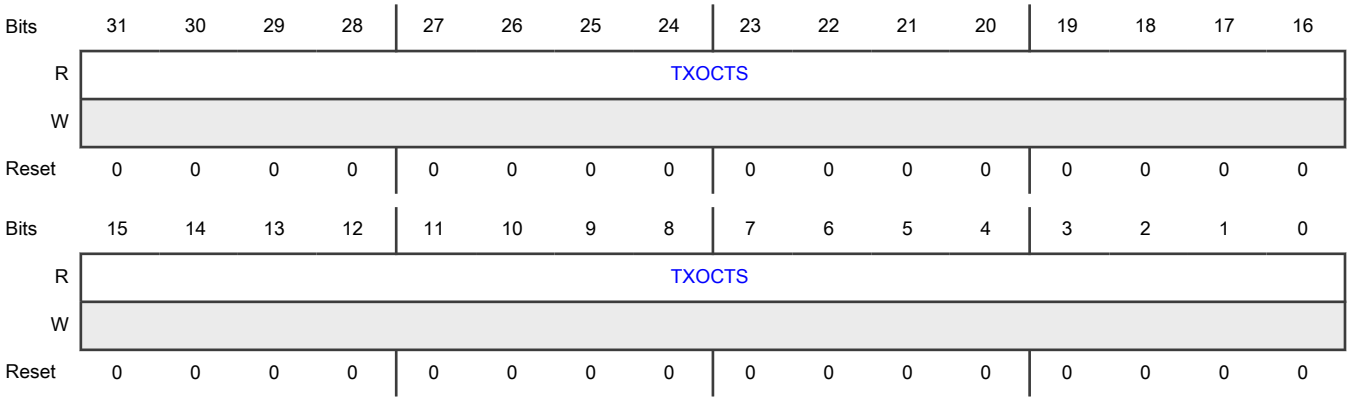
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets greater than 2048 bytes

44.5.1.65 Tx Octets Statistic (RMON_T_OCTETS)

Offset

Register	Offset
RMON_T_OCTETS	244h

Diagram



Fields

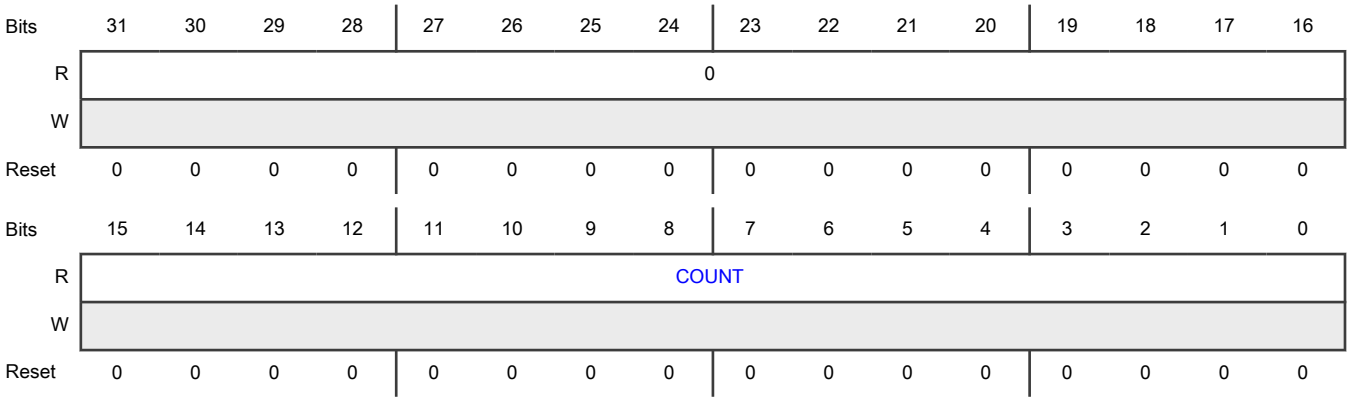
Field	Description
31-0 TXOCTS	Number of transmit octets

44.5.1.66 Frames Transmitted OK Statistic (IEEE_T_FRAME_OK)

Offset

Register	Offset
IEEE_T_FRAME_OK	24Ch

Diagram



Fields

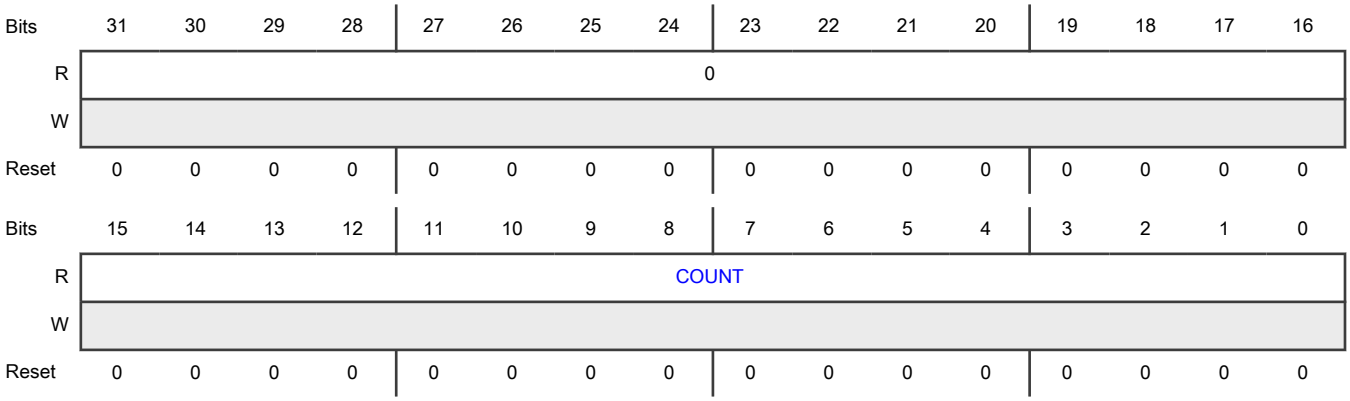
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted OK **NOTE** Does not increment for the broadcast frames when broadcast reject is enabled and promiscuous mode is disabled within the receive control register (RCR).

44.5.1.67 Frames Transmitted with Single Collision Statistic (IEEE_T_1COL)

Offset

Register	Offset
IEEE_T_1COL	250h

Diagram



Fields

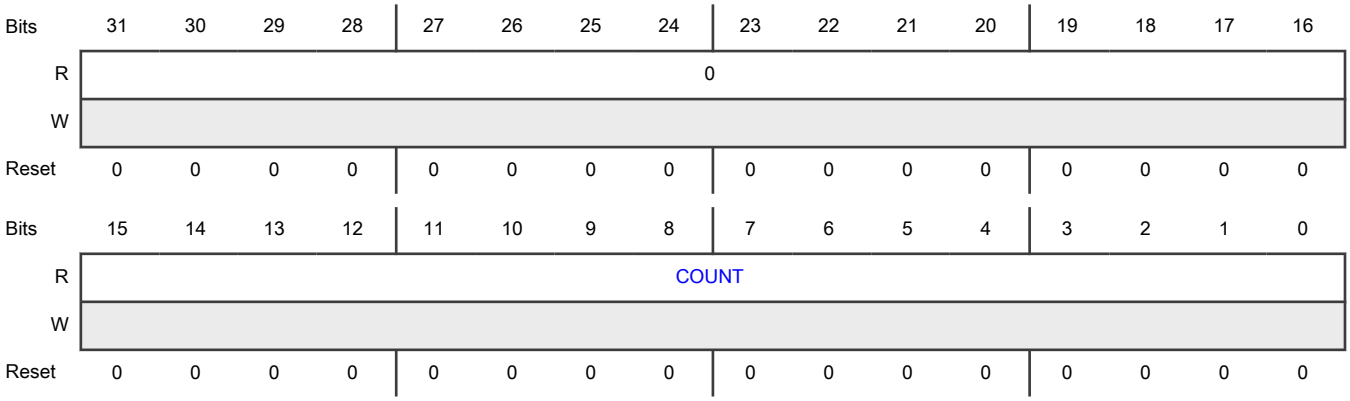
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with one collision

44.5.1.68 Frames Transmitted with Multiple Collisions Statistic (IEEE_T_MCOL)

Offset

Register	Offset
IEEE_T_MCOL	254h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with multiple collisions

44.5.1.69 Frames Transmitted after Deferral Delay Statistic (IEEE_T_DEF)

Offset

Register	Offset
IEEE_T_DEF	258h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with deferral delay

44.5.1.70 Frames Transmitted with Late Collision Statistic (IEEE_T_LCOL)

Offset

Register	Offset
IEEE_T_LCOL	25Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

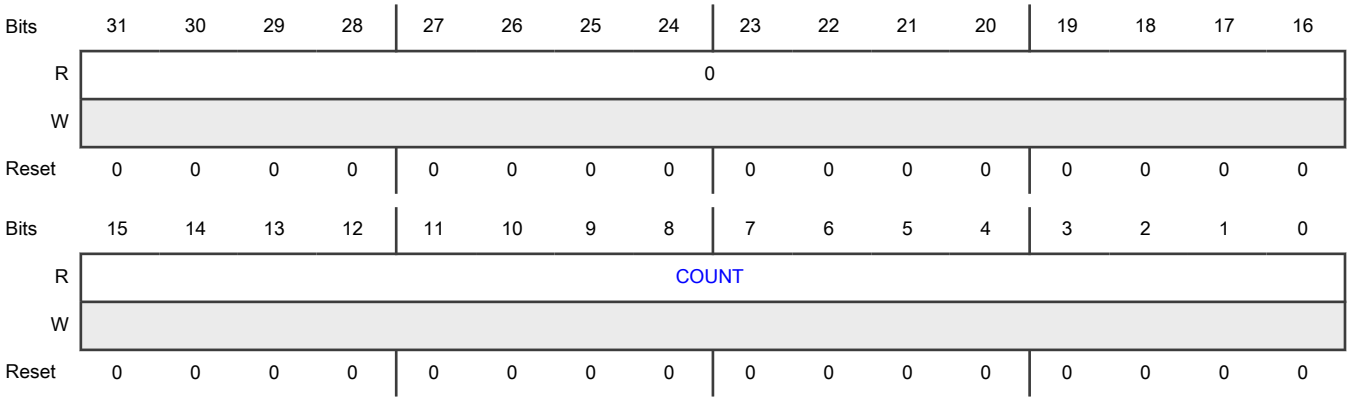
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with late collision

44.5.1.71 Frames Transmitted with Excessive Collisions Statistic (IEEE_T_EXCOL)

Offset

Register	Offset
IEEE_T_EXCOL	260h

Diagram



Fields

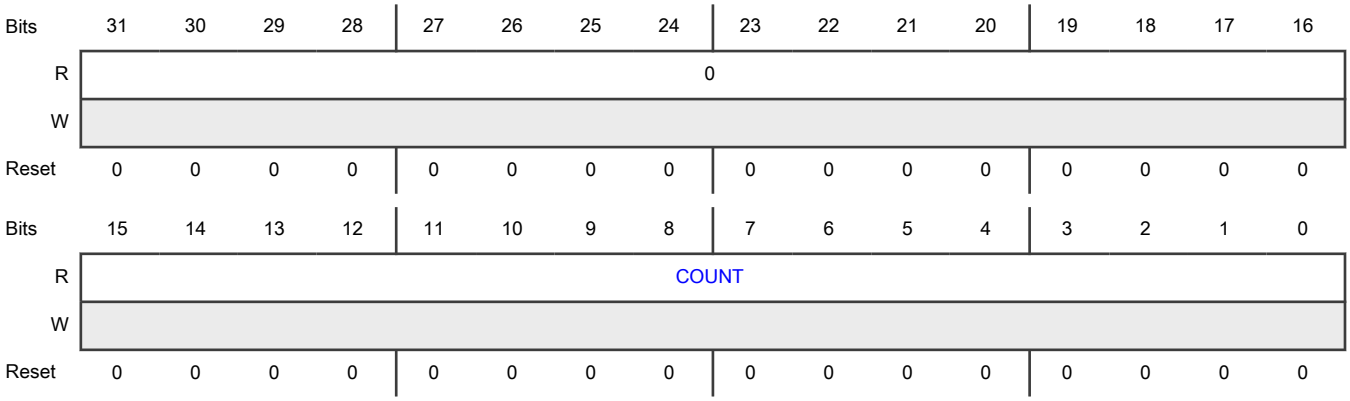
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with excessive collisions

44.5.1.72 Frames Transmitted with Tx FIFO Underrun Statistic (IEEE_T_MACERR)

Offset

Register	Offset
IEEE_T_MACERR	264h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with transmit FIFO underrun

44.5.1.73 Frames Transmitted with Carrier Sense Error Statistic (IEEE_T_CSERR)

Offset

Register	Offset
IEEE_T_CSERR	268h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with carrier sense error

44.5.1.74 Reserved Statistic (IEEE_T_SQE)

Offset

Register	Offset
IEEE_T_SQE	26Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

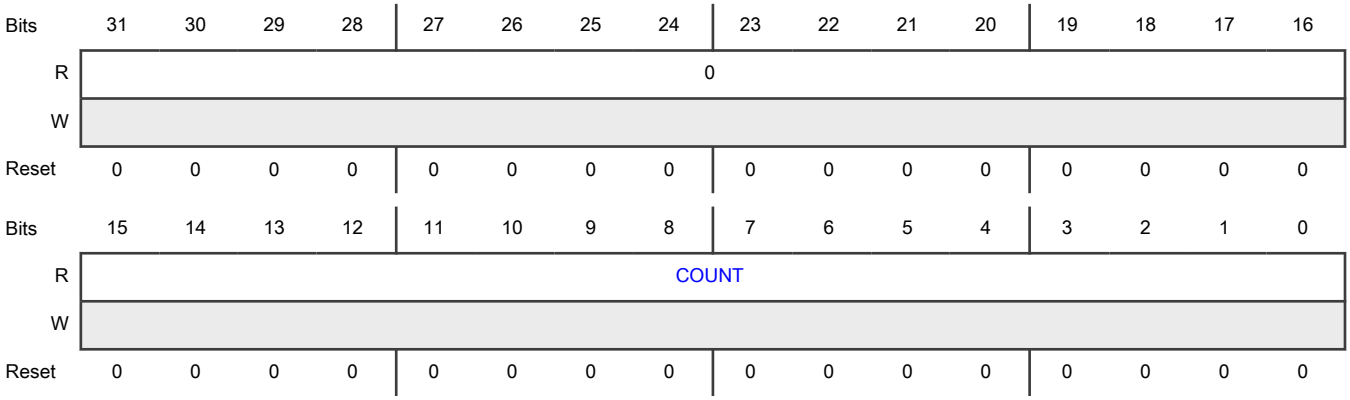
Field	Description
31-16 —	Reserved
15-0 COUNT	This read-only field is reserved and always has the value 0 This read-only field is reserved and always has the value 0. **NOTE** Counter not implemented as no SQE information is available.

44.5.1.75 Flow Control Pause Frames Transmitted Statistic (IEEE_T_FDXFC)

Offset

Register	Offset
IEEE_T_FDXFC	270h

Diagram



Fields

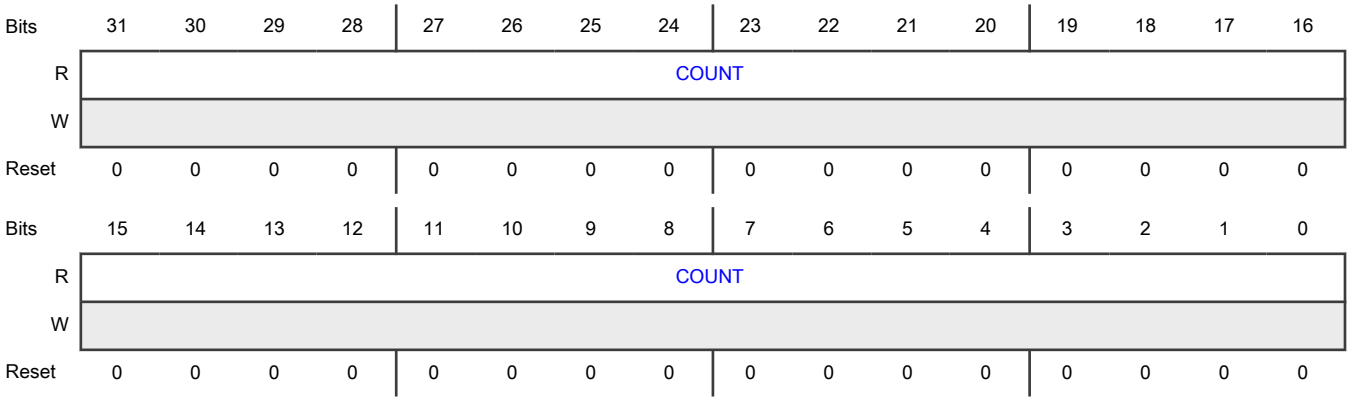
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of flow-control pause frames transmitted

44.5.1.76 Octet Count for Frames Transmitted w/o Error Statistic (IEEE_T_OCTETS_OK)

Offset

Register	Offset
IEEE_T_OCTETS_OK	274h

Diagram



Fields

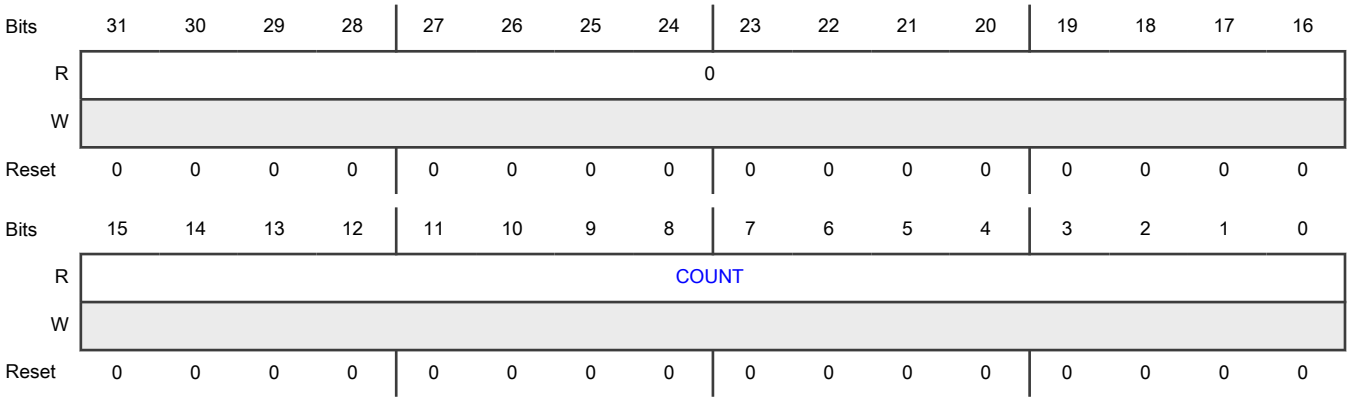
Field	Description
31-0 COUNT	Octet count for frames transmitted without error Counts total octets (includes header and FCS fields).

44.5.1.77 Rx Packet Count Statistic (RMON_R_PACKETS)

Offset

Register	Offset
RMON_R_PACKETS	284h

Diagram



Fields

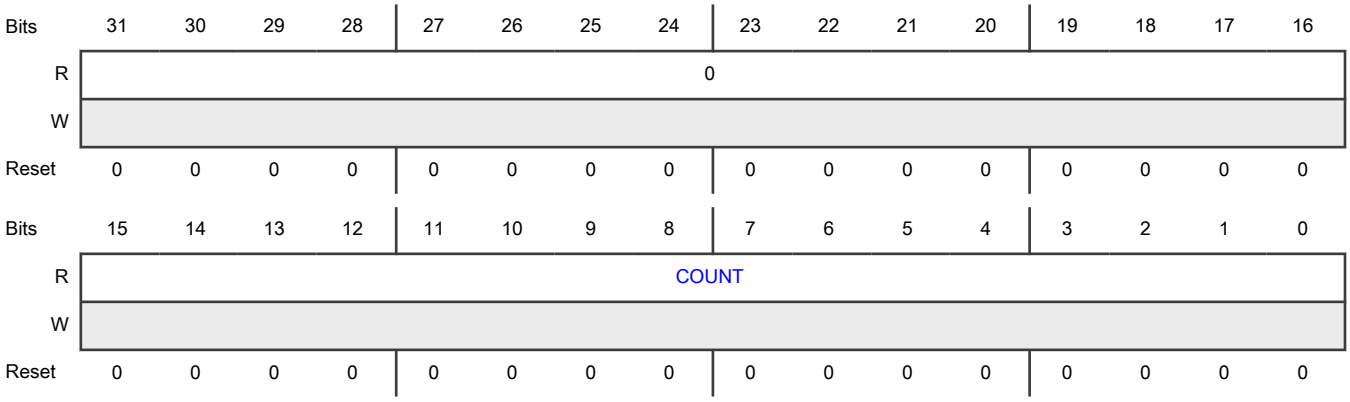
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of packets received

44.5.1.78 Rx Broadcast Packets Statistic (RMON_R_BC_PKT)

Offset

Register	Offset
RMON_R_BC_PKT	288h

Diagram



Fields

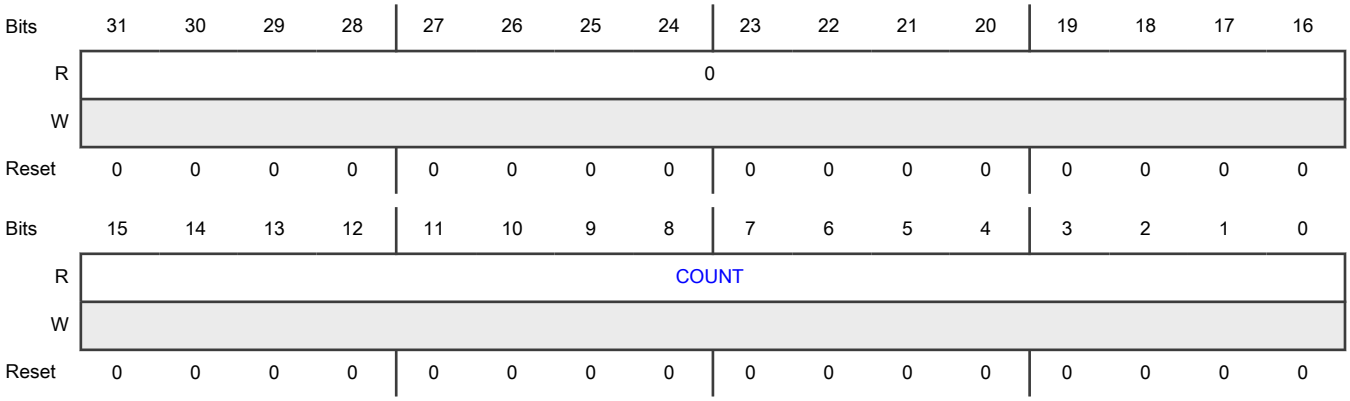
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive broadcast packets

44.5.1.79 Rx Multicast Packets Statistic (RMON_R_MC_PKT)

Offset

Register	Offset
RMON_R_MC_PKT	28Ch

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive multicast packets

44.5.1.80 Rx Packets with CRC/Align Error Statistic (RMON_R_CRC_ALIGN)

Offset

Register	Offset
RMON_R_CRC_ALIGN	290h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive packets with CRC or align error

44.5.1.81 Rx Packets with Less Than 64 Bytes and Good CRC Statistic (RMON_R_UNDERSIZE)

Offset

Register	Offset
RMON_R_UNDERSIZE	294h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

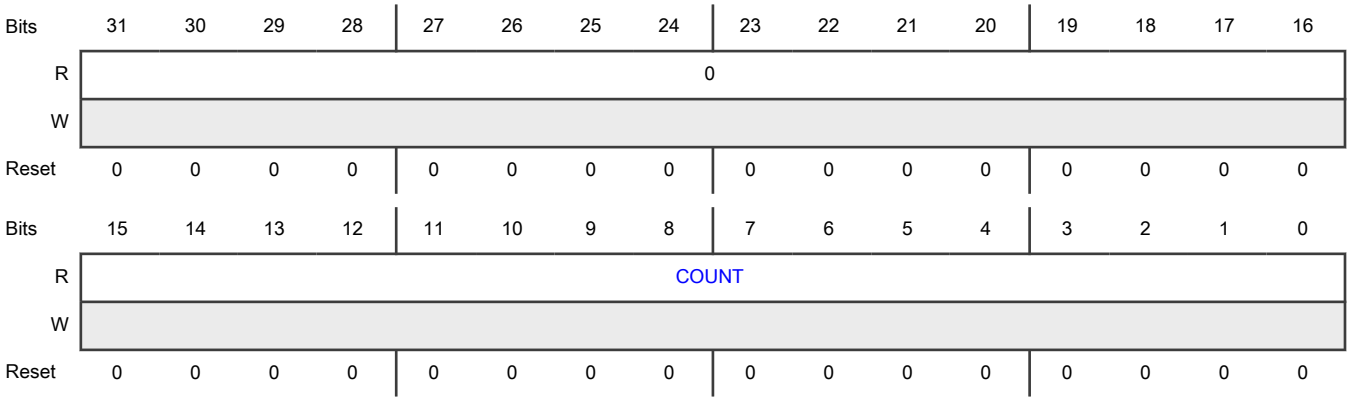
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive packets with less than 64 bytes and good CRC

44.5.1.82 Rx Packets Greater Than MAX_FL and Good CRC Statistic (RMON_R_OVERSIZE)

Offset

Register	Offset
RMON_R_OVERSIZE	298h

Diagram



Fields

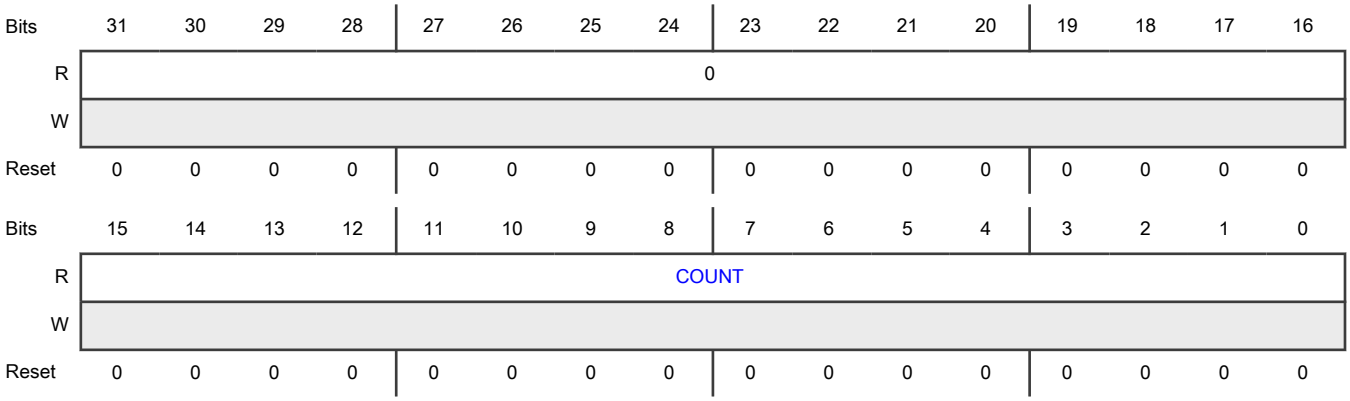
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive packets greater than MAX_FL and good CRC

44.5.1.83 Rx Packets Less Than 64 Bytes and Bad CRC Statistic (RMON_R_FRAG)

Offset

Register	Offset
RMON_R_FRAG	29Ch

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive packets with less than 64 bytes and bad CRC

44.5.1.84 Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic (RMON_R_JAB)

Offset

Register	Offset
RMON_R_JAB	2A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive packets greater than MAX_FL and bad CRC

44.5.1.85 Rx 64-Byte Packets Statistic (RMON_R_P64)

Offset

Register	Offset
RMON_R_P64	2A8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

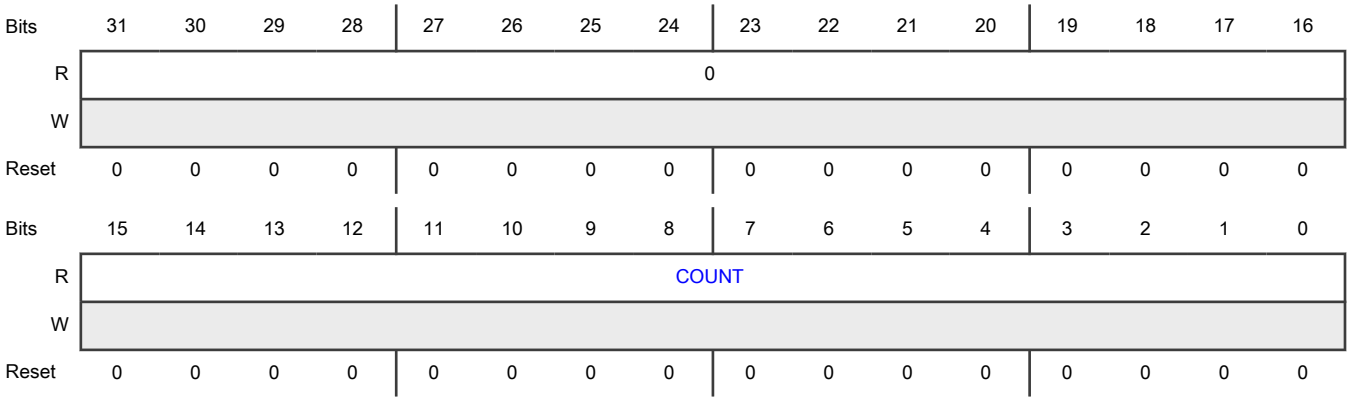
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of 64-byte receive packets

44.5.1.86 Rx 65- to 127-Byte Packets Statistic (RMON_R_P65TO127)

Offset

Register	Offset
RMON_R_P65TO127	2ACh

Diagram



Fields

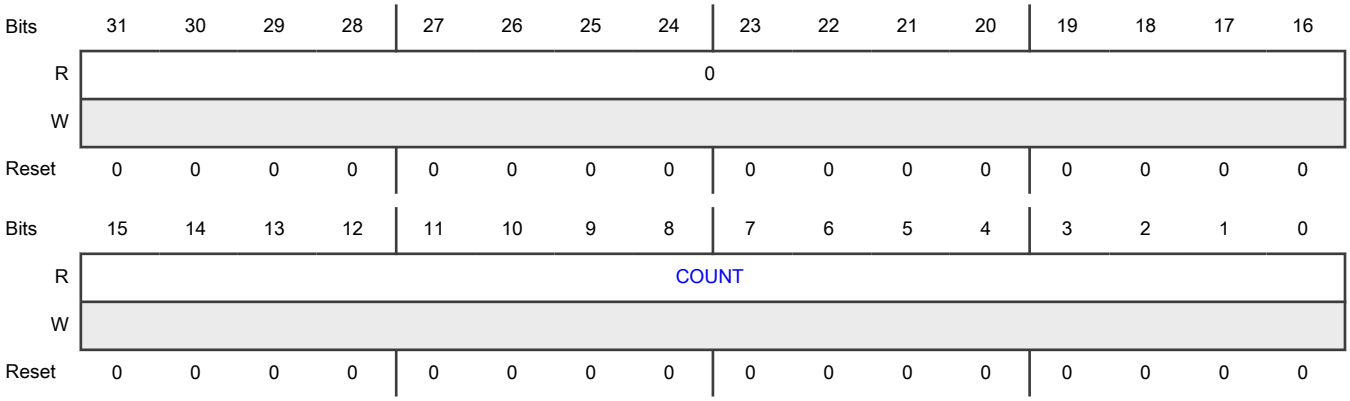
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of 65- to 127-byte receive packets

44.5.1.87 Rx 128- to 255-Byte Packets Statistic (RMON_R_P128TO255)

Offset

Register	Offset
RMON_R_P128TO255	2B0h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of 128- to 255-byte receive packets

44.5.1.88 Rx 256- to 511-Byte Packets Statistic (RMON_R_P256TO511)

Offset

Register	Offset
RMON_R_P256TO511	2B4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of 256- to 511-byte receive packets

44.5.1.89 Rx 512- to 1023-Byte Packets Statistic (RMON_R_P512TO1023)

Offset

Register	Offset
RMON_R_P512TO1023	2B8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

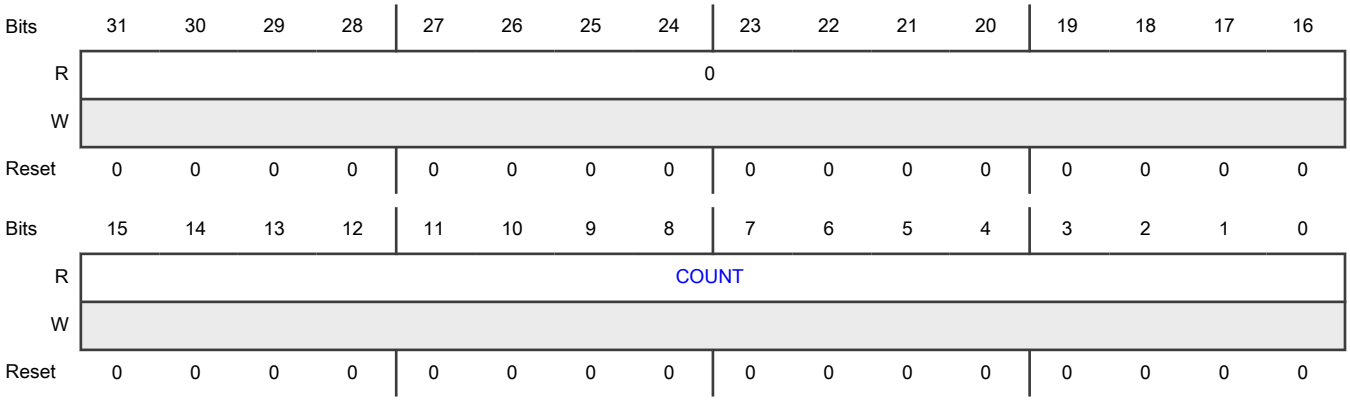
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of 512- to 1023-byte receive packets

44.5.1.90 Rx 1024- to 2047-Byte Packets Statistic (RMON_R_P1024TO2047)

Offset

Register	Offset
RMON_R_P1024TO2047	2BCh

Diagram



Fields

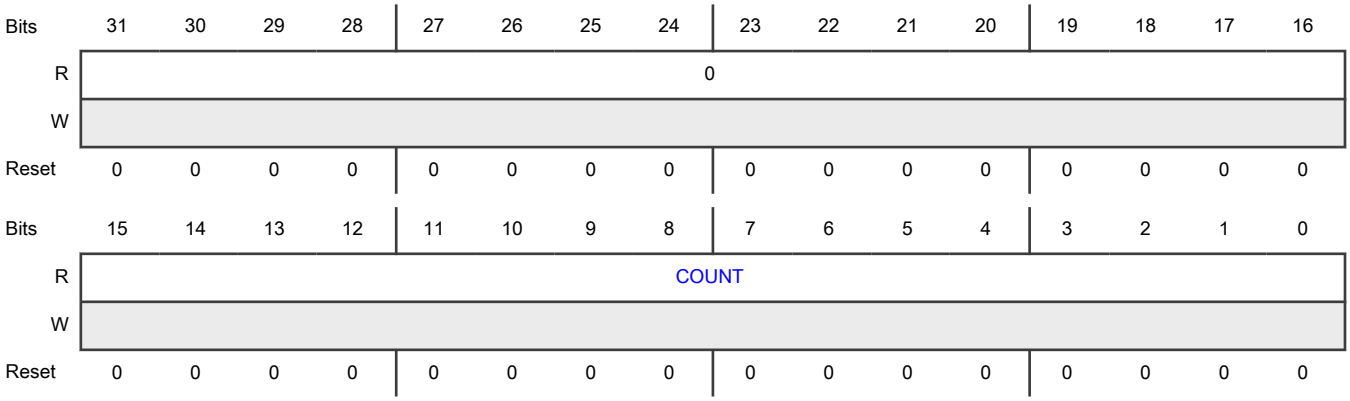
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of 1024- to 2047-byte receive packets

44.5.1.91 Rx Packets Greater than 2048 Bytes Statistic (RMON_R_P_GTE2048)

Offset

Register	Offset
RMON_R_P_GTE2048	2C0h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of greater-than-2048-byte receive packets

44.5.1.92 Rx Octets Statistic (RMON_R_OCTETS)

Offset

Register	Offset
RMON_R_OCTETS	2C4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 COUNT	Number of receive octets

44.5.1.93 Frames not Counted Correctly Statistic (IEEE_R_DROP)

Counter increments if a frame with invalid or missing SFD character is detected and has been dropped. None of the other counters increments if this counter increments.

Offset

Register	Offset
IEEE_R_DROP	2C8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

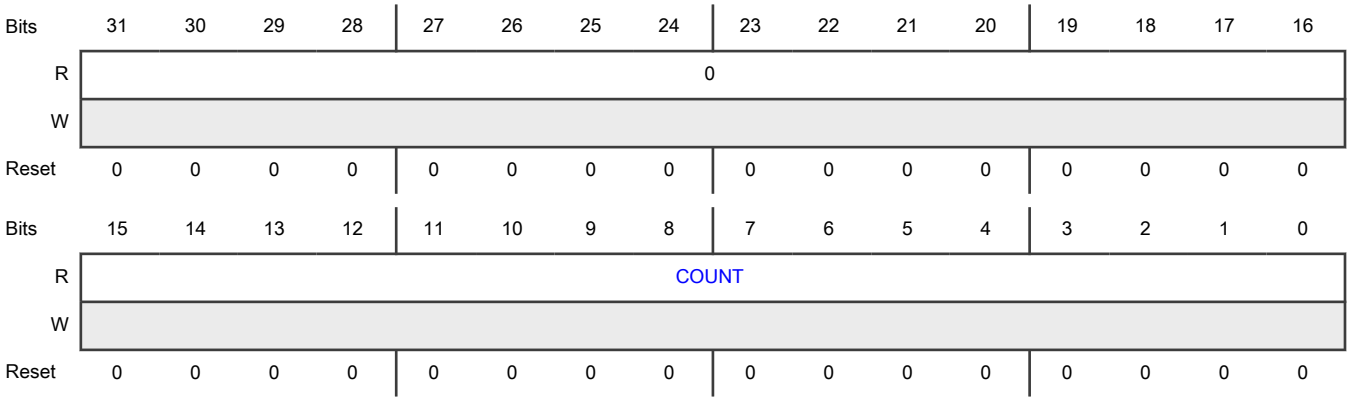
Field	Description
31-16 —	Reserved
15-0 COUNT	Frame count

44.5.1.94 Frames Received OK Statistic (IEEE_R_FRAME_OK)

Offset

Register	Offset
IEEE_R_FRAME_OK	2CCh

Diagram



Fields

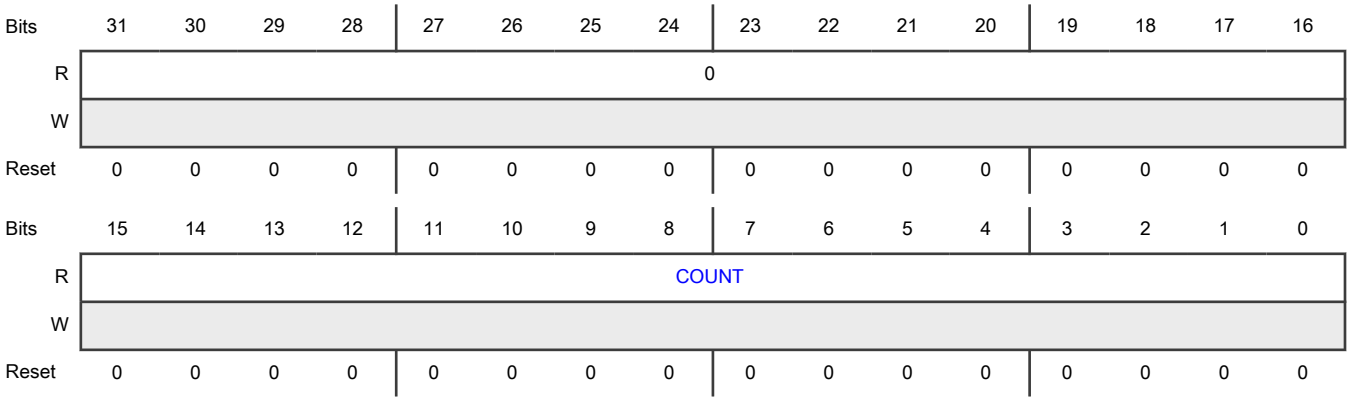
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames received OK

44.5.1.95 Frames Received with CRC Error Statistic (IEEE_R_CRC)

Offset

Register	Offset
IEEE_R_CRC	2D0h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames received with CRC error

44.5.1.96 Frames Received with Alignment Error Statistic (IEEE_R_ALIGN)

Offset

Register	Offset
IEEE_R_ALIGN	2D4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames received with alignment error

44.5.1.97 Receive FIFO Overflow Count Statistic (IEEE_R_MACERR)

Offset

Register	Offset
IEEE_R_MACERR	2D8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

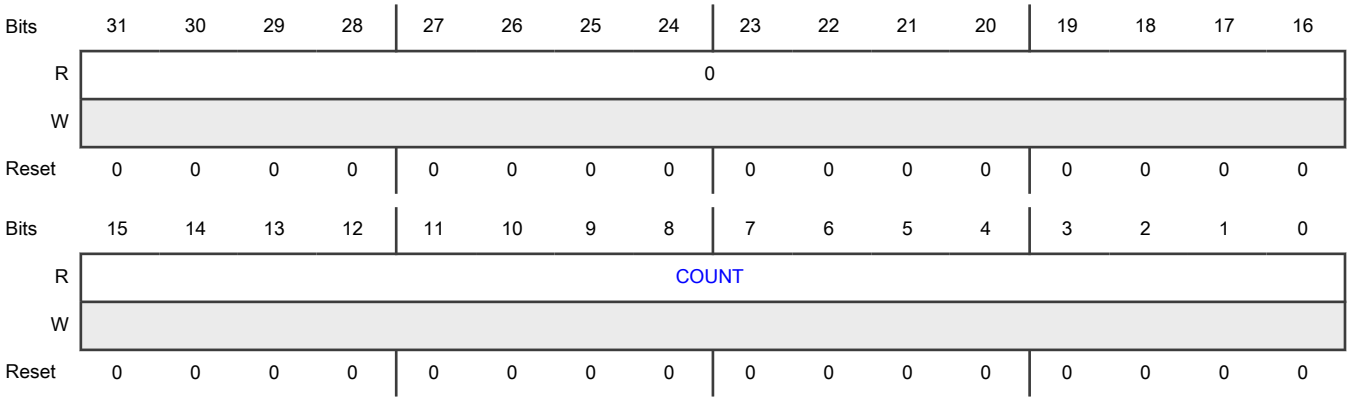
Field	Description
31-16 —	Reserved
15-0 COUNT	Receive FIFO overflow count

44.5.1.98 Flow Control Pause Frames Received Statistic (IEEE_R_FDXFC)

Offset

Register	Offset
IEEE_R_FDXFC	2DCh

Diagram



Fields

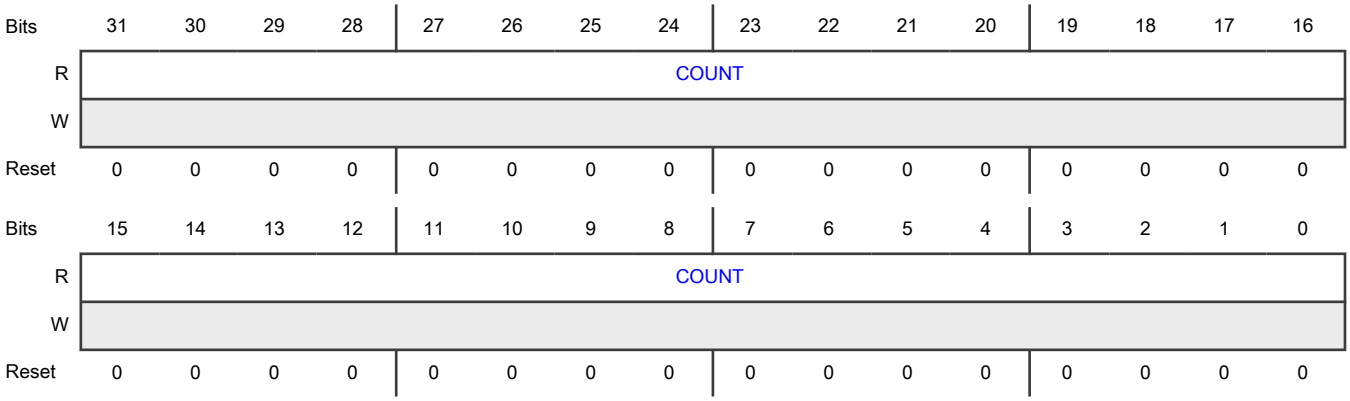
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of flow-control pause frames received

44.5.1.99 Octet Count for Frames Received without Error Statistic (IEEE_R_OCTETS_OK)

Offset

Register	Offset
IEEE_R_OCTETS_OK	2E0h

Diagram



Fields

Field	Description
31-0 COUNT	Number of octets for frames received without error **NOTE** Counts total octets (includes header and FCS fields). Does not increment for the broadcast frames when broadcast reject is enabled and promiscuous mode is disabled within the receive control register (RCR).

44.5.1.100 Adjustable Timer Control (ATCR)

ATCR command fields can trigger the corresponding events directly. It is not necessary to preserve any of the configuration fields when a command field is set in the register, that is, no read-modify-write is required.

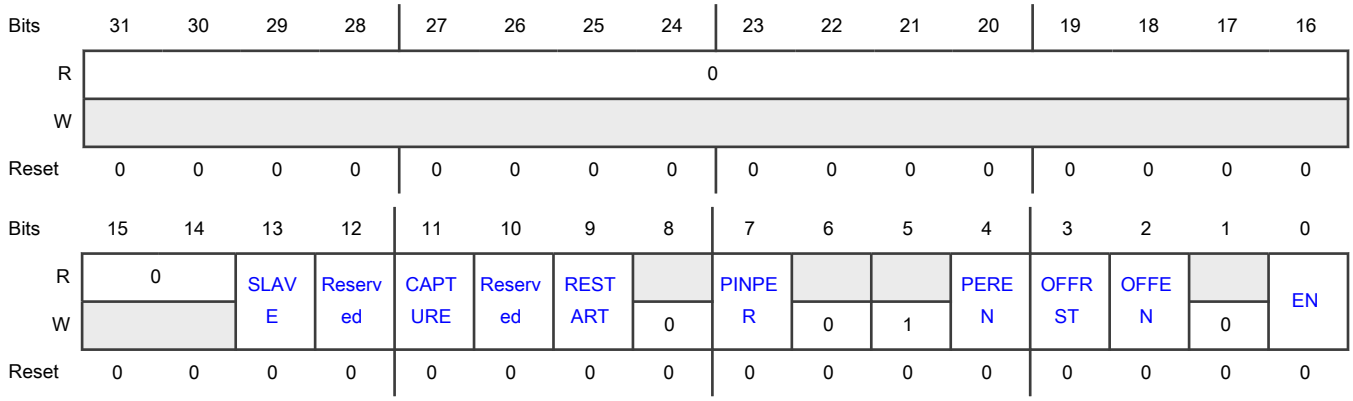
NOTE

The CAPTURE and RESTART fields and bits 12 and 10 must be 0 in order to write to the other fields in this register.

Offset

Register	Offset
ATCR	400h

Diagram



Fields

Field	Description
31-14 —	Reserved
13 SLAVE	Enable Timer Follower Mode 0 - The timer is active and all configuration fields in this register are relevant. 1 - The internal timer is disabled and the externally provided timer value is used. All other fields, except CAPTURE, in this register have no effect. CAPTURE can still be used to capture the current timer value.
12 —	Always write 0 to this field Always write 0 to this field.
11 CAPTURE	Capture Timer Value When this field is set, all other fields are ignored during a write. This field automatically clears to 0 after the command completes. **NOTE** **WARNING:** To ensure that the correct time value is read from the ATVR register, a minimum amount of time must elapse from issuing this command to reading the ATVR register. This minimum time is defined by the greater of either six register clock cycles or six 1588/timestamp clock cycles. 0 - No effect. 1 - The current time is captured and can be read from the ATVR register.
10 —	Always write 0 to this field Always write 0 to this field.
9 RESTART	Reset Timer Resets the timer to zero. This has no effect on the counter enable. If the counter is enabled when this field is set, the timer is reset to zero and starts counting from there. When set, all other fields are ignored

Table continues on the next page...

Table continued from the previous page...

Field	Description
	during a write. This field automatically clears to 0 after the command completes. RESTART should be used when the timer is enabled. **NOTE** **WARNING:** The Reset Timer command requires at least 6 clock cycles of either the register clock or the 1588/timestamp clock, whichever is greater, to complete.
8 —	Reserved
7 PINPER	Enables event signal output external pin frc_evt_period assertion on period event Enables event signal output assertion on period event. **NOTE** Not all devices contain the event signal output. See the chip configuration details. 0 - Disable. 1 - Enable.
6 —	Reserved
5 —	This field must be written always with one **NOTE** This field must be written always with one.
4 PEREN	Enable Periodical Event 0 - Disable. 1 - A period event interrupt can be generated (EIR[TS_TIMER]) and the event signal output is asserted when the timer wraps around according to the periodic setting ATPER. The timer period value must be set before setting this bit. Not all devices contain the event signal output. See the chip configuration details.
3 OFFRST	Reset Timer On Offset Event 0 - The timer is not affected and no action occurs, besides clearing OFFEN, when the offset is reached. 1 - If OFFEN is set, the timer resets to zero when the offset setting is reached. The offset event does not cause a timer interrupt.
2 OFFEN	Enable One-Shot Offset Event 0 - Disable. 1 - The timer can be reset to zero when the given offset time is reached (offset event). The field is cleared when the offset event is reached, so no further event occurs until the field is set again. The timer offset value must be set before setting this field.
1	Reserved

Table continues on the next page...

Table continued from the previous page...

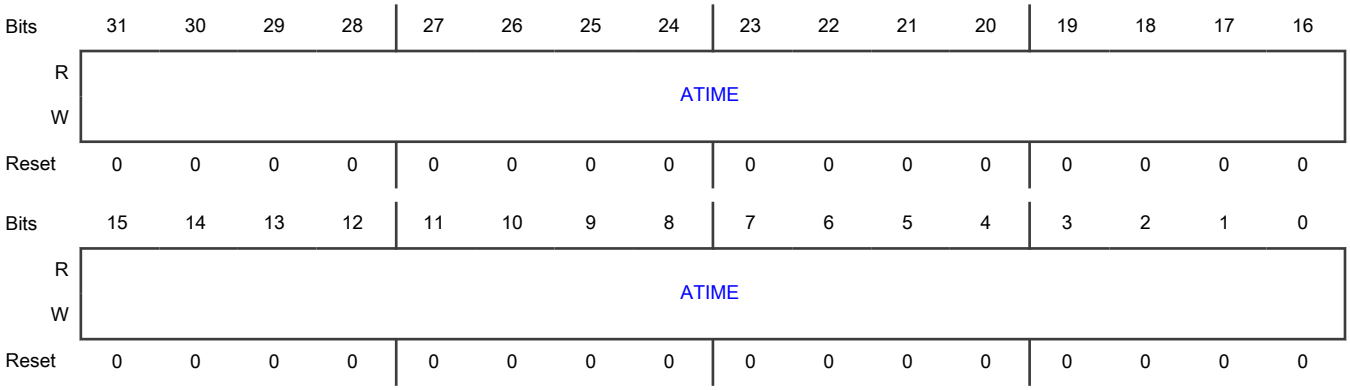
Field	Description
—	
0 EN	Enable Timer 0 - The timer stops at the current value. 1 - The timer starts incrementing.

44.5.1.101 Timer Value (ATVR)

Offset

Register	Offset
ATVR	404h

Diagram



Fields

Field	Description
31-0 ATIME	A write sets the timer. A read returns the last captured value. To read the current value, issue a capture command (i.e., set ATCR[CAPTURE]) prior to reading this register.

44.5.1.102 Timer Offset (ATOFF)

Offset

Register	Offset
ATOFF	408h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OFFSET	Offset value for one-shot event generation. When the timer reaches the value, an event can be generated to reset the counter. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds.

44.5.1.103 Timer Period (ATPER)

Offset

Register	Offset
ATPER	40Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PERIOD															
Reset	0	0	1	1	1	0	1	1	1	0	0	1	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERIOD															
Reset	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0	Value for generating periodic events

Table continues on the next page...

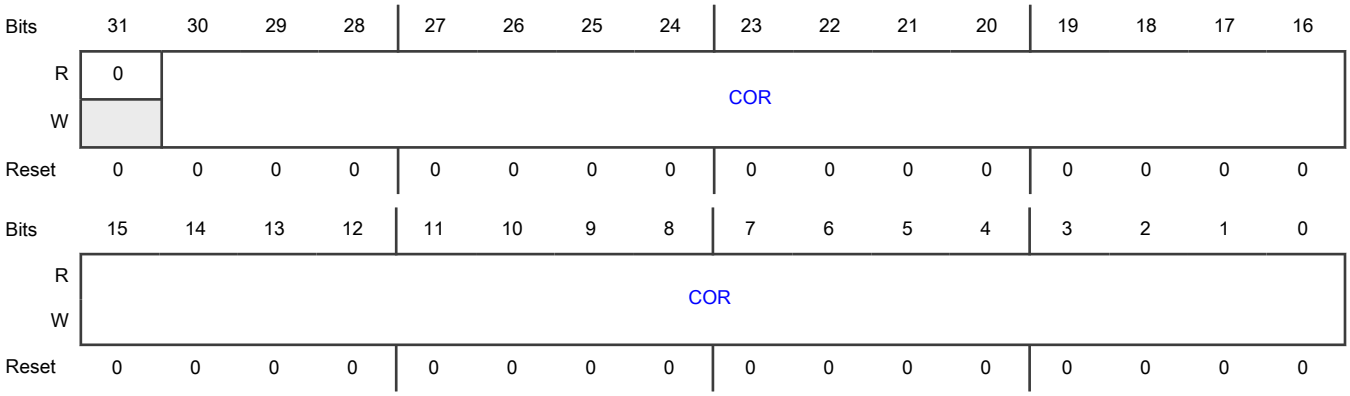
Field	Description
PERIOD	Value for generating periodic events. Each instance the timer reaches this value, the period event occurs and the timer restarts. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds. The value should be initialized to 1,000,000,000 (1×10^9) to represent a timer wrap around of one second. The increment value set in ATINC should be set to the true nanoseconds of the period of clock ts_clk, hence implementing a true 1 second counter. **NOTE** The value of PERIOD has the following constraint: 2^{32} − ENET_ATINC[INC_COR] − 3×ENET_ATINC[INC] ≥ PERIOD > 0.

44.5.1.104 Timer Correction (ATCOR)

Offset

Register	Offset
ATCOR	410h

Diagram



Fields

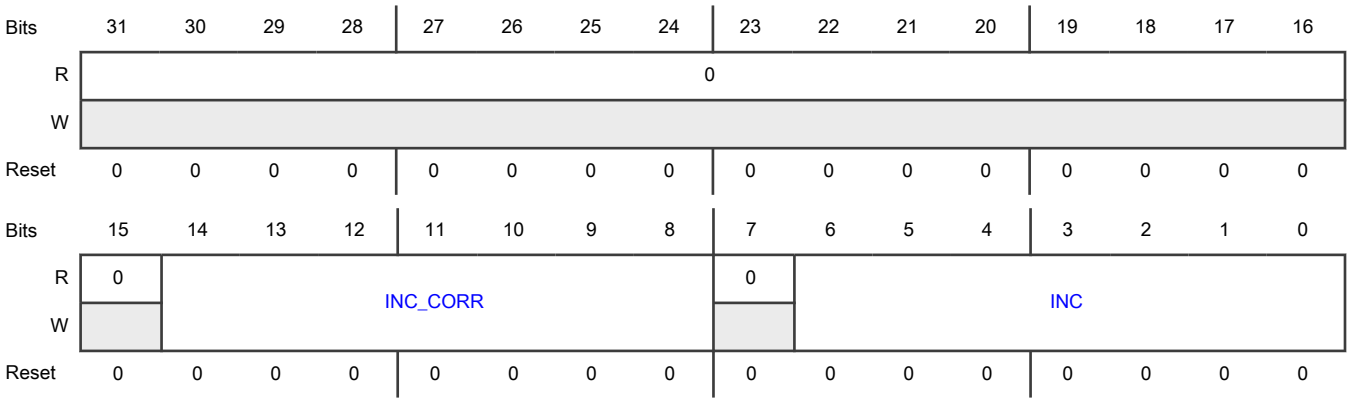
Field	Description
31 —	Reserved
30-0 COR	Correction Counter Wrap-Around Value Defines after how many timer clock cycles (ts_clk) the correction counter should be reset and trigger a correction increment on the timer. The amount of correction is defined in ATINC[INC_CORR]. A value of 0 disables the correction counter and no corrections occur. **NOTE** This value is given in clock cycles, not in nanoseconds as all other values.

44.5.1.105 Time-Stamping Clock Period (ATINC)

Offset

Register	Offset
ATINC	414h

Diagram



Fields

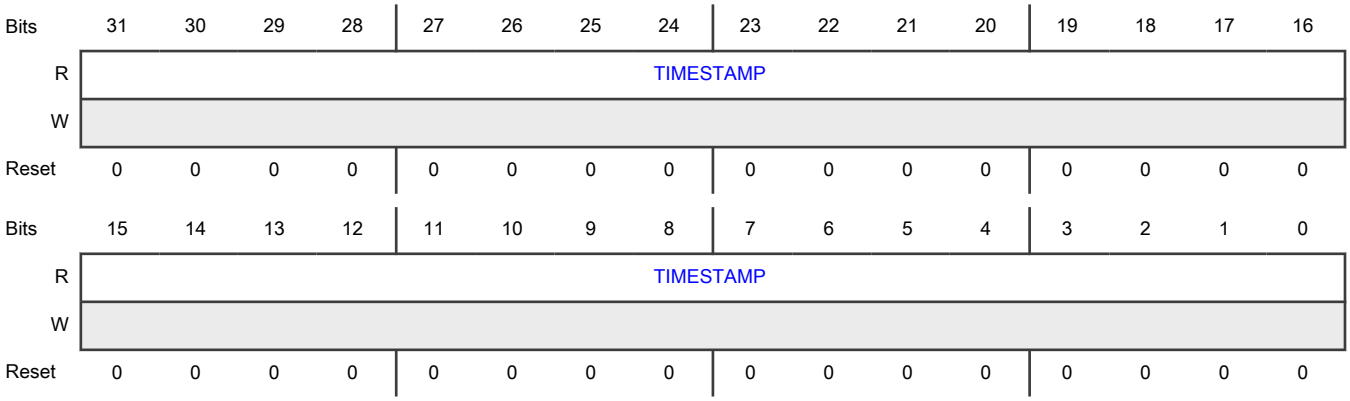
Field	Description
31-15 —	Reserved
14-8 INC_CORR	Correction Increment Value This value is added every time the correction timer expires (every clock cycle given in ATCOR). A value less than INC slows down the timer. A value greater than INC speeds up the timer.
7 —	Reserved
6-0 INC	Clock Period Of The Timestamping Clock (ts_clk) In Nanoseconds The timer increments by this amount each clock cycle. For example, set to 10 for 100 MHz, 8 for 125 MHz, 5 for 200 MHz. **NOTE** For highest precision, use a value that is an integer fraction of the period set in ATPER.

44.5.1.106 Timestamp of Last Transmitted Frame (ATSTMP)

Offset

Register	Offset
ATSTMP	418h

Diagram



Fields

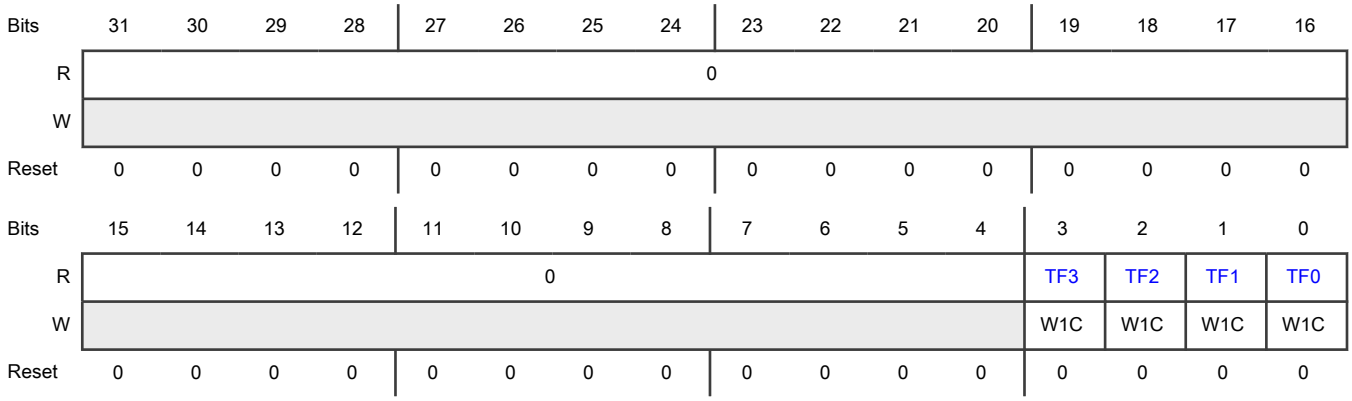
Field	Description
31-0 TIMESTAMP	Timestamp of the last frame transmitted by the core that had TxBD[TS] set the ff_tx_ts_frm signal asserted from the user application Timestamp of the last frame transmitted by the core that had TxBD[TS] set . This register is only valid when EIR[TS_AVAIL] is set.

44.5.1.107 Timer Global Status (TGSR)

Offset

Register	Offset
TGSR	604h

Diagram



Fields

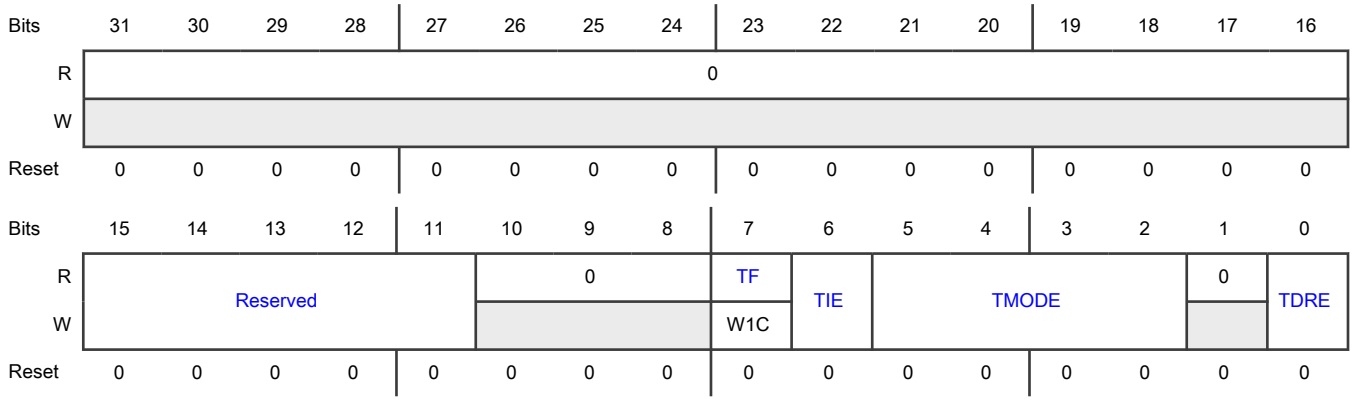
Field	Description
31-4 —	Reserved
3 TF3	Copy Of Timer Flag For Channel 3 0 - Timer Flag for Channel 3 is clear 1 - Timer Flag for Channel 3 is set
2 TF2	Copy Of Timer Flag For Channel 2 0 - Timer Flag for Channel 2 is clear 1 - Timer Flag for Channel 2 is set
1 TF1	Copy Of Timer Flag For Channel 1 0 - Timer Flag for Channel 1 is clear 1 - Timer Flag for Channel 1 is set
0 TF0	Copy Of Timer Flag For Channel 0 0 - Timer Flag for Channel 0 is clear 1 - Timer Flag for Channel 0 is set

44.5.1.108 Timer Control Status (TCSR0 - TCSR3)

Offset

Register	Offset
TCSR0	608h
TCSR1	610h
TCSR2	618h
TCSR3	620h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-11 —	This field must be written with 0.
10-8 —	Reserved
7 TF	Timer Flag Sets when input capture or output compare occurs. This flag is double buffered between the module clock and 1588 clock domains. When this field is 1, it can be cleared to 0 by writing 1 to it. 0 - Input Capture or Output Compare has not occurred. 1 - Input Capture or Output Compare has occurred.
6 TIE	Timer Interrupt Enable 0 - Interrupt is disabled 1 - Interrupt is enabled
5-2 TMODE	Timer Mode Updating the Timer Mode field takes a few cycles to register because it is synchronized to the 1588 clock. The version of Timer Mode returned on a read is from the 1588 clock domain. When changing Timer Mode, always disable the channel and read this register to verify the channel is disabled first. 0000 - Timer Channel is disabled. 0001 - Timer Channel is configured for Input Capture on rising edge. 0010 - Timer Channel is configured for Input Capture on falling edge. 0011 - Timer Channel is configured for Input Capture on both edges. 0100 - Timer Channel is configured for Output Compare - software only.

Table continues on the next page...

Table continued from the previous page...

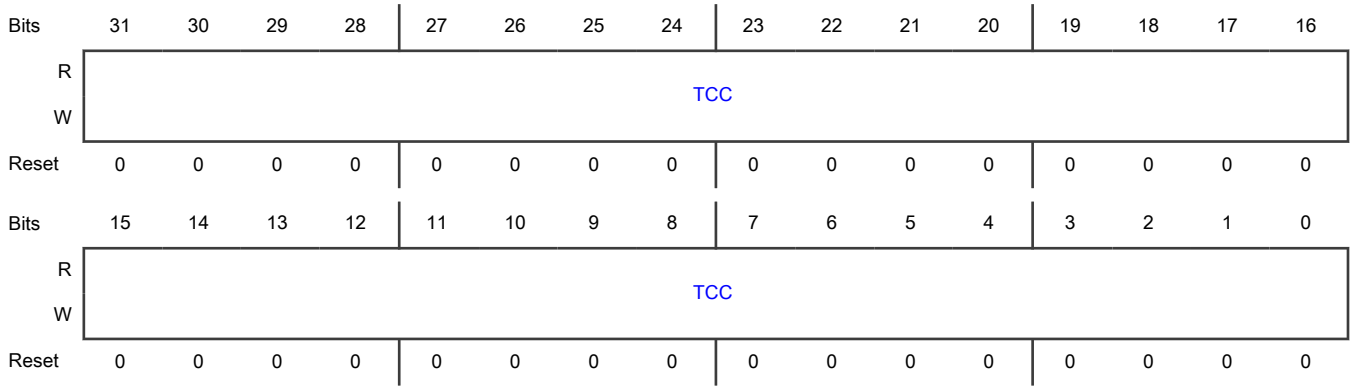
Field	Description
	0101 - Timer Channel is configured for Output Compare - toggle output on compare. 0110 - Timer Channel is configured for Output Compare - clear output on compare. 0111 - Timer Channel is configured for Output Compare - set output on compare. 1000 - Reserved 1010 - Timer Channel is configured for Output Compare - clear output on compare, set output on overflow. 10x1 - Timer Channel is configured for Output Compare - set output on compare, clear output on overflow. 110x - Reserved 1110 - Timer Channel is configured for Output Compare - pulse output low on compare for one 1588-clock cycle. 1111 - Timer Channel is configured for Output Compare - pulse output high on compare for one 1588-clock cycle.
1 —	Reserved
0 TDRE	Timer DMA Request Enable 0 - DMA request is disabled 1 - DMA request is enabled

44.5.1.109 Timer Compare Capture (TCCR0 - TCCR3)

Offset

Register	Offset
TCCR0	60Ch
TCCR1	614h
TCCR2	61Ch
TCCR3	624h

Diagram



Fields

Field	Description
31-0 TCC	Timer Capture Compare This register is double buffered between the module clock and 1588 clock domains. When configured for compare, the 1588 clock domain updates with the value in the module clock domain whenever the Timer Channel is first enabled and on each subsequent compare. Write to this register with the first compare value before enabling the Timer Channel. When the Timer Channel is enabled, write the second compare value either immediately, or at least before the first compare occurs. After each compare, write the next compare value before the previous compare occurs and before clearing the Timer Flag. The compare occurs one 1588 clock cycle after the IEEE 1588 Counter increments past the compare value in the 1588 clock domain. If the compare value is less than the value of the 1588 Counter when the Timer Channel is first enabled, then the compare does not occur until following the next overflow of the 1588 Counter. If the compare value is greater than the IEEE 1588 Counter when the 1588 Counter overflows, or the compare value is less than the value of the IEEE 1588 Counter after the overflow, then the compare occurs one 1588 clock cycle following the overflow. When configured for capture, the value of the IEEE 1588 Counter is captured into the 1588 clock domain and then updated into the module clock domain, provided the Timer Flag is clear. Always read the capture value before clearing the Timer Flag.

Chapter 45
Audio Overview

45.1 Audio Overview

The following audio peripherals are included on this chip.

- Synchronous Audio Interface (SAI)

  The SAI are synchronous serial interfaces used to transfer audio data.

- Pulse Density Modulated Microphone Interface (PDM)

  The PDM Microphone Interface (PDM) is used to receive audio from microphone.

- Sony/Philips digital interface (SPDIF XCVR)

  The audio module is supported through the Audio Transceiver.

- Medium Quality Sound (MQS)

  MQS is used to generate medium quality audio to connect stereo speakers or headphones to a power amplifier.

45.1.1 Audio Interfaces

The audio interfaces are summarized below.

Table 476. Audio Interface Summary

Interface	MIx	Function	RX Lanes	TX Lanes	FIFO size
SAI-1	AONMIX	External audio interface	1	2	16-word TX/RX
SAI-2	WAKEUPMIX	External audio interface	4	4	128 entries per lane
SAI-3	WAKEUPMIX	External audio interface	1	1	128 entries per lane
SPDIF	WAKEUPMIX	External audio interface	1	1	128 Rx and 128 Tx entries
PDM (8 channel)	WAKEUPMIX	External microphone interface	-	4 (8 MIC's)	32 entries
MQS1	AONMIX	Medium quality Audio interface	-	-	-
MQS2	WAKEUPMIX	Medium quality Audio interface	-	-	-

45.1.2 Synchronous Audio Interface (SAI)

- SAI1 supports 2 lanes TX and 1 lane RX.
- SAI2 supports 4 lanes TX and RX.
- SAI3 supports 1 lane TX and RX.
- 16-word TX/RX FIFO and up to 32 words per frame
- One set of Tx clocks and frame sync

45.1.3 Sony Philips Digital Interface (SPDIF) Transceiver

It supports raw capture mode that can save all the digital audio bits into audio buffer.

45.1.4 Pulse Density Modulated Microphone Interface (PDM)

The PDM module receives audio from a microphone. It includes the following features:

- Supports up to 8-microphones (4 lanes)
- Supports frame synchronization

45.1.5 Medium Quality Sound (MQS)

MQS is used to generate medium quality audio via a standard GPIO. It provides only simple audio reproduction. Stereo speakers or headphones can be connected to a power amplifier without an additional DAC chip.

- 2-channel, LSB-valid 16 bit, MSB shift-out first serial data (sdata)
- Signals over 10 kHz have worse THD+N values

Chapter 46
PDM Microphone Interface (PDM)

46.1 Chip-specific PDM information

Table 477. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

This chip includes one instances of PDM in AONMIX.

PDM is clocked by a PDM_CLK_ROOT clock from the CCM or SAI2_MCLK.

46.2 Overview

MICFIL delivers audio from microphones to the processor in several applications, such as mobile telephones. As current digital audio systems use a multi-bit audio signal (also known as multi-bit PCM) to represent the signal, this module implements the required digital interface (a series of filters) to transform a pulse density modulated (PDM) microphone bitstream into a 24-bit PCM signal in the audio band, at a configurable output sampling rate.

The implementation of this digital interface is based on the application of digital signal processing techniques used in hardware. The MICFIL architecture is designed to save gate count and minimize power consumption.

You can implement MICFIL to work in a multi-channel mode. All channels have the same configuration, but you could independently turn on or turn off each input channel.

46.2.1 Block diagram

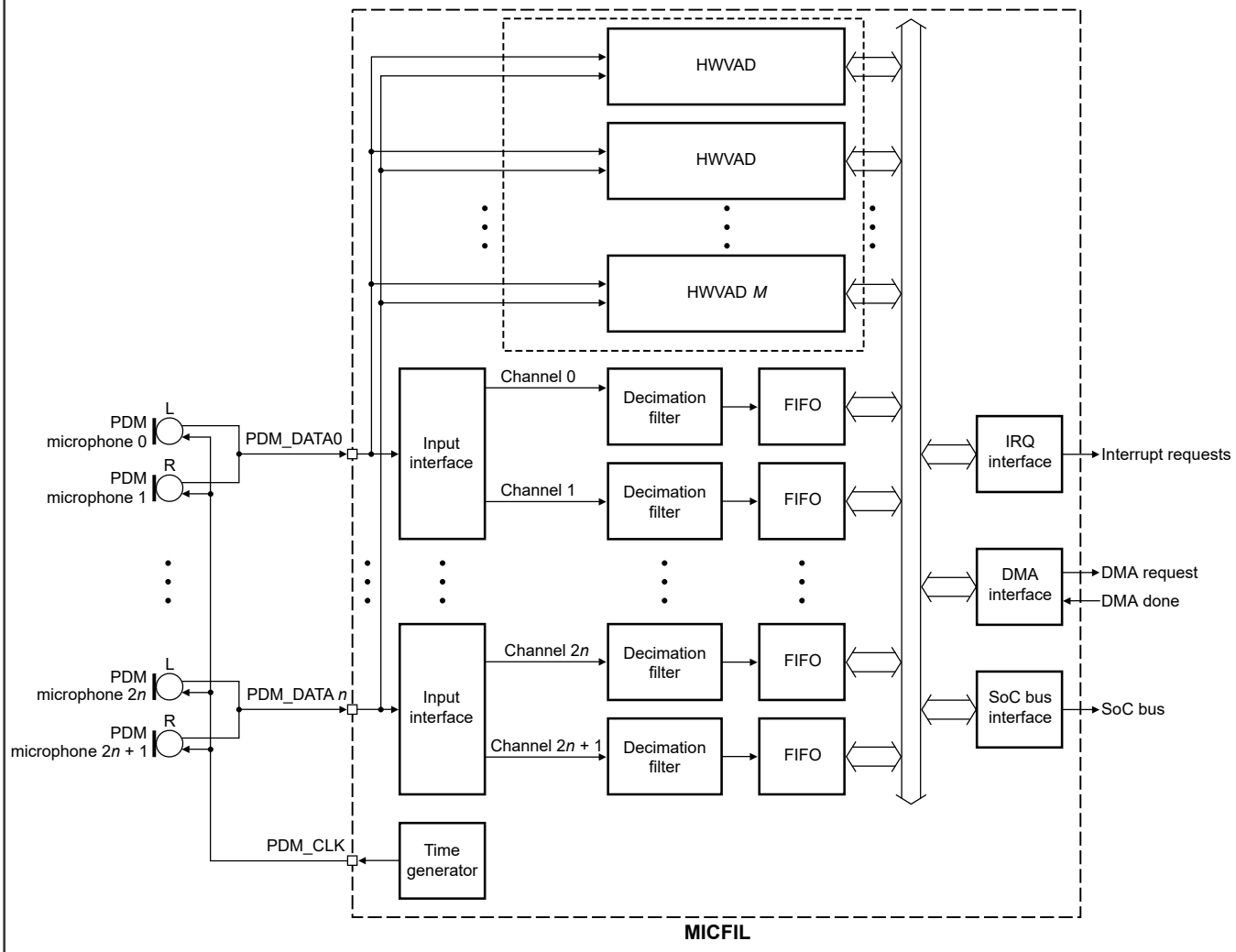


Figure 231. MICFIL block diagram

- HWVAD = hardware voice activity detector

46.2.2 Features

- Decimation filters:
  - Fixed-point filtering
  - 24-bit PCM audio output
  - Internal clock divider for a programmable PDM clock generation:
  - Frame synchronization
  - Full or partial set of channel operations with individual enable controls
  - Programmable decimation rate
  - Programmable DC remover
  - Programmable DC remover at output
  - Range adjustment capability

- FIFOs with interrupt and DMA capability: each FIFO having a length of 32 entries
- Hardware voice activity detector (HWVAD), equipped with:
  - Interrupt capability
  - Zero-crossing detection (ZCD) option

46.3 Functional description

46.3.1 Time generator

The time generator unit:

- Generates the PDM_CLK to microphones. This clock is the same and is active for all the PDM microphones, which means, it is not possible to turn off the PDM_CLK only for one single microphone.
- Delivers the PDM_CLK to all microphones that must operate at the same clock frequency. Each input interface receives a time multiplexed PDM bitstream from two PDM microphones and it separates audio information in two channels: left (0) and right (1). Every decimation filter, corresponding to its channel, does this processing.

NOTE

A soft reset affects the time generator unit (see CTRL_1[SRES]). The PDM_CLK starts with a rising edge after a soft reset is issued.

46.3.1.1 Clock divider

The sampling rate of the microphone input depends on the PDM_CLK rate, which you can trim using an internal clock divider, configuring the quality mode (see Quality modes) or trimming the MICFIL's root clock frequency (MICFIL_CLK_ROOT rate).

The internal clock divider frequency. You must configure CTRL_2[CLKDIV] by using Equation 39.

$$CLKDIV = \frac{MICFIL_CLK_ROOT\ rate}{8 \times OSR \times (output\ rate)}$$

Equation 39. CLKDIV value calculation

After calculating the value of CTRL_2[CLKDIV], you can estimate the generated PDM_CLK rate based on Equation 40, where the "K" factor value depends on the quality mode shown in Table 478 and floor(x) is the function that takes a real number x as input and gives the greatest integer less than or equal to x as output.

$$PDM_CLK\ rate = \frac{MICFIL_CLK_ROOT}{2 \times floor(K \times CLKDIV)}$$

Equation 40. PDM_CLK rate calculation

Table 478. K factor value

Quality mode	K factor
High Quality	1/2
Medium Quality, Very-Low Quality 0	1
Low Quality, Very-Low Quality 1	2
Very-Low Quality 2	4

46.3.1.1.1 Minimum required CLKDIV

NOTE

This minimum CLKDIV condition is required only if any decimation filter channel is enabled and the clock divider is enabled, otherwise this condition should be ignored.

To work properly the CLKDIV value must have a minimum value according to the quality mode configured (see Quality modes), the number of enabled channels (EC), OSR value and K factor (see I/O timing requirements). When this condition is not met, the LOWFREQF bit is asserted and an error interrupt can be issued (in case [ERREN] = 1).

The equations to estimate the minimum required CLKDIV value depending on the configured quality mode are shown in Equation 41 (for medium, high and low quality modes) and Equation 42 (for very low quality modes), on these equations floor(x) is the function that takes as input a real number x and gives as output the greatest integer less than or equal to x.

$$floor(K \times CLKDIV) \geq K \times \frac{10 + 93EC}{8 \times OSR}$$

Equation 41. Minimum required CLKDIV value in Medium, High and Low Quality modes

$$floor(K \times CLKDIV) \geq K \times \frac{10 + 43EC}{8 \times OSR}$$

Equation 42. Minimum required CLKDIV value in Very Low Quality modes

46.3.1.2 MICFIL_CLK_ROOT rate and CLKDIV calculation examples

Example 1:

If it is required an output rate = 16 kHz and enables 4 channels (EC=4). A valid configuration is following:

1. Configure High Quality mode and CIC decimation by 32 (CTRL_2[CICOSR] = 0, then OSR = 16).
2. Check the PDM_CLK rate for High Quality mode in Table 482. PDM_CLK rate = 16 kHz x 8 x OSR = 16 kHz x 8 x 16 = 2.048 MHz.
3. Check the K factor value for High Quality mode in Table 478. K factor is 1/2.
4. Calculate minimum CLKDIV value (see Equation 41): floor(K x CLKDIV) ≥ K x (10 + 93EC)/(8 x OSR) then floor(K x CLKDIV) ≥ 1.4921875. Finally, CLKDIV ≥ 4.
5. Calculate the minimum MICFIL_CLK_ROOT rate:
   - From Equation 39: MICFIL_CLK_ROOT rate = 8 x OSR x CLKDIV x (Output rate)
   - From step 4 result: CLKDIV ≥ 4, then MICFIL_CLK_ROOT rate ≥ 8 x OSR x 4 x 16 kHz.
   - Finally: MICFIL_CLK_ROOT rate ≥ 8.192 MHz
6. Check if selected MICFIL_CLK_ROOT rate and CLKDIV value satisfies the IO timing requirements.
7. Finally, check again if the selected MICFIL_CLK_ROOT rate satisfies Equation 39. If not, adjust CLKDIV or change MICFIL_CLK_ROOT rate accordingly.

Example 2:

If it is required an output rate = 48 kHz and enables 4 channels (EC=4). A valid configuration is following:

1. Configure Medium Quality mode and CIC decimation by 16 (CICOSR = 0, then OSR = 16).
2. Check the PDM_CLK rate for Medium Quality mode in Table 482. PDM_CLK rate = 48 kHz x 4 x OSR = 48 kHz x 4 x 16 = 3.072 MHz.
3. Check the K factor value for Medium Quality mode in Table 478. K factor is 1.
4. Calculate minimum CLKDIV value (see Equation 41): floor(K x CLKDIV) ≥ K x (10 + 93EC)/(8 x OSR) then floor(K x CLKDIV) ≥ 2.984375. Finally, CLKDIV ≥ 3.

5. Calculate the minimum MICFIL_CLK_ROOT rate:
   - From Equation 39: MICFIL_CLK_ROOT rate = 8 x OSR x CLKDIV x (Output rate)
   - From step 4 result: CLKDIV ≥ 3, then MICFIL_CLK_ROOT rate ≥ 8 x OSR x 3 x 48 kHz.
   - Finally: MICFIL_CLK_ROOT rate ≥ 18.432 MHz
   - A different value of CLKDIV may be preferable, CLKDIV = 4 results in MICFIL_CLK_ROOT = 24.576 MHz, a standard audio frequency.
6. Check if selected MICFIL_CLK_ROOT rate and CLKDIV value satisfies the IO timing requirements.
7. Finally, check again if the selected MICFIL_CLK_ROOT rate satisfies Equation 39. If not, adjust CLKDIV or change MICFIL_CLK_ROOT rate accordingly.

46.3.1.3 I/O timing requirements

The PDM microphones must meet the setup and hold timing requirements shown in Table 479 and Figure 232. The "K" factor value in Table 479 depends on the quality mode you select based on Table 478.

NOTE

These timing requirements apply only if the clock divider functionality is enabled (see CTRL_2[CLKDIV]) with ; otherwise, there are no special timing requirements.

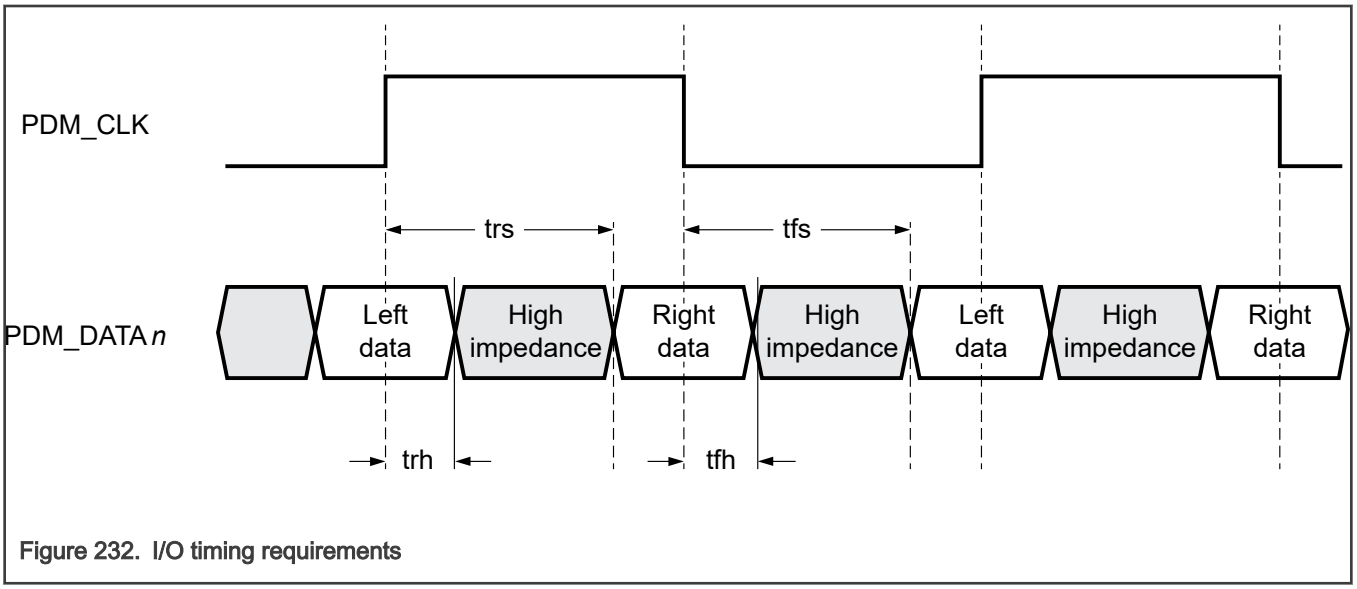


Figure 232. I/O timing requirements

Table 479. Timing parameters

Parameter	Value
trs, tfs	$\leq \frac{floor(K \times CLKDIV) - 1}{MICFIL_CLK_ROOT\ rate}$ Depending on the value of "K," you must ensure that "floor(K x CTRL_2[CLKDIV])" > 1 to avoid timing problems.
trh, tfh	≥ 0

46.3.2 Input interface

Figure 233 shows the timing diagram of the input interface signals . The bitstream incoming from the microphone data input "n" (PDM_DATA*n*) in the first half (right microphone) of the PDM_CLK is directed to channel "2n+1" and the data generated during the last half (left microphone) is directed to channel "2n".

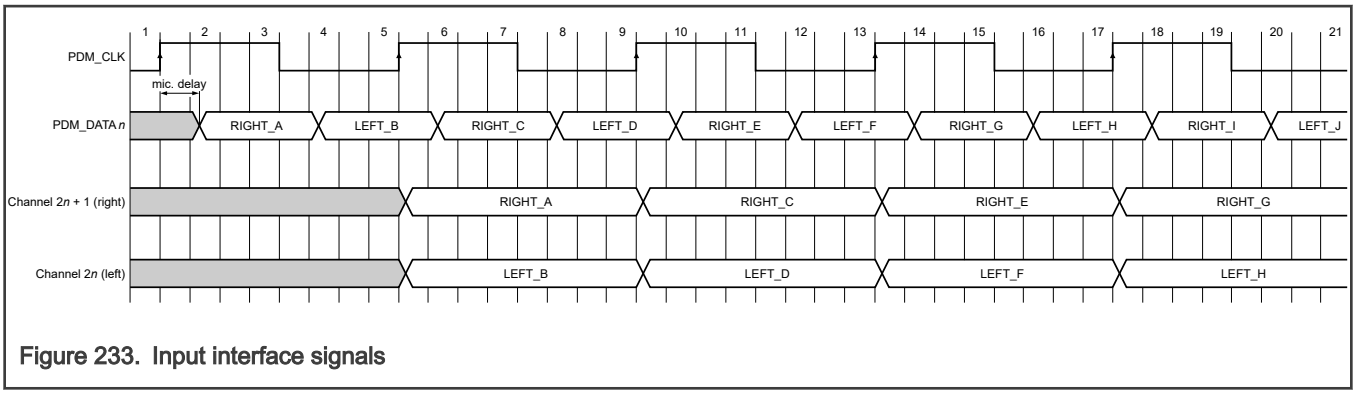


Figure 233. Input interface signals

46.3.3 Decimation filter

A decimation filter facilitates PDM to PCM conversion.

Figure 234 shows the block diagram of the decimation filter. The block consists of:

- A cascade integrator comb (CIC) filter, which is a low-pass filter that you commonly use to filter sigma-delta modulator output signals. The filter is implemented in the audio band (20 Hz–20.0 kHz @48 kHz output sampling rate by default) with a configurable decimation rate. You can implement it using a series of CIC, compensation, and DC remover filters.
- A DC remover, which is a high-pass filter that you use to remove the DC component of the processed signal with a configurable cut-off frequency. Two DC removers exist: one in the input of the decimation filter, embedded in the CIC filter, and another in the last stage of the decimation filter.
- Two compensation filters for each channel, which implement a low-pass digital filter with decimation by 2. You can use them to compensate the high CIC drop in passband. These filters help the FIR filter block to flatten the overall passband response.

You activate the "Sampler" block depending on the quality mode selected (using CTRL_2[QSEL]). In Low-Quality and Very-Low-Quality 2 modes, the sampling rate is doubled.

The decimation filter output depends on the quality mode that you select by using CTRL_2[QSEL], based on different compensation filter output. You can supply the DC remover with the 2 FS signal, instead of the 1 FS signal, to achieve lower-power consumption, bypassing the second compensation decimation filter. The second compensation filter is bypassed when CTRL_2[QSEL] = 110b, 101b, or 100b (in Very-Low Quality modes).

You can adjust the range of each decimation filter by using a range adjustment block so that you can raise an error interrupt in case an adjustment overflow or underflow is detected.

The output of a decimation filter is stored into a FIFO buffer, and each FIFO is mapped to MICFIL Output Result (DATACH0 - DATACH7). It is possible to generate either an interrupt or a DMA request when, in each FIFO of all enabled channels, the number of data stored surpasses a configured watermark.

Also, independently on decimation filters, there is a hardware voice activity detector (HWVAD) which implements voice-detection algorithms to generate wake-up interrupts. See Hardware voice activity detector (HWVAD).

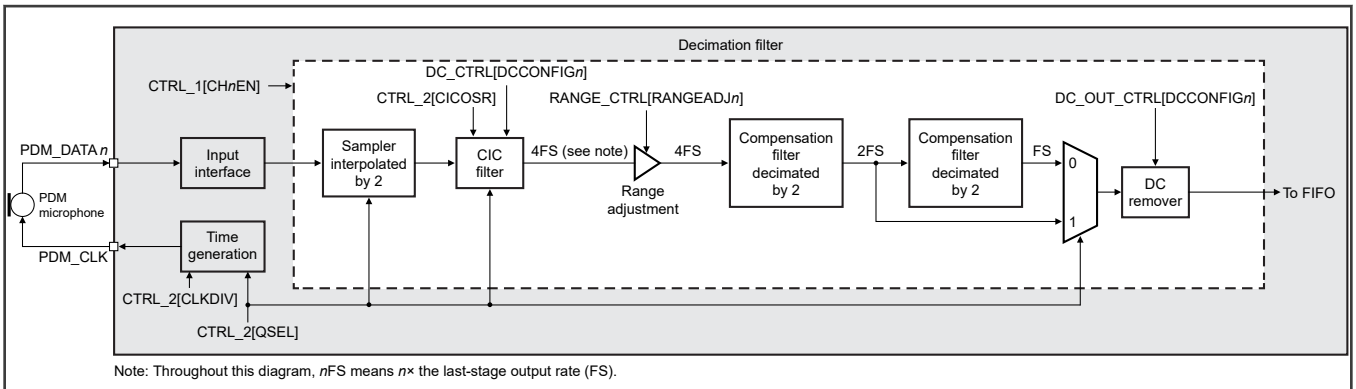


Figure 234. Decimation filter block diagram

46.3.3.1 CIC filter

The CIC filter is an optimized implementation of the low-pass SINC filter, with multiple stages. It is widely used because it requires just adders and subtracters. The CIC filter comprises an integrator and a comb as its sub-blocks. As shown in the following figure, CTRL_2[CICOSR] and the quality mode you select (see Quality modes) define the CIC decimation rate.

$$OSR = 16 - CICOSR$$

$$CIC\ decimation\ rate = \begin{cases} 2 \times OSR\,; If\ HQ,\ VLQ0 \\ OSR\quad\quad ; others \end{cases}$$

Equation 43. CIC decimation rate

46.3.3.2 DC remover

Two DC removers exist in the filter chain, one in the input and another in the output. The former is integrated in the CIC filter, and the latter is placed just before the FIFO. Both can be bypassed independently.

The DC remover eliminates the DC level component of the signal using a high-pass filter with a very low cut-off frequency that you select using DC_CTRL[DCCONFIG*n*]. This filter is bypassed when the value of this field is 11b.

NOTE

See PARAM[DC_OUT_BYPASS] and PARAM[DC_BYPASS] to know whether one of the DC removers is permanently disabled in the chip.

46.3.3.3 Compensation filters

A compensation filter divides the baseband signal into two equal subbands before it could to be decimated by 2. It also compensates the CIC filter drop in passband and helps to flatten the overall passband response. MICFIL includes two compensation filter stages, with the possibility of bypassing the last one.

46.3.3.4 Filter performance

46.3.3.4.1 Overall filter gain

As shown in Equation 44, the overall filter gain depends on:

- The quality mode you select (see Quality modes).
- The CIC decimation rate (see Equation 43).
- Dynamic range adjustment of the CIC filter (see RANGE_CTRL[RANGEADJ*n*]).

NOTE

To get a better audio performance, you must configure and adjust RANGE_CTRL[RANGEADJ*n*] depending on the quality mode selected, as pointed out in the register description.

$$Overall\ filter\ gain\ (dB) = \begin{cases} 120 \times log_{10}(32 - 2 \times CICOSR) + 6.02 \times RANGE_CTRL[RANGEADJn] - 180.60 \ ; \ if\ QSEL \in [HQ,VLQ0] \\ 120 \times log_{10}(16 - CICOSR) + 6.02 \times RANGE_CTRL[RANGEADJn] - 180.60 \quad ; \ others \end{cases}$$

Equation 44. Overall filter gain

Finally, the input and output dBFS levels relate to each other, as shown in Equation 45.

$$Output\ (dBFS) = input\ (dBFS) + overall\ filter\ gain\ (dB)$$

Equation 45. Filter input and output relationship

46.3.3.4.2 Overall passband

The decimation filter passband depends on the quality mode and output rate (FS). For instance, the overall filter passband for different output data rates when MICFIL is in High (HQ), Medium (MQ), and Low-Quality (LQ) modes is shown in Table 480.

Table 480. Filter passband in HQ, MQ, and LQ modes (DC_CTRL[DCCONFIG*n*] = 0)

Output data rate	Overall filter passband
FS	**0.0004FS - 0.4167FS**
16 kHz	6.67 Hz - 6.67 kHz
48 kHz	20 Hz - 20 kHz

The overall filter passband for different output data rates when MICFIL is in Very-Low Quality (VLQ) modes is shown in Table 481.

Table 481. Filter passband in VLQn modes (DC_CTRL[DCCONFIG*n*] = 0)

Output data rate	Overall filter passband
FS	**0.0002FS - 0.2083FS**
16 kHz	3.33 Hz - 3.33 kHz
48 kHz	10.00 Hz - 10.00 kHz
96 kHz	20 Hz - 20 kHz

46.3.3.5 Frame synchronization

If CTRL_1[FSYNCEN] = 1, MICFIL sends data to FIFOs in frame size defined by FSYNC[FSYNCLEN]. In this way, if a channel is enabled while MICFIL is running (CTRL_1[PDMIEN] = 1), MICFIL starts sending data to its respective FIFO after the current frame is sent completely. If the channel is disabled:

- if at least one other channel is still enabled, the corresponding channel stops sending data after the current frame is sent completely.
- if after that no channel is enabled, MICFIL stops sending data immediately, even if the act of sending frames is incomplete.

Below example shows the above behavior when two channels, DATACH0 and DATACH2 are enabled, and then disabled one after the other.

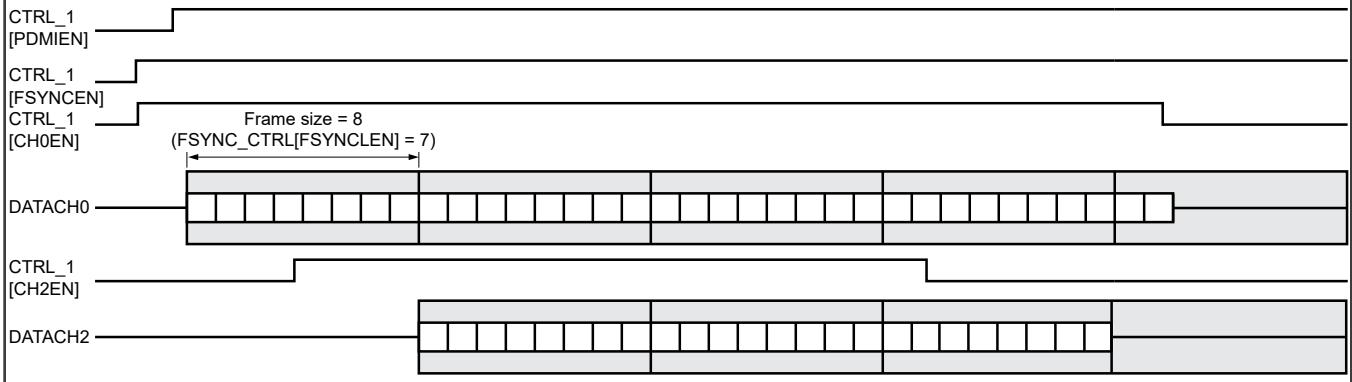


Figure 235. Frame synchronization example

46.3.3.6 Quality modes

If the decimation filter is running independently on the selected power mode, MICFIL can operate in multiple different quality modes that you can select by using CTRL_2[QSEL]. The decimation and interpolation rates in internal blocks are shown in Table 482, where the oversampling rate (OSR) depends on the value of CTRL_2[CICOSR], as shown in the following equation.

$$OSR = 16 - CICOSR$$

Equation 46. OSR

You can configure the selected quality dynamically when the decimation filter is running; the output rate is maintained in every quality selected.

The passband of the decimation filter depends on the quality mode (see Overall passband).

To maintain the same output rate in the decimation filter when the quality is changed, the PDM_CLK rate changes depending on the quality selected. For more on PDM_CLK, see Time generator.

You can enable or disable the sampler and compensation filters according to the selected quality. When they are disabled, you can achieve a low-power consumption. The selected quality also determines whether the CIC filter decimation rate is doubled. For more, see Decimation filter.

With an increase of the clock rate and OSR, you can expect a better signal-to-noise ratio (SNR) performance, which in turn leads to a higher audio quality.

Table 482. Quality modes

Quality mode	CTRL_2[QSEL]	Sampler interpolation	CIC filter decimation	First-compensation filter decimation	Second-compensation filter decimation	PDM_CLK rate	Passband
High Quality	001	—	:(2OSR)	:2	:2	Output rate × 8 × OSR	To ~0.5 × output rate
Medium Quality	000	—	:OSR	:2	:2	Output rate × 4 × OSR	To ~0.5 × output rate
Low Quality	111	× 2	:OSR	:2	:2	Output rate × 2 × OSR	To ~0.5 × output rate

Table continues on the next page...

Table 482. Quality modes (continued)

Quality mode	CTRL_2[QSEL]	Sampler interpolation	CIC filter decimation	First-compensation filter decimation	Second-compensation filter decimation	PDM_CLK rate	Passband
Very-Low Quality 0	110	—	:(2OSR)	:2	—	Output rate × 4 × OSR	To ~0.25 × output rate
Very-Low Quality 1	101	—	:OSR	:2	—	Output rate × 2 × OSR	To ~0.25 × output rate
Very-Low Quality 2	100	× 2	:OSR	:2	—	Output rate × OSR	To ~0.25 × output rate

46.3.3.7 Output rate

The output rate is the sampling rate at the decimation filter output. It depends on the sampling rate of the microphone input and the configured quality mode (see Quality modes). Common audio sampling rates are 48 kHz and 16 kHz. In this chapter, the output rate is sometimes referred to as FS.

46.3.4 Hardware voice activity detector (HWVAD)

HWVAD is a block which detects voice activity in a channel selected by the user as shown in Figure 236 and Figure 237. HWVAD takes data from the input of a selected PDM microphone to detect if there is any voice activity. If HWVAD detects a voice activity, it can issue an interrupt to the system. You can configure HWVAD in Envelope-based or Energy-based mode as described in Initialization.

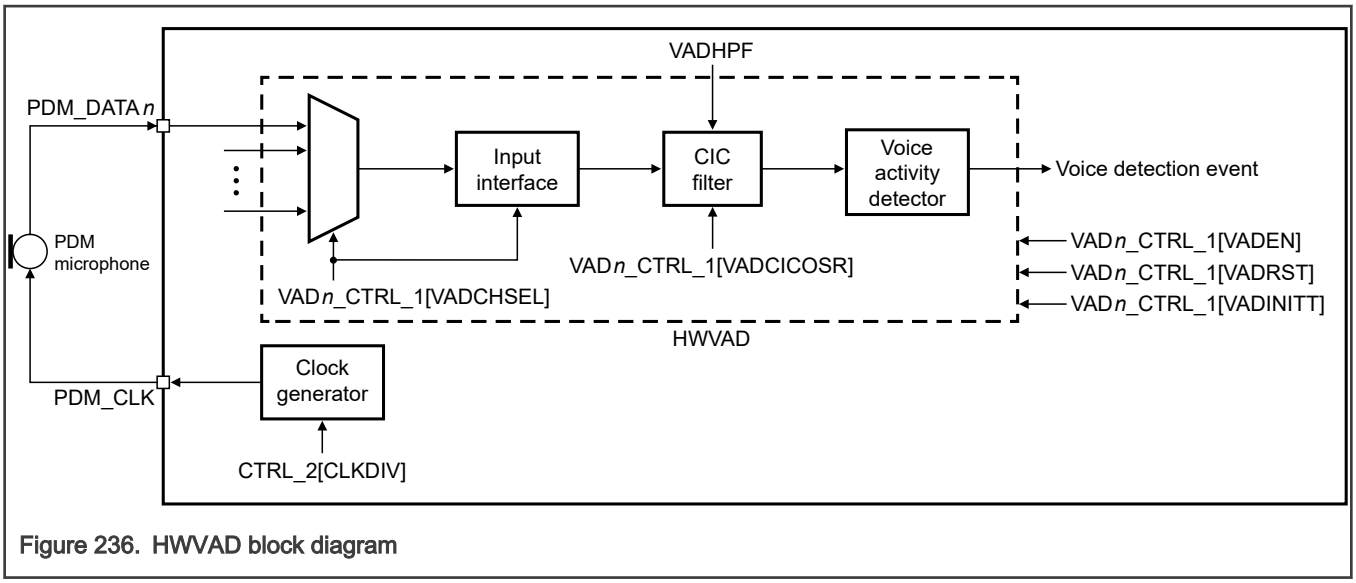


Figure 236. HWVAD block diagram

HWVAD can divide the incoming PCM signal from its internal CIC in frames. The frame energy is used to estimate the background noise. HWVAD examines both frame energy and noise energy. When HWVAD identifies a specific delta between them, it detects a voice activity and then issues an interrupt. Optionally, you can enable a zero-crossing detector (ZCD) to avoid missing of low-energy voiced speech in order to improve the voice detection performance.

You can enable HWVAD by configuring VADn_CTRL_1[VADEN] and reset HWVAD by software through VADn_CTRL_1[VADRST].

NOTE

You can enable or reset HWVAD only when MICFIL isn't running (STAT[BSY_FIL] = 0).

To perform voice detection, HWVAD uses a noise filter and a signal filter as shown in Figure 237. In addition, you can use a minimum block in the output of the noise filter and a maximum block in the output of the signal filter. HWVAD can scale the result of both blocks with gain blocks. Finally, HWVAD compares the estimated signal and noise, and it detects voice when signal is greater than noise .

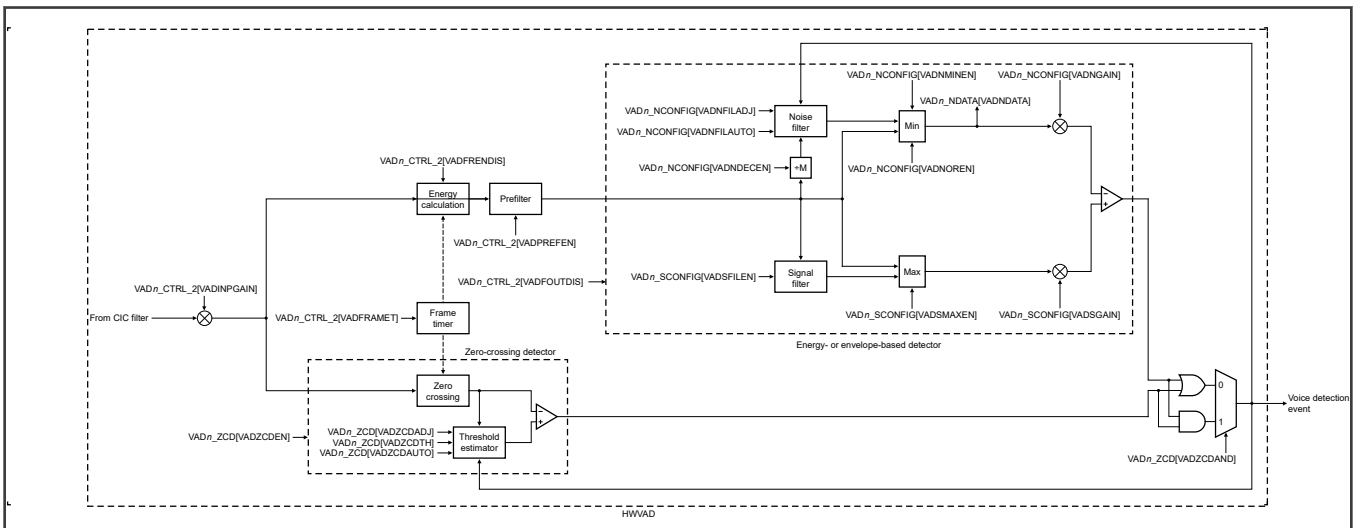


Figure 237. HWVAD block diagram (continued)

46.3.4.1 Input setting

HWVAD input has internally a user-configurable high-pass filter (see VAD*n*_CTRL_2[VADHPF] field description) and a user-configurable gain (see VAD*n*_CTRL_2[VADINPGAIN]). You can use both controls to configure the module based on external noise conditions.

46.3.4.2 Frame time setting

To perform voice detection the signal frames incoming from the CIC filter can be divided in frames whose duration can be configured. The frame time depends on the CIC output rate, the value of VAD*n*_CTRL_2[VADFRAMET] and a constant as defined in Equation 47.

$$frame\ time = \frac{VADFRAMET \times 192}{CIC\ output\ rate}$$

Equation 47. Frame time

For instance, in the typical case of the CIC filter the output rate is 192 kHz, the frame time is 10 ms when [VADFRAMET] is 10.

46.3.4.3 Initialization time

You can configure an initialization by configuring VAD*n*_CTRL_1[VADINITT]. During this initialization time, HWVAD doesn't issue any voice detection event and all detectors are disabled.

$$initialization\ time = (VADINITT + 1) \times frame\ time$$

Equation 48. Initialization time

For instance, in case that frame time is 10 ms and [VADINITT] = 10, the initialization time is 110 ms.

46.3.4.4 Energy calculation

In case VAD*n*_CTRL_2[VADFRENDIS] = 0, HWVAD uses the sum of the absolute values of all the samples of each frame to detect voice. Otherwise, if [VADFRENDIS] = 1, it uses the absolute value of the samples to detect voice, which means that the data incoming from CIC is not divided in frames.

46.3.4.5 Pre-filter

The pre-filter block is a low-pass filter which can be enabled when VAD*n*_CTRL_2[VADPREFEN] = 1.

46.3.4.6 Energy and envelope-based detector

To perform voice detection, HWVAD uses a noise filter and a signal filter as shown in Figure 237. In addition, it can use a minimum block in the output of the noise filter and a maximum block in the output of the signal filter. HWVAD can scale the result of both blocks with gain blocks. Finally, HWVAD compares the estimated signal and noise, and it detects voice when the signal is greater than the noise.

46.3.4.6.1 Noise filter

The noise filter is a low-pass filter . You can configure VAD*n*_NCONFIG[VADNFILADJ] to adjust the filter . In case VAD*n*_NCONFIG[VADNFILAUTO] = 1 the noise filter is enabled only when voice is not detected. When voice is detected the filter output does not change. In case [VADNFILAUTO] = 0 the filter is always enabled.

A decimation stage in the input of the noise filter can be enabled by VAD*n*_NCONFIG[VADNDECEN].

46.3.4.6.2 Minimum noise

To estimate the noise, HWVAD uses a block which calculates the minimum value between the input and output of the noise filter. This block can be enabled by VAD*n*_NCONFIG[VADNMINEN], otherwise the output of the noise filter passes directly.

The estimated noise of the output of this block is stored in VAD*n*_NDATA[VADNDATA] for post-processing via software.

46.3.4.6.3 Signal filter

The signal filter is a low-pass filter . This filter is enabled when VAD*n*_SCONFIG[VADSFILEN] = 1, otherwise the filter is bypassed.

46.3.4.6.4 Maximum signal

To estimate the signal, HWVAD uses a block which calculates the maximum value between the input and output of the signal filter. This block can be enabled by VAD*n*_SCONFIG[VADSMAXEN].

46.3.4.6.5 Filter result gain setting

There are two gain adjustment controls, one for the estimated noise (VAD*n*_NCONFIG[VADNGAIN]) and another for estimated signal (VAD*n*_SCONFIG[VADSGAIN]). Both gain adjustments outputs are compared, the comparison result determines whether or not there is voice in the channel.

46.3.4.7 Zero-crossing detector (ZCD)

Optionally, the use of a ZCD circuit can increase the HWVAD performance. It is enabled by VAD*n*_ZCD[VADZCDEN].

The threshold of the ZCD could be static or calculated dynamically depending on VAD*n*_ZCD[VADZCDAUTO]. When it is dynamically determined ([VADZCDAUTO] = 1) the threshold is calculated automatically taking VAD*n*_ZCD[VADZCDTH] as the initial value, otherwise ([VADZCDAUTO] = 0) the threshold is static in [VADZCDTH] user-configurable value.

You can use AND or OR to combine the ZCD result with the energy-based or envelope-based detection, it is selected by VAD*n*_ZCD[VADZCDAND]. The filter coefficient is equal to ([VADZCDADJ] + 1) × 2^{-4}.

46.3.4.8 Interrupt

HWVAD issues an interrupt when voice activity is detected and VAD*n*_CTRL_1[VADIE] is enabled. The voice activity event also sets the VADIF flag in VAD*n*_STAT[VADIF], this flag is cleared when you write 1 to it.

46.3.4.9 Extended operation

You can use VAD*n*_STAT[VADIF] for an HWVAD-extended operation, that is, using VAD*n*_NDATA[VADNDATA] value to determine a more accurate voice detection event by software. Below is the procedure to follow:

1. HWVAD issues an interrupt, [VADIF] is set.
2. Leave [VADIF] asserted while software is processing [VADNDATA] content.
3. When software finishes to process [VADNDATA] and detects voice activity:
   - If Envelope-based detection is used, pulse VAD*n*_CTRL_1[VADST10] by more than 2 cycles and clear the VADIF flag.
   - If Energy-based detection is used, clear the VADIF flag.

46.3.4.10 VAD soft-reset

You can request the VAD soft-reset by writing 1 to VAD*n*_CTRL_1[VADRST]. It resets the following blocks and registers:

- VAD*n*_STAT[VADIF] and VAD*n*_STAT[VADINSATF].
- All internal VAD buffers.

46.3.5 Bus interface

The bus interface manages DMA transactions, interrupt requests, and user register interface with the chip.

MICFIL can deliver an interrupt request or DMA request. Then, the chip or DMA, respectively, could access the filter results stored in FIFOs via the internal bus interface.

For detailed information on DMA transactions and interrupt requests, see DMA and Interrupts.

46.3.5.1 FIFO

This is the first-in, first-out (FIFO) buffer that saves the overall filtering results. The length of each FIFO is 32 entries, and there exists one FIFO for each channel. A push writes data into the FIFO, and a read pops data from FIFO. The push and pop operations from FIFO are in different clock domains. The push operation uses the MICFIL clock while the pop operation uses the CPU clock. The FIFO has a configurable watermark (see FIFO_CTRL[FIFOWMK]). When the amount of data in all FIFOs of the enabled channels is greater than the watermark value (FIFO_CTRL[FIFOWMK]), a DMA or IRQ is requested, depending on the value of CTRL_1[DISEL]. After CTRL_1[PDMIEN] enables the filters, the initial FIR results are discarded and not written into the FIFOs. Approximately 68 initial FIR results are not stored in FIFOs (FIR_RDY=0). After the filter generates valid data, the results are pushed into the FIFO.

A new filtered result is discarded (not stored) when the FIFO is full and CTRL_1[FIFOOVF*n*] becomes 1. Similarly, when trying a pop operation, if the FIFO is empty, no valid data is read and CTRL_1[FIFOUND*n*] becomes 1. In both cases, when overflow or underflow occurs, they are flagged in MICFIL FIFO Status (FIFO_STAT) to the respective channel, and an error interrupt is generated if CTRL_1[ERREN] is 1.

When you reset the FIFO by writing 1 to CTRL_1[SRES], zero values fill all FIFO positions, as well as push and pop pointers are set to their initial positions.

46.3.6 Modes of operation

This section describes the MICFIL operation modes that depend on:

- Whether MICFIL receives a Stop request.
- The chip's Debug mode.

- The configuration of CTRL_1[MDIS], CTRL_1[DBGE], CTRL_1[DBG],and CTRL_1[DOZEN], as summarized in the following table.

Table 483. Mode selection

Mode	Normal	Debug		Low-Power		Disable/Low-Leakage (DLL)		
CTRL_1[MDIS]	0	0		0		0		1
Doze request and CTRL_1[DOZEN] = 1	0	0		0		0	1	Any
Stop request	0	0		1		1		Any
CTRL_1[DOZEN]	Any	Any		0		1		Any
CTRL_1[DECFILS]	Any	Any		1	0	Any		
Debug request or CTRL_1[DBG]	0	1		Any		Any		
CTRL_1[DBGE]	Any	1	0	Any		Any		
CPU clock	Enabled	Enabled		Gated (Externally)		Gated (Externally)	Enabled	Gated (Internally)
MICFIL internal clock	Enabled	Enabled		Enabled		Gated		
Decimation filter state	Run[1]	Run[1]	Stop	Run[1]	Stop	Stop		
HWVAD*n* state	Run[2]	Run[2]	Stop	Run[2]		Stop		

1. Depends on the configuration of CTRL_1[PDMIEN] and CTRL_1[CH*n*EN].
2. Depends on the configuration of CTRL_1[PDMIEN] and CTRL_1[VADEN].

46.3.6.1 Normal mode

This is the default operational mode for MICFIL. In this mode, you can enable the channel functionality to perform the filtering operation.

46.3.6.2 Debug mode

You could use this mode to check the status of the module, when requested. In this mode, you can stop MICFIL after the remaining data processing is complete, depending on the value of CTRL_1[DBGE]. You can activate this mode by writing 1 to CTRL_1[DBG].

If CTRL_1[DBGE] = 0, and CTRL_1[DBG] = 1 when MICFIL is running, MICFIL stops after processing the remaining data. Otherwise, if [DBGE] = 1, MICFIL continues to run.

MICFIL cannot enter Debug mode after entering DLL or Low-Power mode.

The Debug mode affects all channels in MICFIL. Access to output buffers, as well as their related flags, remains operational.

46.3.6.3 Low-Power mode

MICFIL enters Low-Power mode when MICFIL receives a Stop request and CTRL_1[DOZEN] = 0. In Low-Power mode, MICFIL is able to filter the PDM microphone data and perform voice detection such as in Normal mode, and can generate a wake-up event through asynchronous DM or IRQ requests, depending on the values of CTRL_1[CH*n*EN], VAD*n*_CTRL_1[VADEN], and CTRL_1[PDMIEN].

After MICFIL enters Low-Power mode:

- If CTRL_1[DECFILS] = 1, Stop request is acknowledged immediately and the decimation filter keeps running.
- If CTRL_1[DECFILS] = 0, the decimation filter processes the last input sample. After that, Stop request is acknowledged and the filter is stopped from running.

During this mode, VAD*n*_CTRL_1[VADEN] continues running if it was previously enabled ([VADEN] = 1) during Normal mode.

You can generate DMA and IRQ requests asynchronously to wake up the CPU.

NOTE

The asynchronous DMA or interrupt requests are generated only in this mode.

46.3.6.4 Disable/Low-Leakage (DLL) mode

In this mode, MICFIL stops after the remaining data processing is complete. MICFIL is in DLL mode when:

- MICFIL receives a Stop request and CTRL_1[DOZEN] = 1.
- CTRL_1[MDIS] = 1.

If you request DLL mode when MICFIL is running, MICFIL stops after completion to process the remaining data.

You can write only to CTRL_1 and CTRL_2, with the exception of writes to CTRL_1[DBG] and CTRL_1[SRES], which are ignored. However, you can still read these fields.

46.3.7 Clocking

Table 484. MICFIL clocks

Clock name	Description
CPU clock (MICFIL_CLK, MICFIL_CLK_S)	Clock that controls the registers
MICFIL internal clock (MICFIL_CLK_APP)	Application clock that operates the filter

Besides the above clock signals, MICFIL generates PDM_CLK for PDM microphones based on MICFIL internal clock frequency and the value in CTRL_2[CLKDIV].

46.3.8 Interrupts

If CTRL_1[DISEL] = 10b, an interrupt request indicates that filtering results are stored in the FIFOs to be read by the CPU. In this case, you can read the data in DATACH*n*[DATA], and STAT[CH*n*F] becomes 1 for the enabled channels when the CHn FIFO surpasses the watermark level.

When MICFIL is in Low-Power mode, the CPU clock is disabled, but the MICFIL clock remains enabled. In this case, an asynchronous interrupt is delivered to the system. This wakes up the CPU clock, which returns to Normal mode. The system could read the FIFO data, after which it must clear STAT[CH*n*F] to get a new interrupt request.

An error interrupt request can also be generated when an exceptional condition happens, such as an overflow or underflow in a channel output, or an overflow or underflow of a FIFO buffer, or the MICFIL clock rate is not the minimum value required to the MICFIL working (see Minimum required CLKDIV).

CTRL_1[ERREN] enables error interruption and the respective status flag signalizes the type of error:

- OUTOVFn and OUTUNFn for channel outputs errors
- FIFO_STAT[FIFOOVF*n*] and FIFO_STAT[FIFOUND*n*] for FIFO errors
- STAT[LOWFREQF] for MICFIL clock rate

46.3.9 DMA

If CTRL_1[DISEL] = 1, the FIFO stored samples are read through DMA transactions. Then if there is at least one enabled channel and all FIFOs of all enabled channels reach their watermark levels, the output interface makes a request for a DMA transaction.

If CTRL_1[PDMIEN] = 1 and the filters start to operate, you must wait until the filter results stabilize and no filter results are stored in the FIFOs. Therefore, the first DMA request takes longer to occur than the ones that follow.

STAT[CH*n*F] becomes 0 after a DMA transaction is complete.

If MICFIL is in Low-Power mode and the watermark level is surpassed, an asynchronous DMA request is delivered to the system. Then, the system wakes up the CPU clock, returning to Normal mode to read the FIFO data.

46.4 External signals

Table 485. PDM microphone interface external signals

Signal	Description	Direction
PDM_DATA*n*	Data bitstream received from the nth PDM microphone pair throughout the input interface (see Input interface)	I
PDM_CLK	Clock that is delivered to the PDM microphones from the clock generator (see Time generator)	O

NOTE

MICFIL does not provide metastability protection or filtering for input data. You must generate the data using the provided PDM_CLK.

46.5 Initialization

46.5.1 Procedure without HWVAD

Perform this procedure for MICFIL initialization without enabling HWVAD, (that is, by using only decimation filters):

1. Program CTRL_2[CLKDIV], CTRL_1[CH*n*EN], FIFO_CTRL[FIFOWMK], CTRL_1[DISEL], CTRL_1[ERREN], and so on, as required. For more details see MICFIL_CLK_ROOT rate and CLKDIV calculation examples.
2. Start the module operation by writing 1 to CTRL_1[PDMIEN].

The module is ready to receive data from the input ports of the enabled channels.

46.5.2 Initialization procedure with HWVAD

In this section describes the initialization procedures to initialize MICFIL with HWVAD.

HWVAD can work in two modes: Energy-based mode or Envelope-based mode. Optionally, a ZCD can be enabled to improve the voice detection.

The Energy-based method separates the audio signal in frames and estimates the energy of these frames to detect voice activity. On the other hand, the Envelope-based method calculates the audio signal envelope to detect voice.

Both methods have the same detection performance, but the Energy-based is recommended for low-power applications where the chip can be asleep for long periods. Envelope-based detection is recommended for applications where the chip requires faster wake-up and it is not asleep for long periods.

ZCD is recommended to improve detection with fricatives phonemes for example /f/ or /s/) in noisy environments.

46.5.2.1 HWVAD in Energy-based mode

The sequence of steps to initialize MICFIL with HWVAD in Energy-based mode is as follows:

1. Program the control bits as desired for your application (CTRL_2[CLKDIV], CTRL_1[CH*n*EN], FIFO_CTRL[FIFOWMK], CTRL_1[DISEL], CTRL_1[ERREN] etc.). For more details see MICFIL_CLK_ROOT rate and CLKDIV calculation examples.
2. Configure general HWVAD parameters:
    a. CIC oversampling rate (VAD*n*_CTRL_1[VADCICOSR]).
    b. Configure the voice source channel (VAD*n*_CTRL_1[VADCHSEL]).

c. Configure the internal initialization timer (VAD*n*_CTRL_1[VADINITT]) if required. More details in Initialization time.

d. Write 1 to VAD*n*_CTRL_1[VADEN].

3. Configure input, noise and signal gains according to the environmental noise conditions:
   a. Configure the input gain (VAD*n*_CTRL_2[VADINPGAIN]).
   b. Configure the signal gain (VAD*n*_SCONFIG[VADSGAIN]).
   c. Configure the noise gain (VAD*n*_NCONFIG[VADNGAIN]).
4. Enable the high-pass filter if required (VAD*n*_CTRL_2[VADHPF]).
5. Configure VAD*n*_NCONFIG[VADNFILADJ] according to environmental noise conditions ([VADNFILADJ] = 16 is recommended).
6. Put HWVAD in Energy-based mode (default values):
   a. Keep VAD*n*_CTRL_2[VADFRENDIS] = 0.
   b. Keep VAD*n*_CTRL_2[VADPREFEN] = 0.
   c. Keep VAD*n*_SCONFIG[VADSFILEN] = 0.
   d. Keep VAD*n*_SCONFIG[VADSMAXEN] = 0.
   e. Keep VAD*n*_NCONFIG[VADNFILAUTO] = 1.
   f. Keep VAD*n*_NCONFIG[VADNMINEN] = 0.
   g. Keep VAD*n*_NCONFIG[VADNDECEN] = 0.
   h. Keep VAD*n*_NCONFIG[VADNOREN] = 0.
7. Initialize ZCD (ZCD initialization) if required.
8. Enable interrupts if required (VAD*n*_CTRL_1[VADIE] and [VADERIE]).
9. Start the overall module asserting CTRL_1[PDMIEN].
10. The module is ready to receive data from the input ports of the enabled channels and to trigger interrupts when it detects voice.

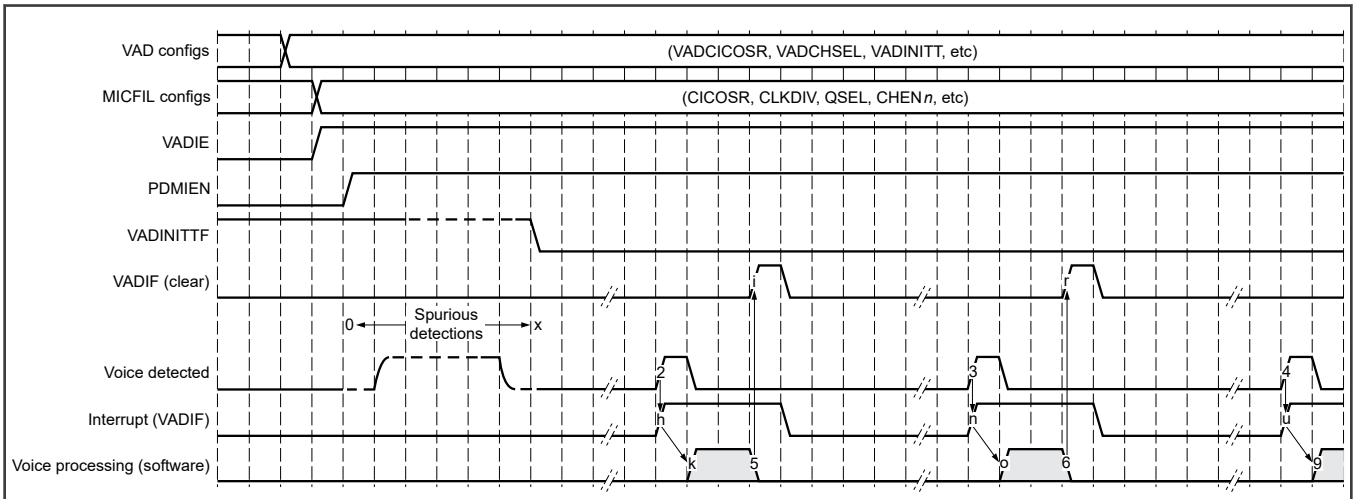


Figure 238. Energy-based initialization sequence

46.5.2.2 HWVAD in Envelope-based mode

The sequence of steps to initialize MICFIL with HWVAD in Envelope-based mode is as follows:

1. Program the control bits as desired for your application (CTRL_2[CLKDIV], CTRL_1[CH*n*EN], FIFO_CTRL[FIFOWMK], CTRL_1[DISEL], CTRL_1[ERREN] etc.). For more details see MICFIL_CLK_ROOT rate and CLKDIV calculation examples.
2. Configure general HWVAD parameters:
    a. CIC oversampling rate (VAD*n*_CTRL_1[VADCICOSR]).
    b. Configure the voice source channel (VAD*n*_CTRL_1[VADCHSEL]).
    c. Configure the internal initialization timer (VAD*n*_CTRL_1[VADINITT]) if required. More details in Initialization time.
    d. Write 1 to VAD*n*_CTRL_1[VADEN] field.
3. Configure input, noise and signal gains according to the environment noise conditions:
    a. Configure the input gain (VAD*n*_CTRL_2[VADINPGAIN]).
    b. Configure the signal gain (VAD*n*_SCONFIG[VADSGAIN]).
    c. Configure the noise gain (VAD*n*_NCONFIG[VADNGAIN]).
4. Enable the high-pass filter if required (VAD*n*_CTRL_2[VADHPF]).
5. Put HWVAD in Envelope-based mode:
    a. Keep VAD*n*_NCONFIG[VADNFILADJ] = 0.
    b. Write 1 to VAD*n*_CTRL_2[VADFRENDIS].
    c. Write 1 to VAD*n*_CTRL_2[VADPREFEN].
    d. Write 1 to VAD*n*_SCONFIG[VADSFILEN].
    e. Write 1 to VAD*n*_SCONFIG[VADSMAXEN].
    f. Write 0 to VAD*n*_NCONFIG[VADNFILAUTO].
    g. Write 1 to VAD*n*_NCONFIG[VADNMINEN].
    h. Write 1 to VAD*n*_NCONFIG[VADNDECEN].
    i. Write 1 to VAD*n*_NCONFIG[VADNOREN].
6. Initialize ZCD (ZCD initialization) if required.
7. Start the overall module asserting the CTRL_1[PDMIEN] bit.
8. Wait until VAD*n*_STAT[VADINITF] becomes 0.
9. Write VAD*n*_CTRL_1[VADST10] three times.
10. Enable interrupts if required (VAD*n*_CTRL_1[VADIE] and [VADERIE]).
11. The module is ready to receive data from the input ports of the enabled channels and to trigger interrupts when it detects voice.

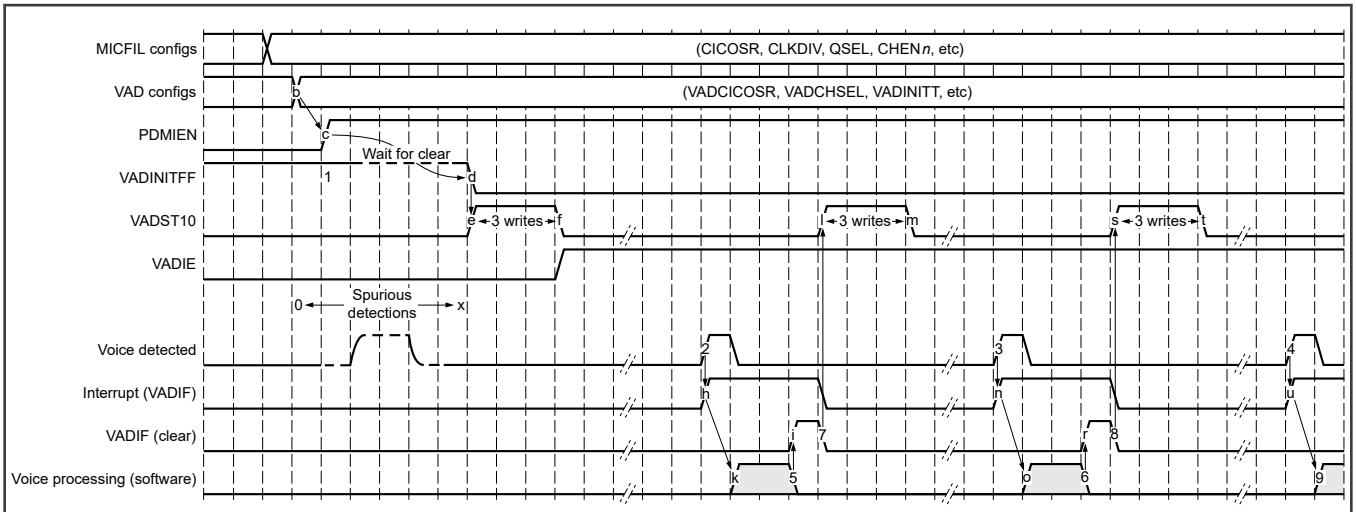


Figure 239. Envelope-based initialization sequence

46.5.2.3 ZCD initialization

The sequence of steps to initialize ZCD in Automatic mode is as follows:

1. Configure VAD*n*_ZCD[VADZCDADJ].
2. Configure an initial threshold in VAD*n*_ZCD[VADZCDTH] according to the environmental noise conditions if required.
3. Configure VAD*n*_ZCD[VADZCDAND].
4. Keep VAD*n*_ZCD[VADZCDAUTO] = 1.
5. Write 1 to VAD*n*_ZCD[VADZCDEN].

The sequence of steps to initialize ZCD in Manual mode is as follows:

1. Configure a threshold in VAD*n*_ZCD[VADZCDTH] according to the environmental noise conditions if required.
2. Configure VAD*n*_ZCD[VADZCDAND].
3. Write 0 to VAD*n*_ZCD[VADZCDAUTO].
4. Write 1 to VAD*n*_ZCD[VADZCDEN].

46.6 Application information

46.6.1 Reconfiguration procedure

Perform this procedure to reconfigure MICFIL:

1. Write 0 to CTRL_1[PDMIEN].
2. Wait until STAT[BSY_FIL] becomes 0.
3. Run a soft-reset cycle.
4. Run an initialization procedure as described in Procedure without HWVAD or Initialization procedure with HWVAD.

46.7 MICFIL register descriptions

This section provides the memory map and detailed description of all MICFIL registers. Any access to reserved areas can issue a target bus error indication.

You must implement the registers allocated in this memory map for up to 4 pairs of microphones, indicating that 8 channels are available.

46.7.1 MICFIL memory map

MICFIL base address: 4452_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	MICFIL Control 1 (CTRL_1)	32	RW	0000_0000
4	MICFIL Control 2 (CTRL_2)	32	RW	0000_0000
8	MICFIL Status (STAT)	32	RW	0000_0000
10	MICFIL FIFO Control (FIFO_CTRL)	32	RW	0000_001F
14	MICFIL FIFO Status (FIFO_STAT)	32	RW	0000_0000
24 - 40	MICFIL Output Result (DATACH0 - DATACH7)	32	R	0000_0000
64	MICFIL DC Remover Control (DC_CTRL)	32	RW	0000_0000
68	MICFIL Output DC Remover Control (DC_OUT_CTRL)	32	RW	0000_0000
74	MICFIL Range Control (RANGE_CTRL)	32	RW	0000_0000
7C	MICFIL Range Status (RANGE_STAT)	32	RW	0000_0000
80	Frame Synchronization Control (FSYNC_CTRL)	32	RW	0000_0000
84	Version ID (VERID)	32	R	020F_0000
88	Parameter (PARAM)	32	R	010B_0154
90	Voice Activity Detector 0 Control (VAD0_CTRL_1)	32	RW	0000_0000
94	Voice Activity Detector 0 Control (VAD0_CTRL_2)	32	RW	000A_0000
98	Voice Activity Detector 0 Status (VAD0_STAT)	32	RW	8000_0000
9C	Voice Activity Detector 0 Signal Configuration (VAD0_SCONFIG)	32	RW	0000_0000
A0	Voice Activity Detector 0 Noise Configuration (VAD0_NCONFIG)	32	RW	8000_0000
A4	Voice Activity Detector 0 Noise Data (VAD0_NDATA)	32	R	0000_0000
A8	Voice Activity Detector 0 Zero-Crossing Detector (VAD0_ZCD)	32	RW	0000_0004

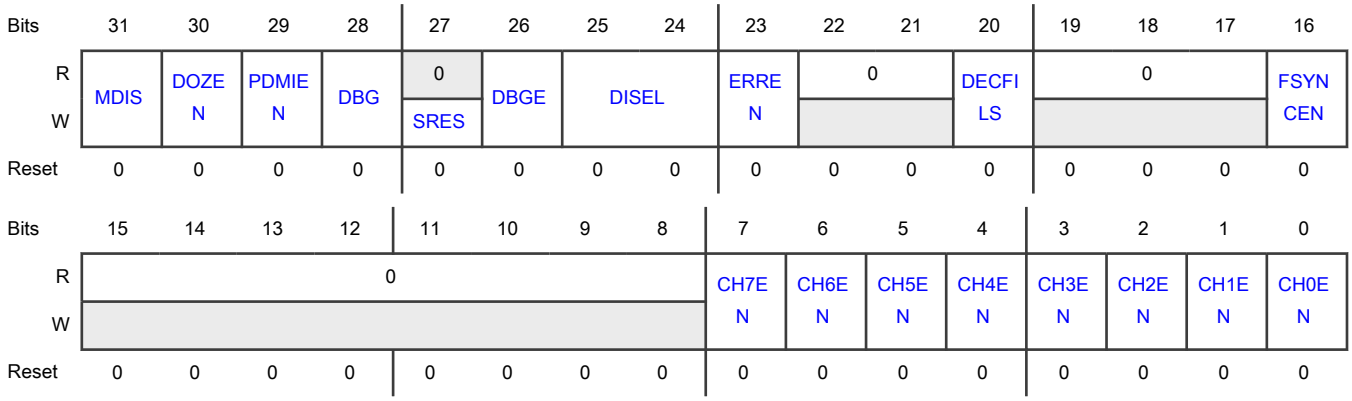
46.7.2 MICFIL Control 1 (CTRL_1)

Contains control fields for MICFIL.

Offset

Register	Offset
CTRL_1	0h

Diagram



Fields

Field	Description
31 MDIS	Module Disable Places MICFIL in DLL mode (see Disable/Low-Leakage (DLL) mode). The generated PDM_CLK stops, and consequently, the microphone generates no data. **NOTE** When this field = 1, only writes to MICFIL Control 1 (CTRL_1) and MICFIL Control 2 (CTRL_2) are allowed with the exception of writes to CTRL_1[DBG] and CTRL_1[SRES]. A target bus error indication is issued if you write to any other register. 0 - Normal mode 1 - DLL mode
30 DOZEN	Stop Enable When MICFIL receives a Stop request and this field is active, MICFIL enters DLL mode (see Disable/Low-Leakage (DLL) mode). 0 - Disables 1 - Enables
29 PDMIEN	MICFIL Enable Starts the module operation. Only the channels that CTRL_1[CH*n*EN] enables or VAD enabled by VADEN start operating. When this field becomes 0, MICFIL stops after it completes the remaining filter processing. **NOTE** You must enable this field as the last step when initializing the module (see Initialization). 0 - Stops MICFIL operation 1 - Starts MICFIL operation
28 DBG	Debug Mode Specifies the mode (Debug or Normal) that MICFIL operates in.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	MICFIL operates in Debug mode, depending on the value of CTRL_1[DBGE]. When Debug mode is disabled, MICFIL operates in Normal mode. See Debug mode for more on Debug mode. **NOTE** You can write to this field only when CTRL_1[MDIS] = 0. 0 - Normal 1 - Debug
27 SRES	Software Reset Provides the CPU with the capability to initialize MICFIL through the target-bus interface. The field always reads 0, and is effective only when CTRL_1[MDIS] = 0. It is a self-clearing field. **NOTE** You must write 1 to this field only when STAT[BSY_FIL] = 0. You can use this field to clear data in all filters. It resets the following blocks and registers: • All internal buffers in decimation filters and voice activity detectors • All FIFOs • STAT[CH*n*F] and LOWFREQF flag and MICFIL FIFO Status (FIFO_STAT) • Time generator 0 - No action 1 - Software reset
26 DBGE	Module Enable in Debug Enables operation in Debug mode (see Debug mode). 0 - Disables after completing the current frame 1 - Enables operation
25-24 DISEL	DMA Interrupt Selection Specifies the type of interrupt requests (DMA or filter) that MICFIL performs when the number of FIFO elements surpasses the configured watermark. Note that requests are only generated from the channels for which CTRL_1[CH*n*EN] = 1. **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 00 - Disables DMA and interrupt requests 01 - Enables DMA requests 10 - Enables interrupt requests 11 - Reserved
23	Error Interruption Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
ERREN	Enables error interrupts (exceptions). The following conditions can lead to errors: • An overflow or underflow in the FIFOs (indicated by FIFO_STAT[FIFOOVF*n*] or FIFO_STAT[FIFOUND*n*], respectively) • An overflow or underflow in the decimations filters (indicated by RANGE_STAT[RANGEOVF*n*] or RANGE_STAT[RANGEUNF*n*], respectively) • not enough MICFIL internal clock frequency (LOWFREQF bit). **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 0 - Disables 1 - Enables
22-21 —	Reserved
20 DECFILS	Decimation Filter Enable in Stop Enables decimation filters to run when MICFIL receives a Stop request. 0 - Stops decimation filter 1 - Keeps decimation filter running
19-17 —	Reserved
16 FSYNCEN	Frame Synchronization Enable Enables frame synchronization (see Frame synchronization). **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 0 - Disables 1 - Enables
15-8 —	Reserved
7-0 CHnEN	Channel n Enable Enables the decimation filter channel n (see Initialization for more information).

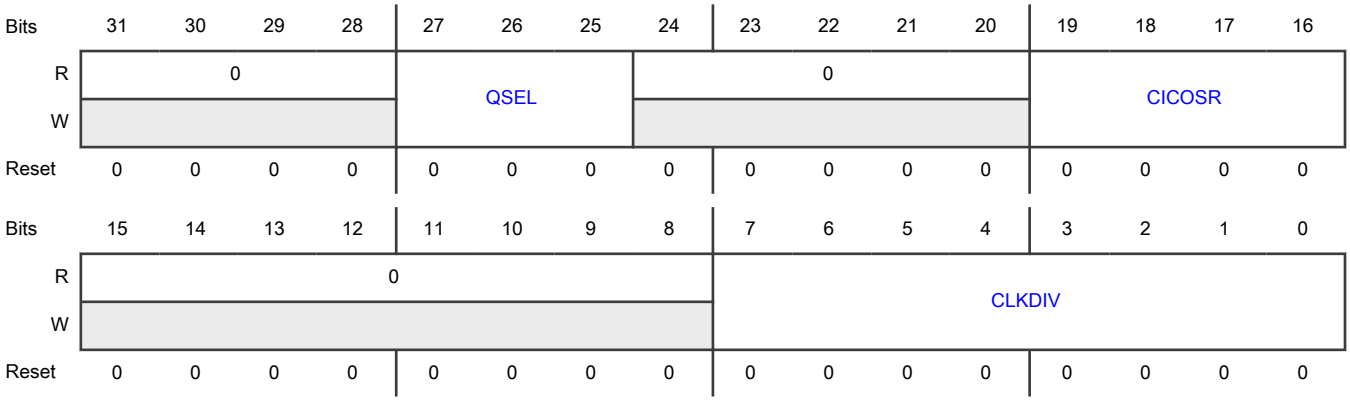
46.7.3 MICFIL Control 2 (CTRL_2)

Contains control fields for MICFIL.

Offset

Register	Offset
CTRL_2	4h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-25 QSEL	Quality Mode Defines the actual quality mode (see Quality modes) of the decimation filter. 000 - Medium-Quality mode 001 - High-Quality mode 100 - Very-Low-Quality 2 mode 101 - Very-Low-Quality 1 mode 110 - Very-Low-Quality 0 mode 111 - Low-Quality mode
24-20 —	Reserved
19-16 CICOSR	CIC Decimation Rate Configures the CIC filter decimation rate (see Equation 43 for more information). **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 0000 - CIC oversampling rate = 0

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0001 - CIC oversampling rate = 1 0010-1110 - ... 1111 - CIC oversampling rate = 15
15-8 —	Reserved
7-0 CLKDIV	Clock Divider Configures the internal clock divider value (see Clock divider for more information) using the following equation: $$PDM_CLK\ rate = \frac{MICFIL_CLK_ROOT}{2 \times floor(K \times CLKDIV)}$$ **NOTE** • You must not change the value of this field when STAT[BSY_FIL] = 1. • [CLKDIV] = 0 has the same effect as [CLKDIV] = 1. In Medium Quality mode and Very-Low Quality 0 mode, [CLKDIV] = 0 or 1 has the same effect as [CLKDIV] = 2. In High Quality mode, [CLKDIV] = 0 or 1 is not allowed. 0000_0000 - Internal clock divider value = 0 0000_0001 - Internal clock divider value = 1 0000_0010-1111_1110 - ... 1111_1111 - Internal clock divider value = 255

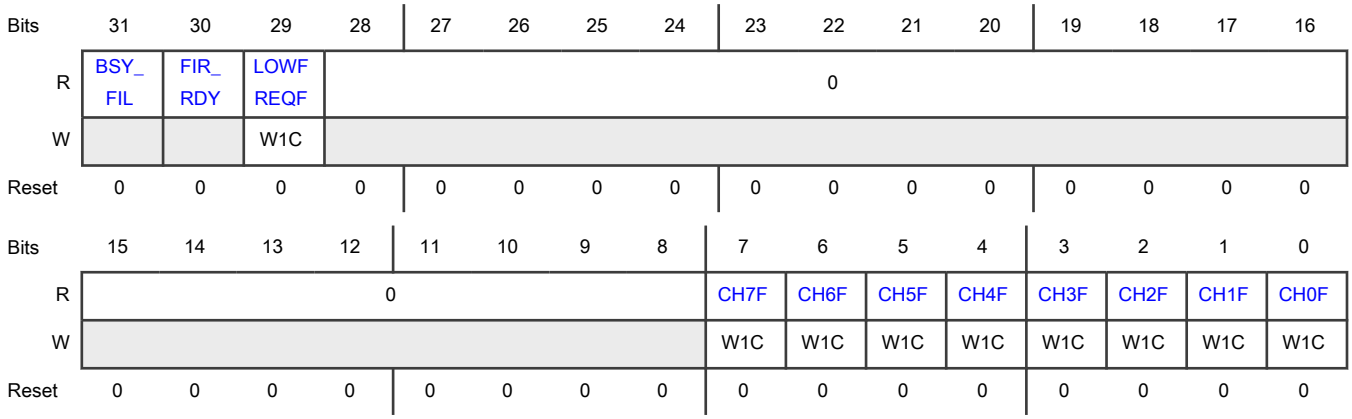
46.7.4 MICFIL Status (STAT)

Contains status flags for MICFIL.

Offset

Register	Offset
STAT	8h

Diagram



Fields

Field	Description
31 BSY_FIL	Busy Flag Signalizes when at least a decimation filter channel or at least a HWVAD is running and the PDM_CLK is being generated. You could use this field as a confirmation that all decimation filters are stopped in a transition to Debug mode (when CTRL_1[DBGE] = 0) or DLL mode. 0 - MICFIL is stopped 1 - MICFIL is running
30 FIR_RDY	Filter Data Ready Indicates whether all the decimation filter buffers are updated, and all filter data provided from now on is reliable. 0 - Not reliable 1 - Reliable
29 LOWFREQF	Low Frequency Flag Indicates that the CLKDIV field value is not high enough to filter all enabled channels , and the compensation filters didn't finish their operations before the next sample arrived. See Minimum required CLKDIV. Write a logic 1 to this field to clear this flag. **NOTE** The filtering results in this condition are unpredictable. 0 - CLKDIV value is OK 1 - CLKDIV value is too low
28-16 —	Reserved
15-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7-0 CHnF	Channel n Output Data Flag Indicates whether this channel's FIFO has surpassed its watermark value and the data is available in MICFIL Output Result (DATACH0 - DATACH7). This flag contributes to generate an interrupt or DMA request if CTRL_1[DISEL] enables it. Write 1 to CTRL_1[DISEL] to clear this flag. If the DMA request is set (CTRL_1[DISEL] = 1), [CHnF] is set according to the watermark level . In other cases (IRQ included), [CHnF] is set according to the watermark level and is cleared when you write 1 to it. **NOTE** If DMA is enabled, this flag clears itself after the DMA transaction is complete. 0 - Not surpassed 1 - Surpassed

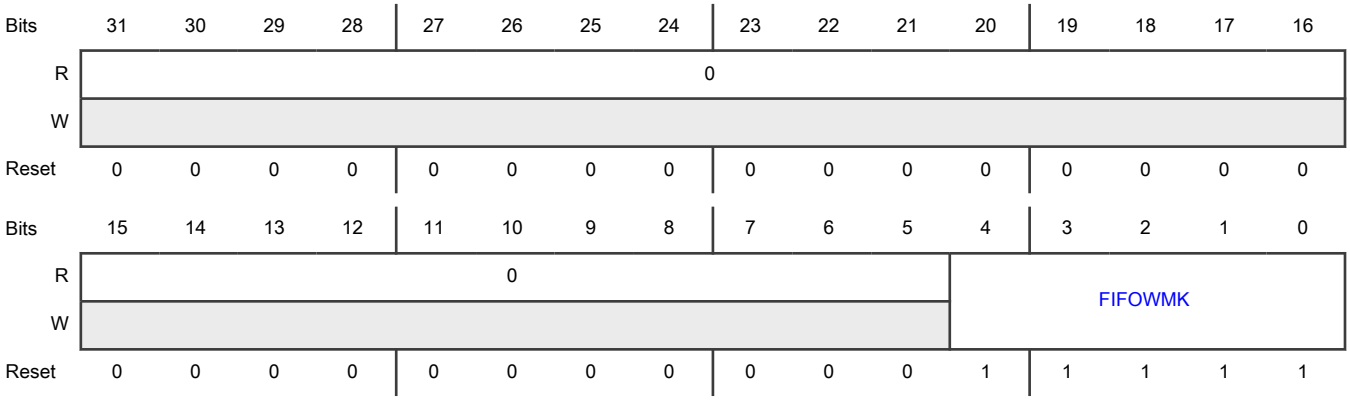
46.7.5 MICFIL FIFO Control (FIFO_CTRL)

Contains MICFIL FIFO control fields.

Offset

Register	Offset
FIFO_CTRL	10h

Diagram



Fields

Field	Description
31-5	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
4-0 FIFOWMK	FIFO Watermark Control Controls the watermark of the FIFO used to set STAT[CH*n*F]. If the number of results in the FIFO is greater than the value of this field, STAT[CH*n*F] is set. A DMA request or interrupt can be also generated according to the configuration of CTRL_1[DISEL]. **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. Before changing the watermark value, you must service all filtered data and clear channel flags.

46.7.6 MICFIL FIFO Status (FIFO_STAT)

Contains MICFIL FIFO status flags.

Offset

Register	Offset
FIFO_STAT	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFOUND7	FIFOUND6	FIFOUND5	FIFOUND4	FIFOUND3	FIFOUND2	FIFOUND1	FIFOUND0	FIFOOVF7	FIFOOVF6	FIFOOVF5	FIFOOVF4	FIFOOVF3	FIFOOVF2	FIFOOVF1	FIFOOVF0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-8 FIFOUNDn	FIFO Underflow Exception Flag for Channel n

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Indicates an exceptional underflow condition in the FIFO. If the FIFO is empty and a read operation is requested, then this flag is set. 0 - No exception by FIFO underflow 1 - Exception by FIFO underflow
7-0 FIFOOVFn	FIFO Overflow Exception Flag for Channel n Indicates an exceptional overflow condition in the FIFO. If the FIFO is full and you receive an additional result to be written, this flag is set. 0 - No exception by FIFO overflow 1 - Exception by FIFO overflow

46.7.7 MICFIL Output Result (DATACH0 - DATACH7)

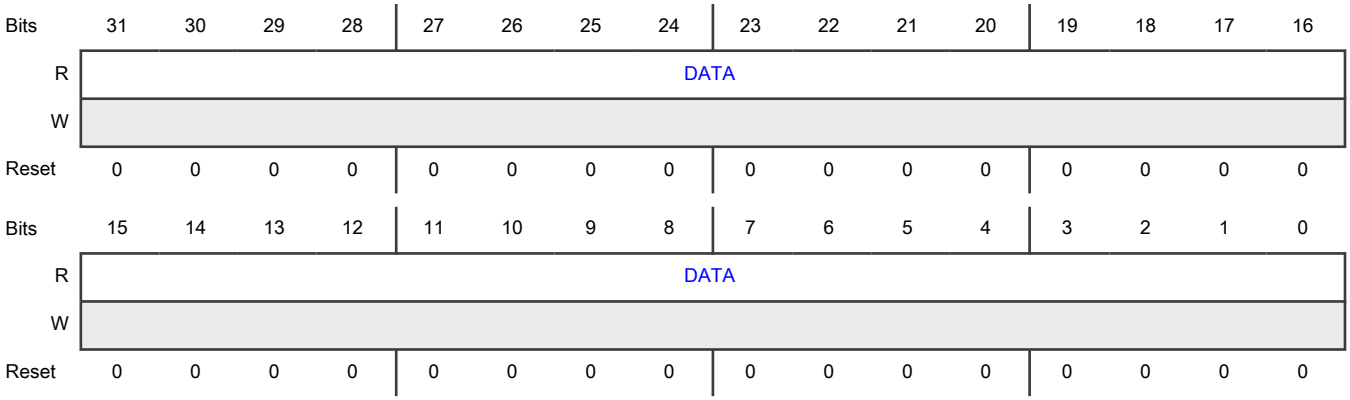
Contains the word value stored at the top of the channel's n FIFO.

Offset

For n = 0 to 7:

Register	Offset
DATACHn	24h + (n × 4h)

Diagram



Fields

Field	Description
31-0 DATA	Channel n Data Reflects the word value stored at the top of the channel's n FIFO.

Table continues on the next page...

Field	Description
	The samples are integer-signed numbers in the two's-complement format. **NOTE** Even if the FIFO data width is 32-bit, only the 24 more significant bits have information, and the other bits are always 0.

46.7.8 MICFIL DC Remover Control (DC_CTRL)

Contains DC remover control fields.

Offset

Register	Offset
DC_CTRL	64h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DCCONFIG7		DCCONFIG6		DCCONFIG5		DCCONFIG4		DCCONFIG3		DCCONFIG2		DCCONFIG1		DCCONFIG0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-14: DCCONFIG7 13-12: DCCONFIG6 11-10: DCCONFIG5 9-8: DCCONFIG4	Channel n DC Remover Configuration Defines the value of the cut-off frequency of the DC remover. 00 - 20 Hz (PDM_CLK = 3.072 MHz) 01 - 13.3 Hz (PDM_CLK = 3.072 MHz) 10 - 40 Hz (PDM_CLK = 3.072 MHz) 11 - DC remover is bypassed

Table continues on the next page...

Table continued from the previous page...

Field	Description
7-6: DCCONFIG3 5-4: DCCONFIG2 3-2: DCCONFIG1 1-0: DCCONFIG0	

46.7.9 MICFIL Output DC Remover Control (DC_OUT_CTRL)

Contains output DC remover configuration fields.

Offset

Register	Offset
DC_OUT_CTRL	68h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DCCONFIG7		DCCONFIG6		DCCONFIG5		DCCONFIG4		DCCONFIG3		DCCONFIG2		DCCONFIG1		DCCONFIG0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-14: DCCONFIG7 13-12: DCCONFIG6	Channel n DC Remover Configuration Defines the value of the cut-off frequency of the DC remover. 00 - 20 Hz (FS = 48 kHz)

Table continues on the next page...

Table continued from the previous page...

Field	Description
11-10: DCCONFIG5 9-8: DCCONFIG4 7-6: DCCONFIG3 5-4: DCCONFIG2 3-2: DCCONFIG1 1-0: DCCONFIG0	01 - 13.3 Hz (FS = 48 kHz) 10 - 40 Hz (FS = 48 kHz) 11 - DC remover is bypassed

46.7.10 MICFIL Range Control (RANGE_CTRL)

Contains range configuration information.

Offset

Register	Offset
RANGE_CTRL	74h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RANGEADJ7				RANGEADJ6				RANGEADJ5				RANGEADJ4			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RANGEADJ3				RANGEADJ2				RANGEADJ1				RANGEADJ0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28: RANGEADJ7	Channel n Range Adjustment

Table continues on the next page...

Field	Description
27-24: RANGEADJ6 23-20: RANGEADJ5 19-16: RANGEADJ4 15-12: RANGEADJ3 11-8: RANGEADJ2 7-4: RANGEADJ1 3-0: RANGEADJ0	Adjusts the dynamic range of the CIC filter. It is an unsigned non-negative value. You must configure this field within the following ranges: • If CTRL_2[QSEL] is in High-Quality and Very-Low-Quality 0 modes (HQ and VLQ0): RANGEADJ*n* ≤ 30 - ceil(6*$\log_2$(2OSR)) • If CTRL_2[QSEL] is in Medium-Quality and Very-Low Quality 1 modes (MQ and VLQ1): RANGEADJ*n* ≤ 30 - ceil(6*$\log_2$(OSR)) • If CTRL_2[QSEL] is in Low Quality and Very-Low-Quality 2 modes (LQ and VLQ2): RANGEADJ*n* ≤ 29 - ceil(6*$\log_2$(OSR)) where OSR = 16 - CICOSR, which is the CIC oversampling rate (see Equation 46), and the ceiling function, ceil(x), maps to the least integer greater than or equal to x.

46.7.11 MICFIL Range Status (RANGE_STAT)

Contains range status information.

Offset

Register	Offset
RANGE_STAT	7Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								RANGEUN...	RANGEUN...	RANGEUN...	RANGEUN...	RANGEUN...	RANGEUN...	RANGEUN...	RANGEUN...
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RANGEOV...	RANGEOV...	RANGEOV...	RANGEOV...	RANGEOV...	RANGEOV...	RANGEOV...	RANGEOV...
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23-16 RANGEUNFn	Channel n Range Underflow Error Flag Indicates an exceptional underflow condition in MICFIL range. **NOTE** If this error flag is asserted, channel n data is not reliable. You must adjust the value of RANGE_CTRL[RANGEADJ*n*] until this flag is not asserted anymore. 0 - No exception by range underflow 1 - Exception by range underflow
15-8 —	Reserved
7-0 RANGEOVFn	Channel n Range Overflow Error Flag Indicates an exceptional overflow condition in MICFIL range. **NOTE** If this error flag is asserted, channel n data is not reliable. You must adjust the value of RANGE_CTRL[RANGEADJ*n*] until this flag is not asserted anymore. 0 - No exception by range overflow 1 - Exception by range overflow

46.7.12 Frame Synchronization Control (FSYNC_CTRL)

Contains the frame synchronization window length value.

Offset

Register	Offset
FSYNC_CTRL	80h

Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	FSYNCLEN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	FSYNCLEN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
0-31 FSYNCLEN	Frame Synchronization Window Length Specifies the frame synchronization window length, which is equal to the value of FSYNC_CTRL[FSYNCLEN] + 1. See Frame synchronization for more information.

46.7.13 Version ID (VERID)

Contains the major number, minor number, and feature specification number.

Offset

Register	Offset
VERID	84h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 MAJOR	Major Version Number Returns the major version number for the module specification.
23-16 MINOR	Minor Version Number Returns the minor version number for the module specification.
15-0 FEATURE	Feature Specification Number Returns the feature set number.

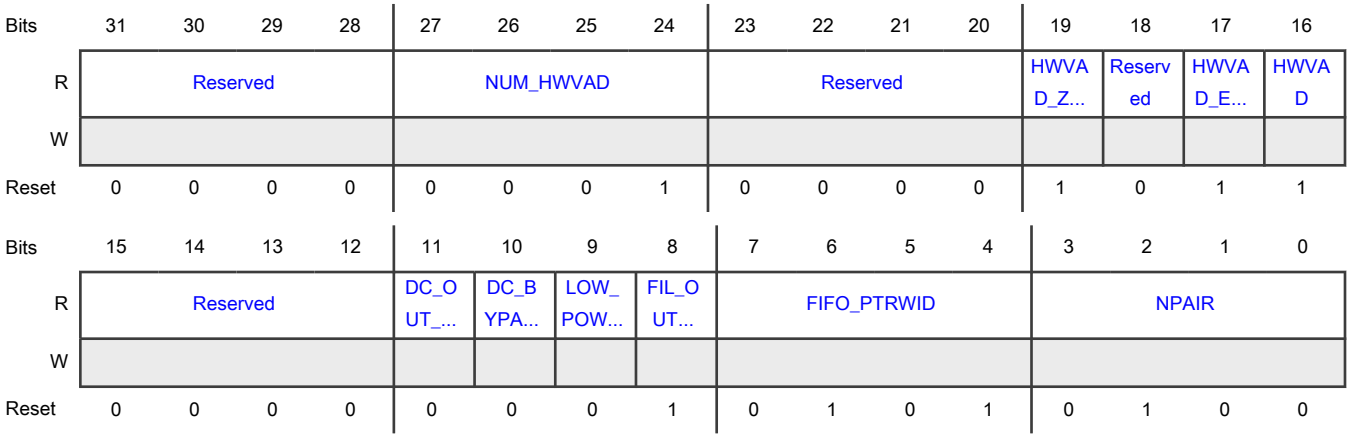
46.7.14 Parameter (PARAM)

Contains MICFIL's general configuration information.

Offset

Register	Offset
PARAM	88h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-24 NUM_HWVAD	Number of HWVADs Indicates the number of HWVADs.

Table continues on the next page...

Table continued from the previous page...

Field	Description
23-20 —	Reserved
19 HWVAD_ZCD	HWVAD ZCD Active 0 - Disabled 1 - Active
18 —	Reserved
17 HWVAD_ENERGY_MODE	HWVAD Energy Mode Active 0 - Disabled 1 - Active
16 HWVAD	HWVAD Active 0 - Disabled 1 - Active
15-12 —	Reserved
11 DC_OUT_BYPASS	Output DC Remover Bypass Indicates whether the DC remover at the output is permanently disabled (bypassed) in silicon. 0 - Active 1 - Disabled
10 DC_BYPASS	Input DC Remover Bypass Indicates whether the DC remover at the input is permanently disabled (bypassed) in silicon. 0 - Active 1 - Disabled
9 LOW_POWER	Low-Power Decimation Filter 0 - Disables 1 - Enables
8 FIL_OUT_WIDTH_24B	Filter Output Width 0 - 16 bits 1 - 24 bits
7-4 FIFO_PTRWID	FIFO Pointer Width 0000 - 0 bits

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0001 - 1 bit 0010 - 2 bits 0011-1110 - ... 1111 - 15 bits
3-0 NPAIR	Number of Microphone Pairs 0000 - None 0001 - 1 pair 0010 - 2 pairs 0011-1110 - ... 1111 - 15 pairs

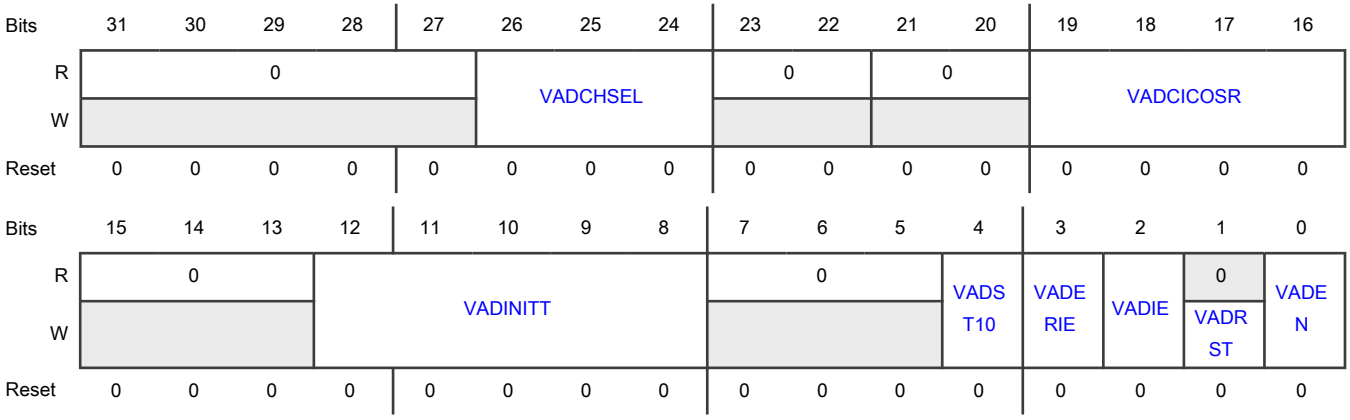
46.7.15 Voice Activity Detector 0 Control (VAD0_CTRL_1)

Contains HWVAD configuration bits including the channel selector, CIC oversampling rate (OSR) and initialization time configuration, internal filters initialization, interrupt, reset, and HWVAD enable bits.

Offset

Register	Offset
VAD0_CTRL_1	90h

Diagram



Fields

Field	Description
31-27	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
26-24 VADCHSEL	Channel Selector Selects the number of channels which HWVAD works with. **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 000 - PDM Microphone 0 Left 001 - PDM Microphone 0 Right 010 - PDM Microphone 1 Left 011 - PDM Microphone 1 Right 100-101 - ... 110 - PDM Microphone 3 Left 111 - PDM Microphone 3 Right
23-22 —	Reserved
21-20 —	Reserved
19-16 VADCICOSR	CIC Oversampling Rate Defines the OSR of the CIC filter. The OSR is (16-CICOSR). **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1.
15-13 —	Reserved
12-8 VADINITT	Initialization Time Selects the number of frames to be used to initialize the voice detection. The initialization time is the time taken by {[VADINITT] + 1} frames. During this period the output of the voice activity detector is forced to inactive. For more information, see Initialization time. **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 0_0000 - 0 0_0001 - 1 0_0010-1_1110 - ... 1_1111 - 31

Table continues on the next page...

Table continued from the previous page...

Field	Description
7-5 —	Reserved
4 VADST10	Internal Filters Initialization Initializes the signal and noise filter with pre-filter value. When [VADST10] = 1, there is no voice activity detection. **NOTE** If the use of this bit is required, it should stay pulsed by more than 2 cycles of the system clock. 0 - Normal operation 1 - Filters initialized
3 VADERIE	Error Interruption Enable Enables Error Interrupts in MICFIL. This error could be caused by an overflow in the input of the HWVAD and it is signalized by the VADINSATF flag. 0 - Disables 1 - Enables
2 VADIE	Interruption Enable Enables interrupts in MICFIL when voice activity event has been detected by HWVAD. See the VADIF for more information. 0 - Disables 1 - Enables
1 VADRST	HWVAD Reset Provides the CPU with the capability to initialize HWVAD through the target-bus interface. This bit always reads zero. This bit is self-cleared. See VAD soft-reset.
0 VADEN	HWVAD Enable Enables the use of HWVAD. **NOTE** You must not change the value of this field when STAT[BSY_FIL] = 1. 0 - Disables 1 - Enables

46.7.16 Voice Activity Detector 0 Control (VAD0_CTRL_2)

Contains HWVAD configuration bits including the frame energy, pre-filter, force output, frame time, input gain, and high-pass filter configuration for the VAD.

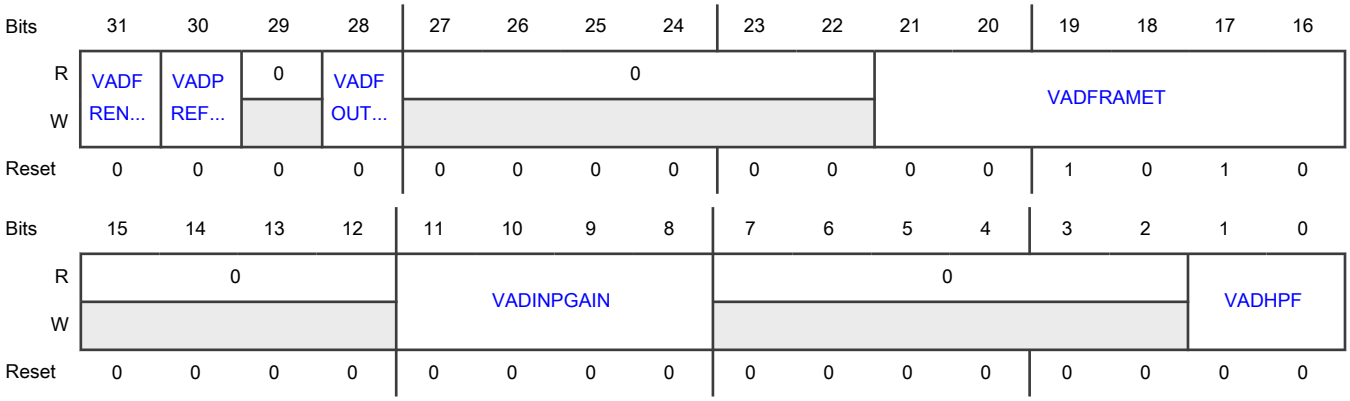
NOTE

You must not change the value of this register when STAT[BSY_FIL] = 1.

Offset

Register	Offset
VAD0_CTRL_2	94h

Diagram



Fields

Field	Description
31 VADFRENDIS	Frame Energy Disable Disables the calculus of energy in a frame. For more information, see Noise filter. 0 - Enables 1 - Disables
30 VADPREFEN	Pre Filter Enable Enables a pre-filter in the input of noise and signal filters. For more information, see Pre-filter. 0 - Pre-filter bypassed 1 - Pre-filter enabled
29 —	Reserved
28 VADFOUTDIS	Force Output Disable Disables the output of the Energy and Envelope-based Detector. 0 - Enables 1 - Disables
27-22 —	Reserved
21-16	Frame Time

Table continues on the next page...

Table continued from the previous page...

Field	Description
VADFRAMET	Selects the scale value to control the frame time duration as described in Frame time setting. **NOTE** [VADFRAMET] = 0 is not allowed. 00_0000 - 1 00_0001 - 2 00_0010-11_1110 - ... 11_1111 - 63
15-12 —	Reserved
11-8 VADINPGAIN	Input Gain Configures the gain value in the HWVAD input as a positive or negative (two's complement) number of bits to shift. 0000 - No shift 0001 - Shift 1 bit to the left 0010 - Shift 2 bits to the left 0011-0110 - ... 0111 - Shift 7 bits to the left 1000 - Shift 8 bits to the right 1001 - Shift 7 bits to the right 1010-1110 - ... 1111 - Shift 1 bits to the right
7-2 —	Reserved
1-0 VADHPF	High-Pass Filter Configures the internal high-pass filter. 00 - Filter bypassed 01 - Cut-off frequency at 1750 Hz 10 - Cut-off frequency at 215 Hz 11 - Cut-off frequency at 102 Hz

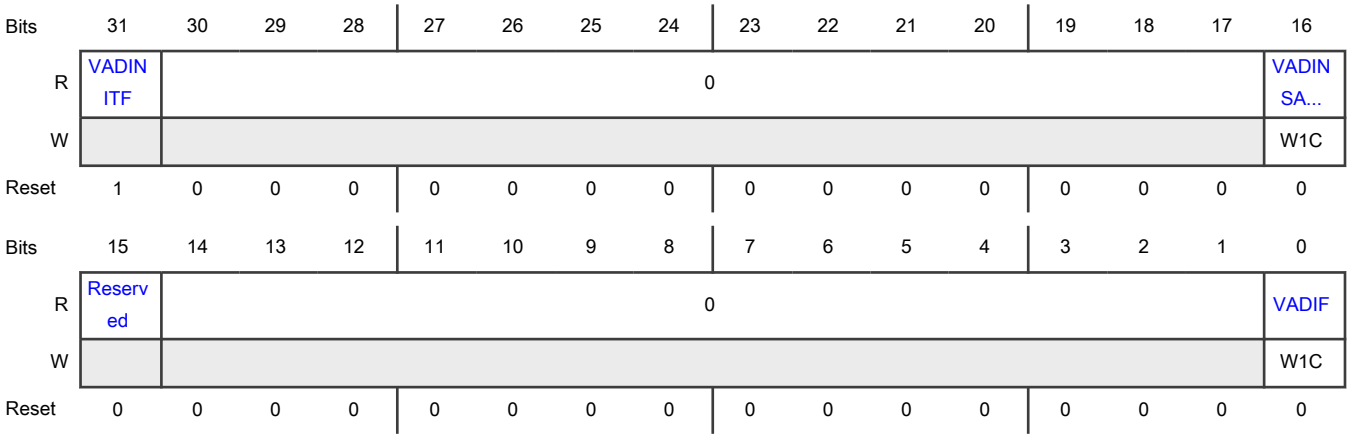
46.7.17 Voice Activity Detector 0 Status (VAD0_STAT)

Indicates the HWVAD status.

Offset

Register	Offset
VAD0_STAT	98h

Diagram



Fields

Field	Description
31 VADINITF	Initialization Flag Signalizes when the HWVAD is being initialized according to VAD*n*_CTRL_1[VADINITT] value. 0 - Not being initialized 1 - Being initialized
30-17 —	Reserved
16 VADINSATF	Input Saturation Flag Indicates an exceptional saturation condition by an overflow or underflow in the HWVAD input. 0 - No exception 1 - Exception
15 —	Reserved
14-1 —	Reserved
0 VADIF	Interrupt Flag Indicates when voice activity has been detected by HWVAD. Write a logic 1 to this field to clear this flag.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Not detected 1 - Detected

46.7.18 Voice Activity Detector 0 Signal Configuration (VAD0_SCONFIG)

Configures the HWVAD signal gain value and the signal filter.

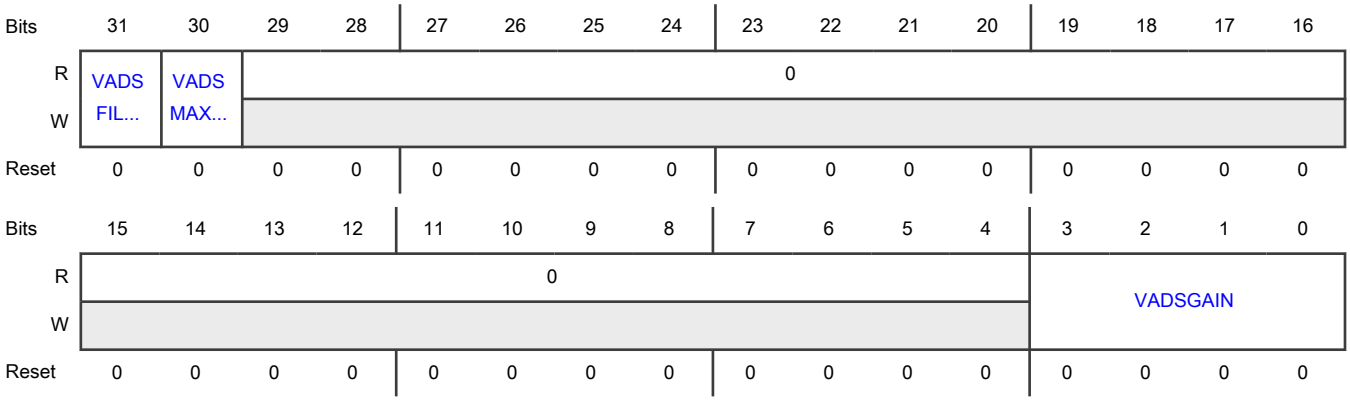
NOTE

These fields must be configured when MICFIL is not running.

Offset

Register	Offset
VAD0_SCONFIG	9Ch

Diagram



Fields

Field	Description
31 VADSFILEN	Signal Filter Enable Enables the signal filter. For more information, see Signal filter. 0 - Disables 1 - Enables
30 VADSMAXEN	Signal Maximum Enable Selects whether or not a block to calculate the maximum value between the input and output of signal filter is used as signal estimation. For more information, see Signal filter. 0 - Maximum block bypassed

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Maximum block enabled
29-4 —	Reserved
3-0 VADSGAIN	Signal Gain Configures the gain value for the signal energy or envelope estimated. It is a unsigned value which is multiplied with the estimated signal data. **NOTE** When [VADSGAIN] = 0, the signal gain corresponds to 1. 0000,0001 - Multiplier = 1 0010 - Multiplier = 2 0011-1110 - ... 1111 - Multiplier = 15

46.7.19 Voice Activity Detector 0 Noise Configuration (VAD0_NCONFIG)

Configures the HWVAD noise gain value and the noise filter.

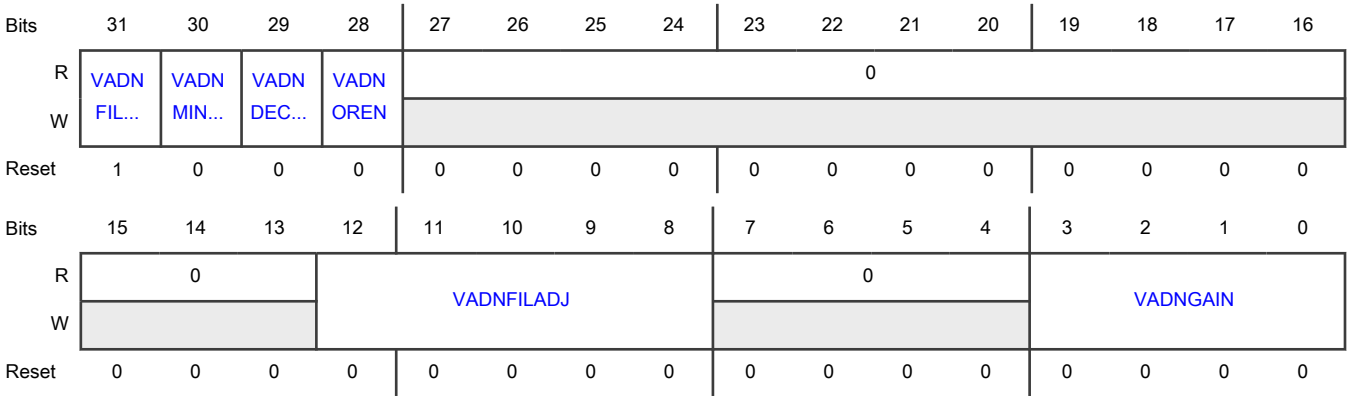
NOTE

These fields must be configured when MICFIL is not running.

Offset

Register	Offset
VAD0_NCONFIG	A0h

Diagram



Fields

Field	Description
31 VADNFILAUTO	Noise Filter Auto Selects whether the noise filter is activated automatically based on voice activity information. For more information, see Noise filter . 0 - Always enabled 1 - Enabled or disabled based on voice activity information
30 VADNMINEN	Noise Minimum Enable Selects whether or not a block to calculate the minimum value between the input and output of noise filter is used as noise estimation. For more information, see Noise filter . 0 - Minimum block bypassed 1 - Minimum block enabled
29 VADNDECEN	Noise Decimation Enable Selects whether or not the input of the noise filter is decimated. For more information, see Noise filter . 0 - Not decimated 1 - Decimated
28 VADNOREN	Noise OR Enable Enables an OR logic in the output of minimum noise estimator block. For more information, see Filter result gain setting . 0 - Not decimated 1 - Decimated
27-13 —	Reserved
12-8 VADNFILADJ	Noise Filter Adjustment Selects the adjustment value of the noise filter. For more information, see Noise filter. 0_0000 - 0 0_0001 - 1 0_0010-1_1110 - ... 1_1111 - 31
7-4 —	Reserved
3-0 VADNGAIN	Noise Gain Configures the gain value for the noise energy or envelope estimated. It is an unsigned value which is multiplied with the estimated noise data.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE When [VADNGAIN] = 0, the noise gain corresponds to 1. 0000,0001 - Multiplier = 1 0010 - Multiplier = 2 0011-1110 - ... 1111 - Multiplier = 15

46.7.20 Voice Activity Detector 0 Noise Data (VAD0_NDATA)

Stores the noise envelope calculated by HWVAD.

Offset

Register	Offset
VAD0_NDATA	A4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VADNDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 VADNDATA	Noise Data Stores the noise energy or noise envelope calculated by HWVAD. It can be used by software for a further voice activity detection.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field value should be ignored when VAD0_CTRL_2[VADFOUTDIS] = 1 or VAD0_CTRL_1[VADST10] = 1.

46.7.21 Voice Activity Detector 0 Zero-Crossing Detector (VAD0_ZCD)

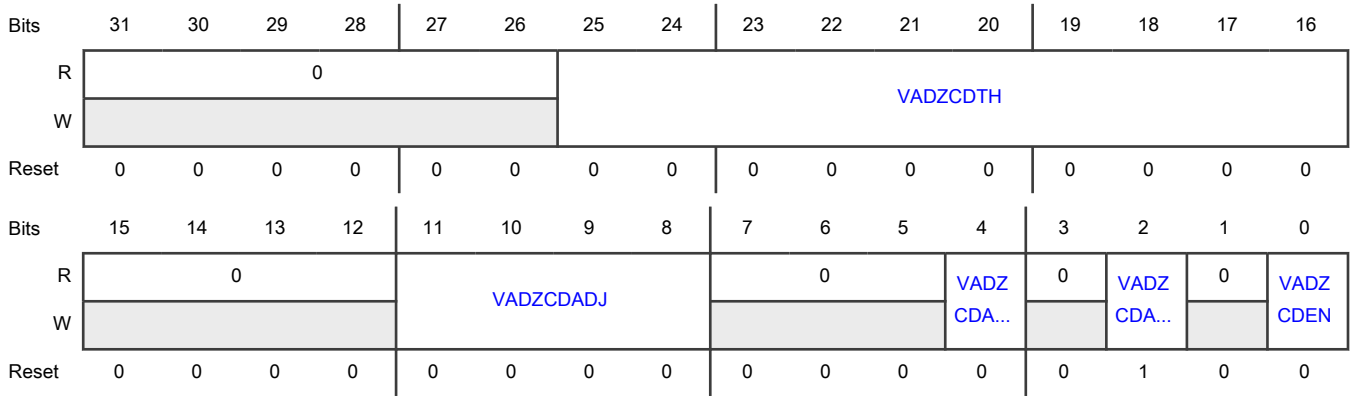
Configures the zero-crossing detector (ZCD).

NOTE
These fields must be configured when MICFIL is not running.

Offset

Register	Offset
VAD0_ZCD	A8h

Diagram



Fields

Field	Description
31-26 —	Reserved
25-16 VADZCDTH	ZCD Threshold Configures the initial value taken to estimate the ZCD threshold when it is estimated automatically ([VADZCDAUTO] = 1), otherwise ([VADZCDAUTO] = 0) this field is the static value of the ZCD threshold.
15-12 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
11-8 VADZCDADJ	ZCD Adjustment Selects the adjustment value of the threshold estimator. For more information, see Zero-crossing detector (ZCD). 0000 - 0 0001 - 1 0010-1110 - ... 1111 - 15
7-5 —	Reserved
4 VADZCDAND	ZCD AND Behavior Selects whether the ZCD results are combined with the energy-based detection using AND or OR to form the voice activity detection event. 0 - OR 1 - AND
3 —	Reserved
2 VADZCDAUTO	ZCD Automatic Threshold Disables the automatic estimation of the ZCD threshold. 0 - Disables 1 - Enables
1 —	Reserved
0 VADZCDEN	ZCD Enable Enables the ZCD in HWVAD. 0 - Disables 1 - Enables

Chapter 47
Synchronous Audio Interface (SAI)

47.1 Chip-specific SAI information

Table 486. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

This chip includes three instances of SAI (SAI1 in AONMIX, SAI2 and SAI3 in WAKEUPMIX). SAI1 supports 2 lanes TX and 1 lane RX. SAI-2 support to 4 lanes TX and 4 lanes RX. SAI-3 support up to 2-channels TX (1 lanes) and 2-channels RX (2 lanes).

47.1.1 Audio Clocking

SAI clocking is shown in the following figure.

NOTE

SAI1 is clocked by SAI1_CLK_ROOT or SAI1_MCLK.

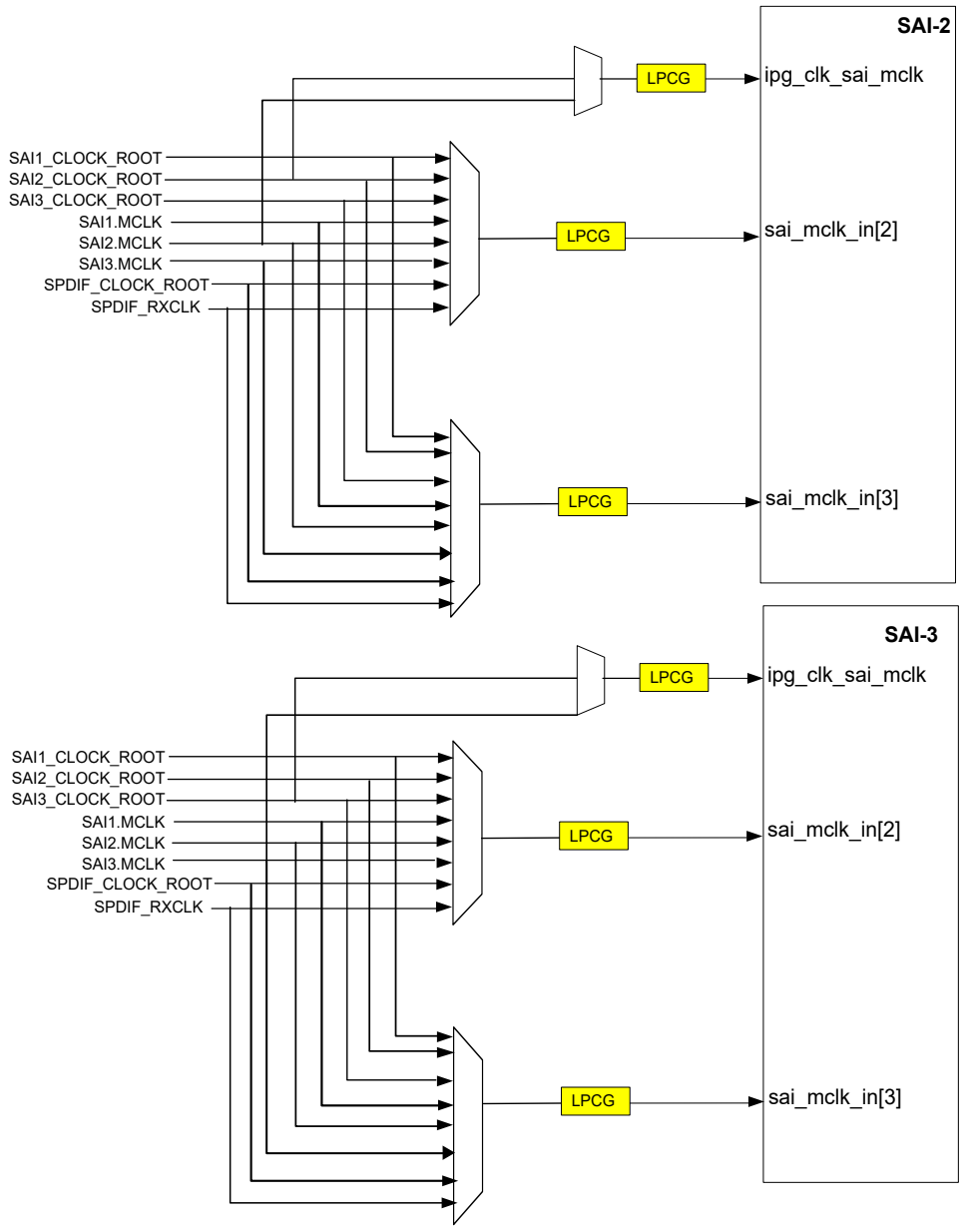


Figure 240. SAI clocking diagram

NOTE

The TX and RX sub-module inside each SAI can independently select one of the clock inputs as its master clock. This allows TX and RX of one SAI to run from different clock source.

47.2 Overview

SAI provides an interface that supports full-duplex serial digital audio interfaces with frame synchronization formats such as I^2S, AC97, TDM, and Codec/DSP interfaces.

47.2.1 Block diagram

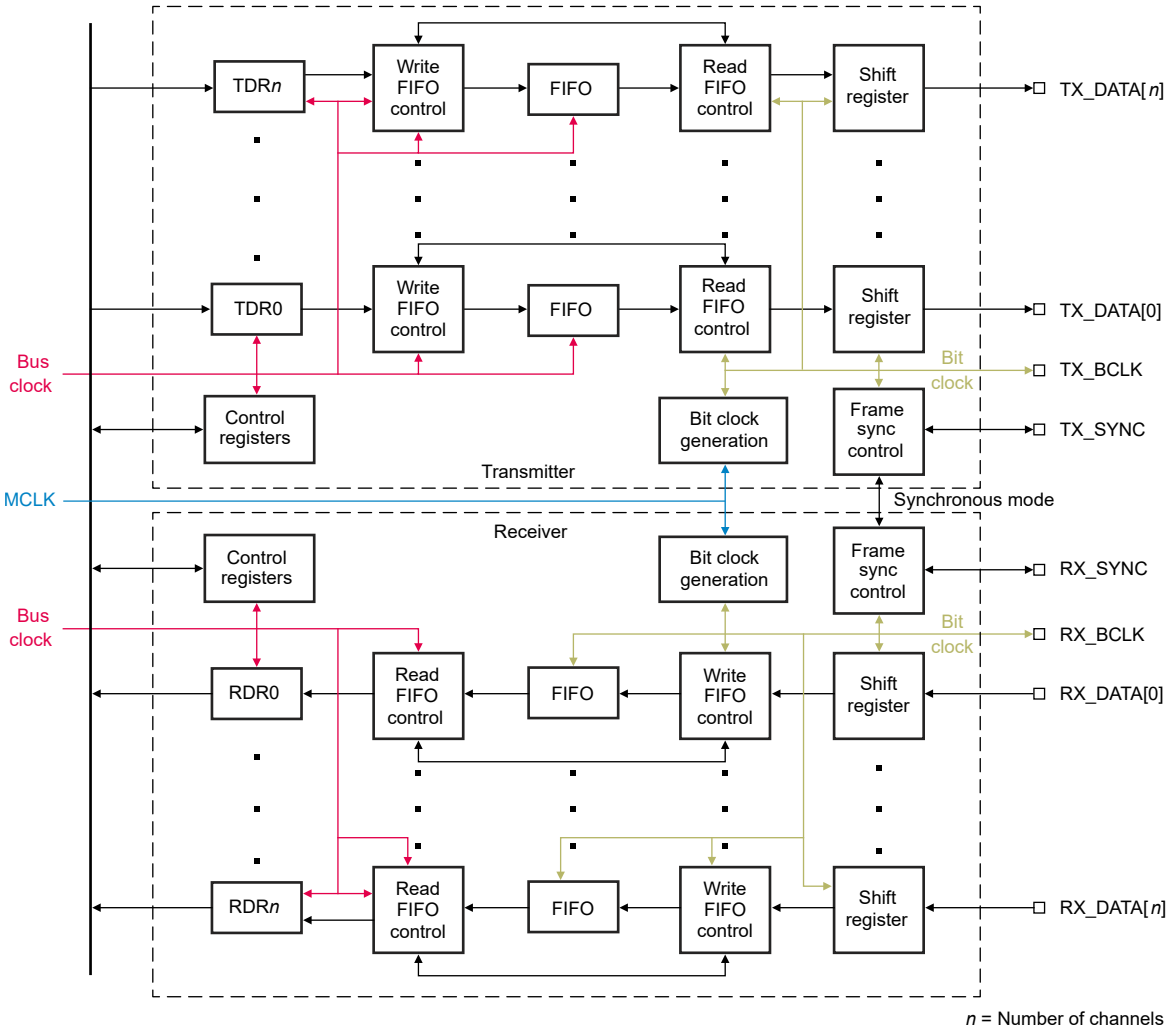


Figure 241. Block diagram

47.2.2 Features

Features can vary among chips and among SAI modules on the same chip. See the chip-specific SAI information at the beginning of this chapter.

- Transmitter with independent bit clock and frame synchronization supporting 4 data lines
- Receiver with independent bit clock and frame synchronization supporting 4 data lines
- Maximum frame size of 32 words per data line
- Word length of 8 to 32 bits
- Word length configured separately for the first word and remaining words in a frame
- Asynchronous 128 × 32-bit FIFO for each transmit and receive data line supports:
  - Graceful restart after FIFO error.
  - Automatic restart after FIFO error without software intervention.
  - 8-bit and 16-bit data packing into each 32-bit FIFO word.

— Multiple-data-line FIFOs combining into single-data-line FIFO.

- Independent 32-bit timestamp counters and bit counters for monitoring transmit and receive progress

47.3 Functional description

This section provides a complete functional description of SAI.

47.3.1 Modes of operation

- Run mode
- Stop mode
- Debug mode

47.3.1.1 Run mode

In Run mode, the SAI transmitter and receiver operate normally.

47.3.1.2 Stop mode

There are two situations in which you can configure the SAI transmitter or receiver to continue operating in Stop mode:

- The SAI transmitter or receiver uses an externally generated bit clock.
- The SAI transmitter or receiver uses an audio clock (MCLK) that operates when the chip is in Stop mode.

Write 1 to TCSR[STOPE] to configure the transmitter to run in Stop mode, and write 1 to RCSR[STOPE] to configure the receiver to run in Stop mode.

The SAI transmitter and receiver can generate an asynchronous interrupt to wake the CPU from Stop mode.

When the chip enters Stop mode, if TCSR[STOPE] = 0, the transmitter is disabled after completing the current transmit frame. Similarly, when the chip enters Stop mode, if RCSR[STOPE] = 0, the receiver is disabled after completing the current receive frame. Entry into Stop mode is blocked (not acknowledged) when waiting for the transmitter and receiver to be disabled at the end of the current frame.

47.3.1.3 Debug mode

You can configure either or both the SAI transmitter and receiver to continue operating in Debug mode:

- Write 1 to TCSR[DBGE] to configure the transmitter to run in Debug mode.
- Write 1 to RCSR[DBGE] to configure the receiver to run in Debug mode.

When TCSR[DBGE] and RCSR[DBGE] are 0 and the chip enters Debug mode, SAI is disabled after completing the current transmit or receive frame. Debug mode does not affect the transmitter and receiver bit clocks.

47.3.2 Synchronous modes

The SAI transmitter and receiver can operate synchronously to each other. You can configure the SAI transmitter and receiver to operate with a synchronous bit clock and frame sync (see TCR2[SYNC] and RCR2[SYNC]).

If both the transmitter and receiver use the transmitter bit clock and frame sync:

- Configure the transmitter for asynchronous operation and the receiver for synchronous operation.
- Enable the receiver in Synchronous mode only after configuring both the transmitter and receiver.
- Enable the transmitter last and disable the transmitter first.

If both the transmitter and receiver use the receiver bit clock and frame sync:

- Configure the receiver for asynchronous operation and the transmitter for synchronous operation.
- Enable the transmitter in Synchronous mode only after configuring both the receiver and transmitter.

- Enable the receiver last and disable the receiver first.

When operating in Synchronous mode, the transmitter and receiver share only the bit clock, frame sync, and transmitter and receiver enable. Otherwise, the transmitter and receiver operate independently, although the configuration register settings must be consistent across both the transmitter and receiver.

47.3.3 Frame sync configuration

When enabled, SAI continuously transmits and receives frames of data. Each frame consists of a fixed number of words, with each word consisting of a fixed number of bits. Within each frame, you can mask any word, causing the receiver to ignore that word and the transmitter to 3-state during that word.

The frame sync signal indicates the start of a frame. A valid frame sync requires the detection of a rising edge (if active high) or falling edge (if active low). The transmitter or receiver cannot be busy with a previous frame. SAI ignores a valid frame sync (in Target mode) or does not generate one (in Controller mode) for the first four bit-clock cycles after enabling the transmitter or receiver.

You can configure the transmitter and receiver frame sync independently by using any of the following options:

- Generate externally or internally
- Configure to be active high or active low
- Assert with the first bit in the frame or one bit early
- Assert for a duration between one bit-clock cycle and the first word length
- Configure the frame length from 1 to 32 words
- Configure the word length from 8 to 32 bits, with separate configurations for the length of the first word and that of the remaining words
- Transmit or receive words MSB first or LSB first

You cannot change these configuration options after enabling the SAI transmitter or receiver.

47.3.4 Data FIFO

Each transmit and receive channel includes a 128 × 32-bit FIFO. Use Transmit Data (TDR0 - TDR3) and Receive Data (RDR0 - RDR3) to access FIFO data.

47.3.4.1 Data alignment

Data in the FIFO can be aligned anywhere within the 32-bit-wide register via the First Bit Shifted (FBT) configuration field. This field selects the bit index (between 31 and 0) of the first bit shifted.

Examples of supported data alignment and the required FBT configuration are shown in Figure 242 for LSB-first configurations and Figure 243 for MSB-first configurations.

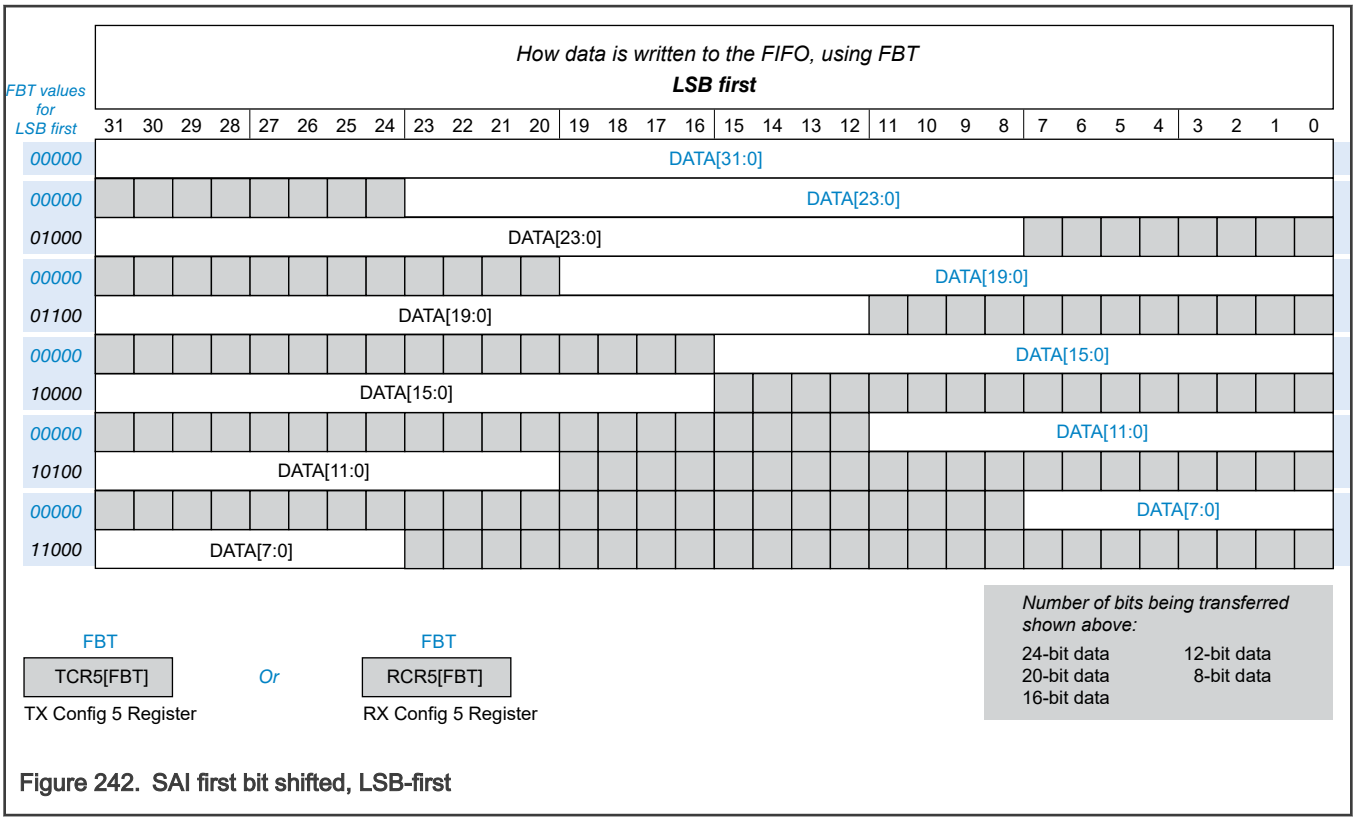


Figure 242. SAI first bit shifted, LSB-first

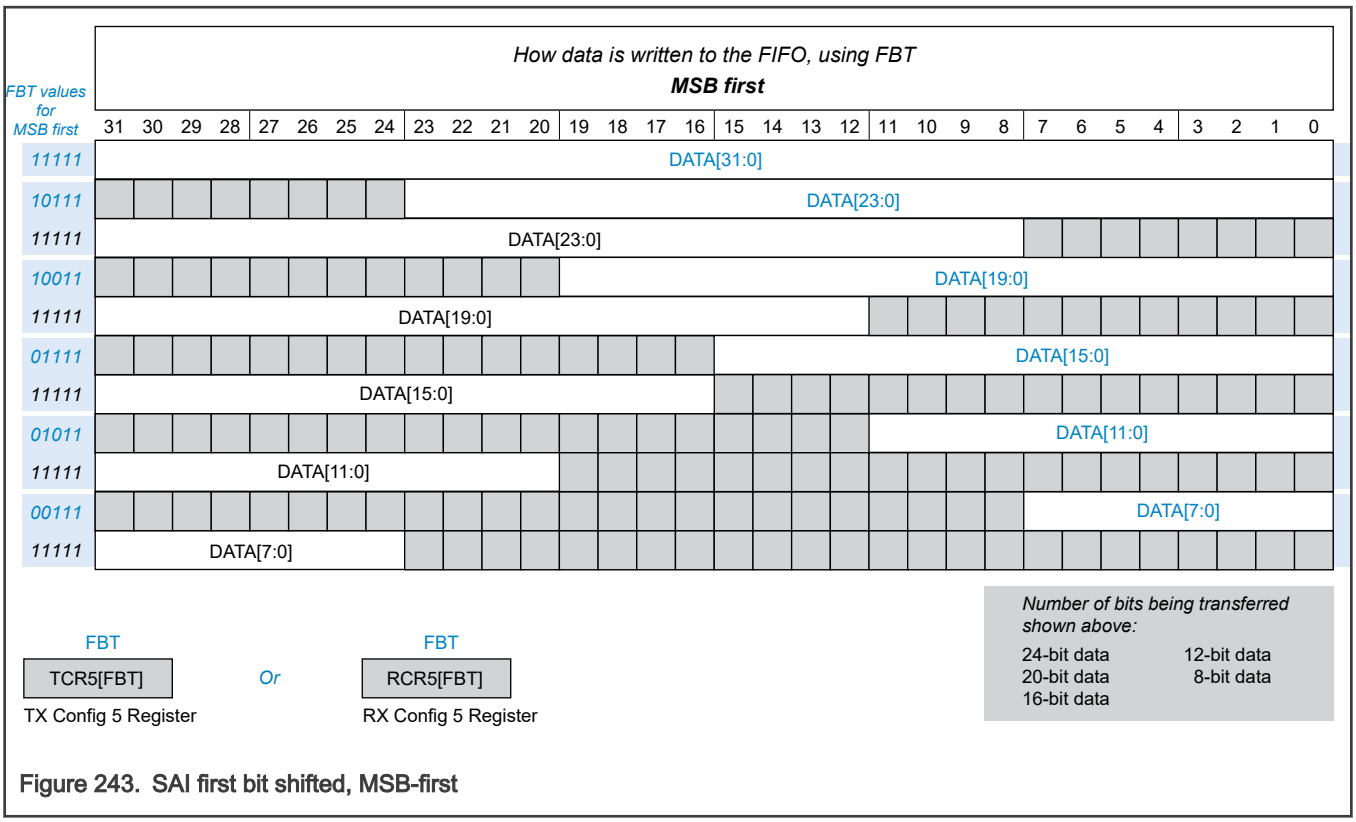


Figure 243. SAI first bit shifted, MSB-first

47.3.4.2 FIFO pointers

When writing to one of the Transmit Data (TDR0 - TDR3) registers, the write FIFO pointer of the corresponding Transmit FIFO register (TFR*n*[WFP]) increments after each valid write. SAI supports 8-bit, 16-bit, and 32-bit writes to TDR*n*, and the FIFO pointer increments after each individual write. Use only 8-bit writes when transmitting up to 8-bit data, and use only 16-bit writes when transmitting up to 16-bit data:

- If the transmit FIFO is full, SAI ignores writes to TDR*n*.
- If the transmit FIFO is empty, write to TDR*n* at least three bit-clock cycles before the start of the next unmasked word. This write avoids a FIFO underrun.
- Before enabling the transmitter, initialize the transmit FIFO with data. The transmitter starts a new frame after enabling the transmitter. If no data is in the FIFO, the transmitter immediately generates an error.

When reading one of the Receive Data (RDR0 - RDR3) registers, the read FIFO pointer of the corresponding Receive FIFO register (RFR*n*[RFP]) increments after each valid read. SAI supports 8-bit, 16-bit, and 32-bit reads from RDR*n*, and the FIFO pointer increments after each individual read. Use only 8-bit reads when receiving up to 8-bit data, and use only 16-bit reads when receiving up to 16-bit data:

- If the receive FIFO is empty, SAI ignores reads to RDR*n*.
- If the receive FIFO is full, read RDR*n* at least three bit-clock cycles before the end of an unmasked word. This read avoids a FIFO overrun.

47.3.4.3 FIFO packing

Using FIFO packing, you can store multiple 8-bit or 16-bit data words in one 32-bit FIFO word for the transmitter or receiver. You could emulate this feature by adjusting the number of bits per word and number of words per frame. FIFO packing, however, does not require even multiples of words per frame, and it fully supports word masking.

When FIFO packing is enabled, the FIFO pointers only increment when the full 32-bit FIFO word has been written (transmit) or read (receive). In this way, FIFO packing supports scenarios where different words within each frame are stored in different areas of memory.

Using 16-bit FIFO packing for transmitting, the transmit shift register loads at the start of each frame and after every second unmasked transmit word. The first transmitted word is taken from the 16-bit word at byte offset 0h. (The first bit is selected by TCR5[FBT], and you must configure it within this 16-bit word.) The second transmitted word is taken from the 16-bit word at byte offset 2h. (The first bit is selected by TCR5[FBT][3:0].) After the 16-bit word has been transmitted, the transmitter transmits logic zeroes until the start of the next word.

Using 16-bit FIFO packing for receiving, the receive shift register is stored after every second unmasked received word. If there is an odd number of unmasked-received words in each frame, the receive register is also stored, at the end of each frame. The first received word is stored in the 16-bit word at byte offset 0h. (The first bit is selected by RCR5[FBT], and you must configure it within this 16-bit word.) The second received word is stored in the 16-bit word at byte offset 2h. (The first bit is selected by RCR5[FBT][3:0].) The receiver ignores received data until the start of the next word after the 16-bit word has been received.

8-bit FIFO packing is similar to 16-bit packing, except four words are loaded or stored into each 32-bit FIFO word. The first word is stored in byte offset 0h, the second word in byte offset 1h, and so on. You must configure TCR5[FBT], RCR5[FBT], or both within byte offset 0h.

47.3.4.4 FIFO Combining mode

This mode enables separate FIFOs for multiple data channels to be used as a single FIFO for software accesses, as a single data channel, or both. The enabled data channels must be contiguous and data channel 0 must be enabled when using FIFO Combining mode.

You can combine FIFOs for software access by writing to transmit FIFO registers and reading from receive FIFO registers. Doing so enables a DMA controller or software to read or write multiple FIFOs without incrementing the address that is accessed. After FIFO Combining mode is enabled (TCR4[FCOMB] > 00b), the first software access to a FIFO register accesses the first enabled channel FIFO. The second access to a FIFO register accesses the second enabled channel FIFO. This process continues until

software accesses the last enabled channel FIFO and the pointer resets to the first enabled channel FIFO. To reset the pointer manually, you can reset the FIFOs or disable FIFO combining on software accesses.

Combining FIFOs for transmit data channels enables one data channel to use the FIFOs of all enabled channel FIFOs. In this case, the data output is identical on each enabled data channel. The transmit shift registers for all enabled data channels load at the start of each frame. The registers also load upon every *n*th unmasked word, where *n* is the number of enabled data channels. The first word transmitted loads from the first enabled channel FIFO; the second word transmitted loads from the second enabled channel FIFO. This loading continues until the end of the frame.

Combining FIFOs for receive data channels enables one data channel to use the FIFOs of all enabled channel FIFOs. In this case, the received data from channel 0 is stored in each enabled data channel. The receive shift registers for all enabled data channels are stored after every *n*th unmasked word, where *n* is the number of enabled data channels. The first word received is stored in the first enabled channel FIFO; the second word received is stored in the second enabled channel FIFO. This storage continues until the end of the frame.

NOTE

The first word in each frame is always stored in the first enabled data channel. NXP recommends that the number of unmasked words in each frame be evenly divisible by the number of enabled data channels.

Combining FIFOs for data channels loads or stores each channel FIFO at the same time. As a result, FIFO error conditions are only checked every *n*th word, where *n* is the number of enabled data channels. If any enabled data channel meets the warning flag or request flag conditions, SAI asserts the FIFO warning and request flags.

47.3.5 Word mask register

The SAI transmitter and receiver each have a word mask register (Transmit Mask (TMR) and Receive Mask (RMR)) that you can use to mask any word in the frame. The word mask register is double buffered. You can update it before the end of each frame to mask a particular word in the next frame.

TMR causes the transmit data pin to be 3-stated for the length of each selected word; the transmit FIFO is not read for masked words.

RMR causes the received data for each selected word to be discarded and not written to the receive FIFO.

47.3.6 Timestamp counter

The transmitter and receiver implement separate timestamp counters and bit counters. The bit counter increments at the end of each bit in a frame when the transmitter or receiver is enabled. The bit counter continues to increment if a word is masked or a FIFO underrun or overrun condition occurs. The bit counter does not increment if an external frame sync is configured and no frame sync is detected. You can reset the bit counter via software. Due to synchronization delays it can take up to 3 bit-clock cycles for the bit clock counter to continue incrementing after a bit counter reset.

The timestamp counter increments on the bus interface clock when it is enabled. You can configure it to synchronize to the bit counter by waiting for the bit counter to increment first (if configured, this synchronization is performed after the time counter is reset). The current value of the timestamp counter is latched whenever the bit counter increments. Reading the bit counter register causes the latched timestamp value to be saved in the bit counter timestamp register. You can reset the timestamp counter via software; a reset via software also resets the latched timestamp value and the bit counter timestamp register.

The timestamp counter can be synchronized between the transmitter and receiver; it can also be synchronized with the transmitter or receiver of another instance. When synchronizing timestamp counters, the controlling timestamp counter must be the last one enabled and the first one disabled. The timestamp counter software reset is not synchronized.

You can use the timestamp counter and bit counter to track the progress of the transmitter and receiver. You can also use it to calculate the relative frequency of the bit clock compared to the bus interface clock.

47.3.7 Clocking

- Audio clock
- Bit clock

- Bus clock

47.3.7.1 Audio clock

The audio clock (MCLK) generates the bit clock when you configure the receiver or transmitter for an internally generated bit clock. The transmitter and receiver can independently select between the bus clock and up to three audio clocks to generate the bit clock.

The SAI peripheral controls the pin direction and postdivider of an audio clock. The postdivider can divide down the clock output on the MCLK signal pin without affecting the audio clock that the transmitter and receiver use.

The audio clock generation and selection is chip-specific. See chip-specific clocking information about how the audio clocks are generated.

47.3.7.2 Bit clock

The SAI transmitter and receiver support asynchronous free-running bit clocks that an audio clock can generate internally or an external source can supply. SAI can also have synchronous bit clock and frame sync operation between the receiver and transmitter, or between multiple SAI peripherals. The transmitter and receiver configuration affects the bit clock and frame sync in the following ways:

- If you configure both the transmitter and receiver for asynchronous operation, each uses its own bit clock and frame sync.
- If you configure the transmitter for asynchronous operation and the receiver for synchronous operation, both use the transmitter bit clock and frame sync.
- If you configure the receiver for asynchronous operation and the transmitter for synchronous operation, both use the receiver bit clock and frame sync.

Your choice of synchronous or asynchronous operation selects the bit clock and frame sync used.

Externally generated bit clocks must be:

- Enabled before the SAI transmitter or receiver is enabled.
- Disabled after the SAI transmitter or receiver is disabled and completes its current frames.

In asynchronous operation, a SAI transmitter or receiver can use a bit clock externally generated by a SAI module instance that is disabled in Stop mode. In this case, disable the transmitter or receiver before entering Stop mode. This issue does not apply when the transmitter or receiver is in synchronous operation because all synchronous SAI modules are enabled and disabled simultaneously.

47.3.7.3 Bus clock

The control and configuration registers use the bus clock to generate synchronous interrupts and DMA requests.

NOTE

Although no minimum bus clock frequency is specified, the frequency must be fast enough (relative to the bit clock frequency) to serve the FIFOs. You must meet this requirement without generating a transmitter FIFO underrun or receiver FIFO overflow condition.

47.3.8 Reset

SAI is asynchronously reset on system reset. SAI has a Software reset and a FIFO reset.

47.3.8.1 Software reset

The SAI transmitter includes a software reset that resets all transmitter internal logic, including the bit clock generation, status flags, and FIFO pointers. The SAI receiver includes a software reset that resets all receiver internal logic, including bit clock generation, status flags, and FIFO pointers.

These software resets do not reset the configuration registers. The software resets remain asserted until you clear them via software.

47.3.8.2 FIFO reset

The SAI transmitter includes a FIFO reset that synchronizes the FIFO write pointer to the value of the FIFO read pointer. This FIFO reset empties the FIFO contents. Use this reset after TCSR[FEF] becomes 1, and before SAI reinitializes the FIFO and TCSR[FEF] becomes 0. SAI asserts the FIFO reset for one cycle only.

The SAI transmitter can also reset the FIFO of individual data channels by writing 1 to the appropriate bit in TCR3[CFR]. Use this reset only when the corresponding bit in TCR3[TCE] is 0.

The SAI receiver includes a FIFO reset that synchronizes the FIFO read pointer to the value of the FIFO write pointer. This FIFO reset empties the FIFO contents. Use this reset after RCSR[FEF] becomes 1 and SAI reads any remaining data from the FIFO, and before RCSR[FEF] becomes 0. SAI asserts the FIFO reset for one cycle only.

The SAI receiver can also reset the FIFO of individual data channels by writing 1 to the appropriate bit in RCR3[RCE]. Use this reset only when the corresponding bit in RCR3[RCE] is 0.

47.3.9 Interrupts and DMA requests

In SAI, the transmitter and receiver generate separate interrupts and DMA requests, but use the same status flags. SAI generates asynchronous versions of the transmitter and receiver interrupts to wake the CPU from Stop mode. SAI only generates asynchronous interrupts when the system clock is gated but the corresponding BCLK signal is active.

47.3.9.1 FIFO request flag

SAI sets the transmit FIFO request flag (TCSR[FRF]) when the number of entries in any enabled transmit FIFO is less than or equal to the transmit FIFO watermark configuration (TCR1[TFW]). SAI clears this flag when the number of entries in each enabled transmit FIFO is greater than TCR1[TFW].

SAI sets the receive FIFO request flag (RCSR[FRF]) when the number of entries in any enabled receive FIFO is greater than the receive FIFO watermark configuration (RCR1[RFW]). SAI clears this flag when the number of entries in each enabled receive FIFO is less than or equal to RCR1[RFW].

The FIFO request flag can generate an interrupt when TCSR[FRIE] = 1. It can generate a DMA request when TCSR[FRDE] = 1.

47.3.9.2 FIFO warning flag

SAI sets the transmit FIFO warning flag (TCSR[FWF]) when the number of entries in any of the enabled transmit FIFOs is empty. SAI clears this flag when the number of entries in each enabled transmit FIFO is not empty.

SAI sets the receive FIFO warning flag (RCSR[FWF]) when the number of entries in any of the enabled receive FIFOs is full. SAI clears this flag when the number of entries in each enabled receive FIFO is not full.

The FIFO warning flag can generate an interrupt when TCSR[FWIE] = 1. The flag can generate a DMA request when TCSR[FWDE] = 1.

47.3.9.3 FIFO error flag

SAI sets the transmit FIFO error flag (TCSR[FEF]) when any enabled transmit FIFOs underrun. After the error flag is set, all enabled transmit channels transmit zero data before SAI clears TCSR[FEF].

When you write 1 to TCR4[FCONT], the FIFO continues transmitting data following an underrun without software intervention. To ensure that data transmits in the correct order, the transmitter continues from the same word number in the frame that caused the FIFO to underrun. It continues only after new data is written to the transmit FIFO. In this case, clear TCSR[FEF] without reinitializing the transmit FIFOs.

SAI sets the receive FIFO error flag (RCSR[FEF]) when any enabled receive FIFO overflows. After the flag is set, all enabled receive channels discard their received data until SAI clears RCSR[FEF] and the next receive frame starts. Empty all enabled receive FIFOs before SAI clears RCSR[FEF].

When you write 1 to RCR4[FCONT], the FIFO continues receiving data following an overflow without software intervention. To ensure that data is received in the correct order, the receiver continues from the same word number in the frame that caused the

FIFO to overflow. It continues only after data has been read from the receive FIFO. In this case, clear RCSR[FEF] without emptying the receive FIFOs.

TCSR[FEF] and RSCR[FEF] can only generate an interrupt.

47.3.9.4 Sync error flag

SAI sets a sync error flag (TCSR[SEF] or RCSR[SEF]) when both of these conditions are true:

- The frame sync is generated externally.
- The external frame sync asserts when the transmitter or receiver is busy with the previous frame.

SAI ignores the external frame sync assertion and sets the sync error flag. When this flag is set, the transmitter or receiver continues checking for frame sync assertion when idle or at the end of each frame.

TCSR[SEF] and RCSR[SEF] can only generate an interrupt.

47.3.9.5 Word start flag

SAI sets the word start flag (TCSR[WSF]) at the start of the second bit-clock cycle for the selected word. You can select this word via TCR3[WDFL].

TCSR[WSF] can only generate an interrupt.

47.4 External signals

Name	Function	I/O
TX_BCLK	Transmit bit clock: the bit clock is an input when externally generated and an output when internally generated.	I/O
TX_SYNC	Transmit frame sync: the frame sync is an input sampled synchronously by the bit clock when externally generated. It is an output generated synchronously by the bit clock when internally generated.	I/O
TX_DATA[3:0]	Transmit data: the bit clock synchronously generates the transmit data; this signal is 3-stated when not transmitting a word.	O
RX_BCLK	Receive bit clock: the bit clock is an input when externally generated and an output when internally generated.	I/O
RX_SYNC	Receive frame sync: the frame sync is an input sampled synchronously by the bit clock when externally generated. It is an output generated synchronously by the bit clock when internally generated.	I/O
RX_DATA[3:0]	Receive data: the bit clock synchronously samples the receive data.	I
MCLK	Audio clock: the audio clock is an input when externally generated and an output when internally generated.	I/O

47.5 Initialization

To initialize SAI:

1. Enable the clock to SAI.
2. Reset the internal transmitter logic and receiver logic by writing 1 to TCSR[SR] and RCSR[SR], respectively.

47.6 Memory map and register definition

A read or write access to an address from offset 108h and above results in a bus error.

47.6.1 SAI register descriptions

SAI provides an interface that supports full-duplex serial digital audio interfaces with frame synchronization formats.

47.6.1.1 SAI memory map

SAI1 base address: 443B_0000h

SAI2 base address: 4265_0000h

SAI3 base address: 4266_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VERID)	32	R	0302_0002
4	Parameter (PARAM)	32	R	See section
8	Transmit Control (TCSR)	32	RW	0000_0000
C	Transmit Configuration 1 (TCR1)	32	RW	0000_0000
10	Transmit Configuration 2 (TCR2)	32	RW	0000_0000
14	Transmit Configuration 3 (TCR3)	32	RW	0000_0000
18	Transmit Configuration 4 (TCR4)	32	RW	0000_0000
1C	Transmit Configuration 5 (TCR5)	32	RW	0000_0000
20 - 2C	Transmit Data (TDR0 - TDR3)	32	RW	See section
40 - 4C	Transmit FIFO (TFR0 - TFR3)	32	R	See section
60	Transmit Mask (TMR)	32	RW	0000_0000
70	Transmit Timestamp Control (TTCR)	32	RW	0000_0000
74	Transmit Timestamp (TTSR)	32	R	0000_0000
78	Transmit Bit Count (TBCR)	32	R	0000_0000
7C	Transmit Bit Count Timestamp (TBCTR)	32	R	0000_0000
88	Receive Control (RCSR)	32	RW	0000_0000
8C	Receive Configuration 1 (RCR1)	32	RW	0000_0000
90	Receive Configuration 2 (RCR2)	32	RW	0000_0000
94	Receive Configuration 3 (RCR3)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
98	Receive Configuration 4 (RCR4)	32	RW	0000_0000
9C	Receive Configuration 5 (RCR5)	32	RW	0000_0000
A0 - AC	Receive Data (RDR0 - RDR3)	32	R	See section
C0 - CC	Receive FIFO (RFR0 - RFR3)	32	R	See section
E0	Receive Mask (RMR)	32	RW	0000_0000
F0	Receive Timestamp Control (RTCR)	32	RW	0000_0000
F4	Receive Timestamp (RTSR)	32	R	0000_0000
F8	Receive Bit Count (RBCR)	32	R	0000_0000
FC	Receive Bit Count Timestamp (RBCTR)	32	R	0000_0000
100	MCLK Control (MCR)	32	RW	0000_0000

47.6.1.2 Version ID (VERID)

Contains version numbers for the module design and feature set.

Offset

Register	Offset
VERID	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fields

Field	Description
31-24	Major Version Number

Table continues on the next page...

Table continued from the previous page...

Field	Description
MAJOR	Indicates the major version number for the specification.
23-16 MINOR	Minor Version Number Indicates the minor version number for the specification.
15-0 FEATURE	Feature Specification Number Indicates the feature set number. 0000_0000_0000_0000 - Standard feature set 0000_0000_0000_0010 - Standard feature set with timestamp registers

47.6.1.3 Parameter (PARAM)

Contains parameter values implemented in the module.

Offset

Register	Offset
PARAM	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												FRAME			
W																
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FIFO				0				DATALINE			
W																
Reset	See Register reset values.															

Register reset values

Register	Reset value
PARAM	SAI1: 0005_0502h SAI2: 0005_0704h SAI3: 0005_0701h

Fields

Field	Description
31-20 —	Reserved
19-16 FRAME	Frame Size Describes the maximum number of words per frame, which is 2^FRAME.
15-12 —	Reserved
11-8 FIFO	FIFO Size Describes the number of words in each FIFO, which is 2^FIFO.
7-4 —	Reserved
3-0 DATALINE	Number of Data Lines Contains the number of data lines implemented.

47.6.1.4 Transmit Control (TCSR)

Contains transmitter enable fields including resets, error and interrupt enable fields, and error flag fields.

Offset

Register	Offset
TCSR	8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TE	STOPE	DBGE	BCE	0		0	SR	0			WSF	SEF	FEF	FWF	FRF
W							FR					W1C	W1C	W1C		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WSIE	SEIE	FEIE	FWIE	FRIE	0			0			FWDE	FRDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 TE	Transmitter Enable Enables the transmitter. When you write 0 to this field, the transmitter remains enabled, and this field remains 1 until the end of the current frame. 0 - Disable 1 - Enable (or transmitter has been disabled and has not yet reached the end of the frame)
30 STOPE	Stop Enable Enables transmitter operation in Stop mode. 0 - Disable 1 - Enable
29 DBGE	Debug Enable Enables transmitter operation in Debug mode, which does not affect the transmit bit clock. If you write 0 to this field, the transmitter operation is disabled after completing the current frame. 0 - Disable 1 - Enable
28 BCE	Bit Clock Enable Enables the transmit bit clock, separately from TCSR[TE]. This field automatically becomes 1 when TCSR[TE] becomes 1. When you write 0 to this field, the transmit bit clock remains enabled, and the field remains 1 until the end of the current frame. 0 - Disable 1 - Enable
27-26 —	Reserved
25 FR	FIFO Reset Empties the FIFO and sets the FIFO read and write pointers to the same value, which may or may not be zero. **NOTE** The FIFO reset is asserted for one cycle only. Reading this field always returns zero. SAI resets the FIFO pointers when the transmitter is disabled or the FIFO error flag is set. 0 - No effect 1 - FIFO reset
24 SR	Software Reset Resets the internal transmitter logic, including the FIFO read and write pointers.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Software-visible registers are not affected, except for the status registers. **NOTE** The software reset remains asserted until software writes 0 to the field. 0 - No effect 1 - Software reset
23-21 —	Reserved
20 WSF	Word Start Flag Indicates whether the start of the configured word has been detected. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not detected 1 - Detected When writing 0 - No effect 1 - Clear the flag
19 SEF	Sync Error Flag Indicates whether an error in the externally generated frame sync has been detected. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not detected 1 - Detected When writing 0 - No effect 1 - Clear the flag
18 FEF	FIFO Error Flag Indicates whether an enabled transmit FIFO has underrun. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Not detected 1 - Detected When writing 0 - No effect 1 - Clear the flag
17 FWF	FIFO Warning Flag Indicates whether an enabled transmit FIFO is empty. 0 - Not empty 1 - Empty
16 FRF	FIFO Request Flag Indicates whether the number of words in an enabled transmit channel FIFO is less than or equal to the transmit FIFO watermark. 0 - Watermark not reached 1 - Watermark reached
15-13 —	Reserved
12 WSIE	Word Start Interrupt Enable Enables word start interrupts. 0 - Disable 1 - Enable
11 SEIE	Sync Error Interrupt Enable Enables sync error interrupts. 0 - Disable 1 - Enable
10 FEIE	FIFO Error Interrupt Enable Enables FIFO error interrupts. 0 - Disable 1 - Enable
9 FWIE	FIFO Warning Interrupt Enable Enables FIFO warning interrupts. 0 - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable
8 FRIE	FIFO Request Interrupt Enable Enables FIFO request interrupts. 0 - Disable 1 - Enable
7-5 —	Reserved
4-2 —	Reserved
1 FWDE	FIFO Warning DMA Enable Enables the DMA warning. 0 - Disable 1 - Enable
0 FRDE	FIFO Request DMA Enable Enables DMA requests. 0 - Disable 1 - Enable

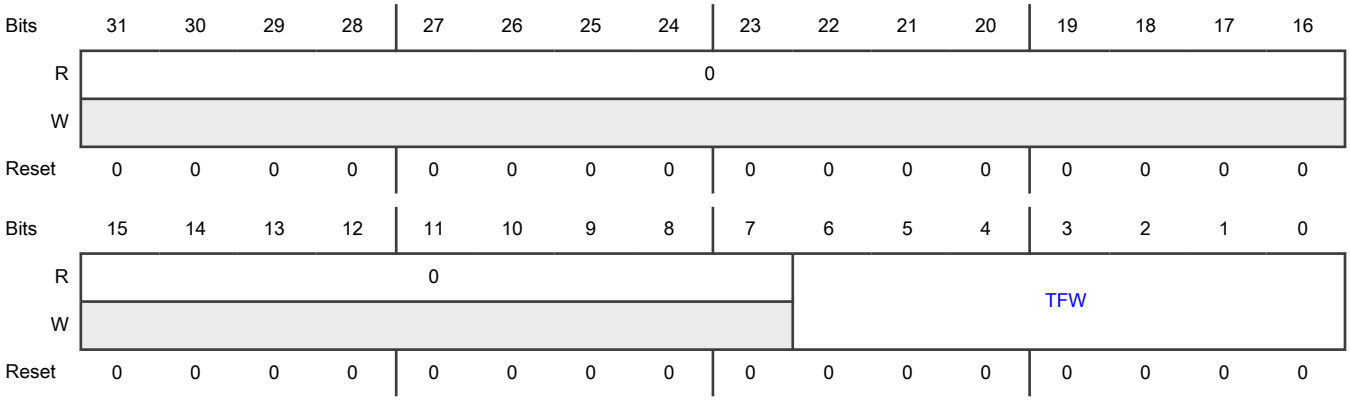
47.6.1.5 Transmit Configuration 1 (TCR1)

Configures the watermark level for all enabled transmit channels.

Offset

Register	Offset
TCR1	Ch

Diagram



Fields

Field	Description
31-7 —	Reserved
6-0 TFW	Transmit FIFO Watermark Indicates the number of 32-bit FIFO words in the watermark level of the transmit FIFO. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
SAI1	TCR1[4–0]	TCR1[6–5]
SAI2	TCR1	—
SAI3	TCR1	—

NOTE
The descriptions of the field settings vary by module instance.

Instance	Field value and description
SAI1	0_0000 - 0 0_0001 - 1 0_0010-1_1110 - (TFW value) 1_1111 - 31
SAI2 SAI3	000_0000 - 0 000_0001 - 1

Table continued from the previous page...

Field	Description
	<table><tr><th>Instance</th><th>Field value and description</th></tr><tr><td></td><td>000_0010-111_1110 - (TFW value) 111_1111 - 127</td></tr></table>

47.6.1.6 Transmit Configuration 2 (TCR2)

Contains the SYNC mode and clock setting fields.

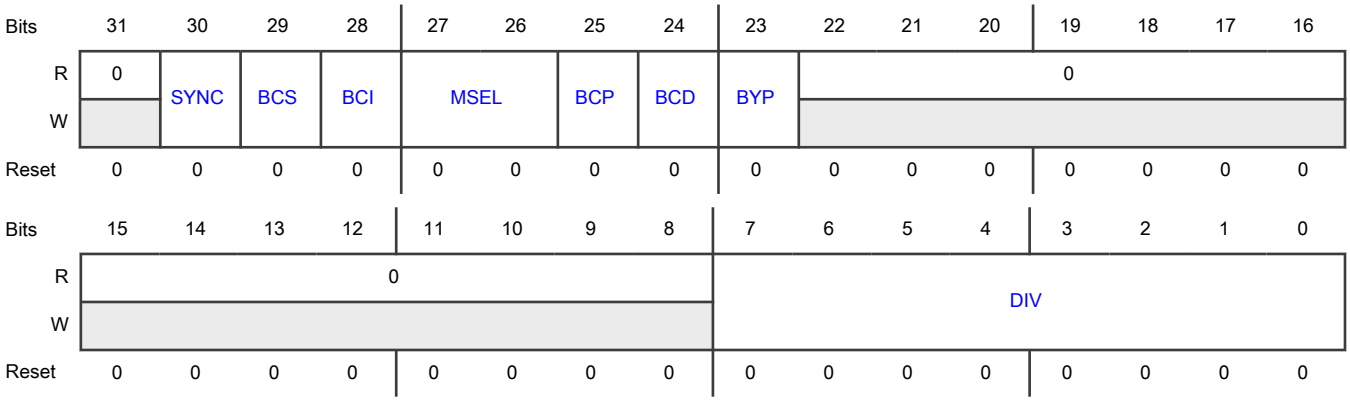
NOTE

Do not alter this register when TCSR[TE] is 1.

Offset

Register	Offset
TCR2	10h

Diagram



Fields

Field	Description
31 —	Reserved
30 SYNC	Synchronous Mode Configures Asynchronous and Synchronous modes of operation. When configured for Synchronous mode, you must configure the receiver for asynchronous operations.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Asynchronous mode 1 - Synchronous with receiver
29 BCS	Bit Clock Swap Swaps the bit clock used by the transmitter. When you configure the transmitter in Asynchronous mode and this field is 1, the receiver bit clock (RX_BCLK) clocks the transmitter. This configuration allows the transmitter and receiver to share a bit clock, while the transmitter continues to use the transmit frame sync (TX_SYNC). When you configure the transmitter in Synchronous mode, you must write the same value to the transmitter BCS field and the receiver BCS field. When both are 1, the transmitter and receiver are both clocked by the transmitter bit clock (TX_BCLK) but both use the receiver frame sync (RX_SYNC). 0 - Use the normal bit clock source 1 - Swap the bit clock source
28 BCI	Bit Clock Input Enables the internal logic to be clocked as if the bit clock is generated externally. When this field is 1 while using an internally generated bit clock in Synchronous or Asynchronous mode, the bit clock used by the transmitter is delayed by the pad output delay. (The pad input clocks the transmitter as if the clock is externally generated.) This setting decreases the data input setup time, but increases the data output valid time. Use the Target mode timing from the data sheet for the transmitter when this field is 1. In Synchronous mode, this field allows the transmitter to use the Target mode timing from the data sheet, while the receiver uses the Controller mode timing. This field has no effect when configured for an externally generated bit clock. **NOTE** When BCI = 1, both the input buffer and output buffer must be enabled for the BCLK pad. 0 - Disable 1 - Enable
27-26 MSEL	MCLK Select Selects the audio clock option used to generate an internally generated bit clock. This field has no effect when SAI is configured for an externally generated bit clock. **NOTE** Depending on the chip, some audio clock options might not be available. See the chip-specific information for the meaning of each option. 00 - Bus clock 01 - Audio clock (MCLK) option 1 10 - Audio clock (MCLK) option 2 11 - Audio clock (MCLK) option 3

Table continues on the next page...

Table continued from the previous page...

Field	Description
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. If you write 0 to this field, the bit clock becomes active high with drive outputs on rising edge and sample inputs on falling edge. If you write 1 to this field, the bit clock becomes active low with drive outputs on falling edge and sample inputs on rising edge. 0 - Active high 1 - Active low
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0 - Generate externally in Target mode 1 - Generate internally in Controller mode
23 BYP	Bit Clock Bypass Bypasses the bit clock divider. When bypassed, the internal bit clock is divide-by-one of the audio clock. When not bypassed, the internal bit clock is generated from the bit clock divider. 0 - Disable 1 - Enable
22-8 —	Reserved
7-0 DIV	Bit Clock Divide Determines the value by which SAI divides down the audio clock to generate the bit clock (when configured for an internal bit clock). The division value is (DIV + 1) × 2.

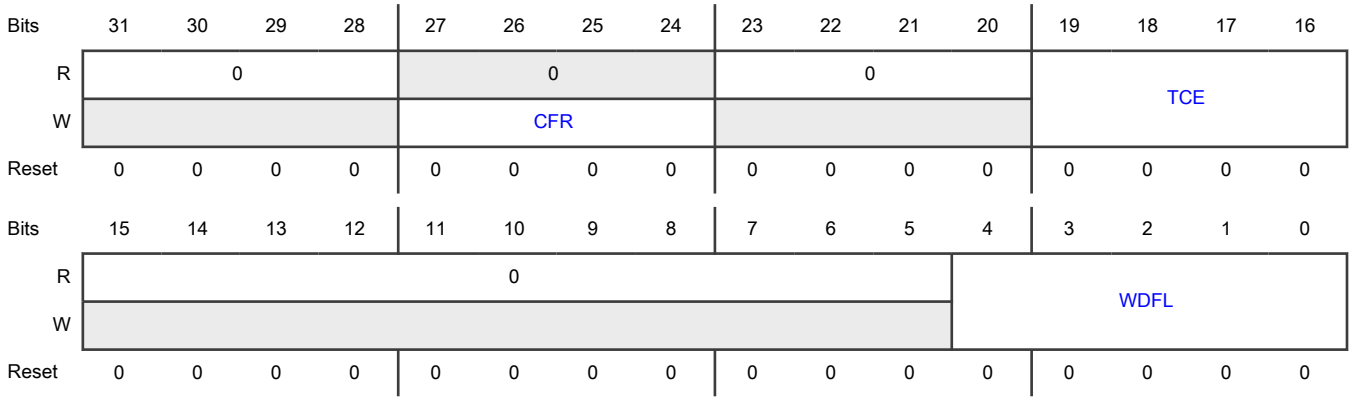
47.6.1.7 Transmit Configuration 3 (TCR3)

Contains the transmit channel settings.

Offset

Register	Offset
TCR3	14h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-24 CFR	Channel FIFO Reset Resets the FIFO pointers for a specific channel. Reading this field always returns zero. You must reset FIFO pointers only when a channel is disabled or the FIFO error flag is set. The width of this field = the number of transmit channels (call it *n*). For example, if this field is 2 bits wide, bit position 24 refers to the transmit channel 1 FIFO pointer and bit position 25 refers to the transmit channel 2 FIFO pointer. Writing 1 to bit 24 resets the transmit channel 1 FIFO pointer, and writing 1 to bit 25 resets the transmit channel 2 FIFO pointer. Writing 1 to bit *n* resets the transmit channel *n* FIFO pointer. **NOTE** When there is only one channel, you do not need the individual channel FIFO reset (TCR3[CFR]). For a single channel, use the global FIFO reset (TCSR[FR]). • 0b – No effect • 1b – Reset transmit data channel *n* FIFO **NOTE** This field is not supported in every instance. The following table includes only supported registers.
23-20	Reserved

Instance	Field supported in	Field not supported in
SAI1	TCR3[25–24]	TCR3[27–26]
SAI2	TCR3	—
SAI3	—	TCR3

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>—</td><td></td></tr>
<tr><td>19-16
TCE</td><td>Transmit Channel Enable
Enables the corresponding data channel for transmit operations. Changing this field takes effect immediately for generating the FIFO request and warning flags. It takes effect at the end of each frame for transmit operations.
The width of this field = the number of transmit channels (call it n). For example, if this field is two bits wide, bit position 16 refers to transmit channel 1 and bit position 17 refers to transmit channel 2. Writing 1 to bit 16 enables transmit channel 1, and setting bit 17 enables transmit channel 2. Writing 1 to bit n enables transmit channel n. For each transmit data channel:
• 0b – Disable
• 1b – Enable
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>SAI1</td><td>TCR3[17–16]</td><td>TCR3[19–18]</td></tr><tr><td>SAI2</td><td>TCR3</td><td>—</td></tr><tr><td>SAI3</td><td>TCR3[16]</td><td>TCR3[19–17]</td></tr></table></td></tr>
<tr><td>15-5
—</td><td>Reserved</td></tr>
<tr><td>4-0
WDFL</td><td>Word Flag Configuration
Selects the word that sets the word start flag (TCSR[WSF]). The value must be one less than the word number. For example, writing 0 selects the first word in the frame. When you configure this field with a value greater than TCR4[FRSZ], TCSR[WSF] is never set.</td></tr>
</table>

47.6.1.8 Transmit Configuration 4 (TCR4)

Contains the transmit fields for FIFO Combine mode, FIFO Packing mode, and frame sync settings.

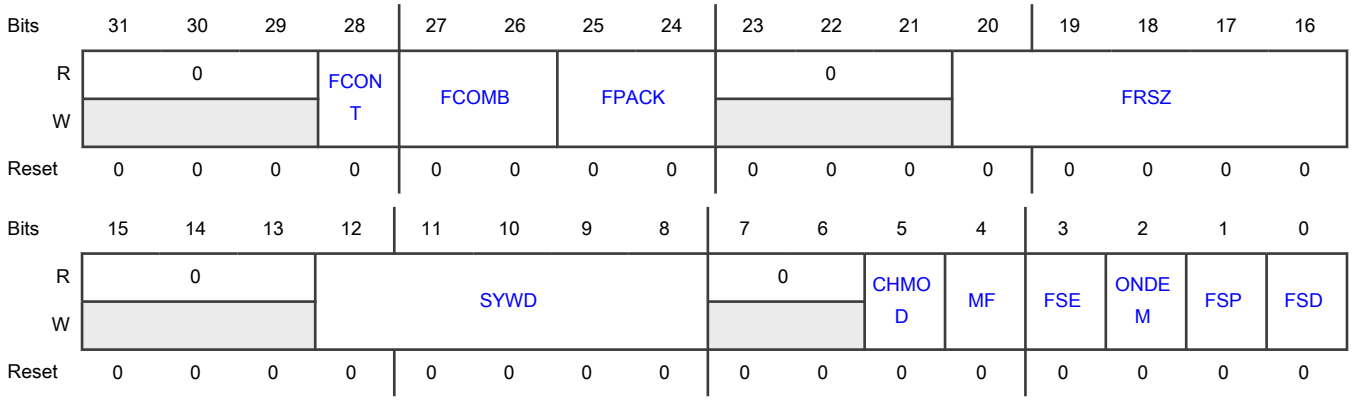
NOTE

Do not alter this register when TCSR[TE] is 1.

Offset

Register	Offset
TCR4	18h

Diagram



Fields

Field	Description
31-29 —	Reserved
28 FCONT	FIFO Continue on Error Configures when SAI must continue transmitting after a FIFO error is detected. When this field is 0 and a FIFO error occurs, SAI continues from the start of the next frame after the FIFO error flag clears. When this field is 1 and a FIFO error occurs, SAI continues from the same word that caused the FIFO error after the FIFO warning flag clears. 0 - Continue from the start of the next frame 1 - Continue from the same word that caused the FIFO error
27-26 FCOMB	FIFO Combine Mode Enables FIFO Combine mode for specified operations. When FIFO Combine mode is enabled for FIFO writes, software writing to any FIFO data register alternates the write among the enabled data channel FIFOs. For example, if two data channels are enabled: • The first write is performed to the first enabled data channel FIFO. • The second write is performed to the second enabled data channel FIFO. Resetting the FIFO or disabling FIFO Combine mode for FIFO writes resets the pointer back to the first enabled data channel. When FIFO Combine mode is enabled for FIFO reads from the transmit shift registers, the transmit data channel output alternates between the enabled data channel FIFOs. For example, if two data channels are enabled: • The first unmasked word is transmitted from the first enabled data channel FIFO. • The second unmasked word is transmitted from the second enabled data channel FIFO. Because the first word of the frame is always transmitted from the first enabled data channel FIFO, NXP recommends that the number of unmasked words per frame is evenly divisible by the number of enabled data channels.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>SAI1</td><td>TCR4</td><td>—</td></tr>
<tr><td>SAI2</td><td>TCR4</td><td>—</td></tr>
<tr><td>SAI3</td><td>—</td><td>TCR4</td></tr>
</table>
00 - Disable
01 - Enable on FIFO reads (from transmit shift registers)
10 - Enable on FIFO writes (by software)
11 - Enable on FIFO reads (from transmit shift registers) and writes (by software)
</td></tr>
<tr><td>25-24
FPACK</td><td>FIFO Packing Mode
Enables packing of 8-bit data or 16-bit data into each 32-bit FIFO word. If the word size is greater than 8 bits or 16 bits, only the first 8 bits or 16 bits are loaded from the FIFO. The first word in each frame always starts with a new 32-bit FIFO word and the first bit shifted must be configured within the first packed word. When FIFO packing is enabled, the FIFO write pointer only increments when you write the full 32-bit FIFO word.
00 - Disable FIFO packing
01 - Reserved
10 - Enable 8-bit FIFO packing
11 - Enable 16-bit FIFO packing</td></tr>
<tr><td>23-21
—</td><td>Reserved</td></tr>
<tr><td>20-16
FRSZ</td><td>Frame Size
Configures the number of words in each frame. The value must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 32 words.</td></tr>
<tr><td>15-13
—</td><td>Reserved</td></tr>
<tr><td>12-8
SYWD</td><td>Sync Width
Configures the length of the frame sync in number of bit-clock cycles. The value must be one less than the number of bit-clock cycles. For example, write 0 for the frame sync to assert for one bit-clock cycle only. You cannot configure the sync width to be longer than the first word of the frame.</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Description
7-6 —	Reserved
5 CHMOD	Channel Mode Specifies the mode of transmit data pins. In TDM mode, transmit data pins are 3-stated when slots are masked or channels are disabled. In Output mode, transmit data pins are never 3-stated, and they output zero when slots are masked or channels are disabled. 0 - TDM mode 1 - Output mode
4 MF	MSB First Configures what is transmitted first: LSB or MSB. 0 - LSB 1 - MSB
3 FSE	Frame Sync Early Determines when frame sync is asserted. 0 - First bit of the frame 1 - One bit before the first bit of the frame
2 ONDEM	On-Demand Mode Determines when the internal frame sync is generated. When this field is 1, and the frame sync is generated internally, a frame sync is only generated after the FIFO warning flag is cleared. 0 - Generated continuously 1 - Generated after the FIFO warning flag is cleared
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0 - Active high 1 - Active low
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0 - Generated externally in Target mode 1 - Generated internally in Controller mode

47.6.1.9 Transmit Configuration 5 (TCR5)

Contains transmit word width and bit index settings.

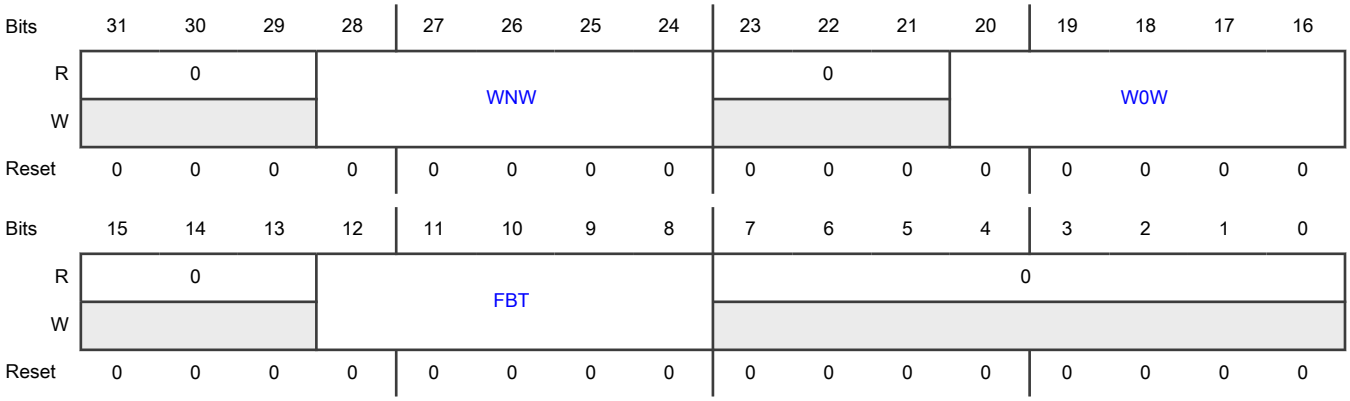
NOTE

Do not alter this register when TCSR[TE] is 1.

Offset

Register	Offset
TCR5	1Ch

Diagram



Fields

Field	Description
31-29 —	Reserved
28-24 WNW	Word N Width Configures the number of bits in each word, except for the first one in the frame. The value written must be one less than the number of bits per word. SAI does not support word widths of less than 8 bits. SAI does not support bit settings 0–6 (0b–0_0110b). 0_0111 - 8 0_1000 - 9 0_1001-1_1110 - (WNW value + 1) 1_1111 - 32
23-21 —	Reserved
20-16 W0W	Word 0 Width Configures the number of bits in the first word of each frame. The value written must be one less than the number of bits in the first word. SAI does not support word widths of less than 8 bits when there is only one word per frame. SAI does not support bit settings 0–6 (0b–0_0110b). 0_0111 - 8 0_1000 - 9

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0_1001-1_1110 - (W0W value + 1) 1_1111 - 32
15-13 —	Reserved
12-8 FBT	First Bit Shifted Configures the bit index for the first bit transmitted for each word in the frame. If configured for MSB-first, the index of the next bit transmitted is one less than the current bit transmitted. If configured for LSB-first, the index of the next bit transmitted is one more than the current bit transmitted. The value written must be greater than or equal to the word width when configured for MSB-first. The value written must be less than or equal to 31-word width when configured for LSB-first. See Data alignment for details. 0_0000 - 0 0_0001-1_1110 - FBT 1_1111 - 31
7-0 —	Reserved

47.6.1.10 Transmit Data (TDR0 - TDR3)

Contains transmit data.

When the transmit FIFO is not full, writes to this register push the data written into the transmit data FIFO. When the transmit FIFO is full, SAI ignores writes to this register.

Offset

Register	Offset
TDR0	20h
TDR1	24h
TDR2	28h
TDR3	2Ch

NOTE

Each module instance supports a different number of registers.

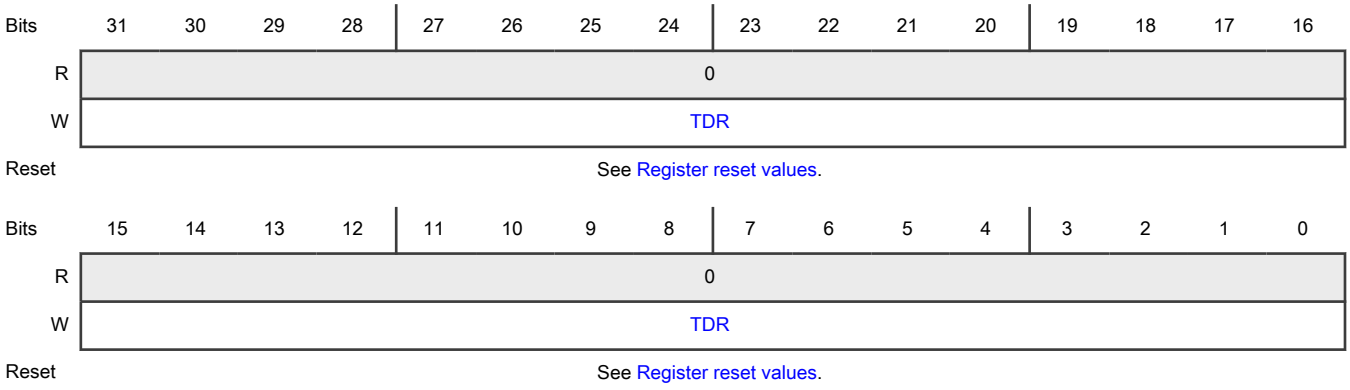
Instance	Register supported	Register not supported
SAI1	TDR0–TDR1	TDR2–TDR3

Table continues on the next page...

Table continued from the previous page...

Instance	Register supported	Register not supported
SAI2	TDR0–TDR3	—
SAI3	TDR0	TDR1–TDR3

Diagram



Register reset values

Register	Reset value
TDR0	SAI1–SAI3: 0000_0000h
TDR1	SAI1,SAI2: 0000_0000h SAI3: Register not supported
TDR2–TDR3	0000_0000h

Fields

Field	Description
31-0 TDR	Transmit Data

47.6.1.11 Transmit FIFO (TFR0 - TFR3)

Contains the transmit FIFO pointers. The MSB of the read and write pointers distinguishes between FIFO full and empty conditions. If the read and write pointers are identical, the FIFO is empty. If the read and write pointers are identical except for the MSB, the FIFO is full.

Offset

Register	Offset
TFR0	40h
TFR1	44h
TFR2	48h
TFR3	4Ch

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
SAI1	TFR0–TFR1	TFR2–TFR3
SAI2	TFR0–TFR3	—
SAI3	TFR0	TFR1–TFR3

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WCP	0							WFP							
W																
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RFP							
W																
Reset	See Register reset values.															

Register reset values

Register	Reset value
TFR0	SAI1–SAI3: 0000_0000h
TFR1	SAI1,SAI2: 0000_0000h SAI3: Register not supported
TFR2–TFR3	0000_0000h

Fields

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>31
WCP</td><td>Write Channel Pointer
Indicates whether this data channel is the next FIFO to be written when FIFO Combine mode is enabled for write operations.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>SAI1</td><td>TFR0–TFR1</td><td>—</td></tr><tr><td>SAI2</td><td>TFR0–TFR3</td><td>—</td></tr><tr><td>SAI3</td><td>—</td><td>TFR0</td></tr></table>
0 - No effect
1 - Next FIFO to be written</td></tr>
<tr><td>30-24
—</td><td>Reserved</td></tr>
<tr><td>23-16
WFP</td><td>Write FIFO Pointer
Indicates the FIFO write pointer for the transmit data channel.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>SAI1</td><td>TFR0–TFR1[21–16]</td><td>TFR0–TFR1[23–22]</td></tr><tr><td>SAI2</td><td>TFR0–TFR3</td><td>—</td></tr><tr><td>SAI3</td><td>TFR0</td><td>—</td></tr></table></td></tr>
<tr><td>15-8
—</td><td>Reserved</td></tr>
<tr><td>7-0
RFP</td><td>Read FIFO Pointer
Indicates the FIFO read pointer for the transmit data channel.</td></tr>
</table>

Table continued from the previous page...

Field	Description
	NOTE This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
SAI1	TFR0–TFR1[5–0]	TFR0–TFR1[7–6]
SAI2	TFR0–TFR3	—
SAI3	TFR0	—

47.6.1.12 Transmit Mask (TMR)

Contains the mask for the transmit word. This register is double-buffered and updates:

- When TCSR[TE] first becomes 1.
- At the end of each frame.

This setup allows the masked words in each frame to change from frame to frame.

Offset

Register	Offset
TMR	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TWM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TWM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 TWM	Transmit Word Mask Configures whether the transmit word *n* is masked for the corresponding word in the frame. In this case, being masked means that transmit data pins are 3-stated or driven zero and transmit data is not read from the FIFO. 0000_0000_0000_0000_0000_0000_0000_0000 - Enable 0000_0000_0000_0000_0000_0000_0000_0001 - Mask

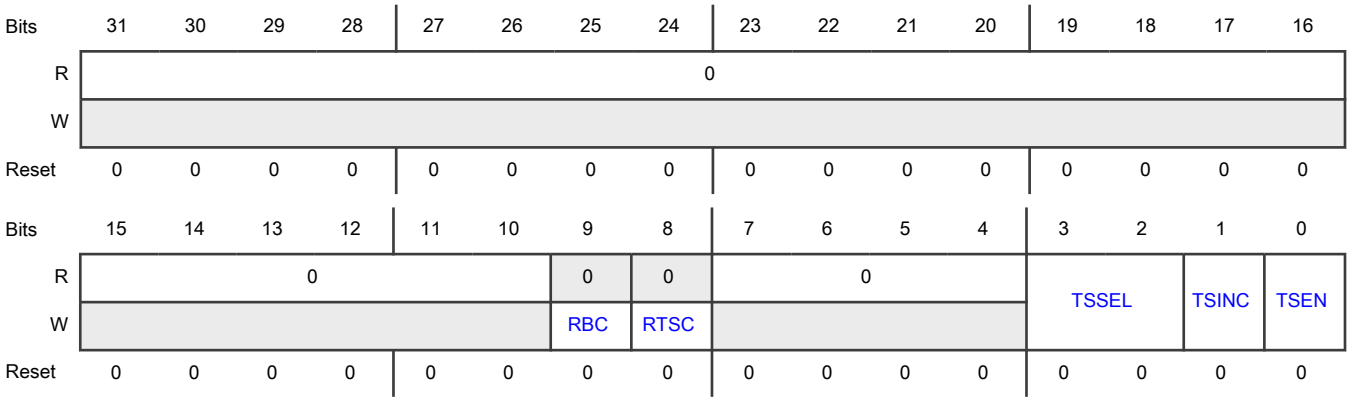
47.6.1.13 Transmit Timestamp Control (TTCR)

Controls the transmit timestamp counter and bit clock counter.

Offset

Register	Offset
TTCR	70h

Diagram



Fields

Field	Description
31-10 —	Reserved
9 RBC	Reset Bit Counter Resets the bit counter. 0 - No effect 1 - Reset

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 RTSC	Reset Timestamp Counter Resets the timestamp counter. 0 - No effect 1 - Reset
7-4 —	Reserved
3-2 TSSEL	Timestamp Select Configures whether the timestamp counter is synchronized with another timestamp counter. 00 - Increment when enabled 01 - Increment when the receive timestamp counter is enabled 10 - Increment when the transmit timestamp counter on another instance is enabled 11 - Increment when the receive timestamp counter on another instance is enabled
1 TSINC	Timestamp Increment Configures when the timestamp counter starts incrementing. 0 - When enabled and after the bit counter has incremented 1 - When enabled
0 TSEN	Timestamp Enable Enables the timestamp counter to increment on the bus interface clock. 0 - Disable 1 - Enable

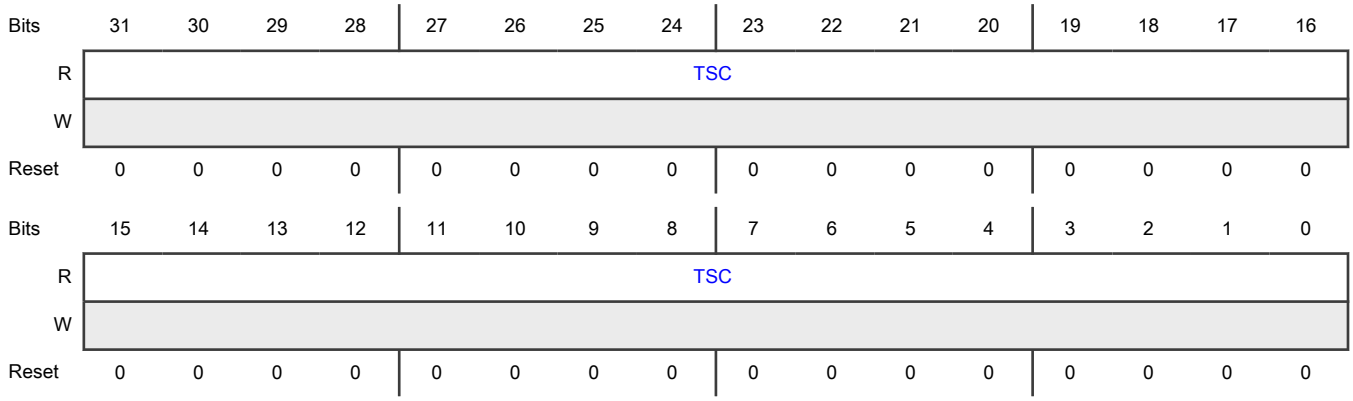
47.6.1.14 Transmit Timestamp (TTSR)

Indicates the current value of the transmit timestamp counter.

Offset

Register	Offset
TTSR	74h

Diagram



Fields

Field	Description
31-0 TSC	Timestamp Counter Indicates the value of the timestamp counter.

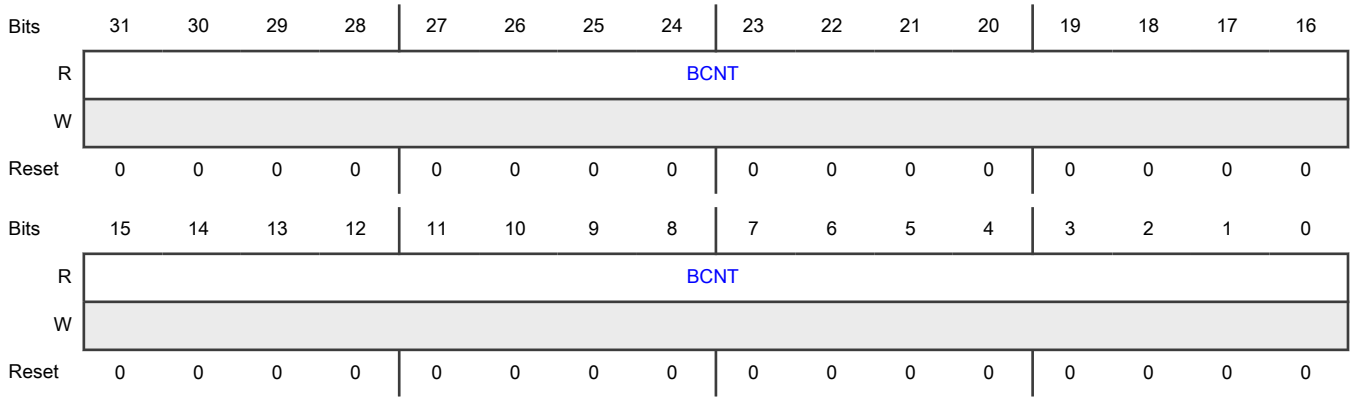
47.6.1.15 Transmit Bit Count (TBCR)

Indicates the current value of the transmit bit counter. The bit counter increments for each bit in a frame when the SAI transmitter is enabled. Reading this register causes Transmit Bit Count Timestamp (TBCTR) to update with the value of the timestamp counter that corresponds to the bit count register value.

Offset

Register	Offset
TBCR	78h

Diagram



Fields

Field	Description
31-0 BCNT	Bit Counter Indicates the current value of the bit counter.

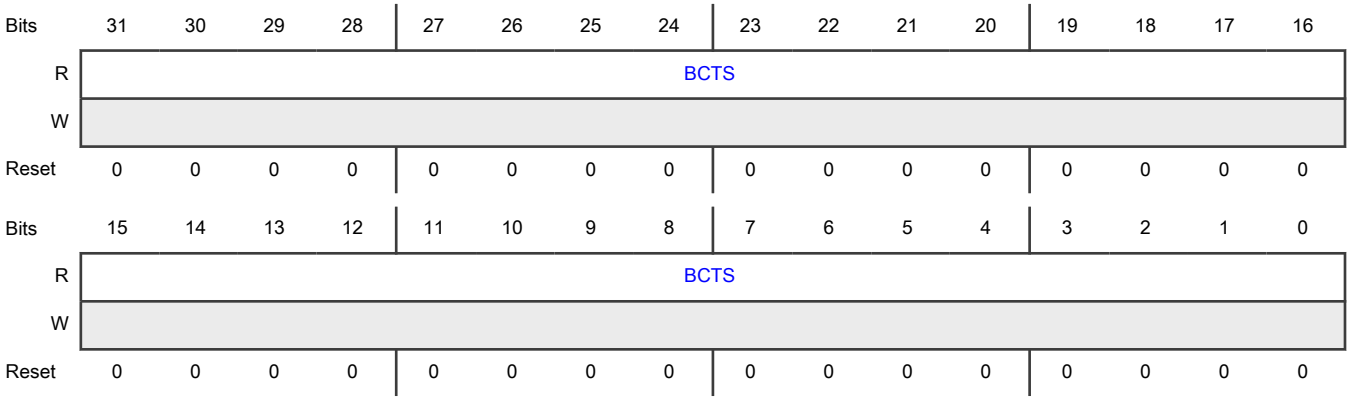
47.6.1.16 Transmit Bit Count Timestamp (TBCTR)

Indicates the value of the timestamp register that corresponds to the last value read from Transmit Bit Count (TBCR).

Offset

Register	Offset
TBCTR	7Ch

Diagram



Fields

Field	Description
31-0 BCTS	Bit Timestamp Indicates the timestamp counter value that matches the last read from Transmit Bit Count (TBCR).

47.6.1.17 Receive Control (RCSR)

Contains receiver enable fields including resets, error and interrupt enable fields, and error flag fields.

Offset

Register	Offset
RCSR	88h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RE	STOPE	DBGE	BCE	0		0	SR	0			WSF	SEF	FEF	FWF	FRF
W							FR					W1C	W1C	W1C		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WSIE	SEIE	FEIE	FWIE	FRIE	0			0			FWDE	FRDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 RE	Receiver Enable Enables the receiver. When you write 0 to this field, the receiver remains enabled and the field remains 1 until the end of the current frame. 0 - Disable 1 - Enable (or receiver disabled and not yet reached end of frame)
30 STOPE	Stop Enable Enables receiver operation in Stop mode. 0 - Disable 1 - Enable
29 DBGE	Debug Enable Enables receiver operation in Debug mode, which does not affect the receive bit clock. 0 - Disable after completing the current frame 1 - Enable
28 BCE	Bit Clock Enable Enables the receive bit clock, separately from RCSR[RE]. This field becomes 1 automatically when RCSR[RE] becomes 1. When you write 0 to this field, the receive bit clock remains enabled, and this field remains 1 until the end of the current frame. 0 - Disable 1 - Enable
27-26 —	Reserved
25	FIFO Reset

Table continues on the next page...

Table continued from the previous page...

Field	Description
FR	Empties the FIFO, and sets the FIFO read and write pointers to the same value, which can be zero. Reading this field always returns zero. **NOTE** The FIFO reset is asserted for one cycle only. You must reset FIFO pointers only when the receiver is disabled or the FIFO error flag is 1. 0 - No effect 1 - Reset
24 SR	Software Reset Resets the internal receiver logic including the FIFO pointers. Software-visible registers are not affected, except for the status registers. **NOTE** The software reset remains asserted until you clear it. 0 - No effect 1 - Software reset
23-21 —	Reserved
20 WSF	Word Start Flag Indicates whether SAI has detected the start of the configured word. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not detected 1 - Detected When writing 0 - No effect 1 - Clear the flag
19 SEF	Sync Error Flag Indicates whether SAI has detected an error in the externally generated frame sync. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not detected

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Detected When writing 0 - No effect 1 - Clear the flag
18 FEF	FIFO Error Flag Indicates whether an enabled receive FIFO has overflowed. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Receive overflow detected When writing 0 - No effect 1 - Clear the flag
17 FWF	FIFO Warning Flag Indicates whether an enabled receive FIFO is full. 0 - Not full 1 - Full
16 FRF	FIFO Request Flag Indicates whether the number of words in an enabled receive channel FIFO is greater than the receive FIFO watermark. 0 - Watermark not reached 1 - Watermark reached
15-13 —	Reserved
12 WSIE	Word Start Interrupt Enable Enables word start interrupts. 0 - Disable 1 - Enable
11 SEIE	Sync Error Interrupt Enable Enables sync error interrupts. 0 - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable
10 FEIE	FIFO Error Interrupt Enable Enables FIFO error interrupts. 0 - Disable 1 - Enable
9 FWIE	FIFO Warning Interrupt Enable Enables FIFO warning interrupts. 0 - Disable 1 - Enable
8 FRIE	FIFO Request Interrupt Enable Enables FIFO request interrupts. 0 - Disable 1 - Enable
7-5 —	Reserved
4-2 —	Reserved
1 FWDE	FIFO Warning DMA Enable Enables DMA warnings. 0 - Disable 1 - Enable
0 FRDE	FIFO Request DMA Enable Enables DMA requests. 0 - Disable 1 - Enable

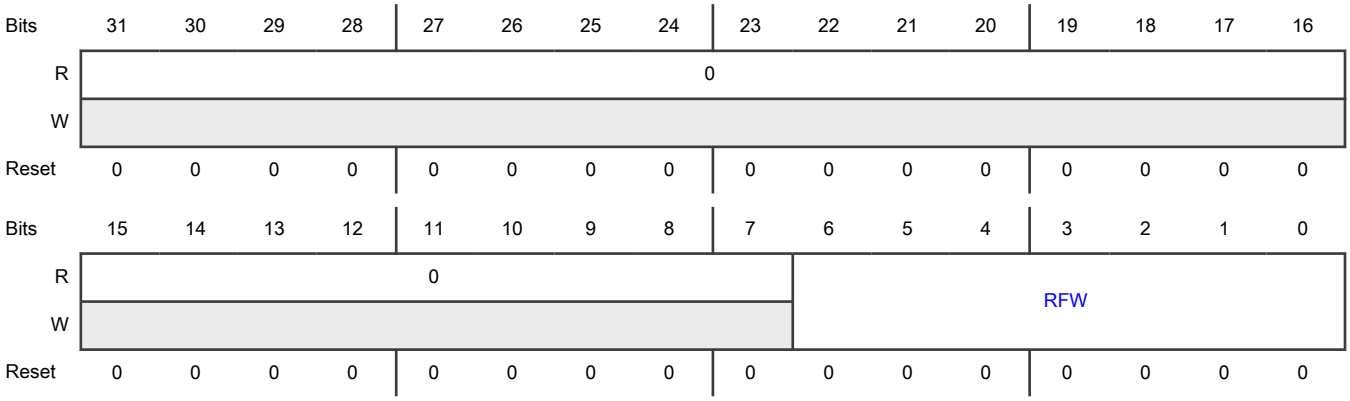
47.6.1.18 Receive Configuration 1 (RCR1)

Configures the watermark level for all enabled receiver channels.

Offset

Register	Offset
RCR1	8Ch

Diagram



Fields

Field	Description
31-7 —	Reserved
6-0 RFW	Receive FIFO Watermark Configures the level of the receive FIFO watermark, in 32-bit FIFO words. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
SAI1	RCR1[4–0]	RCR1[6–5]
SAI2	RCR1	—
SAI3	RCR1	—

NOTE

The descriptions of the field settings vary by module instance.

Instance	Field value and description
SAI1	0_0000 - 1 0_0001 - 2 0_0010-1_1110 - (RFW value + 1) 1_1111 - 32
SAI2 SAI3	000_0000 - 1 000_0001 - 2

Table continued from the previous page...

Field	Description
	<table><tr><th>Instance</th><th>Field value and description</th></tr><tr><td></td><td>000_0010-111_1110 - (RFW value + 1) 111_1111 - 128</td></tr></table>

47.6.1.19 Receive Configuration 2 (RCR2)

Contains the SYNC mode and clock setting fields.

NOTE

Do not alter this register when RCSR[RE] is 1.

Offset

Register	Offset
RCR2	90h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	SYNC	BCS	BCI	MSEL		BCP	BCD	BYP	0						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30 SYNC	Synchronous Mode Configures Asynchronous and Synchronous modes of operation. When configured for a Synchronous mode of operation, you must configure the transmitter for asynchronous operation.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Asynchronous mode 1 - Synchronous with transmitter
29 BCS	Bit Clock Swap Swaps the bit clock used by the receiver. When the receiver is configured in Asynchronous mode and this field is 1, the transmitter bit clock (TX_BCLK) clocks the receiver. This setting allows the transmitter and receiver to share one bit clock, while the receiver continues to use the receiver frame sync (RX_SYNC). When you configure the receiver in Synchronous mode, you must write the same value to the transmitter BCS field and receiver BCS field. When both are 1, the transmitter and receiver are both clocked by the receiver bit clock (RX_BCLK) but both use the transmitter frame sync (TX_SYNC). 0 - Use the normal bit clock source 1 - Swap the bit clock source
28 BCI	Bit Clock Input Enables the internal logic to be clocked as if the bit clock is generated externally. When this field is 1 and when using an internally generated bit clock in Synchronous or Asynchronous mode, the bit clock used by the receiver is delayed by the pad output delay. (The pad input clocks the receiver as if the clock is externally generated.) This setting decreases the data input setup time, but increases the data output valid time. Use the Target mode timing from the data sheet for the receiver when this field is 1. In Synchronous mode, this field allows the receiver to use the Target mode timing from the data sheet, while the transmitter uses the Controller mode timing. This field has no effect when configured for an externally generated bit clock. **NOTE** When this field is 1, both the input buffer and output buffer must be enabled for the BCLK pad. 0 - Disable 1 - Enable
27-26 MSEL	MCLK Select Selects the audio clock option used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. **NOTE** Some audio clock options are not available for some chips. See the chip-specific information for the availability and chip-specific meaning of each option. 00 - Bus clock 01 - Audio clock (MCLK) option 1 10 - Audio clock (MCLK) option 2 11 - Audio clock (MCLK) option 3
25	Bit Clock Polarity

Table continues on the next page...

Table continued from the previous page...

Field	Description
BCP	Configures the polarity of the bit clock. If you write 0 to this field, the bit clock becomes active high with drive outputs on the rising edge and sample inputs on the falling edge. If you write 1 to this field, the bit clock becomes active low with drive outputs on the falling edge and sample inputs on the rising edge. 0 - Active high 1 - Active low
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0 - Generated externally in Target mode 1 - Generated internally in Controller mode
23 BYP	Bit Clock Bypass Enables the bypass of the bit clock divider. When enabled, the internal bit clock is divide-by-one of the audio clock. When disabled, the internal bit clock is generated from the bit clock divider. 0 - Disable 1 - Enable
22-8 —	Reserved
7-0 DIV	Bit Clock Divide Determines the value by which the audio clock is divided to generate the bit clock, when configured for an internal bit clock. The division value is (DIV + 1) × 2.

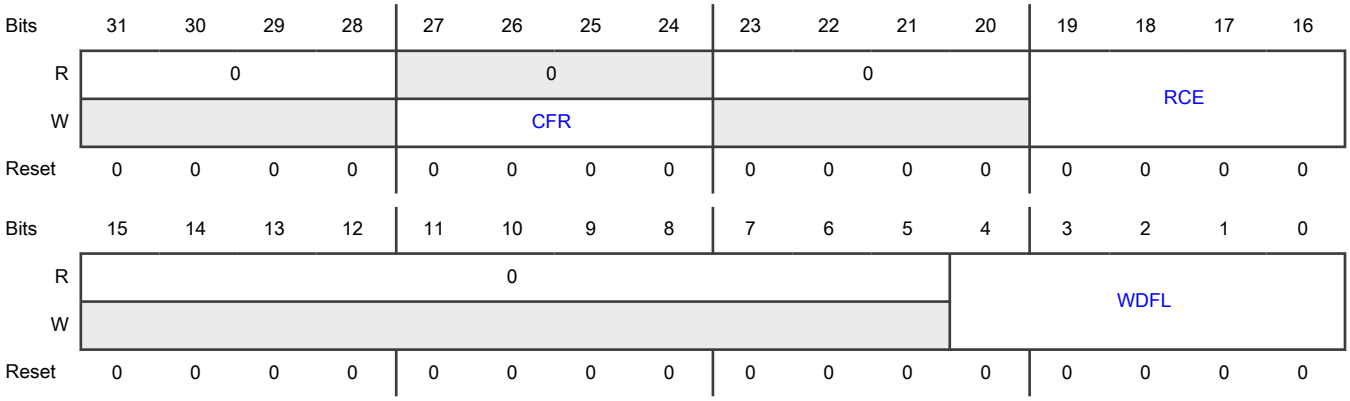
47.6.1.20 Receive Configuration 3 (RCR3)

Contains the receive channel settings.

Offset

Register	Offset
RCR3	94h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-24 CFR	Channel FIFO Reset Resets the FIFO pointers for a specific channel. Reading this field always returns zero. Reset FIFO pointers only when a channel is disabled or the FIFO error flag is set. The width of this field = the number of receive channels (call it *n*). For example, if this field is two bits wide, bit position 24 refers to the receive channel 1 FIFO pointer and bit position 25 refers to the receive channel 2 FIFO pointer. Writing 1 to bit 24 resets the receive channel 1 FIFO pointer, and writing 1 to bit 25 enables the receive channel 2 FIFO pointer. Writing 1 to bit *n* resets receive channel *n* FIFO pointer. • 0b – No effect • 1b – Reset receive data channel *n* FIFO **NOTE** This field is not supported in every instance. The following table includes only supported registers.
23-20 —	Reserved
19-16	Receive Channel Enable

Instance	Field supported in	Field not supported in
SAI1	RCR3[25–24]	RCR3[27–26]
SAI2	RCR3	—
SAI3	—	RCR3

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>RCE</td><td>Enables the corresponding data channel for receive operation. Changing this field takes effect immediately for generating the FIFO request and warning flags. It takes effect at the end of each frame for receive operations.

The width of RCE = the number of receive channels (call it n). For example, if RCE is two bits wide, then bit position 16 refers to receive channel 1 and bit position 17 refers to receive channel 2. Writing 1 to bit 16 enables receive channel 1, and writing 1 to bit 17 enables receive channel 2. Writing 1 to bit n enables receive channel n.

NOTE
When there is only a single channel, you do not need an individual channel FIFO reset (RCR3[CFR]). In that case, use the global FIFO reset (RCSR[FR]).

• 0b – Disable receive data channel n
• 1b – Enable receive data channel n

NOTE
This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>SAI1</td><td>RCR3[17–16]</td><td>RCR3[19–18]</td></tr>
<tr><td>SAI2</td><td>RCR3</td><td>—</td></tr>
<tr><td>SAI3</td><td>RCR3[16]</td><td>RCR3[19–17]</td></tr>
</table>
</td></tr>
<tr><td>15-5
—</td><td>Reserved</td></tr>
<tr><td>4-0
WDFL</td><td>Word Flag Configuration

Configures which word sets the word start flag (RCSR[WSF]). The value must be one less than the word number (for example, write zero to select the first word in the frame). If you configure this field to a value greater than RCR4[FRSZ], RCSR[WSF] is never set.

0_0000 - Word 1

0_0001 - Word 2

0_0010-1_1110 - Word (WDFL value + 1)

1_1111 - Word 32</td></tr>
</table>

47.6.1.21 Receive Configuration 4 (RCR4)

Contains the receive fields for FIFO Combine mode, FIFO Packing mode, and frame sync settings.

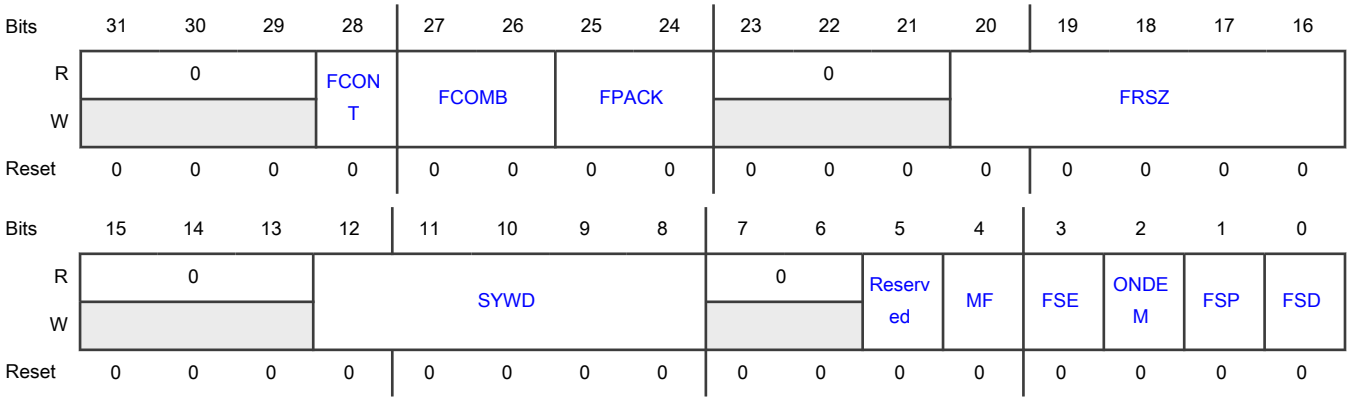
NOTE

Do not alter this register when RCSR[RE] is 1.

Offset

Register	Offset
RCR4	98h

Diagram



Fields

Field	Description
31-29 —	Reserved
28 FCONT	FIFO Continue on Error Configures when SAI continues receiving data after a FIFO error is detected. 0 - From the start of the next frame after the FIFO error flag is cleared 1 - From the same word that caused the FIFO error to become 1 after the FIFO warning flag is cleared
27-26 FCOMB	FIFO Combine Mode Enables FIFO Combine mode for specified operations. When FIFO Combine mode is enabled for FIFO reads, software reading any FIFO data register alternates the read among the enabled data channel FIFOs. For example, if two data channels are enabled: • The first read is performed to the first enabled data channel FIFO. • The second read is performed to the second enabled data channel FIFO. Resetting the FIFO or disabling FIFO Combine mode for FIFO reads resets the pointer back to the first enabled data channel. When FIFO Combine mode is enabled for FIFO writes from the receive shift registers, the first enabled data channel input alternates between the enabled data channel FIFOs. For example, if two data channels are enabled: • The first unmasked received word is stored in the first enabled data channel FIFO.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
The second unmasked received word is stored in the second enabled data channel FIFO.
Because the first word of the frame is always stored in the first enabled data channel FIFO, NXP recommends that the number of unmasked words per frame is evenly divisible by the number of enabled data channels.

NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>SAI1</td><td>RCR4</td><td>—</td></tr>
<tr><td>SAI2</td><td>RCR4</td><td>—</td></tr>
<tr><td>SAI3</td><td>—</td><td>RCR4</td></tr>
</table>
00 - Disable
01 - Enable on FIFO writes (from receive shift registers)
10 - Enable on FIFO reads (by software)
11 - Enable on FIFO writes (from receive shift registers) and reads (by software)
</td></tr>
<tr><td>25-24
FPACK</td><td>FIFO Packing Mode
Enables packing of 8-bit data or 16-bit data into each 32-bit FIFO word. If the word size is greater than 8 bits or 16 bits, only the first 8 bits or 16 bits are stored in the FIFO. The first word in each frame always starts with a new 32-bit FIFO word and the first bit shifted must be configured within the first packed word. When FIFO packing is enabled, the FIFO read pointer only increments when software reads the full 32-bit FIFO word.
00 - Disable
01 - Reserved
10 - Enable 8-bit FIFO packing
11 - Enable 16-bit FIFO packing</td></tr>
<tr><td>23-21
—</td><td>Reserved</td></tr>
<tr><td>20-16
FRSZ</td><td>Frame Size
Configures the number of words in each frame. The value must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 32 words.
0_0000 - 1
0_0001 - 2
0_0010-1_1110 - (FRSZ value + 1)</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1_1111 - 32
15-13 —	Reserved
12-8 SYWD	Sync Width Configures the length of the frame sync in number of bit-clock cycles. The value must be one less than the number of bit-clock cycles. For example, write 0 for the frame sync to assert for one bit-clock cycle only. You cannot configure this field to be longer than the first word of the frame. 0_0000 - 1 0_0001 - 2 0_0010-1_1110 - (SYWD value + 1) 1_1111 - 32
7-6 —	Reserved
5 —	Reserved Write only zero to this field.
4 MF	MSB First Configures what is received first: LSB or MSB. 0 - LSB 1 - MSB
3 FSE	Frame Sync Early Determines when frame sync is asserted. 0 - First bit of the frame 1 - One bit before the first bit of the frame
2 ONDEM	On-Demand Mode Determines when the internal frame sync is generated. When this field is 1, and the frame sync is generated internally, a frame sync is only generated when the FIFO warning flag is 0. 0 - Generated continuously 1 - Generated when the FIFO warning flag is 0
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0 - Active high 1 - Active low

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0 - Generated externally in Target mode 1 - Generated internally in Controller mode

47.6.1.22 Receive Configuration 5 (RCR5)

Contains receive word and bit index settings.

NOTE

Do not alter this register when RCSR[RE] is 1.

Offset

Register	Offset
RCR5	9Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				WNW				0				W0W			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FBT				0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28-24 WNW	Word N Width Configures the number of bits in each word except for the first one in the frame. The value must be one less than the number of bits per word. SAI does not support word widths less than 8 bits. SAI does not support bit settings 0–6 (0b–0_0110b).

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0_0111 - 8 0_1000 - 9 0_1001-1_1110 - (WNW value + 1) 1_1111 - 32
23-21 —	Reserved
20-16 W0W	Word 0 Width Configures the number of bits in the first word of each frame. The value must be one less than the number of bits in the first word. SAI does not support word widths less than 8 bits when there is only one word per frame. 0_0000 - 1 0_0001 - 2 0_0010-1_1110 - (W0W value + 1) 1_1111 - 32
15-13 —	Reserved
12-8 FBT	First Bit Shifted Configures the bit index for the first bit received for each word in the frame. If configured for MSB-first, the index of the next bit received is one less than the current bit received. If configured for LSB-first, the index of the next bit received is one more than the current bit received. The value must be greater than or equal to the word width when configured for MSB-first. The value must be less than or equal to 31-word width when configured for LSB-first. See Data alignment for details. 0_0000 - 0 0_0001-1_1110 - FBT value 1_1111 - 31
7-0 —	Reserved

47.6.1.23 Receive Data (RDR0 - RDR3)

Contains receive data.

When the receive FIFO is not empty, reads from this register return the data from the top of the receive FIFO. When the receive FIFO is empty, SAI ignores reads from this register.

Reading this register introduces one more peripheral clock wait state on each read.

Offset

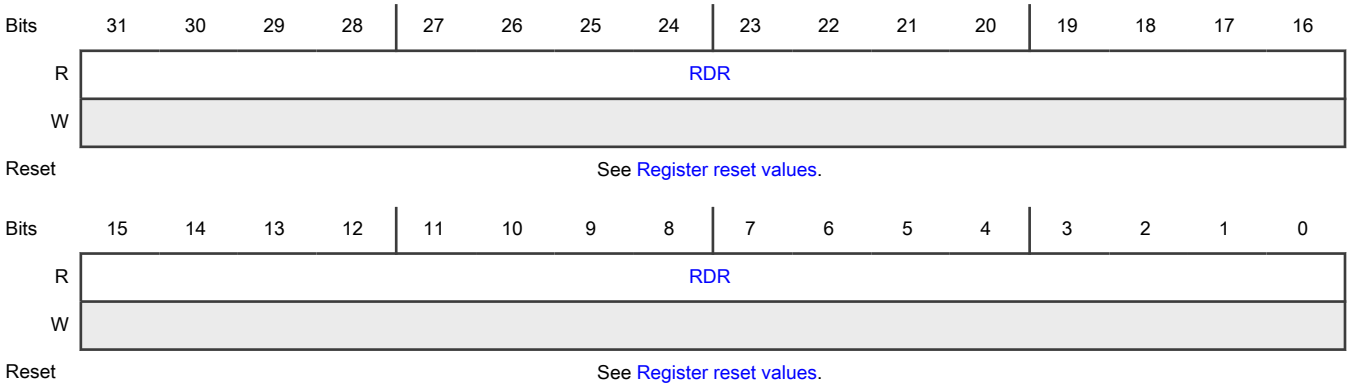
Register	Offset
RDR0	A0h
RDR1	A4h
RDR2	A8h
RDR3	ACh

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
SAI1	RDR0–RDR1	RDR2–RDR3
SAI2	RDR0–RDR3	—
SAI3	RDR0	RDR1–RDR3

Diagram



Register reset values

Register	Reset value
RDR0	SAI1–SAI3: 0000_0000h
RDR1	SAI1,SAI2: 0000_0000h SAI3: Register not supported
RDR2–RDR3	0000_0000h

Fields

Field	Description
31-0 RDR	Receive Data

47.6.1.24 Receive FIFO (RFR0 - RFR3)

Contains the receive FIFO pointers. The MSB of the read and write pointers distinguishes between FIFO full and empty conditions. If the read and write pointers are identical, the FIFO is empty. If the read and write pointers are identical except for the MSB, the FIFO is full.

Offset

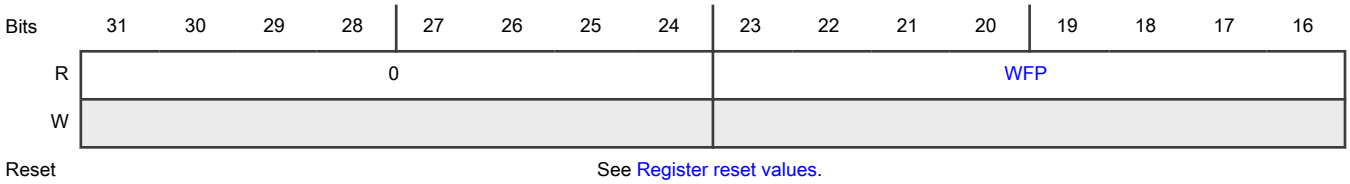
Register	Offset
RFR0	C0h
RFR1	C4h
RFR2	C8h
RFR3	CCh

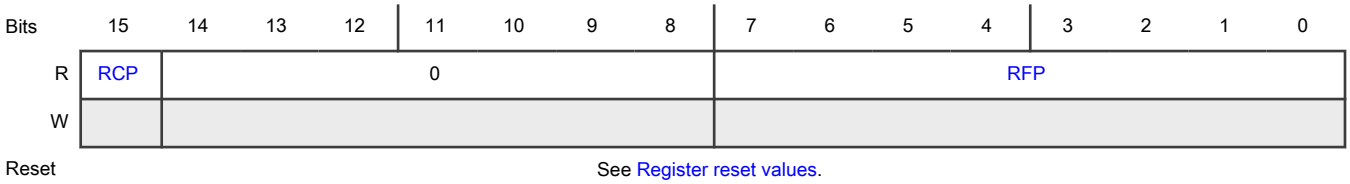
NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
SAI1	RFR0–RFR1	RFR2–RFR3
SAI2	RFR0–RFR3	—
SAI3	RFR0	RFR1–RFR3

Diagram





Register reset values

Register	Reset value
RFR0	SAI1–SAI3: 0000_0000h
RFR1	SAI1,SAI2: 0000_0000h SAI3: Register not supported
RFR2–RFR3	0000_0000h

Fields

Field	Description
31-24 —	Reserved
23-16 WFP	Write FIFO Pointer Indicates the FIFO write pointer for the receive data channel. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
SAI1	RFR0–RFR1[21–16]	RFR0–RFR1[23–22]
SAI2	RFR0–RFR3	—
SAI3	RFR0	—

Field	Description
15 RCP	Read Channel Pointer Indicates whether this data channel is the next FIFO to be read when FIFO Combine mode is enabled for read operations. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
SAI1	RFR0–RFR1	—
SAI2	RFR0–RFR3	—
SAI3	—	RFR0

0 - No effect

Table continued from the previous page...

Field	Description
	1 - Next FIFO to be read
14-8 —	Reserved
7-0 RFP	Read FIFO Pointer Indicates the FIFO read pointer for the receive data channel. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
SAI1	RFR0–RFR1[5–0]	RFR0–RFR1[7–6]
SAI2	RFR0–RFR3	—
SAI3	RFR0	—

47.6.1.25 Receive Mask (RMR)

Contains the mask for the receive word. This register is double-buffered and updates:

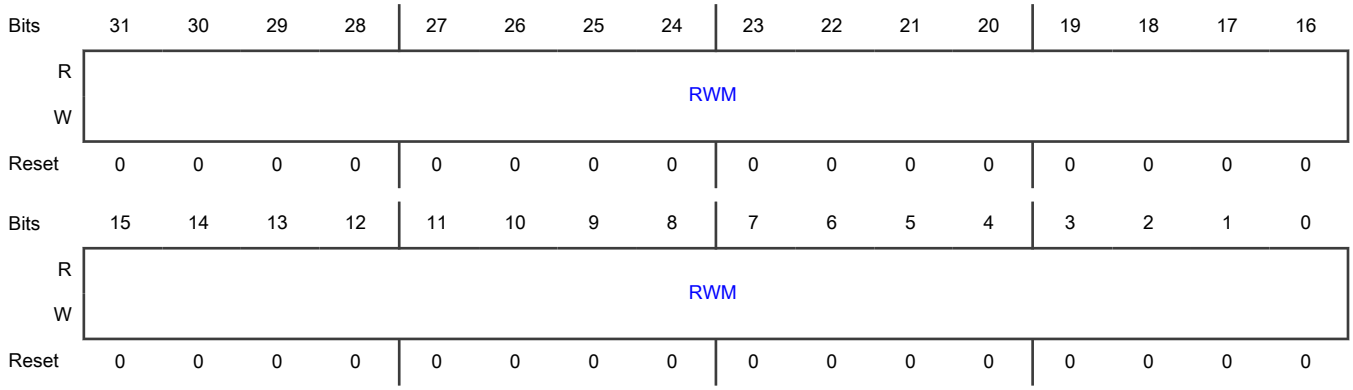
- When RCSR[RE] first becomes 1.
- At the end of each frame.

This setup allows the masked words in each frame to change from frame to frame.

Offset

Register	Offset
RMR	E0h

Diagram



Fields

Field	Description
31-0 RWM	Receive Word Mask Configures whether the receive word is masked (received data ignored and not written to receive FIFO) for the corresponding word in the frame. 0000_0000_0000_0000_0000_0000_0000_0000 - Enable 0000_0000_0000_0000_0000_0000_0000_0001 - Mask

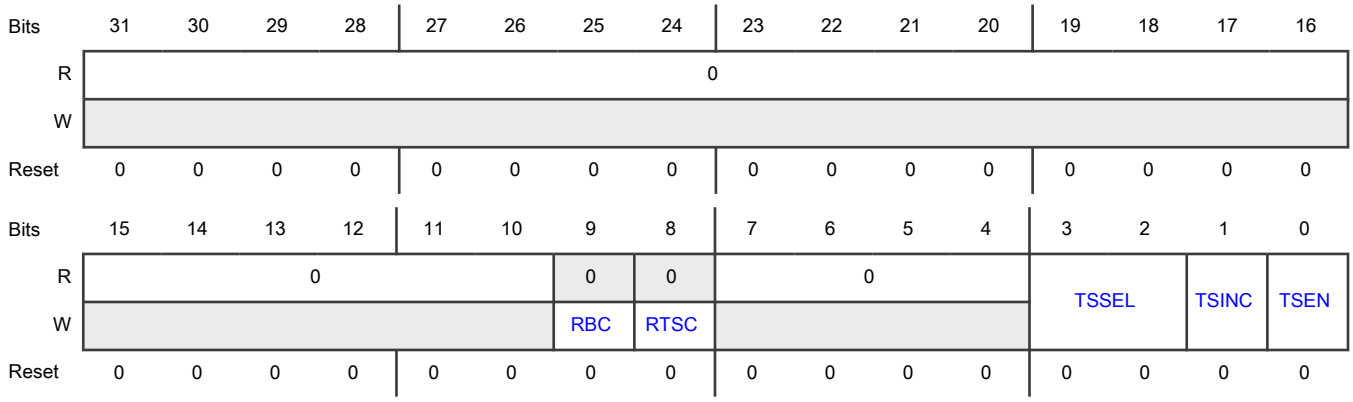
47.6.1.26 Receive Timestamp Control (RTCR)

Controls the receive timestamp counter and bit clock counter.

Offset

Register	Offset
RTCR	F0h

Diagram



Fields

Field	Description
31-10 —	Reserved
9 RBC	Reset Bit Counter Resets the receive bit counter. 0 - No effect 1 - Reset
8 RTSC	Reset Timestamp Counter Resets the receive timestamp counter. 0 - No effect 1 - Reset
7-4 —	Reserved
3-2 TSSEL	Timestamp Select Configures whether the timestamp counter is synchronized with another timestamp counter. 00 - Increment when enabled 01 - Increment when the transmit timestamp counter is enabled 10 - Increment when the receive timestamp counter on another instance is enabled 11 - Increment when the transmit timestamp counter on another instance is enabled
1 TSINC	Timestamp Increment Configures when the timestamp counter starts incrementing. 0 - When enabled and after the bit counter has incremented 1 - When enabled
0 TSEN	Timestamp Enable Enables the timestamp counter to increment on the bus interface clock. 0 - Disable 1 - Enable

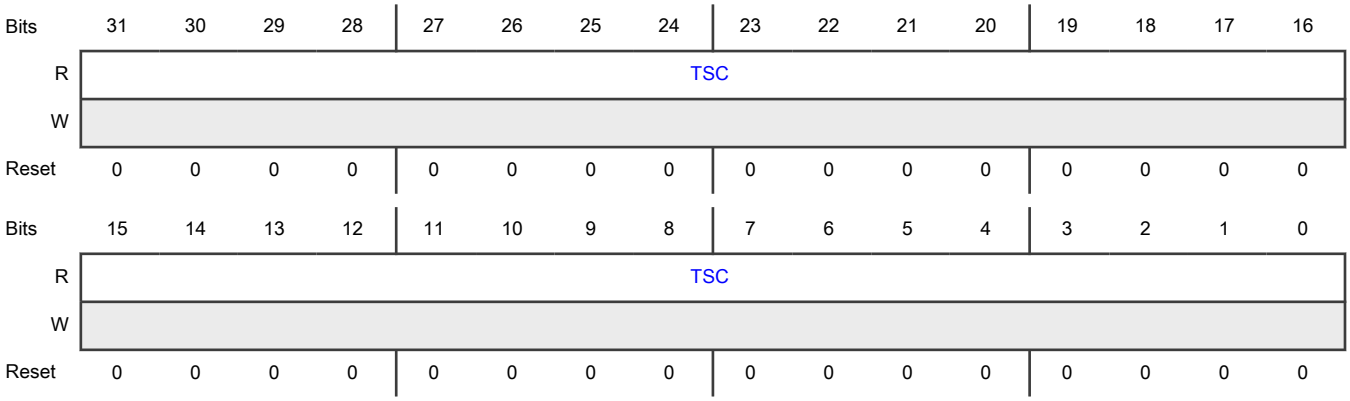
47.6.1.27 Receive Timestamp (RTSR)

Indicates the current value of the receive timestamp counter.

Offset

Register	Offset
RTSR	F4h

Diagram



Fields

Field	Description
31-0 TSC	Timestamp Counter Indicates the current value of the timestamp counter.

47.6.1.28 Receive Bit Count (RBCR)

Indicates the current value of the receive bit counter. The bit counter increments for each bit in a frame when the SAI receiver is enabled. Reading this register causes Receive Bit Count Timestamp (RBCTR) to update with the value of the timestamp counter that corresponds to the bit count register value.

Offset

Register	Offset
RBCR	F8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BCNT	Bit Counter Indicates the current value of the bit counter.

47.6.1.29 Receive Bit Count Timestamp (RBCTR)

Indicates the value of the timestamp register that corresponds to the value last read from Receive Bit Count (RBCR).

Offset

Register	Offset
RBCTR	FCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BCTS	Bit Timestamp Indicates the timestamp counter value that matches the last value read from Receive Bit Count (RBCR).

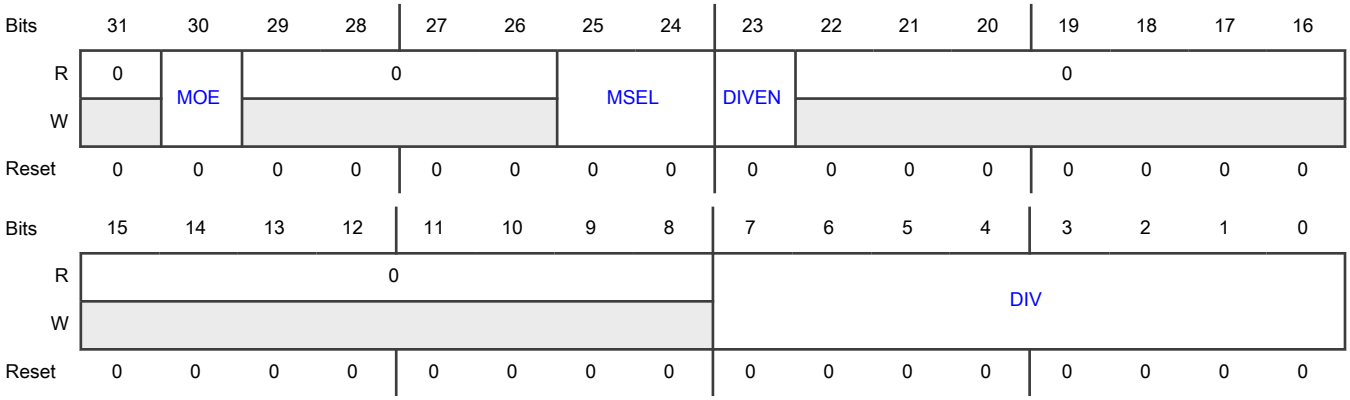
47.6.1.30 MCLK Control (MCR)

Controls the audio clock.

Offset

Register	Offset
MCR	100h

Diagram



Fields

Field	Description
31 —	Reserved
30 MOE	MCLK Output Enable Configures the MCLK signal pin as an input or output. 0 - Input 1 - Output
29-26 —	Reserved
25-24	MCLK Select

Table continues on the next page...

Table continued from the previous page...

Field	Description
MSEL	Selects the clock used by the MCLK postdivider. You cannot change this field when the MCLK postdivider is enabled. See the chip-specific information for the connections to these inputs. 00 - Audio clock (MCLK) option 1 01 - Reserved 10 - Audio clock (MCLK) option 2 11 - Audio clock (MCLK) option 3
23 DIVEN	MCLK Post Divide Enable Enables the postdivider that is output on the MCLK signal pin. When enabled, the output on the MCLK signal pin is a postdivided version of the audio clock. When disabled, the output is the audio clock. The output divider does not impact the audio clock used by the transmitter or receiver. 0 - Disable 1 - Enable
22-8 —	Reserved
7-0 DIV	MCLK Post Divide Specifies the postdivide value for the audio clock that only divides the output to the MCLK signal pin. The division value is (DIV + 1) × 2.

47.7 Glossary

FIFO	First-in first-out
LSB	Least significant bit
MSB	Most significant bit

Chapter 48
Audio Transceiver (SPDIF XCVR)

48.1 Overview

SPDIF_XCVR module supports SPDIF TX and RX functions in full duplex mode.

48.1.1 Block diagram

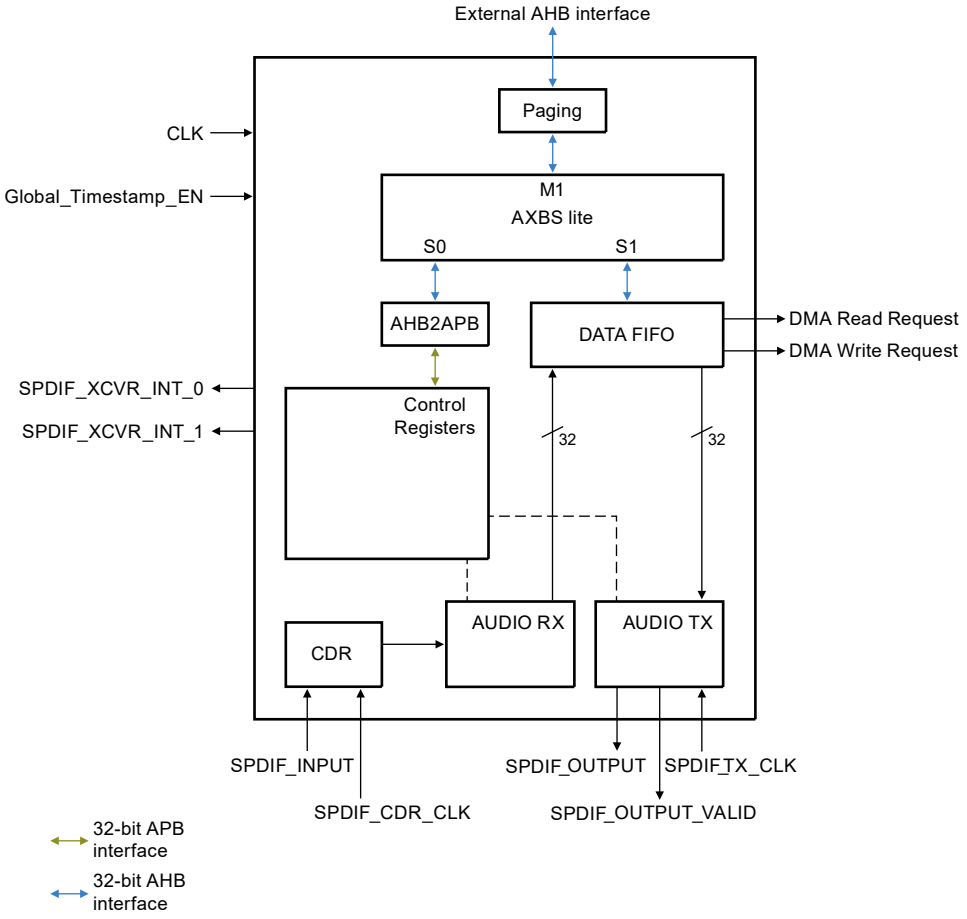


Figure 244. SPDIF_XCVR block diagram

48.1.2 Features

SPDIF_XCVR module can operate in one or both of the following modes:

- SPDIF Receiver:
  - Support for 32 kHz - 192 kHz sample rate
- SPDIF Transmitter:
  - Support for 32 kHz - 192 kHz data TX rate

48.2 Functional description

This section describes functional details of the SPDIF_XCVR.

48.2.1 Audio data format support

All SPDIF_XCVR message formats use IEC60958 framing. Data is stored in the FIFO either in sub-frame format or raw data format programmed in RX_DATAPATH_CTRL.RX_DATA_FMT or TX_DATAPATH_CTRL.TX_FORMAT

1. IEC60958 data in big endian format:
   - 31:28 - Preamble. For TX, the preamble can be added by HW or stored in the FIFO with the correct value. For RX, the 3rd bit will be replaced by MUTE and 4th bit will be replaced by Parity error.
   - 27:4 - TX data
   - 3:0 - V,U,C and P. For TX, these bits can be added by HW or stored in the FIFO with the correct value.

   Each sub-frame can be stored in memory in the reverse order, bit 31 is stored in position 0 and so on using register bit : FRM_FMT in TX_DATAPATH_CTRL and STORE_FMT in RX_dATAPATH_CTRL.
2. Raw data format:
   - 23:0 / 19:0 / 15:0 - Raw data for each audio data format is stored in these bits for TX and RX.

SPDIF_XCVR supports L-PCM and compressed audio within the IEC60958-1 framing structure.

Table 487. Supported data formats

Interface	Type	Channels	Payload Size	Frame Format
SPDIF_XCVR	2-ch LPCM	2	16, 20, or 24	IEC60958
	IEC61937	N/A	16	

48.2.2 SPDIF_XCVR data rates

SPDIF_XCVR supports the following audio bit rates.

- Uncompressed audio at 2.048Mbps to 12.288Mbps for 2-ch
- Compressed audio up to 12.288Mbps

Maximum audio bit rates is 12.288Mbps, or 192kHz IEC60958-1 Frame rate.

NOTE

- The table uses permissible Fs values for SPDIF_XCVR
- L-PCM and compressed audio sample rates are maximum 192kHz

Table 488. Supported SPDIF_XCVR Data Rates

Type	Channels	Fs (kHz)	Data (Mbps)
2-ch LPCM	2	32	2.048
		44.1	2.8224
		48	3.072
		64	4.096
		88.2	5.6448
		96	6.144
		128	8.192
		176.4	11.2896

Table continues on the next page...

Table 488. Supported SPDIF_XCVR Data Rates (continued)

Type	Channels	Fs (kHz)	Data (Mbps)
		192	12.288
IEC61937 Compressed	N/A	32	2.048
		44.1	2.8224
		48	3.072
		64	4.096
		88.2	5.6448
		96	6.144
		128	8.192
		176.4	11.2896
		192	12.288

48.2.3 Operation

SW will need to release the TX and or RX datapaths from reset to enable SPDIF TX / RX datapaths. Control registers will have to be programmed based on the mode needed.

SPDIF_XCVR can be operated in SPDIF TX and/or RX modes. In order to set up the Control registers refer to various options available in RX_DATAPATH_CTRL and TX_DATAPATH_CTRL registers.

- SPDIF TX mode: In order to use the SPDIF_XCVR in SPDIF TX mode, set the input tx_clk to frequency = 2*TX data rate clock. Once the clock is available, release the TX datapath from reset by writing to EXT_CTRL.TX_DPATH_RESET. Once the datapath is out of reset, other control registers can be set. Refer to Audio Transmitter Overview for more information on the control registers. Out of reset, the TX datapath is programmed to have the IEC60958 bit 0 in bit position 31. Insertion of Channel status, user data, presample and parity are disabled. The sub-frame in the TX FIFO is considered to be a complete IEC60958 sub-frame.
- SPDIF RX mode: In order to use the SPDIF_XCVR in SPDIF RX mode, release the RX datapath from reset by writing to EXT_CTRL.RX_DPATH_RESET. Once the datapath is out of reset, other control registers can be set. Refer to Audio Receiver Overview for more information on the control registers. Ou of reset, the RX datapath is programmed to sync to the first recognized B preamble, and store the complete received sub-frame in receive FIFO. Bit 0 of the received sub-frame is stored in bit position 0 of the FIFO data.

48.2.4 SPDIF_XCVR Low power mode

SPDIF_XCVR low power mode is when the SPDIF_XCVR input is used as a wakeup source. Follow the below steps to use SPDIF input as a wakeup source:

- Set SPDIF_XCVR toggle count to the value desired in RX_DATAPATH_CTRL. Default is 4 toggles. Max is 7.
- Enable the interrupt either in IER0 or IER1.
- Enable SPDIF_XCVR wakeup function using : EXT_CTRL.EN_SPDIF_WAKEUP
- Switch ipg_clk source to 24MHz clock source.

48.2.5 Test mode

The main test mode that is supported by SPDIF_XCVR controller is the digital loopback mode. It can be enabled by setting TX_DATAPATH_CTRL.SPDIF_LOOPBACK_EN. Enable RX datapath first, followed by setting the TX control registers and then as a final step, enable data transmit using TX_DATAPATH_CTRL.STRT_DATA_TX.

48.2.5.1 SPDIF_XCVR loopback mode

The SPDIF TX and RX datapaths can be set up as required. The registers ideally should be set as follows: Transmitter should be set to insert parity, and preamble before transmitting data. The receiver will receive the complete sub-frame out of reset. This is the desired format for accurate loopback testing.

- Start out by enabling the transmit PLL for the desired transmit data rate.
- Release TX and RX datapaths from reset. EXT_CTRL_CLR = 0x18000000.
- Program the TXDATAPAT_CTRL registers.
- If a shorter test time is desired, enable RX datapath to start storing entries starting the first recognized preamble compared to storing entries starting with a B preamble match. This can be programmed in RX_DATAPATH_CTRL.FSM
- Load the transmit FIFO with transmit data, by writing to offset 0xE00. This location will accept upto 128 entries of transmit data. Since SPDIF_XCVR DLL will take a while to lock to the input data rate, SW will need to program at least 2 blocks worth of sub-frames. Dma_wr_req can be used to add more data to the transmit FIFO.
- Once the data is written to the transmit FIFO, start transmit of data using TX_DATAPATH_CTRL.STRT_DATA_TX
- Once 64 entries are available in the receive FIFO, SDMA_RD_REQ will be asserted and the received data can be checked against the transmitted data, by reading it from address 12'hC00.

48.2.6 Clocking

This section describes clocks and special clocking requirements of the SPDIF_XCVR module.

SPDIF_XCVR clock inputs are:

- CLK - This clock should be around 400 MHz, though it can depend on the Subsystem.
- SPDIF_TX_CLK - This is clock source from Audio PLL.

48.2.7 Reset

The SPDIF_XCVR block has one input reset It is also combined with the software reset to allow of power saving in "low power mode" when certain parts of the HW are held in reset.

48.2.8 Interrupts

This section describes all the interrupts that the SPDIF_XCVR module generates.

- RX_NEW_CH_STAT- New channel status was received. This interrupt is asserted if the Channel status for the current block does not match the channel status received for the last block of data.
- RX_USR_DATA - Non Zero user data received. This interrupt is set if a non-zero user data is received. If all 0 user data is received, it doe snot need to be acknowledged by SW.
- MUTE_DET - Interrupt to indicate change in mute channel status was detected.
- PREAMBLE_MISMATCH - Asserted when there is an unexpected preamble received / when a preamble was expected but one was not received.
- FIFO_OFLOW_UFLOW_ERR - This interrupt is asserted if the RX / TX FIFO overflows / underflows.
- RX_NO_DATA_REC - Set after valid frames were received, but then valid data was not received for over 1us.

- DMA_RD_REQ - Asserted when RX FIFO entries reaches the programmed threshold. (Default = half full)
- DMA_WR_REQ - Asserted when TX FIFO entries reaches the programmed threshold. (Default = half empty)
- RX_BME_BIT_ERR - This interrupt indicates there was BME violation other than for a recognized preamble.
- PREAMBLE_MATCH_INT - This interrupt is asserted every time a PA PB / CD match is received in compressed data. The information for packet count when this was received and which match occurred can be read from
- M_W_PRE_MISMATCH - Asserted when there is a preamble error in the received sub-frame. e.g.: M was expected and W was received.
- B_PRE_MISMATCH - Asserted when there is a B preamble error in the received sub-frame. Set when an unexpected B is received or an expected B is not received.
- UNEXP_PRE_REC - This interrupt is set if an unexpected preamble is received in the middle of a sub-frame. If this is received, Host software should reset the RX datapath and wait for the next B preamble to be received.
- CS_OR_UD_OFLOW - Asserted when received Channel Status / user data is not acknowledged by SW and HW will have to drop the new channel status received as the alternate buffer is full.
- NEW_BLK_RCVD - Asserted every time a new block of data is received.
- SPDIF_WAKEUP_REC - Asserted when SPDIF_XCVR is used as a wakeup source. Once cleared, the clock source should be switched to 400MHz for ipg_clk and datapaths should be release from reset.

48.3 External signals

This table describes the SPDIF_XCVR module signals.

Signal	Description
SPDIF_INPUT	SPDIF RX data
SPDIF_OUTPUT	SPDIF TX data
SPDIF_OUTPUT_VALID	SPDIF TX data valid. If necessary, it can be used to indicate that SPDIF data is being transmitted.

48.4 Initialization

To initialize the SPDIF_XCVR mode use the following sequence:

48.4.1 PLL Programming Sequence

PLL programmed frequency value will be:

$$PLLfrequency = 24 \cdot I(Integer) + \frac{N(Numerator)}{D(Denominator)}$$

PLL can be programmed using the below sequence.

1. Set the frequency values
   - If SPDIF_XCVR mode,(198MHz) then AV_PLL_DIV_ADDR (0x0060) = 0x00842010
2. Wait for 5 us.
3. Power up PLL - PLL_CTRL[0]
   - AV_PLL_CTRL_ADDR (0x0000) = 0x00000001
4. Wait 100 us OR poll on PLL lock. (PLL_STATUS[0] (0x00F0))

5. Enable output clock, PLL_Ctrl[1]
    - AV_PLL_CTRL_ADDR (0x0000) = 0x00000003

48.5 Application information

This section describes the applications supported by the SPDIF_XCVR and how to program the registers to support these features.

48.5.1 Audio Receiver Overview

The Audio RX datapath is held in reset until host core decides to release the HW out of reset to start receiving IEC60958 data. This feature helps save power when specific datapaths are not being used.

- FSM: The input data stream can be received when aligned to either the B preamble, M preamble, or the M/W preambles. By default, the input stream is set to sync to the B preamble. Change this register setting to align to M/W input preamble.
- STORE_FMT: Format to store the received data in memory. If this bit is 0, the first bit received in a sub frame is stored in bit position 31. Else, it is stored in bit position 0. The rest of the bits follow the first bit.
- RX_DATA_FMT : These bits control the format of data stored in FIFO memory. Default value of 2'00 indicates the data will be stored in the raw sub-frame received. Other combinations possible are 24-bit audio, 20-bit audio and 16-bit audio.
- EN_PARITY_CALC: This bit enables parity evaluation. If Parity evaluation fails, bit 4 of the preamble is replaced with 1.
- MUTE_MODE: Mute mode selects between HW detection of MUTE mode and FW enabling Mute. By default, HW mode is set. In this mode, mute is detected by looking at channel status of frame number programmed in MBIT. If SW control is enabled, mute is asserted using MUTE_CTRL. Preamble bit 1 is replaced by
- Performance counter: The Audio Receiver path supports a performance/progress counter. This is essentially 2 counters, one running at the highest clock speed available to the block and the other at the audio bit rate. The high speed counter is latched twice, once on every bit clock transition and again when the bit clock counter is read to guarantee a synchronized read.
- Packet counter: The packet counter keeps track of all the sub-frames written to the output FIFO. It is a rolling counter used to keep a reference of the number of frames received.

Channel status and user data buffers are stored per block. An interrupt to the host core is asserted if the Channel status does not match the last channel status value. Since user data is infrequent, an interrupt is only asserted if the value received is non-zero. Zero value of previous user data received can be found in the EXT_STATUS register.

48.5.2 Audio Transmitter Overview

The Audio transmitter can output data in the SPDIF_XCVR format. In order to transmit data, the spdif_tx_clk frequency needs to be programmed to data TX rate. If transmit datapath needs to be enabled, it should be released from reset using EXT_CTRL.TX_DPATH_RESET. If TX datapath is not released from reset, data entries should not be written to the TX FIFO. TX can now be enabled using TX_DATAPATH_CTRL.STRT_DATA_TX. Once this bit is set, HW will start transmitting data in the TX FIFO. If the FIFO does not have watermark number of entries, HW will assert a dma_wr_req. Care should be taken to make sure the FIFO does not under run.

The Audio transmitter consists of the following features that can be enabled in TX_DATAPATH_CTRL register.

- EN_PREAMBLE: If this bit is set, preamble matching the frame count is inserted in the TX sub-frame. TX always starts with a B preamble.
- CS_MOD and UD_MOD: These register bits are used to modify the channel status and user data values programmed in the TX FIFO. The channel status and user data replacement values should be programmed in TX_CS_DATA_BITS and TX_USER_DATA_BITS buffers on a block boundary. These registers have an ACK value in UD_ACK and CS_ACK. FW should not write to the data buffers if the ACK bits are not 0. Once FW writes to the registers, the ACK bits should be set so HW can read the new values when it starts transmitting the next block of data. These bits will be cleared by HW when the value in TX_CS_DATA_BITS and TX_USER_DATA_BITS have been moved to a secondary register.

- VLD_MOD and FRM_VLD: These register bits are used to modify the valid field programmed in the TX FIFO. If VLD_MOD is set, Valid bit in the TX frames is replaced by the value in FRM_VLD.
- EN_PARITY: This bit is used to calculate the parity over the data and VUC bits and replace the parity bit of the frame.
- FRM_FMT: This bit indicates if the preamble is stored in bits 3:0 or 31:28
- TX_FORMAT: This bit indicates if the data stored in the transmit FIFO has been formatted to a match a sub-frame or the data is in raw data format

Since Channel Status is fairly static, a new value is not expected every block and the same Channel status will be transmitted unless the value in the registers is updated. A new value is not expected for every block.

48.6 Memory Map and register definition

This section includes the SPDIF_XCVR module memory map and detailed descriptions of all registers.

48.6.1 EARC_SPDIF_XCVR register descriptions

48.6.1.1 EARC_SPDIF_XCVR memory map

SPDIF base address: 4268_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version control register (VERSION)	32	RW	0000_0000
10	External control register (EXT_CTRL)	32	RW	1800_4040
14	External control register (EXT_CTRL_SET)	32	RW	1800_4040
18	External control register (EXT_CTRL_CLR)	32	RW	1800_4040
1C	External control register (EXT_CTRL_TOG)	32	RW	1800_4040
20	External Status register (EXT_STATUS)	32	RW	0000_0000
24	External Status register (EXT_STATUS_SET)	32	RW	0000_0000
28	External Status register (EXT_STATUS_CLR)	32	RW	0000_0000
2C	External Status register (EXT_STATUS_TOG)	32	RW	0000_0000
30	Interrupt enables for interrupt 0 (EXT_IER0)	32	RW	0000_0000
34	Interrupt enables for interrupt 0 (EXT_IER0_SET)	32	RW	0000_0000
38	Interrupt enables for interrupt 0 (EXT_IER0_CLR)	32	RW	0000_0000
3C	Interrupt enables for interrupt 0 (EXT_IER0_TOG)	32	RW	0000_0000
40	Interrupt enables for interrupt 1 (EXT_IER1)	32	RW	0000_0000
44	Interrupt enables for interrupt 1 (EXT_IER1_SET)	32	RW	0000_0000
48	Interrupt enables for interrupt 1 (EXT_IER1_CLR)	32	RW	0000_0000
4C	Interrupt enables for interrupt 1 (EXT_IER1_TOG)	32	RW	0000_0000
50	External Interrupt Status register (EXT_ISR)	32	RW	0000_0000
54	External Interrupt Status register (EXT_ISR_SET)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
58	External Interrupt Status register (EXT_ISR_CLR)	32	RW	0000_0000
5C	External Interrupt Status register (EXT_ISR_TOG)	32	RW	0000_0000
A8	Audio XCVR datapath status (DPATH_STATUS)	32	R	0000_0000
B0	Clock control register (CLK_CTRL)	32	RW	0000_018F
180	Data path control register (RX_DATAPATH_CTRL)	32	RW	0004_2CC1
184	Data path control register (RX_DATAPATH_CTRL_SET)	32	RW	0004_2CC1
188	Data path control register (RX_DATAPATH_CTRL_CLR)	32	RW	0004_2CC1
18C	Data path control register (RX_DATAPATH_CTRL_TOG)	32	RW	0004_2CC1
190 - 1A4	Channel staus bits (RX_CS_DATA_BITS_0 - RX_CS_DATA_BITS_5)	32	R	0000_0000
1A8 - 1BC	User data bits (RX_USER_DATA_BITS_0 - RX_USER_DATA_BITS_5)	32	R	0000_0000
1C0	Receive Datapath counter control register (RX_DPATH_CNTR_CTRL)	32	RW	0000_0000
1C4	Receive Datapath counter control register (RX_DPATH_CNTR_CTRL_SET)	32	RW	0000_0000
1C8	Receive Datapath counter control register (RX_DPATH_CNTR_CTRL_CLR)	32	RW	0000_0000
1CC	Receive Datapath counter control register (RX_DPATH_CNTR_CTRL_TOG)	32	RW	0000_0000
1D0	Receive Datapath Timestamp Counter Register (RX_DPATH_TSCR)	32	R	0000_0000
1D4	Receive Datapath Bit counter register (RX_DPATH_BCR)	32	R	0000_0000
1D8	Receive datapath Bit count timestamp register. (RX_DPATH_BCTR)	32	R	0000_0000
1DC	Receive datapath Bit read timestamp register. (RX_DPATH_BCRR)	32	R	0000_0000
1E0	Preamble match value register (PRE_MATCH_VAL)	32	RW	4E1F_F872
1E4	Preamble match value register (PRE_MATCH_VAL_SET)	32	RW	4E1F_F872
1E8	Preamble match value register (PRE_MATCH_VAL_CLR)	32	RW	4E1F_F872
1EC	Preamble match value register (PRE_MATCH_VAL_TOG)	32	RW	4E1F_F872
1F0	Preamble match value register (DTS_PRE_MATCH_VAL)	32	RW	1387_FE1C
1F4	Preamble match value register (DTS_PRE_MATCH_VAL_SET)	32	RW	1387_FE1C
1F8	Preamble match value register (DTS_PRE_MATCH_VAL_CLR)	32	RW	1387_FE1C
1FC	Preamble match value register (DTS_PRE_MATCH_VAL_TOG)	32	RW	1387_FE1C
200	Error count for IEC60958-1 Block Synchronization. (RX_DPATH_PRE_ERR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
204	Parity Error count for IEC60958-1 Blocks. (RX_DPATH_PARITY_ERR)	32	RW	0000_0000
210	Receive Data packet count. (RX_DPATH_PKT_CNT)	32	R	0000_0000
218	Preamble match offset value register (PRE_MATCH_OFFSET)	32	R	0000_0000
220	Transmit Data path control register (TX_DATAPATH_CTRL)	32	RW	0000_0000
224	Transmit Data path control register (TX_DATAPATH_CTRL_SET)	32	RW	0000_0000
228	Transmit Data path control register (TX_DATAPATH_CTRL_CLR)	32	RW	0000_0000
22C	Transmit Data path control register (TX_DATAPATH_CTRL_TOG)	32	RW	0000_0000
230 - 244	Channel staus bits (TX_CS_DATA_BITS_0 - TX_CS_DATA_BITS_5)	32	RW	0000_0000
248 - 25C	User data bits (TX_USER_DATA_BITS_0 - TX_USER_DATA_BITS_5)	32	RW	0000_0000
260	Transmit datapath counter control register (TX_DPATH_CNTR_CTRL)	32	RW	0000_0000
264	Transmit datapath counter control register (TX_DPATH_CNTR_CTRL_SET)	32	RW	0000_0000
268	Transmit datapath counter control register (TX_DPATH_CNTR_CTRL_CLR)	32	RW	0000_0000
26C	Transmit datapath counter control register (TX_DPATH_CNTR_CTRL_TOG)	32	RW	0000_0000
270	Transmit Datapath Timestamp Counter Register (TX_DPATH_TSCR)	32	R	0000_0000
274	Transmit Datapath Bit counter register (TX_DPATH_BCR)	32	R	0000_0000
278	Transmit datapath Bit count timestamp register. (TX_DPATH_BCTR)	32	R	0000_0000
27C	Transmmit datapath Bit read timestamp register. (TX_DPATH_BCRR)	32	R	0000_0000

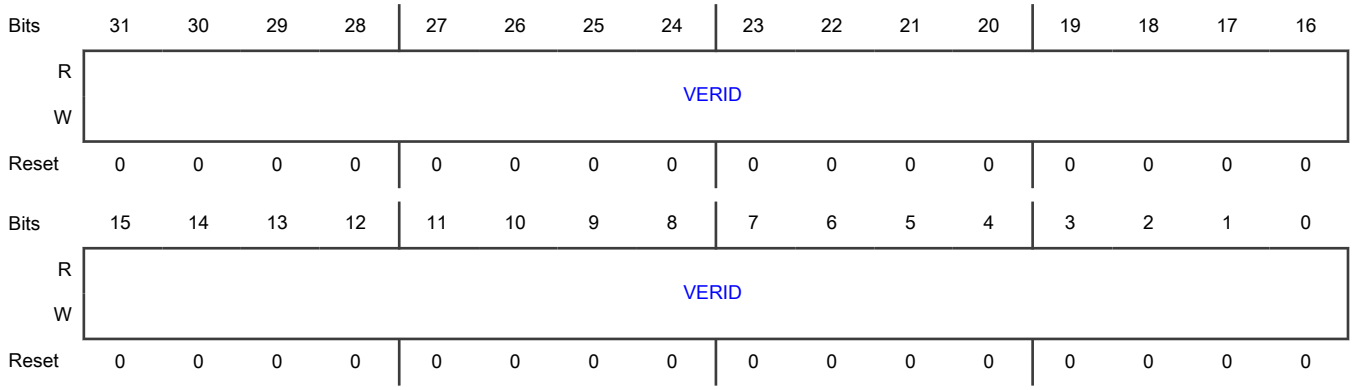
48.6.1.2 Version control register (VERSION)

Indicates the version of Audio XCVR

Offset

Register	Offset
VERSION	0h

Diagram



Fields

Field	Description
31-0 VERID	Version ID Indicates the version of SPDIF XCVR

48.6.1.3 External control register (EXT_CTRL)

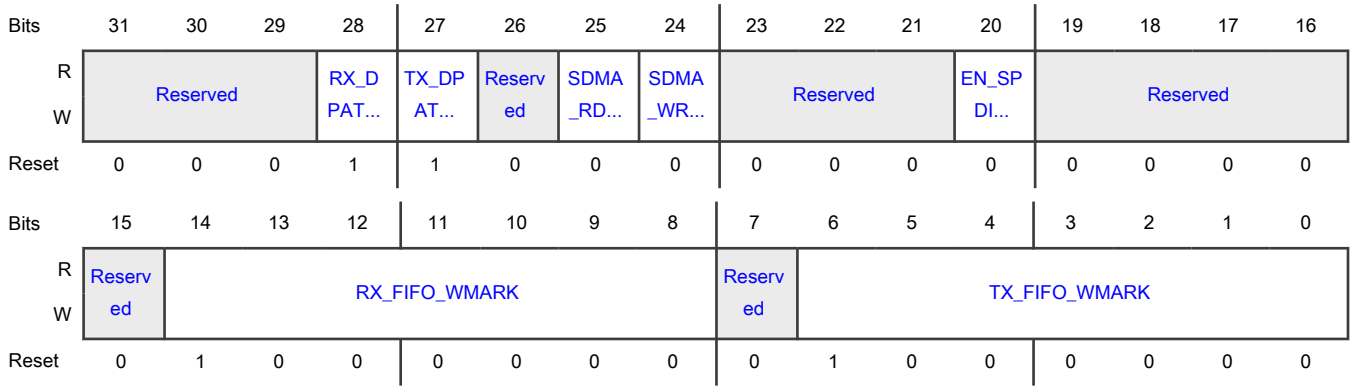
This register provides host control registers for the Audio XCVR

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
EXT_CTRL	10h	External control register
EXT_CTRL_SET	14h	Writing 1 to a bit in this register ensures that the corresponding bit in EXT_CTRL is 1
EXT_CTRL_CLR	18h	Writing 1 to a bit in this register ensures that the corresponding bit in EXT_CTRL is 0
EXT_CTRL_TOG	1Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in EXT_CTRL

Diagram



Fields

Field	Description
31-29 —	Reserved
28 RX_DPATH_RESET	Soft reset to SPDIF Receive datapath Receive datapath is reset when '1' is written to this bit. Software must allow 8 or more clock cycles before writing a '0' to this bit to allow for reset synchronization and propagation.
27 TX_DPATH_RESET	Soft reset to SPDIF Transmit datapath Transmit datapath is reset when '1' is written to this bit. Software must allow 8 or more clock cycles before writing a '0' to this bit to allow for reset synchronization and propagation.
26 —	Reserved
25 SDMA_RD_REQ_DIS	SDMA RD REQ disable This bit field is used to disable SDMA read requests.
24 SDMA_WR_REQ_DIS	SDMA WR REQ disable This bit field is used to disable SDMA write requests.
23-21 —	Reserved
20 EN_SPDIF_WAKEUP	Enable SPDIF wakeup interrupt If this bit is set, when SPDIF_TGL_CNT number of input toggles are received, the SPDIF wakeup interrupt is asserted.
19-15	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
14-8 RX_FIFO_WMARK	Audio Receive FIFO Watermark Level Indicates the watermark level (number of 32-bit words) at and above which a FIFO read request will be generated Reset to 1/2 the FIFO size. • 0 - Not Applicable • 1 - 127 - Valid values
7 —	reserved
6-0 TX_FIFO_WMARK	Audio Transmit FIFO Watermark Level Indicates the watermark level (number of 32-bit words) at and below which a FIFO write request will be generated. Reset to 1/2 the FIFO size. • 0 - Not Applicable • 1 - 127 - Valid values

48.6.1.4 External Status register (EXT_STATUS)

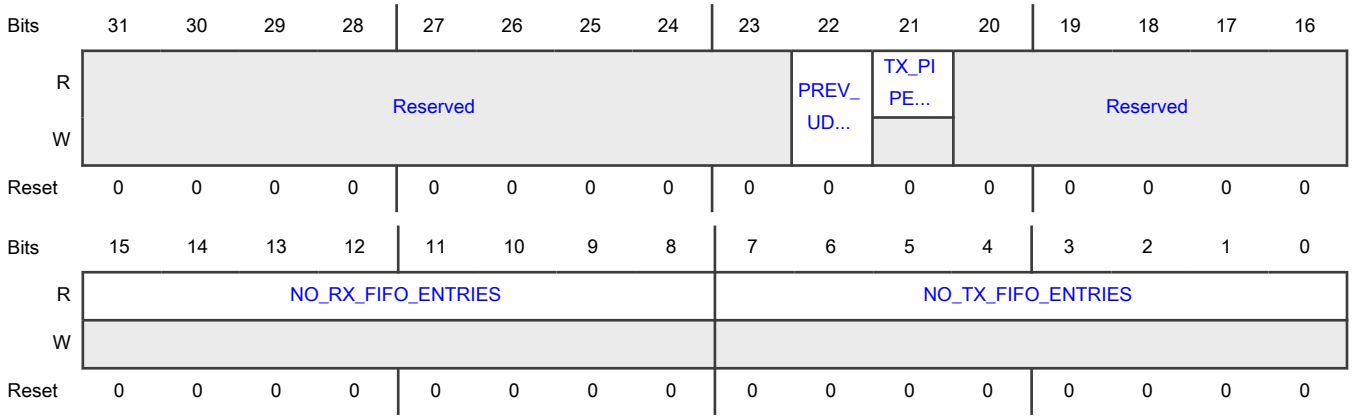
This register provides status to the host controller

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
EXT_STATUS	20h	External Status register
EXT_STATUS_SET	24h	Writing 1 to a bit in this register ensures that the corresponding bit in EXT_STATUS is 1
EXT_STATUS_CLR	28h	Writing 1 to a bit in this register ensures that the corresponding bit in EXT_STATUS is 0
EXT_STATUS_TOG	2Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in EXT_STATUS

Diagram



Fields

Field	Description
31-23 —	Reserved
22 PREV_UD_0	Last User data received was all 0 This bit indicates If the last user data received was all 0. SW can clear this bit if needed.
21 TX_PIPE_EMPTY	Indicates TX pipe status. This status bit is used to indicate TX pipeline is empty. This can be used to determine if new block of data can be sent / Frequency can be updated.
20-16 —	Reserved
15-8 NO_RX_FIFO_ENTRIES	RX FIFO entries This status provides the number of entries in the receiver FIFO
7-0 NO_TX_FIFO_ENTRIES	TX FIFO entries This status provides the number of entries in the transmit FIFO

48.6.1.5 Interrupt enables for interrupt 0 (EXT_IER0)

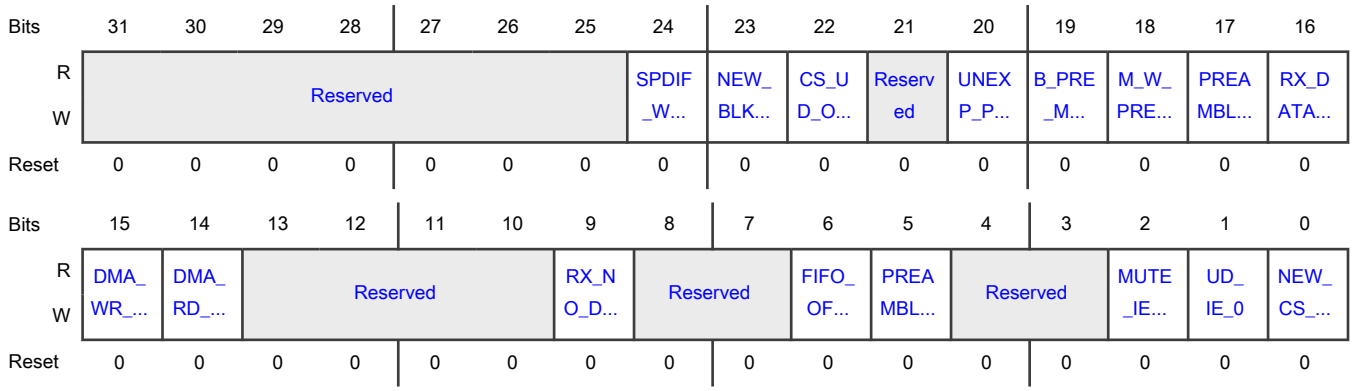
Provides enable bits to enable interrupt 0 generation. All enables are active high. Interrupt enables for bits 25 - 31 are reserved for future use

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
EXT_IER0	30h	Interrupt enables for interrupt 0
EXT_IER0_SET	34h	Writing 1 to a bit in this register ensures that the corresponding bit in EXT_IER0 is 1
EXT_IER0_CLR	38h	Writing 1 to a bit in this register ensures that the corresponding bit in EXT_IER0 is 0
EXT_IER0_TOG	3Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in EXT_IER0

Diagram



Fields

Field	Description
31-25 —	Reserved
24 SPDIF_WAKEUP_IE_0	SPDIF Wakeup interrupt enable. This interrupt enable allows asserting SPDF wakeup interrupt. When this interrupt is enabled, SPDIF should be in low power mode, and clock input should be a 24MHz low frequency clock to reduce power. Refer to EXT_CTRL.SPDIF_WAKEUP_CNT for how many toggles will be detected, before an interrupt is asserted.
23 NEW_BLK_RCVD_IE_0	New block of data was received. This interrupt enable allows asserting an interrupt when a block of data is received.
22 CS_UD_OFLOW_IE_0	Channel status or used data could not be stored. This interrupt enables channel status or user data overflow interrupt. This interrupt is asserted when HW is unable to store user data or channel status because the previous entries were not acknowledged.
21 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
20 UNEXP_PRE_REC_IE_0	Interrupt enable for Unexpected preamble received. This enables the interrupt to be asserted everytime HW detects a preamble was not received when expected or more than 8 bits were received before a preamble was detected.
19 B_PRE_MISMATCH_IE_0	Interrupt enable for sub-frame B preamble mismatch received. This enables the interrupt to be asserted everytime HW detects a B sub-frame preamble mismatch.
18 M_W_PRE_MISMATCH_IE_0	Interrupt enable for sub-frame M/W preamble mismatch received. This enables the interrupt to be asserted everytime HW detects a M/W sub-frame preamble mismatch.
17 PREAMBLE_MATCH_IE_0	Interrupt enable for preamble match received. This enables the interrupt to be asserted everytime HW detects a preamble match.
16 RX_DATA_BME_ERR_IE_0	Bi-phase mark encoding error This enables the interrupt to be asserted everytime Bi-phase mark encoded data received has an error. i.e. No toggle after 2 bits during data phase of sub-frame.
15 DMA_WR_REQ_IE_0	Request to write data to FIFO. This enables the interrupt to be asserted everytime only Watermark number of sub-frames are available to be transmitted in the tx FIFO.
14 DMA_RD_REQ_IE_0	Request to read data from FIFO. This enables the interrupt to be asserted everytime Watermark number of entries are available to be read.
13-10 —	Reserved
9 RX_NO_DATA_REC_IE_0	Indicates no data is received. No input data received for over 2 ms.
8-7 —	Reserved
6 FIFO_OFLOW_UFLOW_ERR_IE_0	Receive FIFO overflow error interrupt enable. Enables assertion of interrupt when FIFO underflows (in TX mode) / overflows (in RX mode).
5	Preamble mismatch interrupt enable.

Table continues on the next page...

Table continued from the previous page...

Field	Description
PREAMBLE_MISMATCH_IE_0	Enables assertion of interrupt when the expected preamble is not received. Usually when a W does not follow an M / an unexpected B preamble is received. Once an unexpected preamble is received, the curent block of data is dropped and sub-frame data is synced to the next input B preamble.
4-3 —	Reserved
2 MUTE_IE_0	Enable for Mute detected interrupt Enables assertion of interrupt when HW mute bit is set for received channel status.
1 UD_IE_0	Enable for user data received interrupt Enables assertion of interrupt when non-0 user data is received.
0 NEW_CS_IE_0	Enable for New channel status block received interrupt Enables assertion of interrupt when a new channel status block is received.

48.6.1.6 Interrupt enables for interrupt 1 (EXT_IER1)

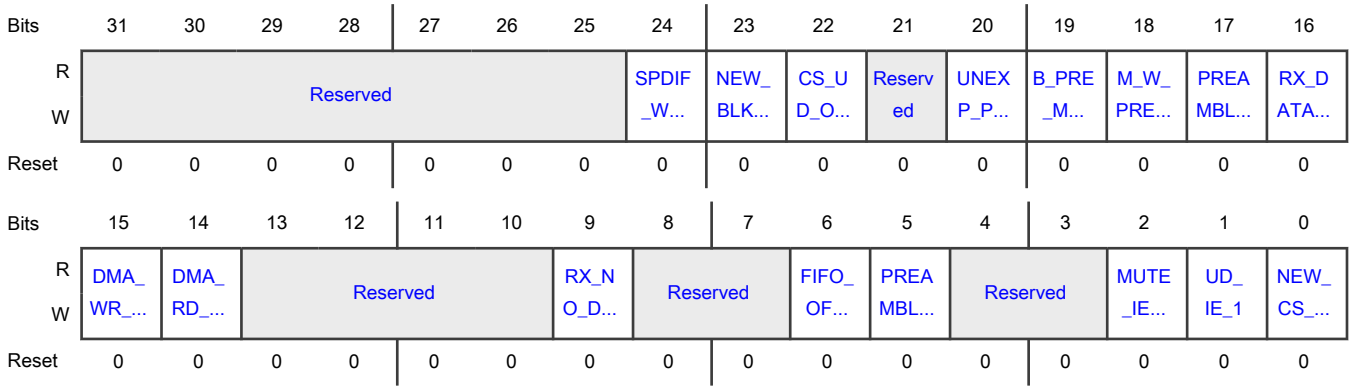
Provides enable bits to enable interrupt 1 generation. All enables are active high. Interrupt enables 25 - 31 are reserved for future use

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
EXT_IER1	40h	Interrupt enables for interrupt 1
EXT_IER1_SET	44h	Writing 1 to a bit in this register ensures that the corresponding bit in EXT_IER1 is 1
EXT_IER1_CLR	48h	Writing 1 to a bit in this register ensures that the corresponding bit in EXT_IER1 is 0
EXT_IER1_TOG	4Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in EXT_IER1

Diagram



Fields

Field	Description
31-25 —	Reserved
24 SPDIF_WAKEUP_IE_1	SPDIF Wakeup interrupt enable. This interrupt enable allows asserting SPDF wakeup interrupt. When this interrupt is enabled, SPDIF should be in low power mode, and clock input should be a 24MHz low frequency clock to reduce power. Refer to RX_DATAPATH.SPDIF_TGL_CNT for how many toggles will be detected, before an interrupt is asserted.
23 NEW_BLK_RCVD_IE_1	New block of data was received. This interrupt enable allows asserting an interrupt when a block of data is received.
22 CS_UD_OFLOW_IE_1	Channel status or used data could not be stored. This interrupt enables channel status or user data overflow interrupt. This interrupt is asserted when HW is unable to store new user data or channel status because the previous entries were not acknowledged.
21 —	Reserved
20 UNEXP_PRE_REC_IE_1	Interrupt enable for Unexpected preamble received. This enables the interrupt to be asserted everytime HW detects a preamble was not received when expected or more than 8 bits were received before a preamble was detected.
19 B_PRE_MISMATCH_IE_1	Interrupt enable for sub-frame B preamble mismatch received. This enables the interrupt to be asserted everytime HW detects a B sub-frame preamble mismatch.
18 M_W_PRE_MISMATCH_IE_1	Interrupt enable for sub-frame M/W preamble mismatch received. This enables the interrupt to be asserted everytime HW detects a M/W sub-frame preamble mismatch.

Table continues on the next page...

Table continued from the previous page...

Field	Description
17 PREAMBLE_MATCH_IE_1	Interrupt enable for preamble match received. This enables the interrupt to be asserted everytime HW detects a preamble match.
16 RX_DATA_BME_ERR_IE_1	Bi-phase mark encoding error This enables the interrupt to be asserted everytime Bi-phase mark encoded data received has an error. i.e. No toggle after 2 bits during data phase of sub-frame.
15 DMA_WR_REQ_IE_1	Request to write data to FIFO. This enables the interrupt to be asserted everytime only Watermark number of sub-frames are available to be transmitted in the tx FIFO.
14 DMA_RD_REQ_IE_1	Request to read data from FIFO. This enables the interrupt to be asserted everytime Watermark number of entries are available to be read.
13-10 —	Reserved
9 RX_NO_DATA_REC_IE_1	Indicates no data is received. No input data for over 2 ms.
8-7 —	Reserved
6 FIFO_OFLOW_UFLOW_ERR_IE_1	Receive FIFO overflow error interrupt enable. Enables assertion of interrupt when FIFO underflows (in TX mode) / overflows (in RX mode).
5 PREAMBLE_MISMATCH_IE_1	Preamble mismatch interrupt enable. Enables assertion of interrupt when the expected preamble is not received. Usually when a W does not follow an M / an unexpected B preamble is received. Once an unexpected preamble is received, the curent block of data is dropped and sub-frame data is synced to the next input B preamble.
4-3 —	Reserved
2 MUTE_IE_1	Enable for Mute detected interrupt Enables assertion of interrupt when the channel status mute bit is set for received channel status.
1 UD_IE_1	Enable for user data received interrupt Enables assertion of interrupt when non-0 user data is received.

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 NEW_CS_IE_1	Enable for New channel status block received interrupt Enables assertion of interrupt when a new channel status block is received.

48.6.1.7 External Interrupt Status register (EXT_ISR)

Interrupt status for the host core

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
EXT_ISR	50h	External Interrupt Status register
EXT_ISR_SET	54h	Writing 1 to a bit in this register ensures that the corresponding bit in EXT_ISR is 1
EXT_ISR_CLR	58h	Writing 1 to a bit in this register ensures that the corresponding bit in EXT_ISR is 0
EXT_ISR_TOG	5Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in EXT_ISR

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved							SPDIF_W...	NEW_BLK...	CS_OR_U...	Reserved	UNEXP_P...	B_PRE_M...	M_W_PRE...	PREAMBL...	RX_BME_...
W								W1C	W1C	W1C		W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMA_WR_...	DMA_RD_...	Reserved				RX_NO_D...	Reserved		FIFO_OF...	PREAMBL...	Reserved		MUTE_DET	RX_USR_...	RX_NEW_...
W	W1C	W1C					W1C			W1C	W1C			W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-25 —	Reserved
24	SPDIF Wakeup received.

Table continues on the next page...

Table continued from the previous page...

Field	Description
SPDIF_WAKEUP_REC	This interrupt allows HW to wake up the core when new toggle on the input is detected in low power mode.
23 NEW_BLK_RCVD	New block of data was received. This interrupt is asserted when a block of data is received, i.e. a new B preamble is detected.
22 CS_OR_UD_OFLOW	Channel status or used data could not be stored. This interrupt indicates channel status or user data previous entries were not acknowledged and so, HW was unable to store a new channel status or User data received..
21 —	Reserved
20 UNEXP_PRE_REC	Set when SPDIF preamble was received after unexpected number of input bits. This interrupt indicates SPDIF subframe received has a preamble error. i.e, a preamble was expected and not received or a preamble was detected before the required number of data bits were received for a sub-frame.
19 B_PRE_MISMATCH	Set when SPDIF B preamble has an error. This interrupt indicates SPDIF subframe received has a B preamble error. i.e, B preamble expected and was not received / B preamble received and was not expected.
18 M_W_PRE_MISMATCH	Set when SPDIF preamble of M/W has an error. This interrupt indicates SPDIF subframe received has a preamble error. M/W sub-frame was nbot detected in teh correct order.
17 PREAMBLE_MATCH_INT	Interrupt to indicate PA PB / DTC CD preamble match was detected. This interrupt is asserted every time a PA PB match is detected by HW. Host core will neeed to go read the Preamble register to find out more information about the sub frame number.
16 RX_BME_BIT_ERR	Set when RX BME data has an error. This interrupt indicates BME data received has an error. After a preamble was received, the input data did not toggle after every 2 bits. This interrupt will have to be cleared.
15 DMA_WR_REQ	Set when DMA write request is asserted. This interrupt indicates Watermark number of entries are available to be filled with new sub-frames. This interrupt will have to be cleared.
14 DMA_RD_REQ	Set when DMA read request is asserted. This interrupt indicates Watermark number of entries are avaialble to be read. This interrupt will have to be cleared.
13-10 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 RX_NO_DATA_REC	No data is received for 1us. This interrupt indicates input hasn't toggled in 1us. This is 4* the slowest data rate. This can either indicate a format change / other issue in the data pipe.
8-7 —	Reserved
6 FIFO_OFLOW_UFLOW_ERR	Receive FIFO overflow error interrupt This interrupt indicates the data receive FIFO is full, or the TX fifo is empty. In RX mode, the new frames will be dropped until the fifo has room for more sub-frames. In TX mode, no data will be sent out on the TX path.
5 PREAMBLE_MISMATCH	Preamble mismatch interrupt This interrupt indicates an unexpected preamble is received. When this interrupt is set, HW clears the RX pipe and waits for a new B preamble to start receiving a fresh block of data.
4-3 —	Reserved
2 MUTE_DET	Interrupt to indicate HW mute bit was detected. This interrupt indicates mute bit was detected in HW or set by FW.
1 RX_USR_DATA	Received User data Information This interrupt indicates user data received for last block of data wasn't all 0's.
0 RX_NEW_CH_STAT	Received new channel status block This interrupt indicates channel status for new block received did not match previous channel status.

48.6.1.8 Audio XCVR datapath status (DPATH_STATUS)

Provides the status for datapath

Offset

Register	Offset
DPATH_STATUS	A8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX_FRM_CNT								RX_FRM_CNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-8 TX_FRM_CNT	Count of transmitted frames in a block This register holds the count of transmitted frames in a block of data.
7-0 RX_FRM_CNT	Count of received frames in a block This register holds the count of received frames in a block of data.

48.6.1.9 Clock control register (CLK_CTRL)

Provides the divider value to get to 1MHz clock based on the current clock speed.

Offset

Register	Offset
CLK_CTRL	B0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						CLKDIV									
W																
Reset	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1

Fields

Field	Description
31-10 —	Reserved
9-0 CLKDIV	Clock divider value Clock divide value to generate the 1 MHz clock rate from the peripheral clock. All internal timers use this as the reference for counting time. Reset value is for 400 MHz peripheral clock.

48.6.1.10 Data path control register (RX_DATAPATH_CTRL)

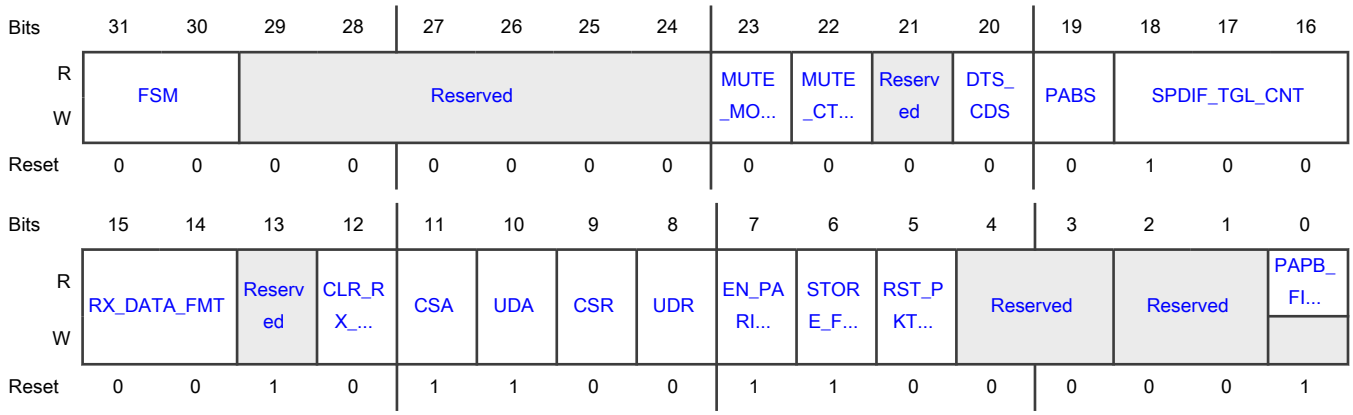
Configuration and control bist for the Receive audio block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
RX_DATAPATH_CTRL	180h	Data path control register
RX_DATAPATH_CTRL_SET	184h	Writing 1 to a bit in this register ensures that the corresponding bit in RX_DATAPATH_CTRL is 1
RX_DATAPATH_CTRL_CLR	188h	Writing 1 to a bit in this register ensures that the corresponding bit in RX_DATAPATH_CTRL is 0
RX_DATAPATH_CTRL_TOG	18Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in RX_DATAPATH_CTRL

Diagram



Fields

Field	Description
31-30	IEC60958-1 Frame Synchronization Mode

Table continues on the next page...

Table continued from the previous page...

Field	Description
FSM	Selects between frame sync or immediate capture of data. • 00 - Normal, Sync to block using B preamble • 01 - Debug mode • 10 - Sync to B or M preamble • 11 - Start with the first received preamble.
29-24 —	reserved
23 MUTE_MODE	Mute mode control Selects between SW or HW control for enabling Mute detection. • 0 - Channel Status control. MBIT field is used to select the bit that will control the mute request • 1 - SW Control
22 MUTE_CTRL	M0+ mute request If M0+ is used to control MUTE assertion, this bit is used to set mute mode. • 0 - Normal operation • 1 - Assert mute request
21 —	reserved
20 DTS_CDS	Enable DTS CD 14 preamble search Enable Preamble Pa Pb search which matches DTS CD 14 format.
19 PABS	Enable preamble search Enable Preamble Pa,Pb search
18-16 SPDIF_TGL_CNT	SPDIF wakeup source toggle count. When SPDIF is identified as a wakeup source, HW will look for the programmed number of toggles before asserting wakeup interrupt
15-14 RX_DATA_FMT	Indicates format of data stored in memory. • 00 - Store complete sub frame • 01 - Store 24 audio data bits of sub-frame. • 10 - Store 20 audio data bits of sub-frame. • 11 - Store 16 audio data bits of sub-frame.
13 —	reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 CLR_RX_FIFO	Clear Receive FIFO Clear receive FIFO. This is a self clear register bit.
11 CSA	Channel Status Acknowledge channel status acknowledge that should be set once channel status is processed.
10 UDA	User Data Acknowledge User data acknowledge that should be set once user data is processed.
9 CSR	Channel Status reset Reset channel status gathering and storage. This is an active high reset.
8 UDR	User data reset Reset user data gathering and storage. This is an active high reset.
7 EN_PARITY_CALC	Enable Parity calculation. This bit enables parity calculation. • 0 - Do not calculate Parity for un-compressed data • 1 - Calculate Parity for un-compressed data
6 STORE_FMT	Receive Data store format. This bit describes the format to store the received data in memory. • 0 - Store data with bit 0 of subframe at bit position 31 in sub-frame. • 1 - Store data with bit 0 of subframe at bit position 0 in sub-frame.
5 RST_PKT_CNT_FIFO	Resets the packet count fifo. This bit when enabled, resets the preamble match packet number FIFO • 0 - Do not reset the FIFO • 1 - Reset FIFO.
4-3 —	reserved
2-1 —	reserved
0 PAPB_FIFO_STATUS	This bit is an empty fifo indicator for PaPb match.

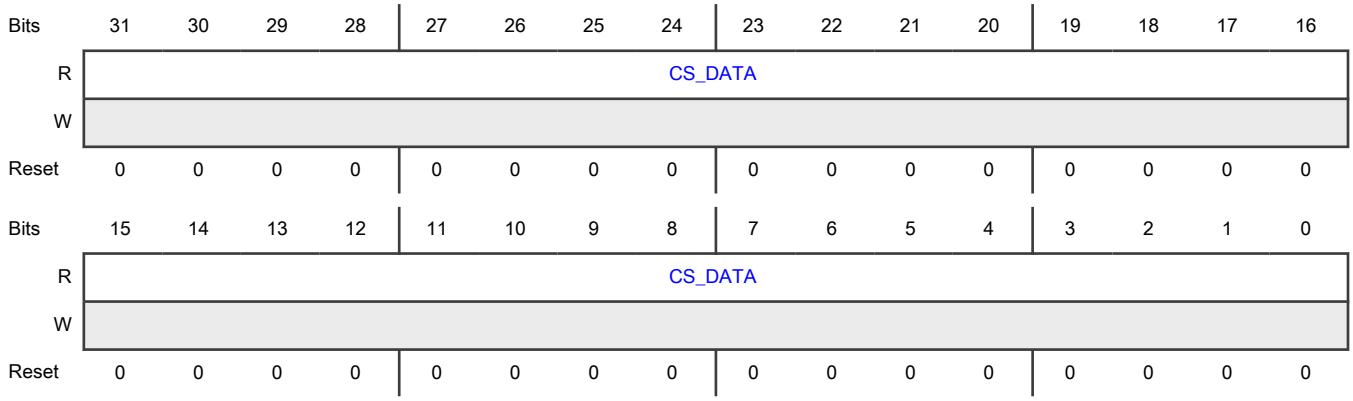
48.6.1.11 Channel staus bits (RX_CS_DATA_BITS_0 - RX_CS_DATA_BITS_5)

Holds the 192 bit channel status for the last received block. The channel status value is updated when a new block is received only if the current value read is acknowledged using RX_DATAPATH_CTRL.CSA.

Offset

Register	Offset
RX_CS_DATA_BITS_0	190h
RX_CS_DATA_BITS_1	194h
RX_CS_DATA_BITS_2	198h
RX_CS_DATA_BITS_3	19Ch
RX_CS_DATA_BITS_4	1A0h
RX_CS_DATA_BITS_5	1A4h

Diagram



Fields

Field	Description
31-0 CS_DATA	Channel Status bits 192 bit channel status / block

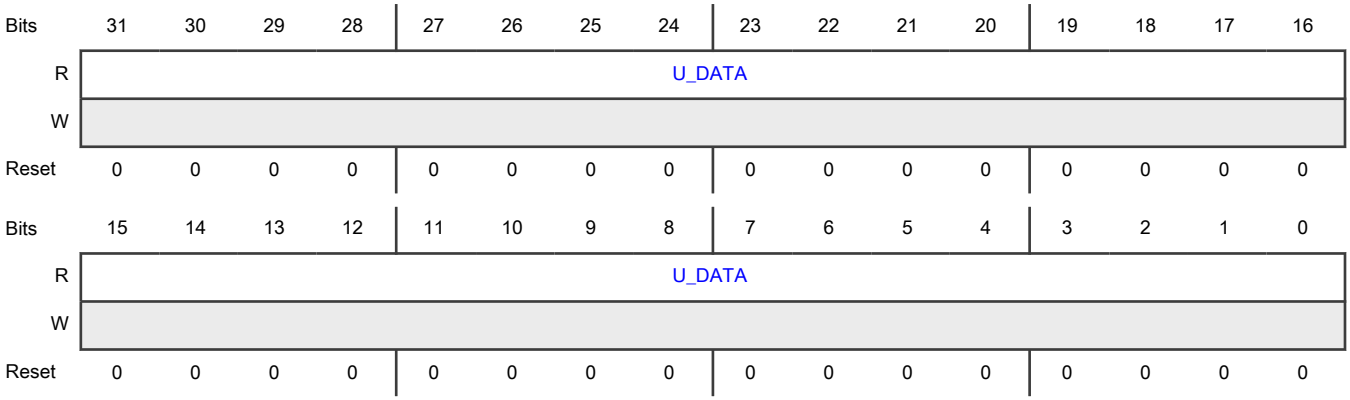
48.6.1.12 User data bits (RX_USER_DATA_BITS_0 - RX_USER_DATA_BITS_5)

Holds the 192 bit user data for the last received block. The user data value is updated when a new non-0 block is received only if the current value read is acknowledged using RX_DATAPATH_CTRL.UDA.

Offset

Register	Offset
RX_USER_DATA_BITS_0	1A8h
RX_USER_DATA_BITS_1	1ACh
RX_USER_DATA_BITS_2	1B0h
RX_USER_DATA_BITS_3	1B4h
RX_USER_DATA_BITS_4	1B8h
RX_USER_DATA_BITS_5	1BCh

Diagram



Fields

Field	Description
31-0 U_DATA	User data bits 192 bit user data / block

48.6.1.13 Receive Datapath counter control register (RX_DPATH_CNTR_CTRL)

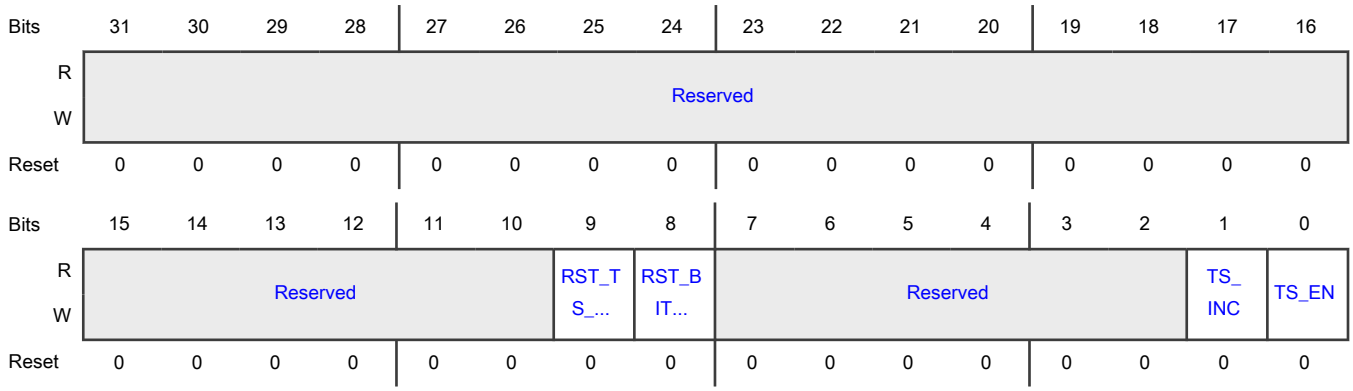
Controls the functioning of the RX datapath internal counters for time stamp and bit count.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
RX_DPATH_CNTR_CTRL	1C0h	Receive Datapath counter control register
RX_DPATH_CNTR_CTRL_SET	1C4h	Writing 1 to a bit in this register ensures that the corresponding bit in RX_DPATH_CNTR_CTRL is 1
RX_DPATH_CNTR_CTRL_CLR	1C8h	Writing 1 to a bit in this register ensures that the corresponding bit in RX_DPATH_CNTR_CTRL is 0
RX_DPATH_CNTR_CTRL_TOG	1CCh	Writing 1 to a bit in this register inverts the value of the corresponding bit in RX_DPATH_CNTR_CTRL

Diagram



Fields

Field	Description
31-10 —	reserved
9 RST_TS_CNTR	Reset timestamp counter. • 0 - Timestamp counter is not reset. • 1 - Timestamp counter is reset.
8 RST_BIT_CNTR	Reset bit counter. • 0 - Bit counter is not reset. • 1 - Bit counter is reset.
7-2 —	reserved
1 TS_INC	Timestamp Increment • 0 - Timestamp counter starts to increment when enabled and the bit counter has incremented.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• 1 - Timestamp counter increments when enabled.
0 TS_EN	Timestamp counter enable • 0 - Timestamp counter is disabled • 1 - Timestamp counter is enabled.

48.6.1.14 Receive Datapath Timestamp Counter Register (RX_DPATH_TSCR)

Current value of the timestamp counter.

Offset

Register	Offset
RX_DPATH_TSCR	1D0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CVAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CVAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CVAL	Timestamp counter value This counter is incremented every cycle. Increment start time is based on the timestamp control register

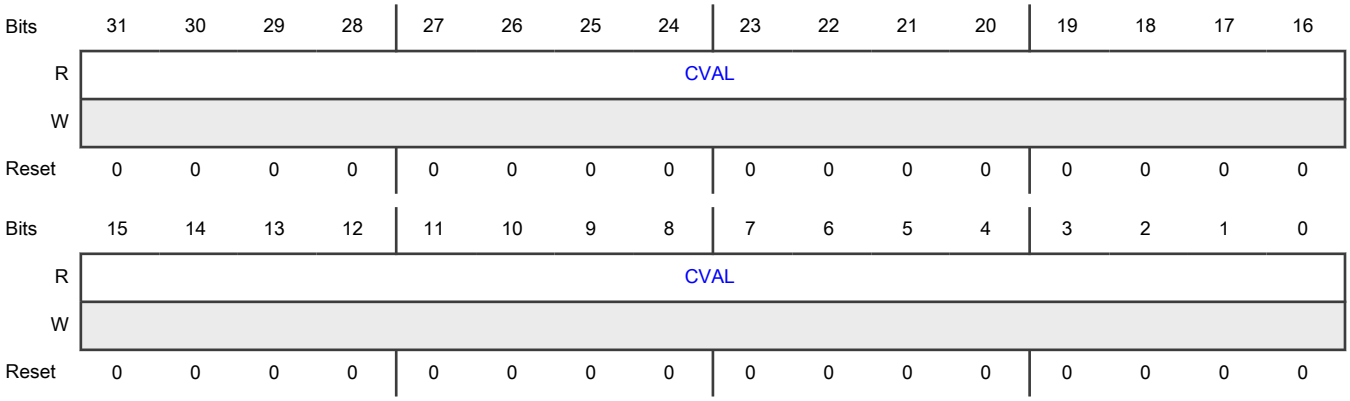
48.6.1.15 Receive Datapath Bit counter register (RX_DPATH_BCR)

This register returns the current value of the bit counter.

Offset

Register	Offset
RX_DPATH_BCR	1D4h

Diagram



Fields

Field	Description
31-0 CVAL	Bit count value The bit counter increments based on a bit count valid sgnal from the audio clock divider when receive datapath is active.

48.6.1.16 Receive datapath Bit count timestamp register. (RX_DPATH_BCTR)

Returns the value of the timestamp counter that corresponds to the value of timestamp when the bit count register is updated.

Offset

Register	Offset
RX_DPATH_BCTR	1D8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BCT_VAL	Bit count timestamp value This counter is updated every time the the bit counter is updated.

48.6.1.17 Receive datapath Bit read timestamp register. (RX_DPATH_BCRR)

Returns the value of the bit count timestamp register that corresponds to the value when bit counter was last read.

Offset

Register	Offset
RX_DPATH_BCRR	1DCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BCT_VAL	Bit count timestamp value This counter is updated every time the the bit counter is read.

48.6.1.18 Preamble match value register (PRE_MATCH_VAL)

Holds the values for IEC61937 Preamble match.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
PRE_MATCH_VAL	1E0h	Preamble match value register
PRE_MATCH_VAL_SET	1E4h	Writing 1 to a bit in this register ensures that the corresponding bit in PRE_MATCH_VAL is 1
PRE_MATCH_VAL_CLR	1E8h	Writing 1 to a bit in this register ensures that the corresponding bit in PRE_MATCH_VAL is 0
PRE_MATCH_VAL_TOG	1ECh	Writing 1 to a bit in this register inverts the value of the corresponding bit in PRE_MATCH_VAL

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PA_VAL															
Reset	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1	1
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PB_VAL															
Reset	1	1	1	1	1	0	0	0	0	1	1	1	0	0	1	0

Fields

Field	Description
31-16 PA_VAL	Preamble PA value Holds the PA value for PA PB match.
15-0 PB_VAL	Preamble PB value Holds the PB value for PA PB match.

48.6.1.19 Preamble match value register (DTS_PRE_MATCH_VAL)

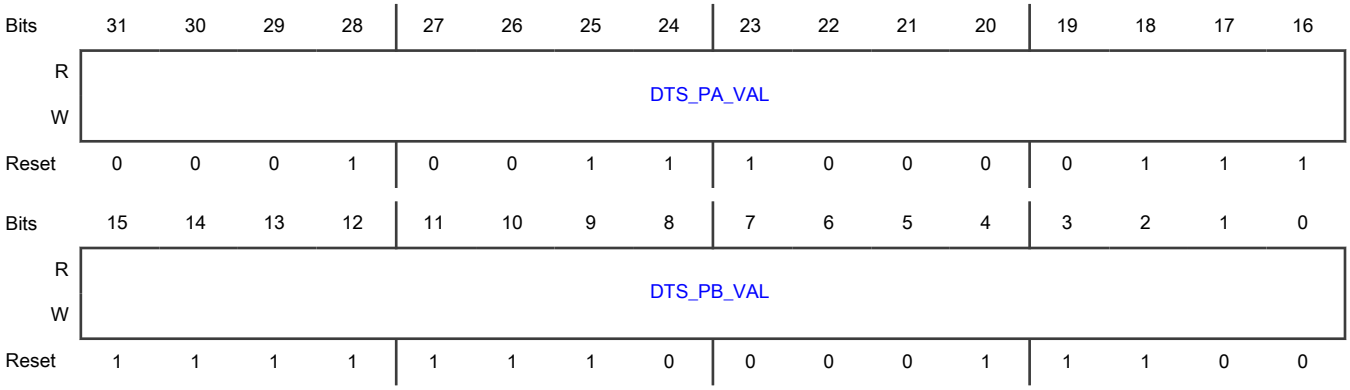
Holds the values for DTS CD 14 IEC61937 Preamble match.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DTS_PRE_MATCH_VAL	1F0h	Preamble match value register
DTS_PRE_MATCH_VAL_SET	1F4h	Writing 1 to a bit in this register ensures that the corresponding bit in DTS_PRE_MATCH_VAL is 1
DTS_PRE_MATCH_VAL_CLR	1F8h	Writing 1 to a bit in this register ensures that the corresponding bit in DTS_PRE_MATCH_VAL is 0
DTS_PRE_MATCH_VAL_TOG	1FCh	Writing 1 to a bit in this register inverts the value of the corresponding bit in DTS_PRE_MATCH_VAL

Diagram



Fields

Field	Description
31-16 DTS_PA_VAL	Preamble value Holds the odd sub-frame match value.
15-0 DTS_PB_VAL	Preamble value Holds the even sub-frame match value.

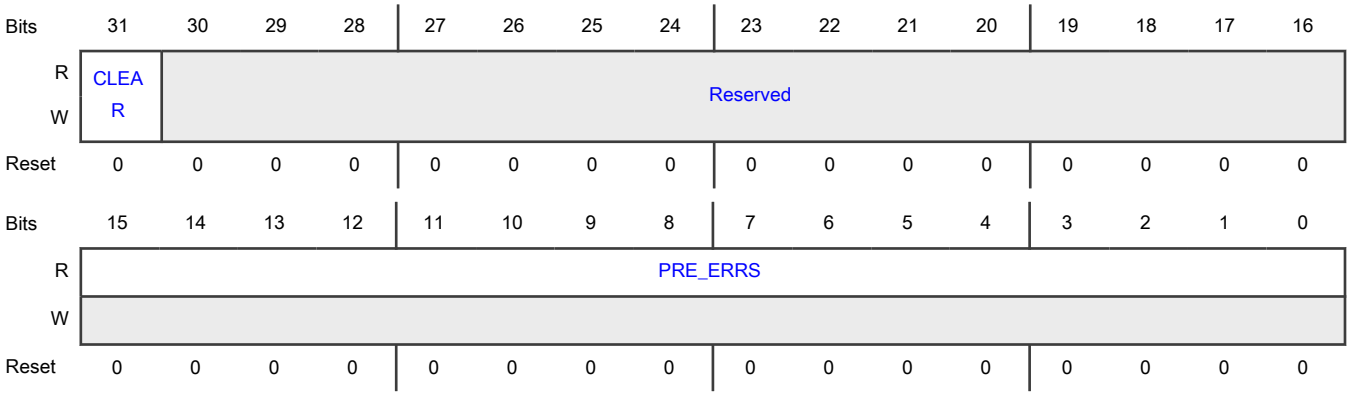
48.6.1.20 Error count for IEC60958-1 Block Synchronization. (RX_DPATH_PRE_ERR)

Error count for IEC60958-1 Block Synchronization. It is incremented when an unexpected B,M or W preamble is detected. Writing to bit 31 will clear this register

Offset

Register	Offset
RX_DPATH_PRE_ERR	200h

Diagram



Fields

Field	Description
31 CLEAR	Clear bit for error counter. When this bit is set, the preamble error counter will be reset.
30-16 —	reserved
15-0 PRE_ERRS	Preamble Error counter This counter is incremented when an unexpected Preamble is detected

48.6.1.21 Parity Error count for IEC60958-1 Blocks. (RX_DPATH_PARITY_ERR)

Parity error count for IEC60958-1 Block Synchronization. It is incremented when a parity error is received. Writing a 1 to bit 31 will clear this register. SW will have to clear bit 31 after writing to it.

Offset

Register	Offset
RX_DPATH_PARITY_ERR	204h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CLEAR	Reserved														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PRE_ERRS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 CLEAR	Clear bit for error counter. When this bit is set, the preamble error counter will be reset.
30-16 —	reserved
15-0 PRE_ERRS	Parity Error counter This counter is incremented when the parity of a received sub-frame does not match the value received.

48.6.1.22 Receive Data packet count. (RX_DPATH_PKT_CNT)

This register provides the Receive data packet count.

Offset

Register	Offset
RX_DPATH_PKT_CNT	210h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	VAL														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30-0 VAL	Data packet counter This status provides the number of received packets.

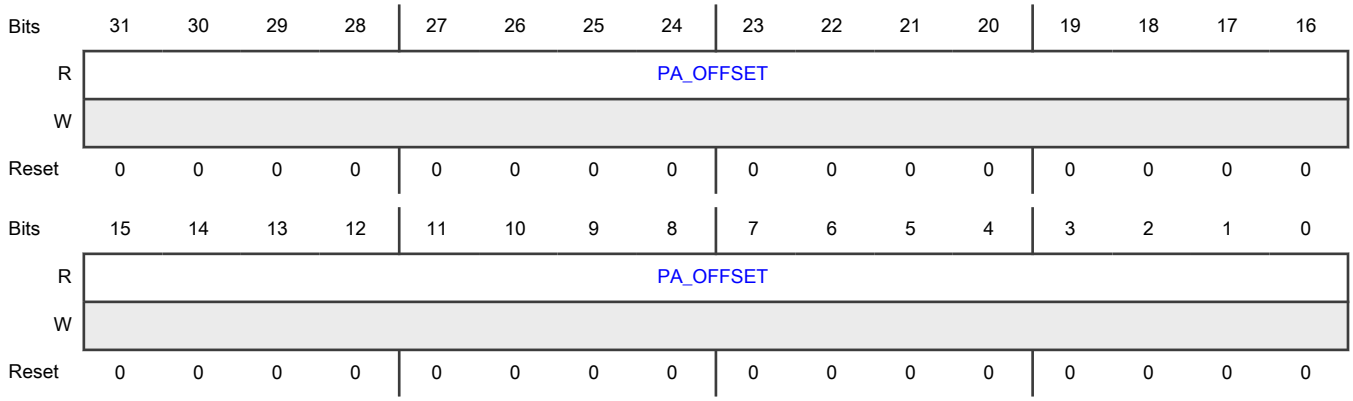
48.6.1.23 Preamble match offset value register (PRE_MATCH_OFFSET)

When an IEC61937 burst preamble is matched the current sample count shall be committed to a FIFO read from this register.

Offset

Register	Offset
PRE_MATCH_OFFSET	218h

Diagram



Fields

Field	Description
31-0 PA_OFFSET	Sample count value for PA offset match This reflects the value of a 32 bit counter that will count all samples received.

48.6.1.24 Transmit Data path control register (TX_DATAPATH_CTRL)

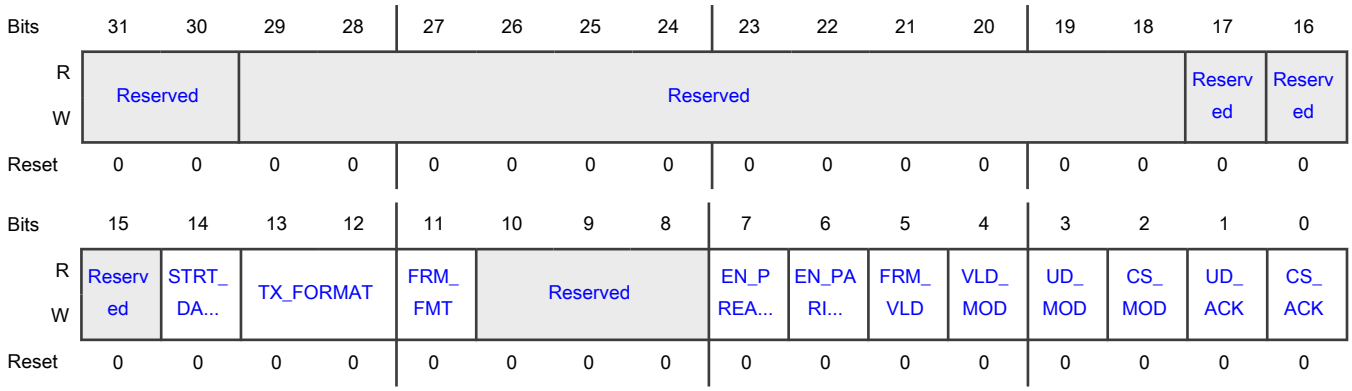
Configures and controls operations for the transmit of 60958 audio block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
TX_DATAPATH_CTRL	220h	Transmit Data path control register
TX_DATAPATH_CTRL_SET	224h	Writing 1 to a bit in this register ensures that the corresponding bit in TX_DATAPATH_CTRL is 1
TX_DATAPATH_CTRL_CLR	228h	Writing 1 to a bit in this register ensures that the corresponding bit in TX_DATAPATH_CTRL is 0
TX_DATAPATH_CTRL_TOG	22Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in TX_DATAPATH_CTRL

Diagram



Fields

Field	Description
31-30 —	reserved
29-18 —	reserved
17 —	reserved
16-15 —	reserved
14 STRT_DATA_TX	Start transmit of data. • 0 - Even though TX FIFO has data, do not transmit it right now. • 1 - As soon as TX_FIFO has data, enable transmitting audio sub-frames.
13-12 TX_FORMAT	Transmit data format

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field describes the number of bits to be transmitted / audio sub-frame. Each FIFO entry only represents one sub-frame worth of data. Bit reversal is indicated by TX_DATAPATH_CTRL.FRM_FMT • 0 - 32 bit data format. The audio data will be positioned within the subframe by SW. • 1 - 24 bit input data stored in bits 23:0 of TX FIFO • 2 - 20 bit input data stored in bits 19:0 of TX FIFO • 3 - 16 bit input data stored in bits 15:0 of TX FIFO
11 FRM_FMT	Frame format of input data • 0 - Frame data will be considered to be aligned to the MSB. i.e, 60958 frame bit 0 is at bit position 31 of the TX data FIFO. • 1 - Output 60958 frame format data will be considered to be aligned to the LSB. i.e, 60958 frame bit 0 is at bit position 0 of the TX data FIFO.
10-8 —	reserved
7 EN_PREAMBLE	Enable preamble insertion If this bit is set, Preamble matching the frame count will be inserted in the TX data path.
6 EN_PARITY	Enable parity insertion If this bit is set, parity will be calculaed and inserted in the TX data path.
5 FRM_VLD	Valid bit value If VLD_MOD bit is set, Valid bit will be replaced in the TX data path with the value in this bit.
4 VLD_MOD	Enable Valid bit insertion If this bit is set, Valid bit will be inserted in the TX data path from the FRM_VLD register bit.
3 UD_MOD	Enable User Data insertion If this bit is set, User data will be inserted in the TX data path from the UD buffer.
2 CS_MOD	Enable Channel Status insertion If this bit is set, Channel status will be inserted in the TX data path from the CS buffer.
1 UD_ACK	User Data ACK The user data registers are double buffered. HW sets this ack to let the host know that new values can be written in the UD buffer. Once the Buffer is updated, host should clear this value.
0 CS_ACK	Channel Status ACK The Channel status registers are double buffered. HW sets this ack to let the host know that new values can be written in the CS buffer. Once the Buffer is updated, host should clear this value.

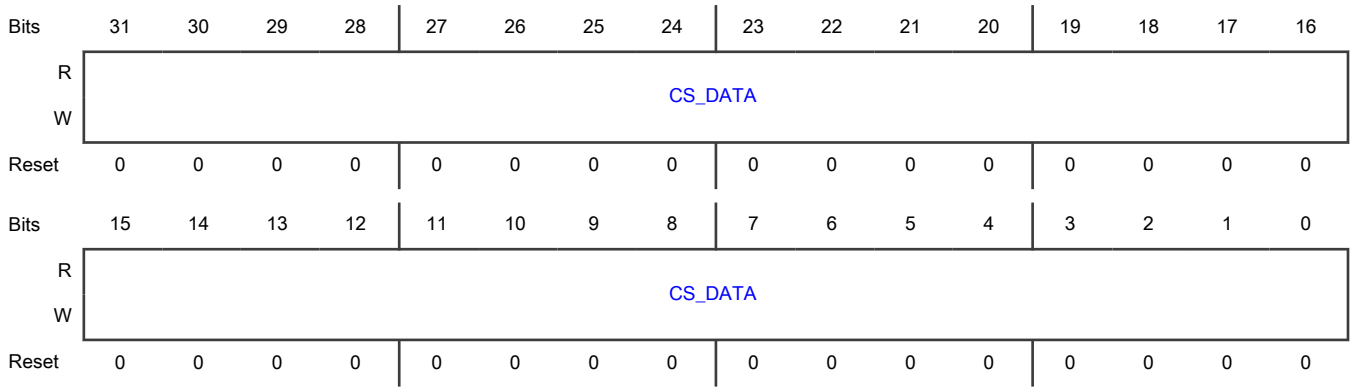
48.6.1.25 Channel staus bits (TX_CS_DATA_BITS_0 - TX_CS_DATA_BITS_5)

Holds the 192 bit channel status for the current block to transmit.

Offset

Register	Offset
TX_CS_DATA_BITS_0	230h
TX_CS_DATA_BITS_1	234h
TX_CS_DATA_BITS_2	238h
TX_CS_DATA_BITS_3	23Ch
TX_CS_DATA_BITS_4	240h
TX_CS_DATA_BITS_5	244h

Diagram



Fields

Field	Description
31-0 CS_DATA	Channel Status bits / block 192-bit channel status field per block of audio data. The value can be updated when TX_DATAPTH_CTRL.CS_ACK is set to 1. It does not need to be updated for every block. The same value will be transmitted if the registers are not updated.

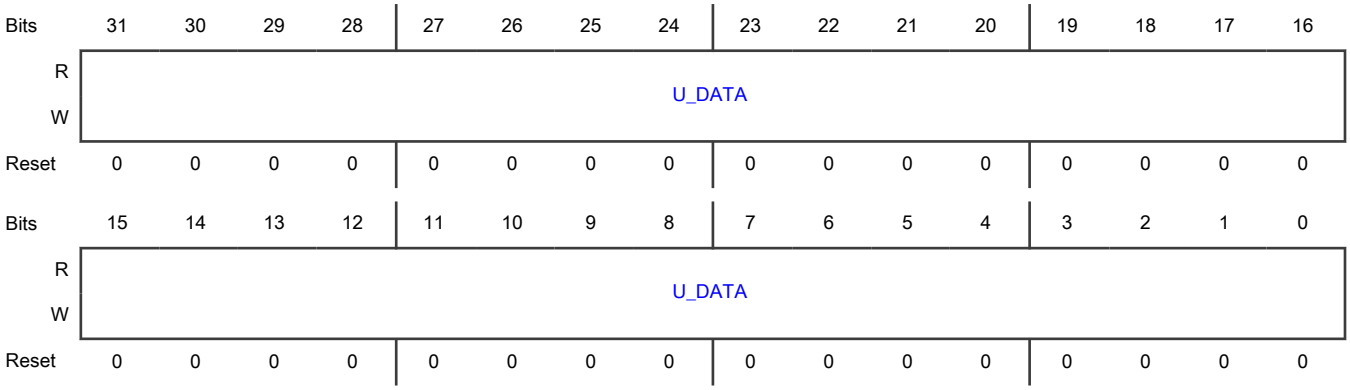
48.6.1.26 User data bits (TX_USER_DATA_BITS_0 - TX_USER_DATA_BITS_5)

Holds the 192 bit user data for the current block to transmit. It can be updated / block of data after the acknowledge field TX_DATAPATH_CTRL.UD_ACK is cleared. Setting this field indicates new user data block is available.

Offset

Register	Offset
TX_USER_DATA_BITS_0	248h
TX_USER_DATA_BITS_1	24Ch
TX_USER_DATA_BITS_2	250h
TX_USER_DATA_BITS_3	254h
TX_USER_DATA_BITS_4	258h
TX_USER_DATA_BITS_5	25Ch

Diagram



Fields

Field	Description
31-0 U_DATA	User data bits/block 192-bit User Data field / block of audio data.

48.6.1.27 Transmit datapath counter control register (TX_DPATH_CNTR_CTRL)

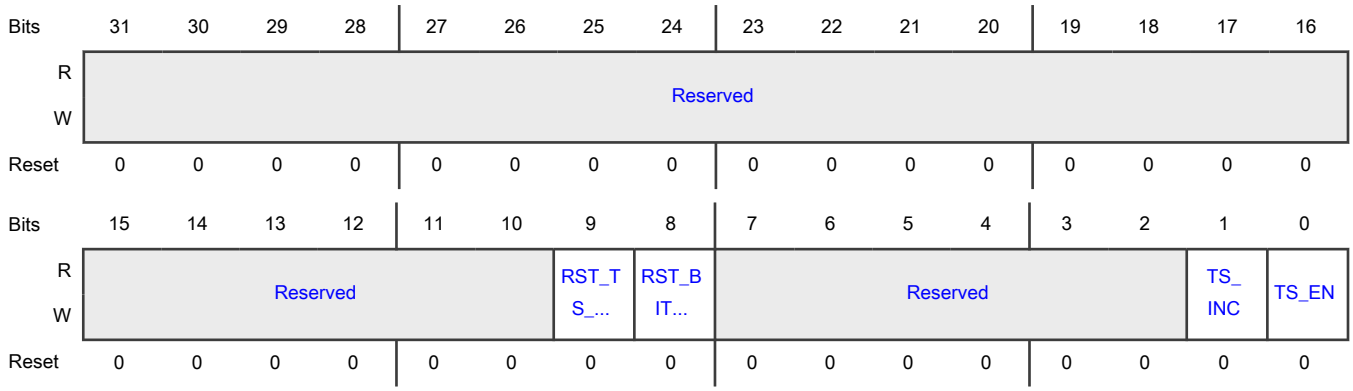
Controls the functioning of the transmit datapath internal counters for time stamp and bit count.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
TX_DPATH_CNTR_CTRL	260h	Transmit datapath counter control register
TX_DPATH_CNTR_CTRL_SET	264h	Writing 1 to a bit in this register ensures that the corresponding bit in TX_DPATH_CNTR_CTRL is 1
TX_DPATH_CNTR_CTRL_CLR	268h	Writing 1 to a bit in this register ensures that the corresponding bit in TX_DPATH_CNTR_CTRL is 0
TX_DPATH_CNTR_CTRL_TOG	26Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in TX_DPATH_CNTR_CTRL

Diagram



Fields

Field	Description
31-10 —	reserved
9 RST_TS_CNTR	Reset timestamp counter. • 0 - Timestamp counter is not reset. • 1 - Timestamp counter is reset.
8 RST_BIT_CNTR	Reset bit counter. • 0 - Bit counter is not reset. • 1 - Bit counter is reset.
7-2 —	reserved
1 TS_INC	Timestamp Increment • 0 - Timestamp counter starts to increment when enabled and the bit counter has incremented.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• 1 - Timestamp counter increments when enabled.
0 TS_EN	Timestamp counter enable • 0 - Timestamp counter is disabled • 1 - Timestamp counter is enabled.

48.6.1.28 Transmit Datapath Timestamp Counter Register (TX_DPATH_TSCR)

Returns the current value of the timestamp counter. This counter is incremented based on the timestamp control register TX_DPATH_CNTR_CTRL.

Offset

Register	Offset
TX_DPATH_TSCR	270h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CVAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CVAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 CVAL	Timestamp counter value

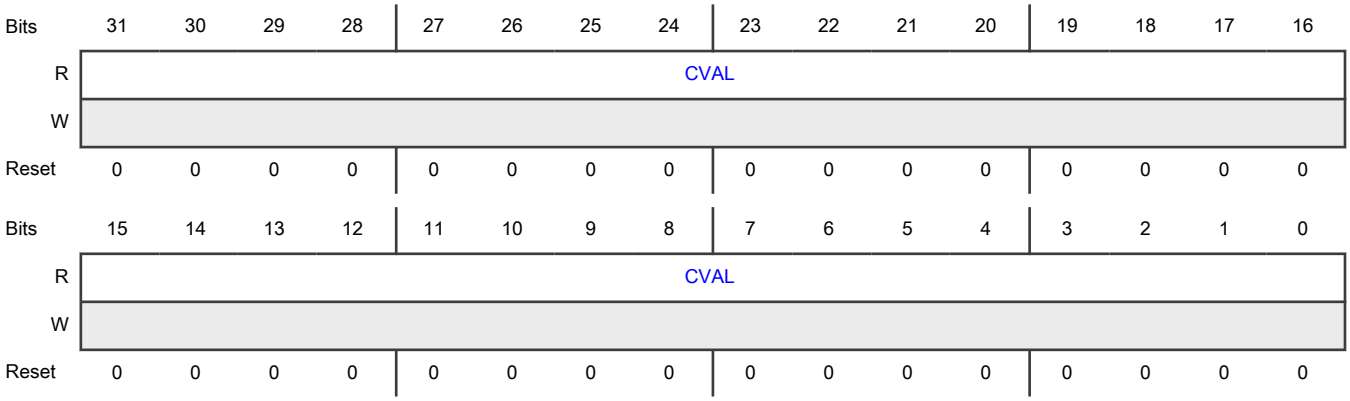
48.6.1.29 Transmit Datapath Bit counter register (TX_DPATH_BCR)

This register returns the current value of the bit counter. The bit counter increments based on a bit count signal from the audio clock divider when the transmit datapath is active.

Offset

Register	Offset
TX_DPATH_BCR	274h

Diagram



Fields

Field	Description
31-0 CVAL	Bit count value

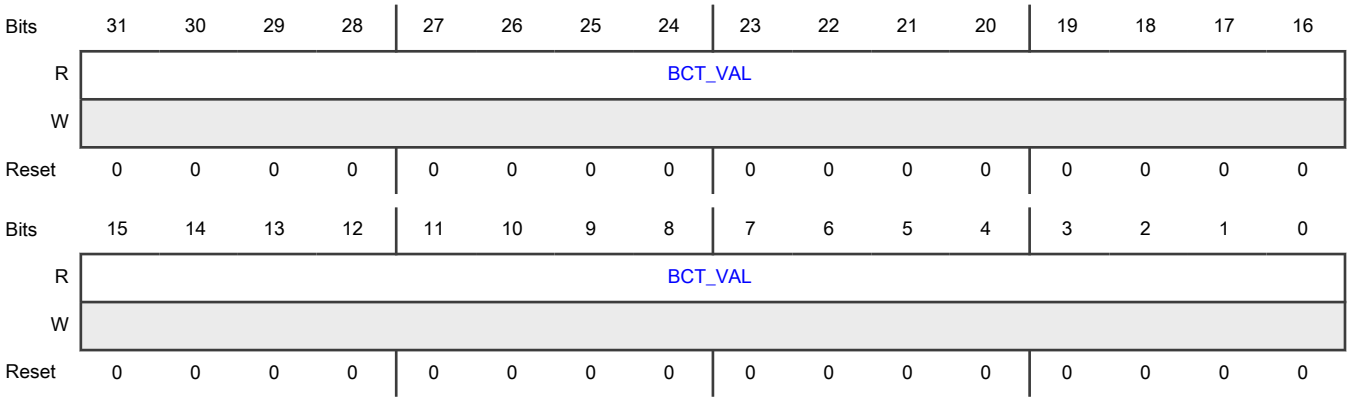
48.6.1.30 Transmit datapath Bit count timestamp register. (TX_DPATH_BCTR)

Returns the value of the timestamp counter that corresponds to the value when the bit count register was last updated.

Offset

Register	Offset
TX_DPATH_BCTR	278h

Diagram



Fields

Field	Description
31-0 BCT_VAL	Bit count timestamp value This counter is updated every time the the bit counter is updated.

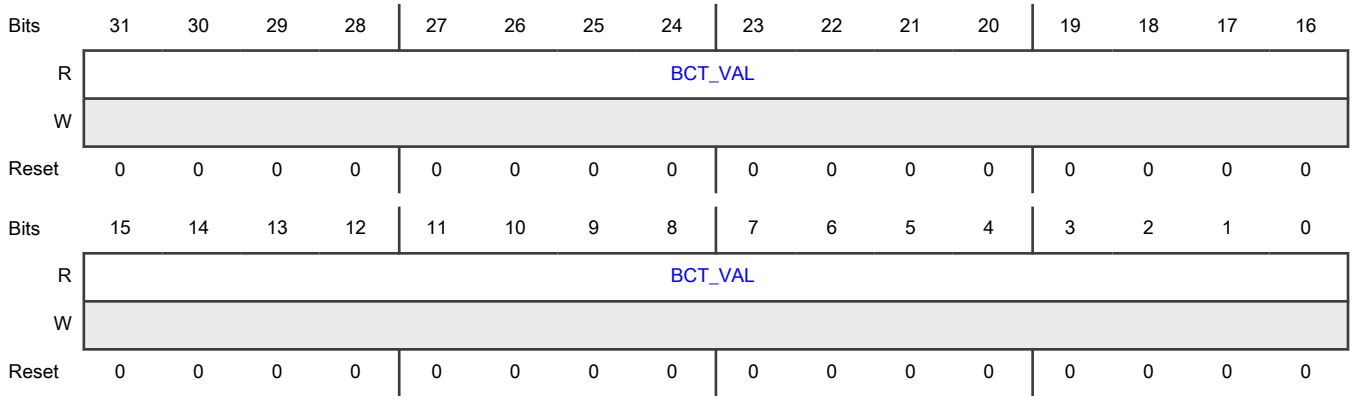
48.6.1.31 Transmmit datapath Bit read timestamp register. (TX_DPATH_BCRR)

Returns the value of the timestamp counter that corresponds to the value when bit counter was last read.

Offset

Register	Offset
TX_DPATH_BCRR	27Ch

Diagram



Fields

Field	Description
31-0 BCT_VAL	Bit count timestamp value This counter is updated every time the the bit counter is read.

48.7 Glossary

HW	Hardware
FW	Firmware
SW	Software

Chapter 49
Medium Quality Sound (MQS)

49.1 Chip-specific MQS information

Table 489. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

This chip includes two instance of MQS (MQS1 in AONMIX and MQS2 in WAKEUPMIX).

MQS1 settings register is in AONMIX and MQS2 settings register is in WAKEUPMIX.

49.2 Overview

Medium quality sound (MQS) is used to generate medium quality audio via a standard GPIO in the pinmux, allowing the user to connect stereo speakers or headphones to a power amplifier without an additional DAC chip.

NOTE

The terminology in this chapter has been updated to align with NXP's inclusive language standards, as shown in the table below.

Table 490. Updated terms

Updated term	Deprecated term
Initiator clock	Master clock

49.2.1 Block Diagram

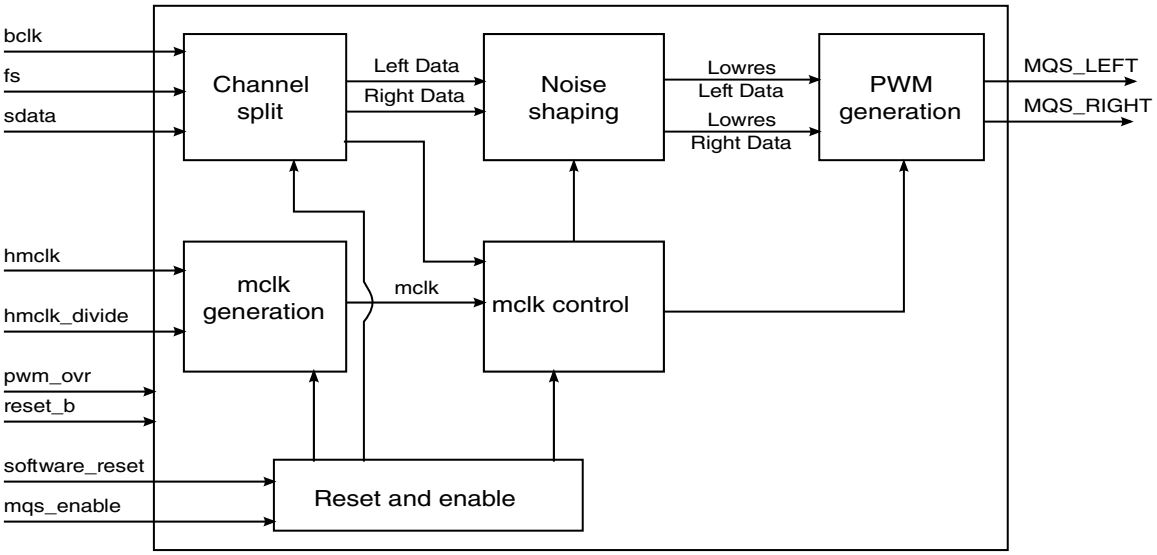


Figure 245. Block Diagram

49.2.2 Features

MQS has the following sub-modules:

- Channel Split
- Noise Shaping
- PWM Generation
- MCLK Generation
- MCLK Control
- Reset and Enable

49.3 Functional Description

MQS accepts the following inputs:

- 2-channel, LSB-valid 16-bit, MSB shift-out first serial data (sdata)
- Frame sync asserting with the first bit of the frame (fs)
- Bit clock used to shift data out on the negative clock edge (bclk)

The signals from the SAI are in left_justified format. MQS provides the SNR target as no more than 20dB for the signals below 10 kHz. The signals above 10 kHz will have worse THD+N values.

MQS provides only simple audio reproduction. No internal pop, click or distortion artifact reduction methods are provided.

The following sections provides brief descriptions of the MQS sub-blocks.

49.3.1 Channel Split

The Channel Split sub-block splits the I2S signals into separate left channel and right channel audio data.

49.3.2 Noise Shaping

The Noise Shaping sub-block uses the sigma-delta algorithm to generate low-resolution, very high sampling audio, while the audio sampling rate is increased.

49.3.3 PWM Generation

Generates the bit stream to the GPIO, which is then used to drive the amplifier and then to drive the external speakers or headphones.

49.3.4 MCLK Generation

The MCLK Generation sub-block is used to generate the initiator clock (mclk). The frequency of mclk is determined by hmclk_divide (BLK_CTRL[MQS_CLK_DIV]).

49.3.5 MCLK Control

The MCLK Control sub-block is used as a metronome to co-ordinate the different functional blocks working synchronously.

49.3.6 Reset and Enable

The Reset and Enable sub-block is used to generate the reset and enable logic to different clock domains.

49.3.7 Clocking

The following table lists the MQS clocks. See Interface Signals for more information.

Table 491.

Clock Name	Description
hmclk	Maximum bit clock
bclk	Bit clock from I2S signal

49.4 External Signals

The following table describes the external signals of MQS:

Table 492. MQS External Signals

Signal	Description	Direction
MQS_LEFT	Left signal output	O
MQS_RIGHT	Right signal output	O

49.4.1 Interface Signals

MQS module has the following interface signals.

Table 493.

Signal Name	In/Out	Bit Width	Description	Comments
reset_b	In	1	asynchronous reset	
software_reset	In	1	Software reset	From BLK_CTRL

Table continues on the next page...

Table 493. (continued)

Signal Name	In/Out	Bit Width	Description	Comments
mqs_enable	In	1	module enable	From BLK_CTRL
pwm_ovr	In	1	PWM oversampling ratio 1=64, 0=32	From BLK_CTRL
hmclk	In	1	Maximum bit clock, used to generate the mclk, divider ratio is controlled by mqs_hmclk_divide	Max 66.5MHzTypical 24.576MHz
hmclk_divide	In	8	Divider ratio control for mclk from hmclk	From BLK_CTRL
bclk	In	1	bit clock from I2S signal	
fs	In	1	frame sync from I2S signal	
sdata	In	1	serial audio data from I2S signal	

49.5 Initialization

The user needs to program the SAI to output 2-channel LSB-16 bit active left-justified signal, and then program the related BLK_CTRL bits.

Due to the different devices connected to MQS, and different high frequency behaviors of the connected analog circuits, the user needs choose the appropriate MQS_CLK_DIV and MQS_OVERSAMPLE values for the best audible effects.

49.6 Application Information

MQS has no internal programmable registers. But it does have some programmability from **BLK_CTRL**.

Register Bitfields	Description
MQS_OVERSAMPLE	Used to control the PWM oversampling rate compared with mclk. 1—64, 0—32.
MQS_EN	MQS enable. 1—Enable MQS, 0—Disable MQS
MQS_SW_RST	MQS software reset. 1—Enable software reset for MQS 0—Exit software reset for MQS

Table continues on the next page...

Table continued from the previous page...

Register Bitfields	Description
MQS_CLK_DIV	Divider ration control for mclk from hmclk. 0—mclk frequency = hmclk frequency; 1—mclk frequency = ½*hmclk frequency; 2—mclk frequency = 1/3*hmclk frequency; …; n—mclk frequency = 1/(n+1)*hmclk frequency

Chapter 50
Multimedia Overview

50.1 Multimedia Overview

The display and camera subsystem consists of modules below:

- LCDIF
- LVDS Display Bridge (LDB LVDS Tx Controller) and LVDS PHY
- Image Sensor Interface (ISI)
- Parallel Camera Interface
- Pixel Pipeline (PXP)
- MIPI-CSI2 controller
- MIPI-DSI controller
- MIPI-DPHY (Use for both MIPI-CSI and MIPI-DSI protocols)

50.1.1 LCD Interface (LCDIF)

The chip includes one instance of LCDIF.

- One layer can support: Programmable plane size (Width/Height/Pitch) and X/Y offset on the panel
- Index color support (1/2/4/8 bpp)
- AXI data bus width (128-bit)
- Support 8-bit / 16-bit / 18-bit / 24-bit / 32-bit pixel depth
- Support DOTCLK mode for MIPI-DPI interface
- One LCDIF can drive any of three displays or drive the same output to multiple displays:
  - MIPI DSI (up to 1920x1200p60)
  - LVDS Tx (up to 1366x768p60 or 1280x800p60)
  - Parallel Display (MIPI-DPI interface)

50.1.2 MIPI DSI

The chip support one 4-lane MIPI DSI display with pixels from the LCDIF. The key features of the MIPI DSI (controller and PHY) include:

- Conforms to MIPI-DSI specification v1.2 and MIPI-DPHY specification v1.2
- Maximum resolution limited to resolutions achievable with a 200MHz pixel clock and active pixel rate of 140Mpixel/s with 24-bit RGB. This includes resolutions such as:
  - 1080p60 or 1920x1200p60
- Support up to 4 Tx data lanes (plus 1 Tx clock lane)
- Support 80Mbps - 1.5Gbps data rate per lane in high speed operation
- Support 10Mbps data rate in low power operation

50.1.3 MIPI CSI2

The chip supports one 2-lane MIPI CSI2 camera inputs. The key features of the MIPI CSI2 (controller and PHY) include:

- Compliant to MIPI-CSI2 specification v1.2 for interfacing to a MIPI-DPHY specification v1.2
- Supports up to 2 Rx data lanes (plus 1 Rx clock lane)

50.1.4 LVDS Tx and Pixel Mapper

The LVDS Display Bridge (LDB) connects to an External LVDS Display Interface. The purpose of the LDB is to support flow of synchronous RGB data to external display devices through the LVDS interface.

- Supports FPD link
- Supports single channel (4 lanes) output at up to 80MHz pixel clock and LVDS clock, with 7:1 ratio from LVDS data to pixel clock, implying up to 560Mbps LVDS data rate. This supports resolutions up to approximately 1366x768p60 or 1280x800p60.
- Supports VESA and JEIDA pixel mapping
- Supports LVDS Transmitter with four 7-bit channels. Each channel sends the 6 pixel bits and one control signal at 7 times the pixel clock rate. The data and control signals are transmitted over an LVDS link.

50.1.5 Image Sensor Interface

The chip supports one instance of ISI configured to support one camera. The ISI is a simple camera interface that supports image processing and transfer via a bus master interface. One camera input can be connected to either of:

- MIPI-CSI camera
- Parallel camera input

Some features included are :

- Many standard pixel formats commonly used in many camera inputs protocols.
- Image processing for
  - 1 processed camera streams at 1080p30
- Image down scaling via decimation and bi-phase filtering.
- Color space conversion.
- Interlaced to progressive conversions.

50.1.6 Parallel Camera Interface

The Parallel Camera Interface module consists of the following:

- Configurable interface logic to support most commonly used parallel CMOS sensors
- Supports up to 80MHz input clock from sensor
- Supports the following input data formats
  - 8-bit/10-bit Camera Sensor Interface (CSI)
  - 8-bit data port for RGB, YCbCr, and YUV data input
  - 8-bit/10-bit data ports for Bayer data input
- Parallel Camera Interface is hooked to the Imaging subsystem via the Pixel Link

50.1.7 Pixel Pipeline (PXP)

The chip supports a PXP for simple composition and acceleration for use by operating systems such as Linux.

Some features included are :

- RGBA and ARGB formats are supported for the PS buffer
- RGBA format is supported for the AS buffer
- One CSC and one rotation engine is supported
- Flexible image composition options – alpha, chroma key
- Porter-Duff operation
- Image rotation (90°, 180°, 270°)
- Image resize
- Color space conversion
- Multiple pixel format support (RGB, YUV444, YUV422, YUV420, YUV400)
- Standard 2D-DMA operation

Chapter 51
LCD Interface (LCDIF)

51.1 Chip-specific LCDIF information

Table 494. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

This chip includes one instance of LCDIF in MEDIAMIX.

NOTE

This chip includes FIFO defines with static support up to 1080P60Hz 4B, while dynamical support is only up to 1080P60Hz 2B. 1080P60Hz 4B is not be supported through Hardware Fast Frequency Change (HWFFC).

51.2 Overview

The LCD Interface (LCDIF) is a system master that fetches graphics stored in memory and display them on a TFT LCD panel. A wide range of panel sizes is supported and the timing of the interface signals is highly configurable. Graphics are read directly from memory. Graphics may be encoded in a variety of formats to optimize memory usage.

51.2.1 Block diagram

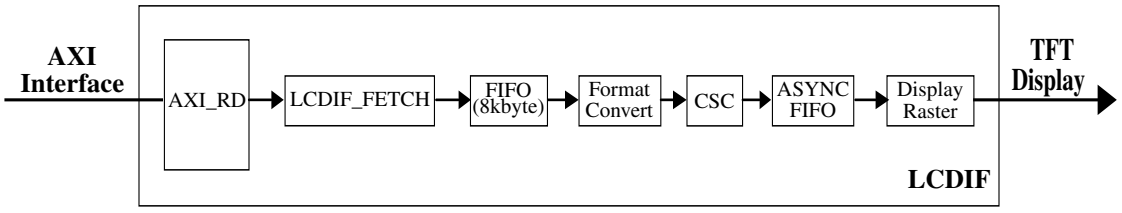


Figure 246. LCDIF block diagram

The above block diagram shows the LCDIF architecuture.

The input source can be configured with:

- Horizontal and vertical size of graphic
- Address of graphic in memory
- Color formats

The values in the registers can be changed at any time, the panel content will be updated automatically in the panel refresh cycle when SHADOW_EN bit is set.

51.2.2 Features

The LCDIF includes the following features:

- The display mode supports DOTCLK mode only.
- One layer can support:
  - Programmable plane size(Width/Height/Pitch) on the panel.

51.3 Functional description

The LCDIF is a master on the bus fabric; it fetches graphic source information directly from memory and sends data to a TFT LCD pannel.

As the LCDIF is a master on the NIC, it can access directly any memory or device connected to the NIC as a slave. Therefore, any compatiable graphic stored anywhere on-chip can be displayed on the connected TFT LCD panel with no further intervention from CPU.

51.3.1 Graphics and data format

The memory location of the graphic that is displayed on the layer is defined by register in the control descriptor for the layer .

This 36-bit value can contain the address of any memory location in the memory map of the system.

The format of the data that describes the graphic is defined by the BPP bit field in the control descriptor for the layer . By choosing an appropriate format, it is possible to optimize the memory required by the graphics in use.

The CSC module receives pixels in any color space and can convert the pixels into any of RGB, YCbCr color spaces.

All coefficients are programmable and in the two's compliment notation.

The following equations indicate the CSC modules ALU architecture.

Selecting RGB output in CSC0_CTRL[CSC_MODE] configures the ALU in the following manner:

R = A1(Y-D1) + A2(U-D2) + A3(V-D3)

G = B1(Y-D1) + B2(U-D2) + B3(V-D3)

B = C1(Y-D1) + C2(U-D2) + C3(V-D3)

Selecting YCbCr output configures the ALU in the alternate manner:

Y = A1*R + A2*G + A3*B + D1

U = B1*R + B2*G + B3*B + D2

V = C1*R + C2*G + C3*B + D3

Saturation of each color channel is checked and corrected for excursions outside the nominal color space. Overflow for the three channels are saturated at 0xFF and underflow is saturated at 0x00.

51.3.2 Clocks

The table below shows the clocks described in this module:

Table 495. Clocks

Clocks	Description
pix_clk	Pixel Clock

Table continues on the next page...

Table 495. Clocks (continued)

Clocks	Description
	This clock should be configured according to TFT panel specification for supported display resolution
apb_clk	System Bus access clock
b_clk	LCDIF peripheral access bus clock

51.3.3 Reset

There is one asynchronous chip reset input to LCDIF, all the internal registers should be forced into their reset state when reset_n is active low.

There is one software reset bit (CTRL[SW_RESET]), which can reset all the internal registers except user configured registers.

51.3.4 Interrupts

This section describes all the interrupts that the LCDIF module generates.

The VSYNC interrupt flag indicates that the LCDIF has begun the next panel refresh period.

The UNDERRUN interrupt flag indicates the TXFIFO for display in underrun condition.

The VS_BLANK interrupt flag gives indication of the start of the vertical blanking period.

There are FIFO_EMPTY, DMA_DONE and DMA_ERR interrupt flags for each layer which indicate the status of pixel FIFO, frame buffer fetch and AXI interface respectively.

51.3.5 External Signals

Table 496. External Signals

Signal	Direction	Description
DOTCLK	OUTPUT	Pixel clock used to drive the display panel.
VSYNC	OUTPUT	Vertical Sync signal, indicating the beginning of a new frame.
HSYNC	OUTPUT	Horizontal Sync signal, indicating the beginning of a new line.
ENABLE	OUTPUT	Data Enable. Indicates when there is valid pixel data.
DATA[23:0]	OUTPUT	Red, green and blue data output.

51.3.6 Application information

The figure below shows the details of software flow for the LCDIF module.

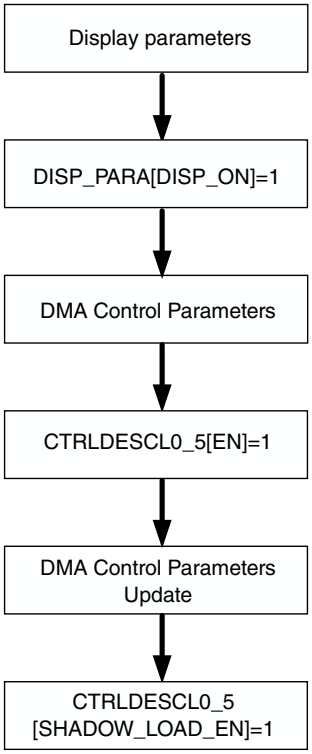


Figure 247. Software flow diagram

51.4 Memory Map and register definition

This section includes the LCDIF module memory map and detailed descriptions of all registers.

51.4.1 LCDIF register descriptions

51.4.1.1 LCDIF memory map

LCDIF base address: 4AE3_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	LCDIF display control Register (CTRL)	32	RW	8000_0000
4	LCDIF display control Register (CTRL_SET)	32	RW	8000_0000
8	LCDIF display control Register (CTRL_CLR)	32	RW	8000_0000
C	LCDIF display control Register (CTRL_TOG)	32	RW	8000_0000
10	Display Parameter Register (DISP_PARA)	32	RW	0000_0000
14	Display Size Register (DISP_SIZE)	32	RW	0000_0000
18	Horizontal Sync Parameter Register (HSYN_PARA)	32	RW	0003_0003
1C	Vertical Sync Parameter Register (VSYN_PARA)	32	RW	0003_0003

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
20	Vertical and Horizontal Pulse Width Parameter Register (VSYN_HSYN_WIDTH)	32	RW	0003_0003
24	Interrupt Status Register for domain 0 (INT_STATUS_D0)	32	RW	0000_0000
28	Interrupt Enable Register for domain 0 (INT_ENABLE_D0)	32	RW	0000_0000
30	Interrupt Status Register for domain 0 (INT_STATUS_D1)	32	RW	0000_0000
34	Interrupt Enable Register for domain 0 (INT_ENABLE_D1)	32	RW	0000_0000
200	Control Descriptor Layer Register 1 (CTRLDESCL0_1)	32	RW	0000_0000
208	Control Descriptor Layer Register 3 (CTRLDESCL0_3)	32	RW	0000_0000
20C	Control Descriptor Layer Register 4 (CTRLDESCL_LOW0_4)	32	RW	0000_0000
210	Control Descriptor Layer Register 4 (CTRLDESCL_HIGH0_4)	32	RW	0000_0000
214	Control Descriptor Layer Register 5 (CTRLDESCL0_5)	32	RW	0000_0000
21C	Color Space Conversion Ctrl Register (CSC0_CTRL)	32	RW	0000_0001
220	Color Space Conversion Coefficient Register 0 (CSC0_COEF0)	32	RW	0000_0000
224	Color Space Conversion Coefficient Register 1 (CSC0_COEF1)	32	RW	0000_0000
228	Color Space Conversion Coefficient Register 2 (CSC0_COEF2)	32	RW	0000_0000
22C	Color Space Conversion Coefficient Register 3 (CSC0_COEF3)	32	RW	0000_0000
230	Color Space Conversion Coefficient Register 4 (CSC0_COEF4)	32	RW	0000_0000
234	Color Space Conversion Coefficient Register 0 (CSC0_COEF5)	32	RW	0000_0000
238	Memory request priority threshold register (PANIC0_THRES)	32	RW	0100_0100

51.4.1.2 LCDIF display control Register (CTRL)

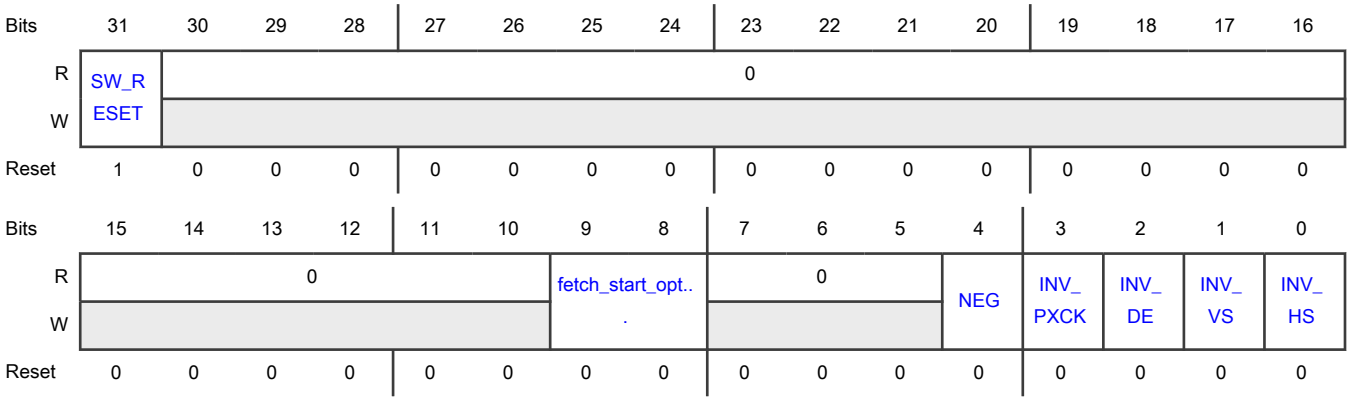
This register control the polarity for HSYNC/VSYNC/DE of display.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
CTRL	0h	LCDIF display control Register
CTRL_SET	4h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 1
CTRL_CLR	8h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 0
CTRL_TOG	Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CTRL

Diagram



Fields

Field	Description
31 SW_RESET	SW_RESET Software reset, high active. When write 1 ,all logical will be reset. 0 - No action 1 - All LCDIF internal registers are forced into their reset state. User registers are not affected.
30-10 —	Reserved
9-8 fetch_start_option	Indicates when to start fetching for new frame. This signals also decide the shadow load, fifo clear time 00 - fetch start as soon as FPV begins(as the end of the data_enable) 01 - fetch start as soon as PWV begins 10 - fetch start as soon as BPV begins 11 - fetch start as soon as RESV begins(still have hsync blanking for margin)
7-5 —	Reserved
4 NEG	Indicates if value at the output (pixel data output) needs to be negated. 0 - Output is to remain same 1 - Output to be negated
3 INV_PXCK	Polarity change of Pixel Clock. 0 - Display samples data on the falling edge 1 - Display samples data on the rising edge
2 INV_DE	Invert Data Enable polarity 0 - Data enable is active high 1 - Data enable is active low

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 INV_VS	Invert Vertical synchronization signal. 0 - VSYNC signal not inverted (active HIGH). 1 - Invert VSYNC signal (active LOW).
0 INV_HS	Invert Horizontal synchronization signal. 0 - HSYNC signal not inverted (active HIGH). 1 - Invert HSYNC signal (active LOW).

51.4.1.3 Display Parameter Register (DISP_PARA)

Offset

Register	Offset
DISP_PARA	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DISP_ON	SWAP_EN	LINE_PATTERN				DISP_MODE		BGND_R							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BGND_G								BGND_B							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 DISP_ON	Display panel On/Off mode. 0 - Display Off. 1 - Display On.
30 SWAP_EN	output data swap enable. 0 - swap disable 1 - swap enbale, output data will swap the high 16bits with the low 16bits.
29-26	LCDIF line output order.

Table continues on the next page...

Table continued from the previous page...

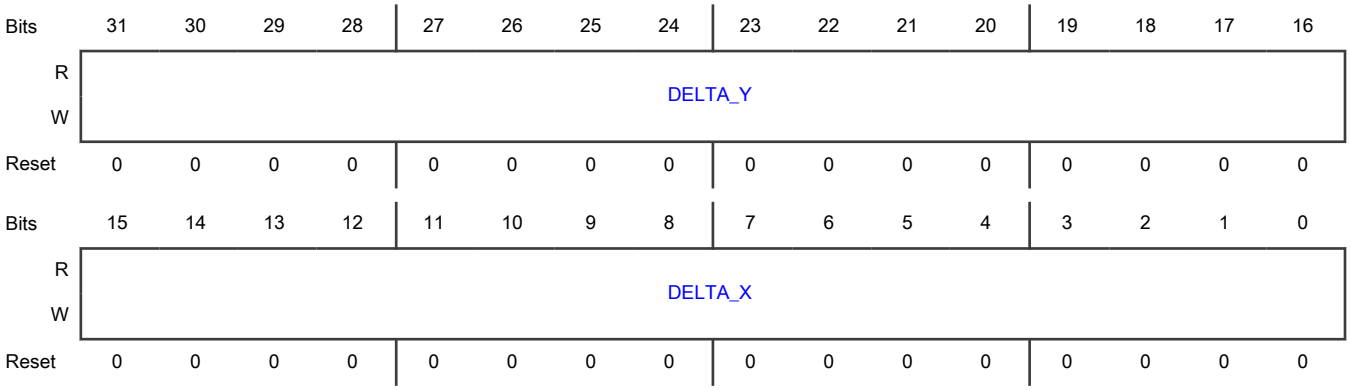
Field	Description
LINE_PATTERN	0000 - RGB/YUV. 0001 - RBG. 0010 - GBR. 0011 - GRB/UYV. 0100 - BRG. 0101 - BGR. 0110 - RGB555. 0111 - RGB565. 1000 - YUYV at [15:0]. For line pattern with "1000 - YUYV at [15:0]，Y is [15:8], U is [7:0], and next 16th bit is Y and V. 1001 - UYVY at [15:0]. 1010 - YVYU at [15:0]. 1011 - YUYV at [15:0]. 1100 - YUYV at [23:8]. 1101 - UYVY at [23:8]. 1110 - YVYU at [23:8]. 1111 - YUYV at [23:8].
25-24 DISP_MODE	LCDIF operating mode. 00 - Normal mode. Panel content controlled by layer configuration. 01 - display Mode1.(BGND Color Display) 10 - display Mode2.(Column Color Bar) 11 - display Mode3.(Row Color Bar)
23-16 BGND_R	Background Red component value in display Mode 1.
15-8 BGND_G	Background Green component value in display Mode 1.
7-0 BGND_B	Background Blue component value in display Mode 1.

51.4.1.4 Display Size Register (DISP_SIZE)

Offset

Register	Offset
DISP_SIZE	14h

Diagram



Fields

Field	Description
31-16 DELTA_Y	Sets the display size vertical resolution in pixels.
15-0 DELTA_X	Sets the display size horizontal resolution in pixels.

51.4.1.5 Horizontal Sync Parameter Register (HSYN_PARA)

HSYN_PARA register sets timing parameters related to the horizontal synchronization signal generation for the TFT. The fields FP_H, BP_H, and PW_H stand for HSYNC signal front-porch, back-porch, and active pulse width, respectively.

Offset

Register	Offset
HSYN_PARA	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BP_H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP_H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Fields

Field	Description
31-16 BP_H	HSYNC back-porch pulse width (in pixel clock cycles). Pulse width has a minimum value of 1.
15-0 FP_H	HSYNC front-porch pulse width (in pixel clock cycles). Pulse width has a minimum value of 1.

51.4.1.6 Vertical Sync Parameter Register (VSYN_PARA)

VSYN_PARA register sets timing parameters of the TFT related to the vertical synchronization signal generation. The fields FP_V, BP_V, and PW_V stand for VSYNC signal front-porch, back-porch, and active pulse width, respectively.

Offset

Register	Offset
VSYN_PARA	1Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BP_V															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP_V															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Fields

Field	Description
31-16 BP_V	VSYNC back-porch pulse width (in horizontal line cycles). Pulse width has a minimum value of 1.
15-0 FP_V	VSYNC front-porch pulse width (in horizontal line cycles). Pulse width has a minimum value of 1.

51.4.1.7 Vertical and Horizontal Pulse Width Parameter Register (VSYN_HSYN_WIDTH)

VSYN_PARA register sets timing parameters of the TFT related to the vertical synchronization signal generation. The fields FP_V, BP_V, and PW_V stand for VSYNC signal front-porch, back-porch, and active pulse width, respectively.

Offset

Register	Offset
VSYN_HSYN_WIDTH	20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PW_V															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PW_H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Fields

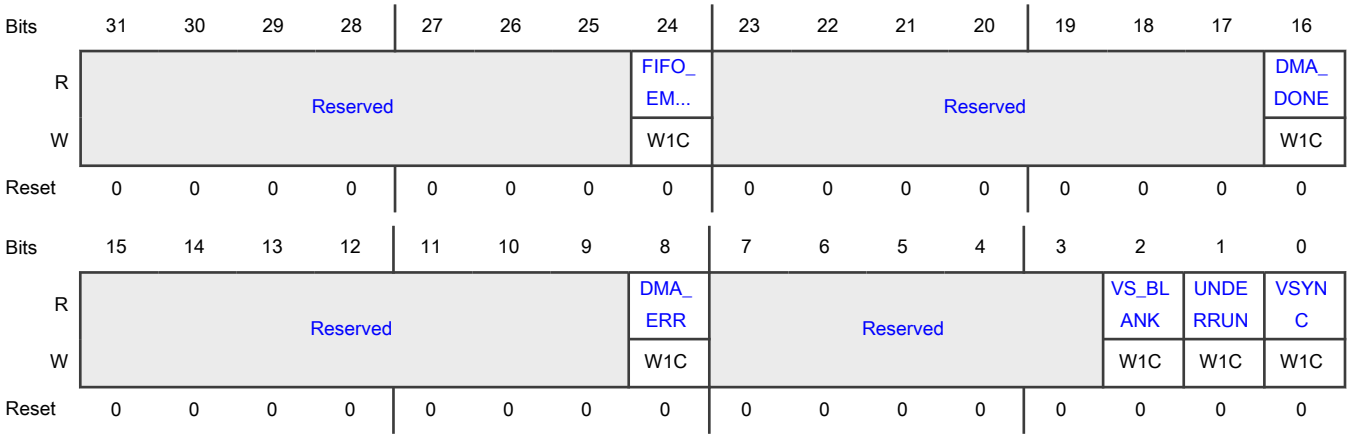
Field	Description
31-16 PW_V	VSYNC active pulse width (in horizontal line cycles). Pulse width has a minimum value of 1.
15-0 PW_H	HSYNC active pulse width (in pixel clock cycles). Pulse width has a minimum value of 1.

51.4.1.8 Interrupt Status Register for domain 0 (INT_STATUS_D0)

Offset

Register	Offset
INT_STATUS_D0	24h

Diagram



Fields

Field	Description
31-25 —	Reserved
24 FIFO_EMPTY	Interrupt flag to indicate that which FIFO in the pixel blending underflowed. Write a logic 1 to this field to clear this flag. • 0b: Did not occur • 1b: Occurred
23-17 —	Reserved
16 DMA_DONE	Interrupt flag to indicate that which PLANE has fetched the last pixel from memory. Write a logic 1 to this field to clear this flag. • 0b: Not finished • 1b: Finished
15-9 —	Reserved
8 DMA_ERR	Interrupt flag to indicate that which PLANE has Read Error on the AXI interface. Write a logic 1 to this field to clear this flag.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• 0b: Did not occur • 1b: Occurred
7-3 —	Reserved
2 VS_BLANK	Interrupt flag to indicate vertical blanking period. Write a logic 1 to this field to clear this flag. • 0b: Not detected • 1b: Detected
1 UNDERRUN	Interrupt flag to indicate the output buffer underrun condition. Write a logic 1 to this field to clear this flag. • 0b: Did not occur • 1b: Occurred
0 VSYNC	Interrupt flag to indicate that the vertical synchronization phase(The beginning of a frame). Write a logic 1 to this field to clear this flag. • 0b: Not detected • 1b: Detected

51.4.1.9 Interrupt Enable Register for domain 0 (INT_ENABLE_D0)

Offset

Register	Offset
INT_ENABLE_D0	28h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved							FIFO_EM...	Reserved							DMA_DON...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							DMA_ERR...	Reserved					VS_BLAN...	UNDERRU...	VSYNC_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

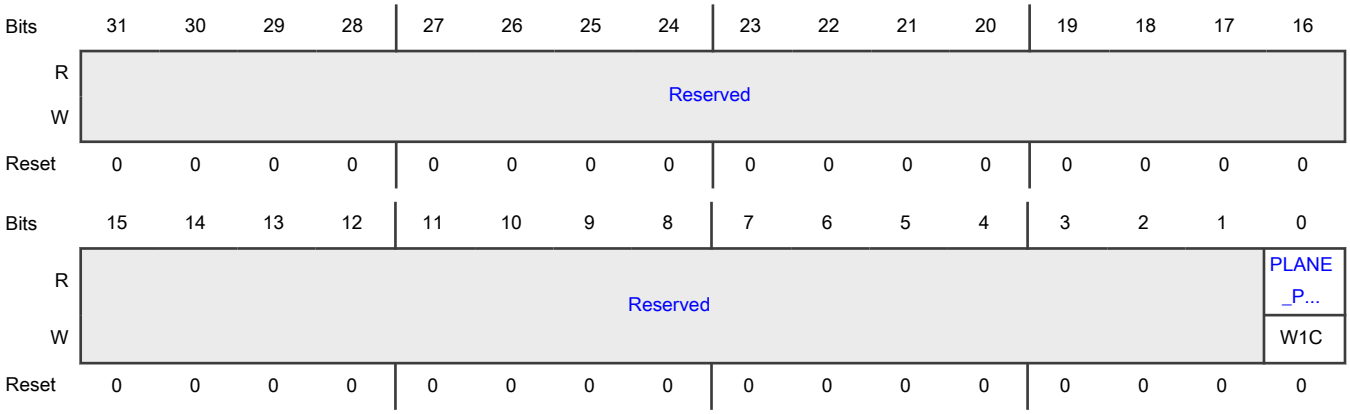
Field	Description
31-25 —	Reserved
24 FIFO_EMPTY_EN	Enable Interrupt flag to indicate that which FIFO in the pixel blending underflowed. • 0b: Disabled • 1b: Enabled
23-17 —	Reserved
16 DMA_DONE_EN	Enable Interrupt flag to indicate that which PLANE has fetched the last pixel from memory. • 0b: Disabled • 1b: Enabled
15-9 —	Reserved
8 DMA_ERR_EN	Enable Interrupt flag to indicate that which PLANE has Read Error on the AXI interface. • 0b: Disabled • 1b: Enabled
7-3 —	Reserved
2 VS_BLANK_EN	Enable Interrupt flag to indicate vertical blanking period. • 0b: Disabled • 1b: Enabled
1 UNDERRUN_EN	Enable Interrupt flag to indicate the output buffer underrun condition. • 0b: Disabled • 1b: Enabled
0 VSYNC_EN	Enable Interrupt flag to indicate that the vertical synchronization phase(The beginning of a frame). • 0b: Disabled • 1b: Enabled

51.4.1.10 Interrupt Status Register for domain 0 (INT_STATUS_D1)

Offset

Register	Offset
INT_STATUS_D1	30h

Diagram



Fields

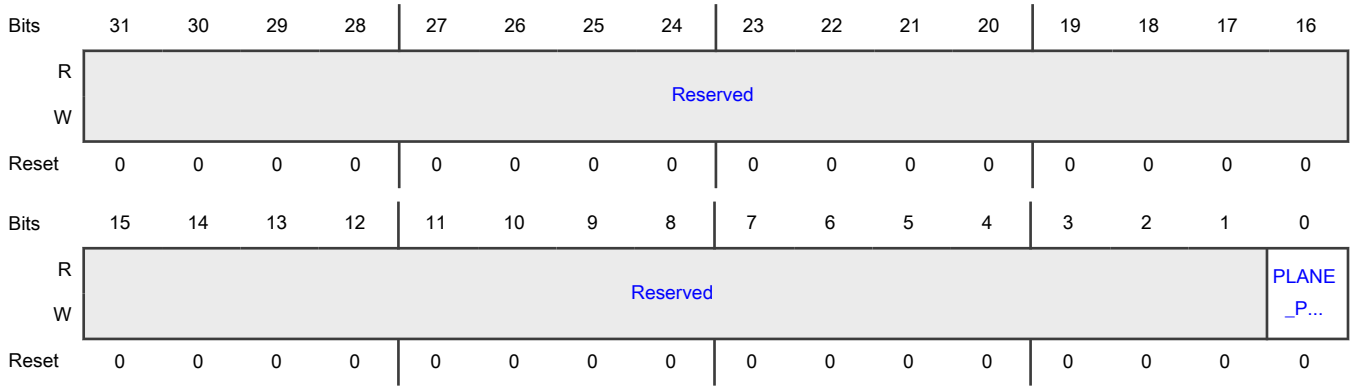
Field	Description
31-1 —	Reserved
0 PLANE_PANIC	Plane panic to indicate that which FIFO reaches the panic threshold. Write a logic 1 to this field to clear this flag. • 0b: Not detected • 1b: Detected

51.4.1.11 Interrupt Enable Register for domain 0 (INT_ENABLE_D1)

Offset

Register	Offset
INT_ENABLE_D1	34h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 PLANE_PANIC_EN	Enable Interrupt flag to indicate that which FIFO in the pixel blending underflowed. • 0b: Disabled • 1b: Enabled

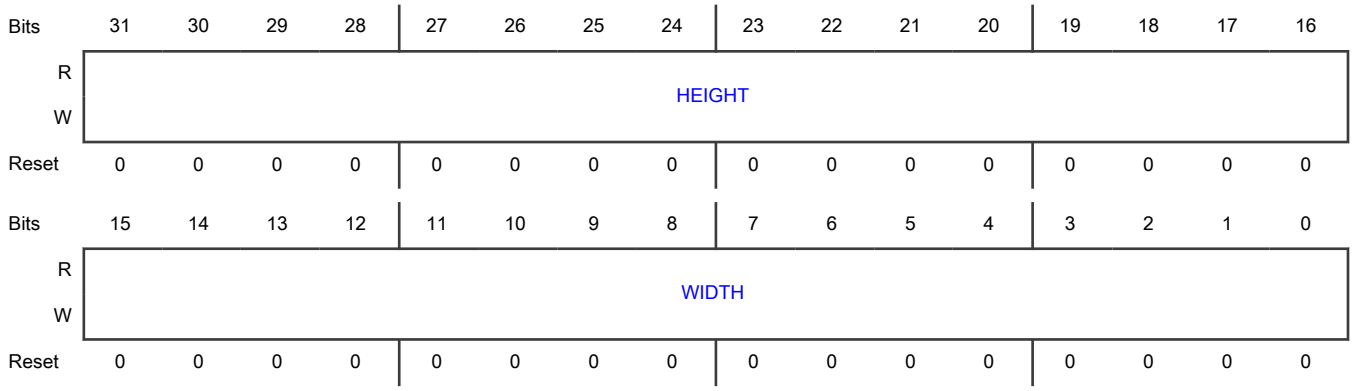
51.4.1.12 Control Descriptor Layer Register 1 (CTRLDESCL0_1)

This register sets the height and width of the layer associated with the register.

Offset

Register	Offset
CTRLDESCL0_1	200h

Diagram



Fields

Field	Description
31-16 HEIGHT	Height of the layer in pixels.
15-0 WIDTH	Width of the layer in pixels. The layer width must be in multiples of the number of pixels that can be stored in 32 bits, and therefore differs depending on color encoding. For example, if 16 bits per pixel format is used, then the layer width must be configured in multiples of 2.

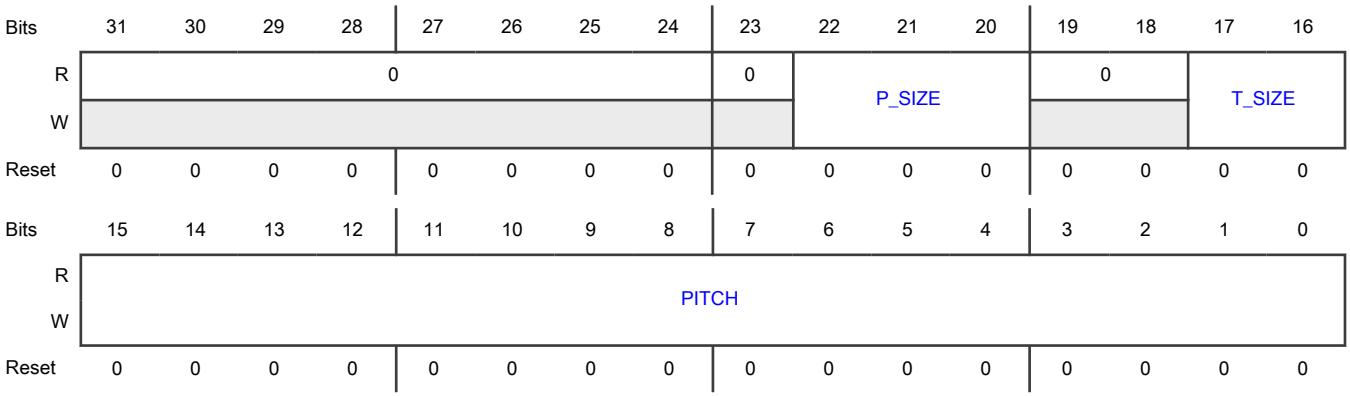
51.4.1.13 Control Descriptor Layer Register 3 (CTRLDESCL0_3)

This register sets the buffer line pitch of layer data.

Offset

Register	Offset
CTRLDESCL0_3	208h

Diagram



Fields

Field	Description
31-24 —	Reserved
23 —	Reserved
22-20 P_SIZE	Payload size.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Holds the payload size. The P_SIZE should never be programmed to be less than T_SIZE. The recommended configuration value is 010b. • 000b - 64 bytes • 001b - 128 bytes • 010b - 256 bytes • 011b - 512 bytes • 100b - Reserved • 101b - Reserved • 110b - Reserved • 111b - Reserved
19-18 —	Reserved
17-16 T_SIZE	Transaction Size Holds the transaction size. This T_SIZE should always be programmed in conjunction with P_SIZE where P_SIZE >= T_SIZE. The recommended configuration value is 10b. • 00b - 64 bytes • 01b - 128 bytes • 10b - 256 bytes • 11b - Reserved
15-0 PITCH	Number of bytes between 2 vertically adjacent pixels in system memory. Byte granularity is supported, but SW should align to 64B boundary.

51.4.1.14 Control Descriptor Layer Register 4 (CTRLDESCL_LOW0_4)

This register sets the low 32bits of the beginning address of layer data.

Offset

Register	Offset
CTRLDESCL_LOW0_4	20Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ADDR_LOW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADDR_LOW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ADDR_LOW	Address of layer data in the memory. The address programmed should be 128-bit aligned.

51.4.1.15 Control Descriptor Layer Register 4 (CTRLDESCL_HIGH0_4)

This register sets the high 4bits of the beginning address of layer data.

Offset

Register	Offset
CTRLDESCL_HIGH0_4	210h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												ADDR_HIGH			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved
3-0 ADDR_HIGH	Address of layer data in the memory.

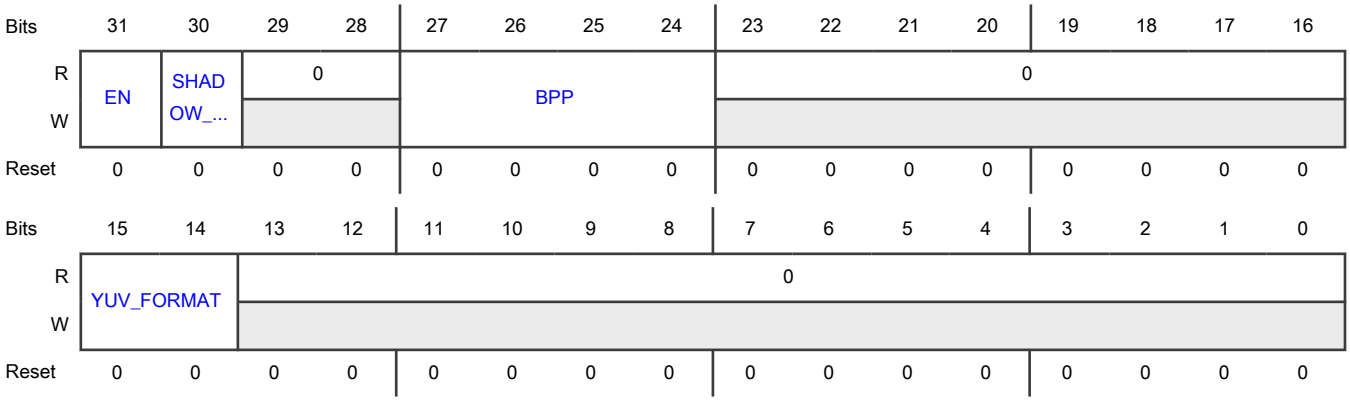
51.4.1.16 Control Descriptor Layer Register 5 (CTRLDESCL0_5)

This register controls various graphics options and whether the layer is enabled.

Offset

Register	Offset
CTRLDESCL0_5	214h

Diagram



Fields

Field	Description
31 EN	Enable the layer for DMA. 0 - OFF 1 - ON
30 SHADOW_LOAD_EN	Shadow Load Enable The SHADOW_LOAD_EN bit is written to 1 by software after all DMA control registers are written. If set to 1, shadowed control registers are updated to the active control registers on VSYNC of next frame. If set to 0, shadowed control registers are not loaded into the active control registers. The previous active control register settings will be used to process the next frame. Hardware will automatically clear this bit, when the shadow registers are loaded to the active control regsisters.

Table continues on the next page...

Table continued from the previous page...

Field	Description
29-28 —	Reserved
27-24 BPP	Layer encoding format (bit per pixel) 0100 - 16 bpp (RGB565) 0101 - 16 bpp (ARGB1555) 0110 - 16 bpp (ARGB4444) 0111 - YCbCr422 1000 - 24 bpp (RGB888) 1001 - 32 bpp (ARGB8888) 1010 - 32 bpp (ABGR8888)
23-16 —	Reserved
15-14 YUV_FORMAT	The YUV422 input format selection. 00 - The YUV422 32bit memory is {Y2,V1,Y1,U1} 01 - The YUV422 32bit memory is {Y2,U1,Y1,V1} 10 - The YUV422 32bit memory is {V1,Y2,U1,Y1} 11 - The YUV422 32bit memory is {U1,Y2,V1,Y1}
13-0 —	Reserved

51.4.1.17 Color Space Conversion Ctrl Register (CSC0_CTRL)

The CSC control register will configure the CSC module to perform color space converison between the RGB/YUV/YCbCr color spaces

// Converting from YUV/YCbCr color spaces to the RGB color space uses the

// R = A1(Y-D1) + A2(U-D2) + A3(V-D3)

// G = B1(Y-D1) + B2(U-D2) + B3(V-D3)

// B = C1(Y-D1) + C2(U-D2) + C3(V-D3)

//Converting from the RGB color space to YCbCr color spaces uses the

//following equation structure:

// Y = A1*R + A2*G + A3*B + D1

// U = B1*R + B2*G + B3*B + D2

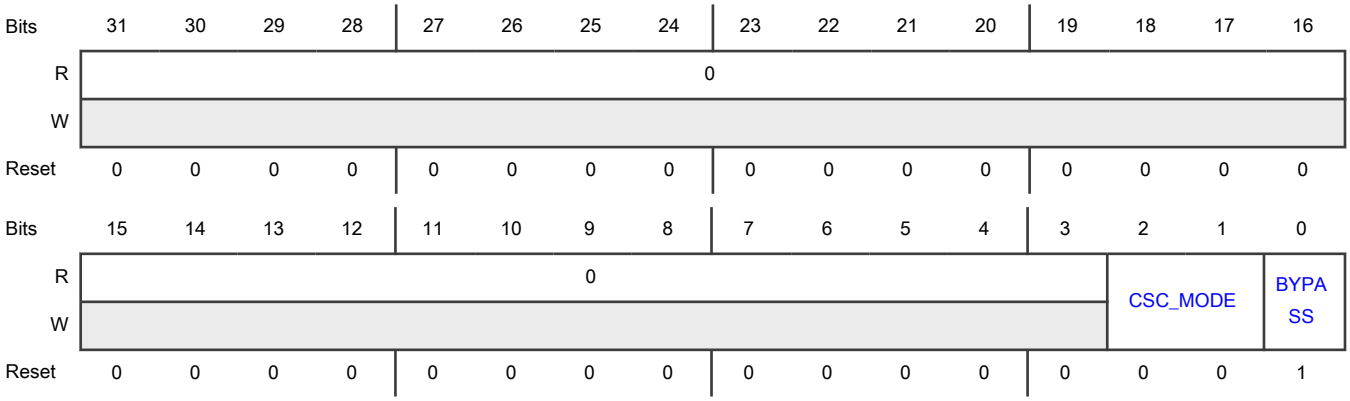
// V = C1*R + C2*G + C3*B + D3

//All math is signed, so all coefficients come in as two's comp numbers

Offset

Register	Offset
CSC0_CTRL	21Ch

Diagram



Fields

Field	Description
31-3 —	Reserved
2-1 CSC_MODE	This field controls how the CSC unit operates on pixels when the CSC is not bypassed. 00 - Convert from YUV to RGB. 01 - Convert from YCbCr to RGB. 10 - Reserved 11 - Convert from RGB to YCbCr.
0 BYPASS	This bit controls whether the pixels entering the CSC2 unit get converted or not. When BYPASS is set, no operations occur on the pixels. When BYPASS is cleared, the selected CSC operation takes place.

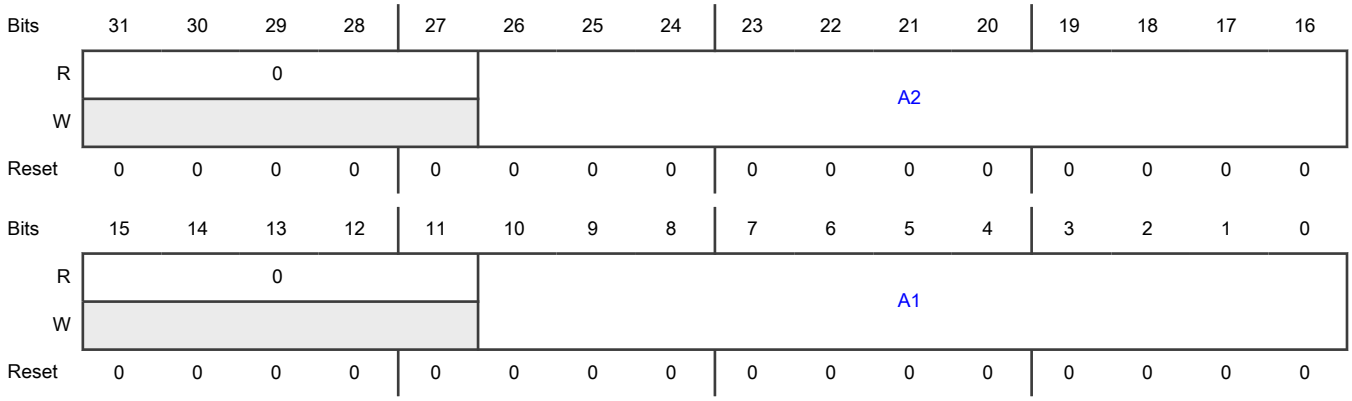
51.4.1.18 Color Space Conversion Coefficient Register 0 (CSC0_COEF0)

This register contains color space conversion coefficients in two's complement notation.

Offset

Register	Offset
CSC0_COEF0	220h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 A2	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 A1	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

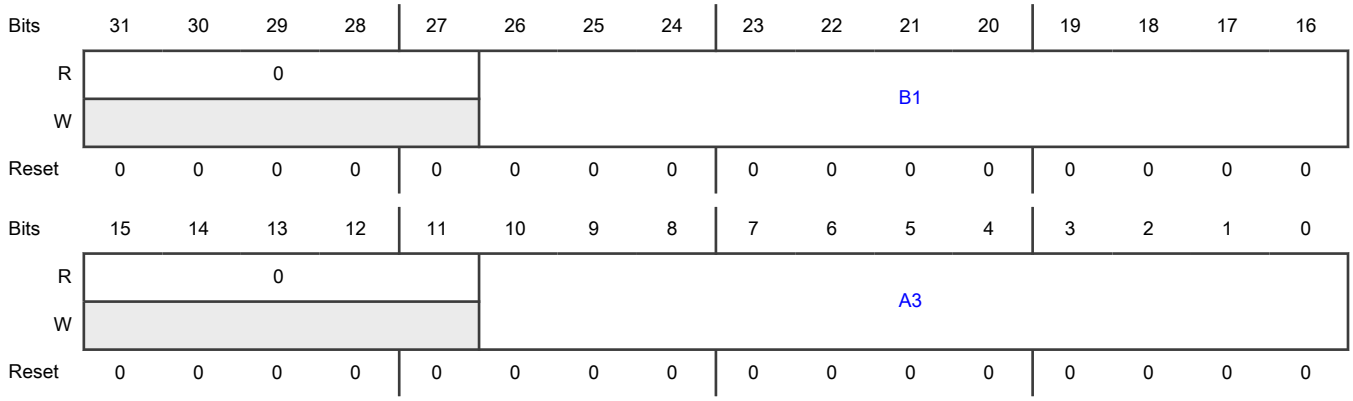
51.4.1.19 Color Space Conversion Coefficient Register 1 (CSC0_COEF1)

This register contains color space conversion coefficients in two's complement notation.

Offset

Register	Offset
CSC0_COEF1	224h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 B1	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 A3	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

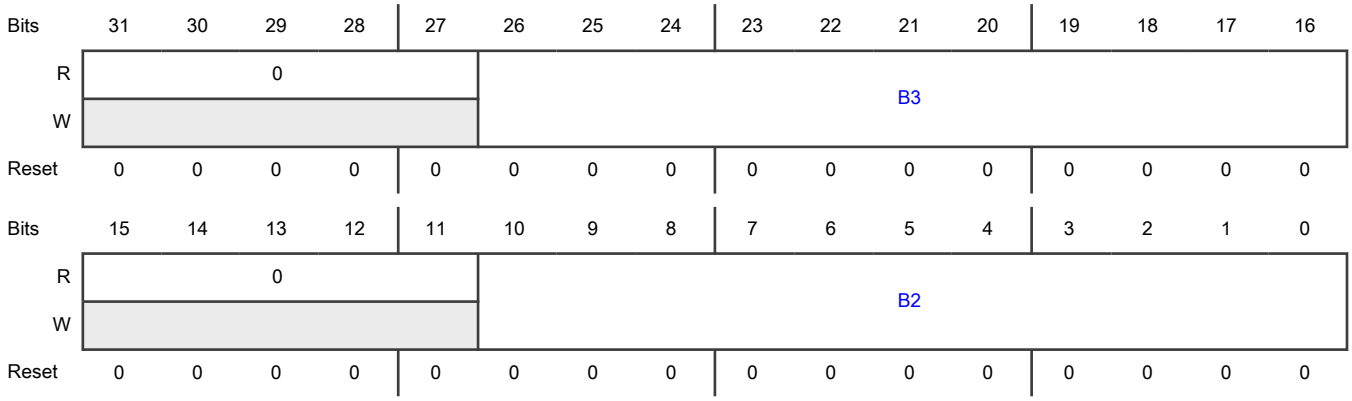
51.4.1.20 Color Space Conversion Coefficient Register 2 (CSC0_COEF2)

This register contains color space conversion coefficients in two's complement notation.

Offset

Register	Offset
CSC0_COEF2	228h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 B3	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 B2	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

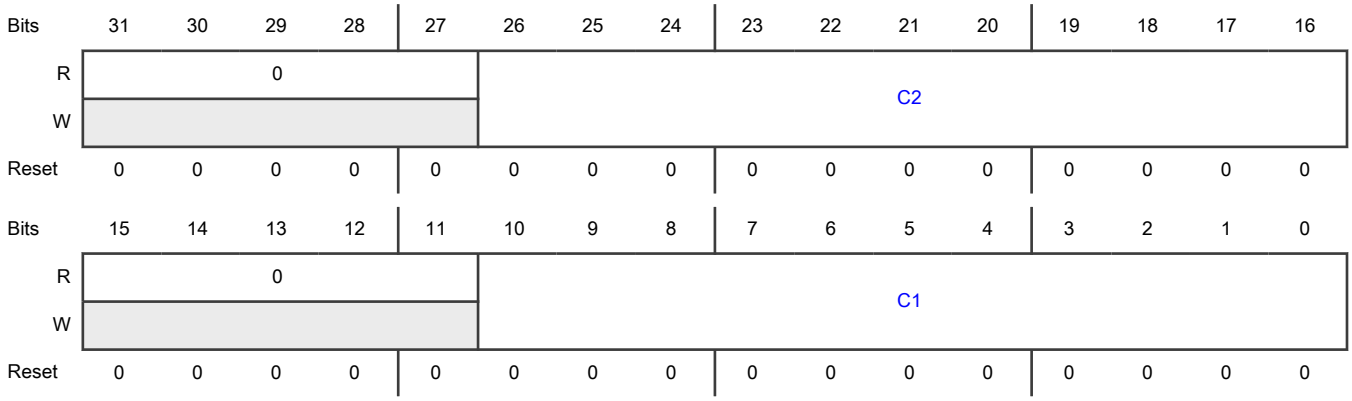
51.4.1.21 Color Space Conversion Coefficient Register 3 (CSC0_COEF3)

This register contains color space conversion coefficients in two's complement notation.

Offset

Register	Offset
CSC0_COEF3	22Ch

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 C2	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 C1	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

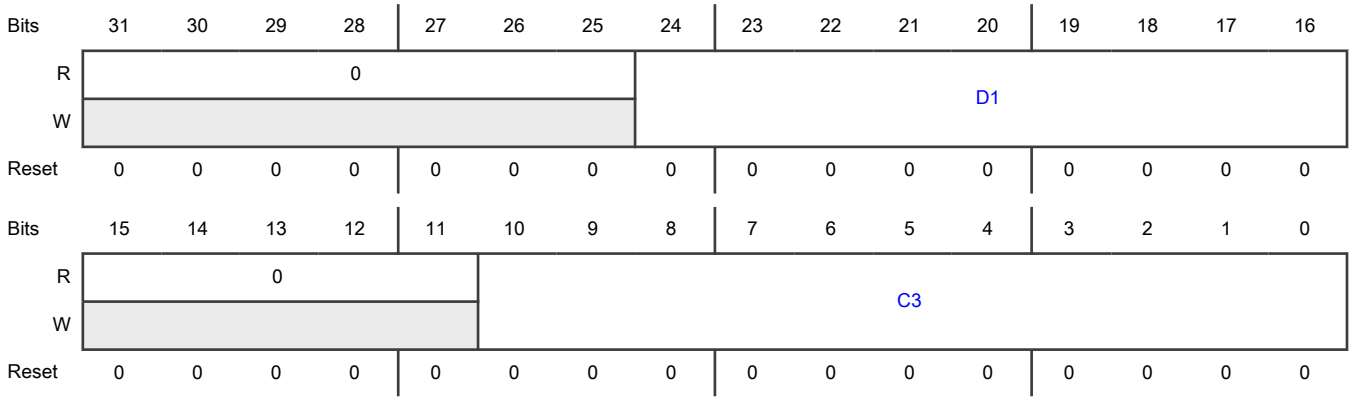
51.4.1.22 Color Space Conversion Coefficient Register 4 (CSC0_COEF4)

This register contains color space conversion coefficients in two's complement notation.

Offset

Register	Offset
CSC0_COEF4	230h

Diagram



Fields

Field	Description
31-25 —	Reserved
24-16 D1	Two's complement D1 coefficient integer offset to be added.
15-11 —	Reserved
10-0 C3	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

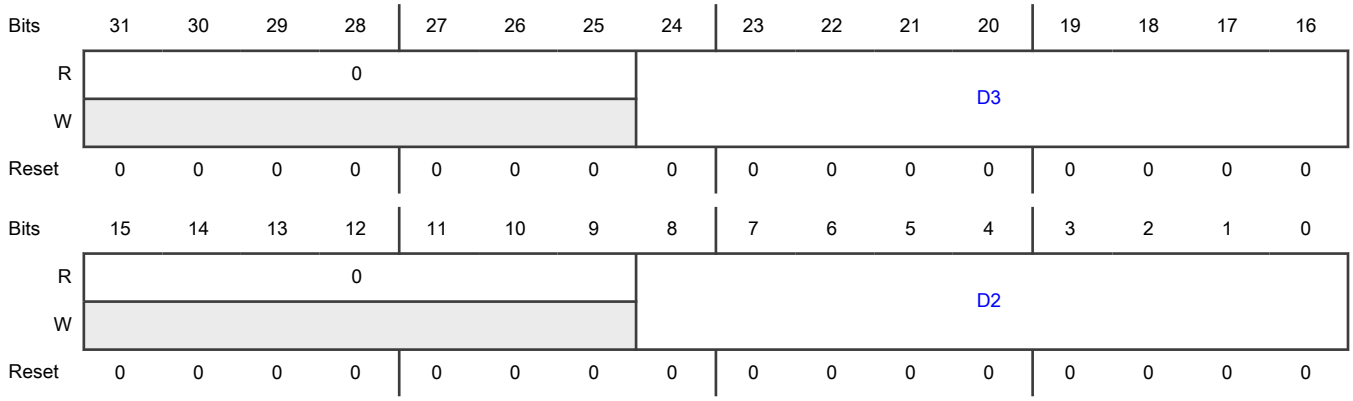
51.4.1.23 Color Space Conversion Coefficient Register 0 (CSC0_COEF5)

This register contains color space conversion coefficients in two's complement notation.

Offset

Register	Offset
CSC0_COEF5	234h

Diagram



Fields

Field	Description
31-25 —	Reserved
24-16 D3	Two's complement D3 coefficient integer offset to be added.
15-9 —	Reserved
8-0 D2	Two's complement D2 coefficient integer offset to be added.

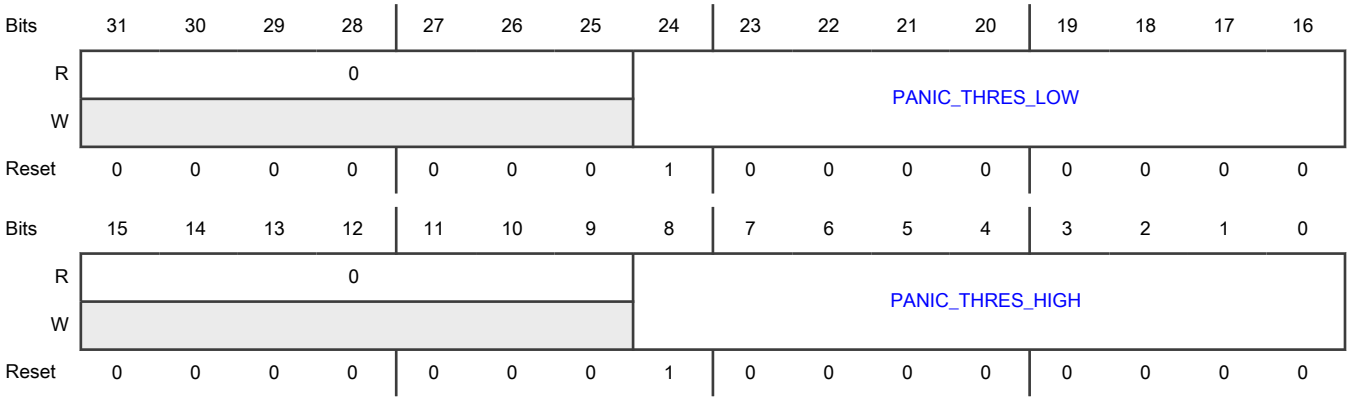
51.4.1.24 Memory request priority threshold register (PANIC0_THRES)

This register is used to activate panic signal when the number of pixels reaches the programmed threshold. It can be used to manipulate access priority to meet the required pixel throughput

Offset

Register	Offset
PANIC0_THRES	238h

Diagram



Fields

Field	Description
31-25 —	Reserved
24-16 PANIC_THRES_LOW	Panic Threshold Low Value This value should be set to a value of pixels, from (0+1)*128bits to (511+1)*128bits. When the number of pixels in the input pixel FIFO is less than this value, the panic control output will be raised.This signal can be used to raise the access LCDIF's access priority. • 0: threshold is 1*128bits • 1: threshold is 2*128bits • . • . • . • 511:threshold is 512*128bits
15-9 —	Reserved
8-0 PANIC_THRES_HIGH	Panic Threshold High Value This value should be set to a value of pixels, from (0+1)*128bits to (511+1)*128bits. When the number of pixels in the input pixel FIFO is more than this value, the panic control output will be removed.This signal can be used to raise the access LCDIF's access priority. • 0: threshold is 1*128bits • 1: threshold is 2*128bits • . • . • . • 511:threshold is 512*128bits

Chapter 52
MIPI-DSI

52.1 Overview

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

MIPI DSI is a digital controller that provides an interface between the system and the MIPI PHY, allowing the communication with a DSI-compliant display.

52.1.1 Block Diagram

The block diagram for the MIPI DSI controller is shown below.

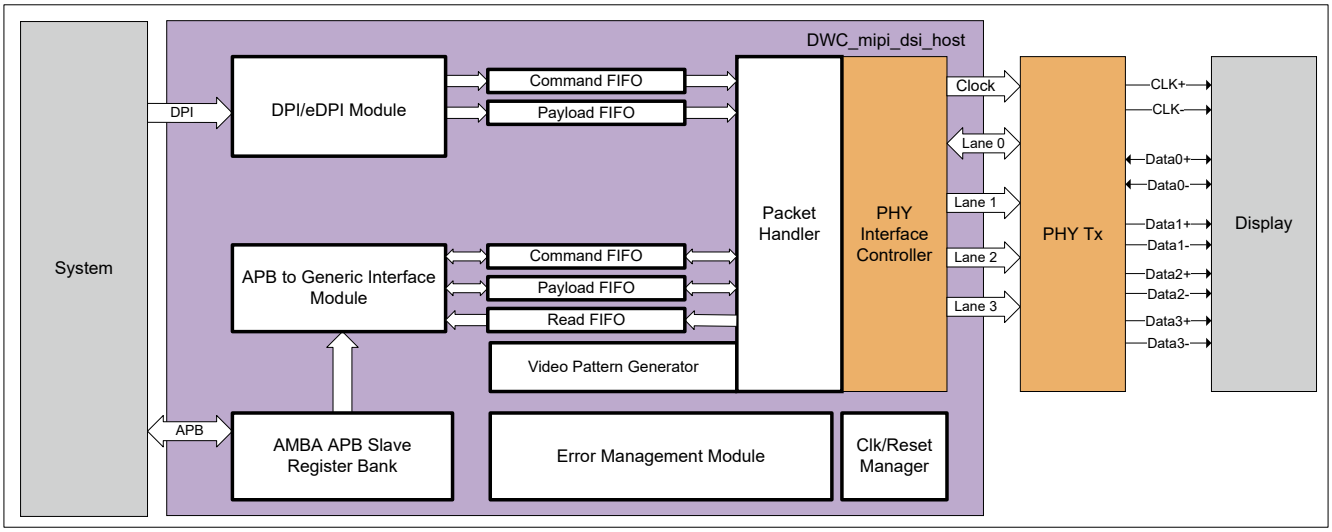


Figure 248. MIPI DSI Controller

The main blocks of the MIPI DSI controller are as follows:

- **DPI/eDPI Module**: Captures the data and control signals and conveys them to two dedicated FIFOs, one for the video control signals, and the other for the pixel data.

  In the Video mode, the controller uses this data to build the Video packets. For more information, see DPI.

- **Register Bank**: Use the standard AMBA-APB slave interface to access the Register Bank. Program the MIPI DSI registers through the AMBA-APB slave interface to configure and control the behavior of the MIPI DSI controller. The MIPI DSI controller provides a fully programmable interrupt generator to manage certain specific events.

- **PHY Interface Controller**: Manages PPI, by acknowledging the current operation and enables the low-power transmission or reception, or high-speed transmission. It also performs data splitting between the available PHY lanes for high-speed transmission.

- **Packet Handler**: Based on the controller interface configuration, and Video transmission mode (Burst or Non-Burst, with sync pulses or sync events), this block generates packets with ECC and CRC codes. This block also performs the following functions:

  — Packet reception

  — Packet header validation by checking the ECC

  — Packet header correction and notification for single-bit errors

- Determine termination of reception
- Error management (detection and notification of errors, for example, multiple header error)
- Packet routing mechanism, for example, this module can route the output data to the Generic interface depending on the packet's virtual channel.

- **APB-to-Generic Interface Module**: Bridges APB operations and FIFOs holding the Generic commands, that is, the command FIFO, write-payload FIFO, and the read-payload FIFO.
- **Error Management Module**: Monitors the system error conditions, and triggers an interrupt to prompt an error notification.

52.1.2 Features

The MIPI DSI controller supports the following features:

- Configurable selection of system interfaces:
  - AMBA APB for control and optional support for Generic and DCS commands
  - Display Pixel Interface (DPI) for Video mode interface (optional)
  - Independently programmable virtual channel ID in the configured system data interface, and APB Slave (optional)
- Fault recovery mechanisms
- Ultra Low-Power mode with PLL disabled
- DPI features:
  - DPI signals programmable polarity
  - Extended resolutions beyond the DPI standard maximum resolution of 800x480 pixels
  - Maximum resolution limited by available DSI physical link bandwidth, which is determined by the number of physical lanes and the maximum speed that they can achieve
- Video pattern generator:
  - Vertical and horizontal color bar generation without DPI stimuli
  - BER pattern without DPI stimuli

52.2 Functional Description

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

52.2.1 Resolution and Frame Rates

The maximum resolution supported for different conditions is detailed in the following table.

Table 497. Video Modes

Display Type							DSI Controller				
Resolution (pixels)	Horizontal		Vertical		Refresh Rate (Hz)	Color Depth (bpp)	Configured System Interface[1]		Physical Interface		
	Active	Total	Active	Total			DSI Bandwidth (Mbps)	DPI Clock (MHz)	Type	Number of Lanes/ Trios[2]	Speed per Lane/ Trios[3]
1920x1080 RB	1920	2080	1080	1111	60	24	3328	138.7	D-PHY	4	1

Table continues on the next page...

Table 497. Video Modes (continued)

Display Type							DSI Controller				
Resolution (pixels)	Horizontal		Vertical		Refresh Rate (Hz)	Color Depth (bpp)	Configured System Interface[1]		Physical Interface		
	Active	Total	Active	Total			DSI Bandwidth (Mbps)	DPI Clock (MHz)	Type	Number of Lanes/ Trios[2]	Speed per Lane/ Trios[3]
1920x1080	1920	2576	1080	1120	60	24	4155	173.1	D-PHY	N/S [4]	1
1920x1200	1920	2592	1200	1245	60	24	4647	193.6	D-PHY	N/S [4]	1
1920x1200 RB	1920	2080	1200	1235	60	24	3699	154.1	D-PHY	4	1
2048X1080 (2K)	2048	2200	1080	1125	60	24	3564	148.5	D-PHY	4	1

1. DPI
2. Number of Lanes for D-PHY
3. For D-PHY, Speed per Lane is in Gbps
4. Not sufficient for DPHY/CPHY bandwidth to support the video mode.

52.2.2 DPI

52.2.2.1 Overview of DPI

The DPI follows the *MIPI DPI-2 Standard* with pixel data bus width up to 24 bits. In Video mode, the controller transmits the information to the peripheral in the form of a real-time pixel stream. This interface allows sending the ShutDown (SD) and ColorMode (CM) commands, which directly trigger the interface pins.

52.2.2.2 Enabling DPI

The `edpihalt` pin of the DPI is an additional pin which is not included in the *MIPI DPI-2 Standard*. The `edpihalt` pin helps the video driver to control the data input rate, when the controller is executing an instruction and cannot accept additional pixel data. For more information about the behavior of the `edpihalt` pin, see Transmission of Commands.

The DPI captures the data and control signals and conveys them to the FIFO interfaces, which subsequently transmit these signals to the DSI link. Two different data streams presented at the interface are - video control signals and pixel data. Depending on the interface color coding, the controller disposes the pixel data differently throughout the `dpipixdata` bus. Figure 249 illustrates the interface pixel color coding.

It is possible to configure the DPI to increase flexibility, and promote correct usage of this interface for several systems. These configuration options are as follows:

- Depending on the system requirements, it is possible to program the control signals to change its polarity.
- After reset, DPI waits for the first VSYNC active transition, to start the signal sampling, including the pixel data. This avoids starting the image data transmission in the middle of a frame.
- Configure the number of pixels in the `vid_pkt_size` register field as a multiple of four, if pixel color coding is 18 bits and the 18-bit loosely packed stream is disabled. This means that, in this mode, the two LSBs in the configuration are always inferred as zero. Additionally, the specification states that, the pixel line size must be a multiple of four.
- To avoid FIFO underflows and overflows, assume that the controller receives the configured number of pixels at all times. This happens even if the `dpidataen` pin is active for more or less time than necessary. A set of bits in the `VID_PKT_STATUS` register report the status of the FIFOs and internal buffers associated with DPI.

- Separate the memory space of different video packets using the number of pixels per packet parameter, to keep the memory organized with respect to the packet scheduling.

For the sampling and transmission of `dpishutdn` and `dpicolorm` signals, the DPI video signaling must be active. Thus, if the commands do not actively generate the video signals like `VSYNC` and `HSYNC`, then these signals are not transmitted through the DSI link. Because of these constraints and to ensure a flawless transmission of commands, the first `VSYNC` active pulse must occur for the command sampling and transmission. Immediately after issuing the command to shutdown the display, DPI video must be kept active for one frame, thus ensuring command transmission.

The *MIPI DPI-2 Standard* does not define a standard way to get certain data formats through the pixel interface. To do so, the bit width of the data bus in DPI-2 has been extended from 24-bit to 30-bit and the location of the color components are defined according to Table 498.

All the uncompressed data types comply with a rate of one pixel per clock cycle, except the 36-bit format and 12-bit YCbCr 4:2:0 format. The exceptions are as follows:

- For 36-bit format, by using the same clock it is not possible to continuously receive one entire pixel per clock cycle, because this data is stored in a 32-bit RAM. Therefore, each clock cycle receives half a pixel, making it {R[11:0], G[5:0]} in one cycle, and the remaining {G[11:6], B[11:0]} of the same pixel in the next one.
- For 12-bit YCbCr 4:2:0, even pixels take 16 bits of information and odd pixels take 8 bits of information. Therefore, it is possible to fit two pixels in 24-bit, transferred in a single clock cycle.

The number of pixels per line must be even for all the YCbCr data types, to meet the number of bytes per packet restrictions, defined in the *MIPI DSI Specification*.

The 30-bit and 36-bit data types are generic and intended to accommodate the data also for non-RGB color spaces in the future.

	20-Bit YCbCr 4:2:2 Loosely Packed		24-Bit YCbCr 4:2:2		16-Bit YCbCr 4:2:2		30-Bit	36-Bit		12-Bit YCbCr 4:2:0		16-Bit Config 1	16-Bit Config 2	16-Bit Config 3	18-Bit Config 1	18-Bit Config 2	24-Bits	DSC24
ORDER	1º Cycle	2º Cycle	1º Cycle	2º Cycle	1º Cycle	2º Cycle		1º Cycle	2º Cycle	Odd line	Even line							
D29							R[9]											
D28							R[8]											
D27							R[7]											
D26							R[6]											
D25							R[5]											
D24							R[4]											
D23	Y1[9]	Y2[9]	Y1[11]	Y2[11]	Y1[7]	Y2[7]	R[3]			Y1[7]	Y1[7]						R[7]	Byte1[7]
D22	Y1[8]	Y2[8]	Y1[10]	Y2[10]	Y1[6]	Y2[6]	R[2]			Y1[6]	Y1[6]						R[6]	Byte1[6]
D21	Y1[7]	Y2[7]	Y1[9]	Y2[9]	Y1[5]	Y2[5]	R[1]			Y1[5]	Y1[5]			R[4]		R[5]	R[5]	Byte1[5]
D20	Y1[6]	Y2[6]	Y1[8]	Y2[8]	Y1[4]	Y2[4]	R[0]			Y1[4]	Y1[4]		R[4]	R[3]		R[4]	R[4]	Byte1[4]
D19	Y1[5]	Y2[5]	Y1[7]	Y2[7]	Y1[3]	Y2[3]	G[9]			Y1[3]	Y1[3]		R[3]	R[2]		R[3]	R[3]	Byte1[3]
D18	Y1[4]	Y2[4]	Y1[6]	Y2[6]	Y1[2]	Y2[2]	G[8]			Y1[2]	Y1[2]		R[2]	R[1]		R[2]	R[2]	Byte1[2]
D17	Y1[3]	Y2[3]	Y1[5]	Y2[5]	Y1[1]	Y2[1]	G[7]	R[11]	G[11]	Y1[1]	Y1[1]		R[1]	R[0]	R[5]	R[1]	R[1]	Byte1[1]
D16	Y1[2]	Y2[2]	Y1[4]	Y2[4]	Y1[0]	Y2[0]	G[6]	R[10]	G[10]	Y1[0]	Y1[0]		R[0]		R[4]	R[0]	R[0]	Byte1[0]
D15	Y1[1]	Y2[1]	Y1[3]	Y2[3]			G[5]	R[9]	G[9]	Y0[7]	Y0[7]	R[4]			R[3]		G[7]	Byte2[7]
D14	Y1[0]	Y2[0]	Y1[2]	Y2[2]			G[4]	R[8]	G[8]	Y0[6]	Y0[6]	R[3]			R[2]		G[6]	Byte2[6]
D13			Y1[1]	Y2[1]			G[3]	R[7]	G[7]	Y0[5]	Y0[5]	R[2]	G[5]	G[5]	R[1]	G[5]	G[5]	Byte2[5]
D12			Y1[0]	Y2[0]			G[2]	R[6]	G[6]	Y0[4]	Y0[4]	R[1]	G[4]	G[4]	R[0]	G[4]	G[4]	Byte2[4]
D11	Cb1[9]	Cr2[9]	Cb1[11]	Cr2[11]	Cb1[7]	Cr2[7]	G[1]	R[5]	B[11]	Y0[3]	Y0[3]	R[0]	G[3]	G[3]	G[5]	G[3]	G[3]	Byte2[3]
D10	Cb1[8]	Cr2[8]	Cb1[10]	Cr2[10]	Cb1[6]	Cr2[6]	G[0]	R[4]	B[10]	Y0[2]	Y0[2]	G[5]	G[2]	G[2]	G[4]	G[2]	G[2]	Byte2[2]
D9	Cb1[7]	Cr2[7]	Cb1[9]	Cr2[9]	Cb1[5]	Cr2[5]	B[9]	R[3]	B[9]	Y0[1]	Y0[1]	G[4]	G[1]	G[1]	G[3]	G[1]	G[1]	Byte2[1]
D8	Cb1[6]	Cr2[6]	Cb1[8]	Cr2[8]	Cb1[4]	Cr2[4]	B[8]	R[2]	B[8]	Y0[0]	Y0[0]	G[3]	G[0]	G[0]	G[2]	G[0]	G[0]	Byte2[0]
D7	Cb1[5]	Cr2[5]	Cb1[7]	Cr2[7]	Cb1[3]	Cr2[3]	B[7]	R[1]	B[7]	Cb[7]	Cr[7]	G[2]			G[1]		B[7]	Byte3[7]
D6	Cb1[4]	Cr2[4]	Cb1[6]	Cr2[6]	Cb1[2]	Cr2[2]	B[6]	R[0]	B[6]	Cb[6]	Cr[6]	G[1]			G[0]		B[6]	Byte3[6]
D5	Cb1[3]	Cr2[3]	Cb1[5]	Cr2[5]	Cb1[1]	Cr2[1]	B[5]	G[5]	B[5]	Cb[5]	Cr[5]	G[0]		B[4]	B[5]	B[5]	B[5]	Byte3[5]
D4	Cb1[2]	Cr2[2]	Cb1[4]	Cr2[4]	Cb1[0]	Cr2[0]	B[4]	G[4]	B[4]	Cb[4]	Cr[4]	B[4]	B[4]	B[3]	B[4]	B[4]	B[4]	Byte3[4]
D3	Cb1[1]	Cr2[1]	Cb1[3]	Cr2[3]			B[3]	G[3]	B[3]	Cb[3]	Cr[3]	B[3]	B[3]	B[2]	B[3]	B[3]	B[3]	Byte3[3]
D2	Cb1[0]	Cr2[0]	Cb1[2]	Cr2[2]			B[2]	G[2]	B[2]	Cb[2]	Cr[2]	B[2]	B[2]	B[1]	B[2]	B[2]	B[2]	Byte3[2]
D1			Cb1[1]	Cr2[1]			B[1]	G[1]	B[1]	Cb[1]	Cr[1]	B[1]	B[1]	B[0]	B[1]	B[1]	B[1]	Byte3[1]
D0			Cb1[0]	Cr2[0]			B[0]	G[0]	B[0]	Cb[0]	Cr[0]	B[0]	B[0]		B[0]	B[0]	B[0]	Byte3[0]

Figure 249. Location of Color Components in an Extension of DPI-2 Pixel Interface

According to Table 498, the number of payload pixels must be restricted to a multiple of a value. For 30-bit and 36-bit formats, the boundaries of the pixels may not always fill all the bits of the last byte of the payload. In that case, the controller automatically fills the remaining bits with 0.

Table 498. Multiplicity of the Payload Size in Pixels for Each Data Type

Value	Data Types
1	• 16-bit • 18-bit loosely packed • 24-bit • 30-bit • 36-bit • DSC24 compressed data
2	• Loosely Packed Pixel Stream, 20-bit YCbCr 4:2:2 • 24-bit YCbCr 4:2:2 • 16-bit YCbCr 4:2:2 • 12-bit YCbCr 4:2:0
4	• 18-bit non-loosely packed

52.2.2.3 Video Mode with Stereoscopic Image Data

Figure 250 illustrates the bytes in a short packet.

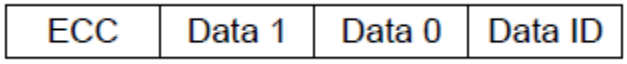


Figure 250. Bytes in a Short Packet

On configuring the SDF_3D register, it is possible to sent the next VSS packet, if necessary, as listed in Table 499. The controller sends this packet, with data 0 byte set to 0x08, as bit 3 indicates the presence of 3D control payload in data 1 byte (the other bits of data 0 are reserved). The data 1 includes a copy of SDF_3D[5:0], the 3D control configuration.

Table 499. VSS with 3D Information

3D was Active (SDF_3D[1:0] was 2'b10 or 2'b01)	3D Activated (SDF_3D[1:0] switched to 2'b10 or 2'b01)	SDF_3D Updated	send_3d_cfg (SDF_3D [16])	Next VSS with 3D Information
*	*	0	0	False
False	False	1	0	False
False	True	1	0	True
True	False	1	0	True
True *	True *	1 *	0 1	True True

The 3D control also indicates that the 2D mode is on. When switching from a 3D mode to 2D mode (`SDF_3D[1:0]` register field set to 2'b00), the next VSS packet contains data 0 set to 0x08 and data 1 has the two LSB set to '00' to indicate that the 3D and the 2D mode are off.

Setting the `send_3d_cfg` field of the `SDF_3D` register, sends the 3D control configuration information in every VSS packet. If not, the controller sends the VSS with 3D information only in the next frame.

On updating the `SDF_3D` register with 3D disabled and without activating the 3D information, then the next VSS does not have the 3D information. This behavior ensures compatibility with devices that do not understand this packet.

52.2.2.4 Video Transmission Modes

The different video transmission modes are as follows:

- Burst mode
- Non-Burst mode
  - Non-Burst mode with sync pulse
  - Non-Burst mode with sync event

52.2.2.4.1 Burst Mode

In this mode, the controller buffers the entire active pixel line into a FIFO and transmits it in a single packet with no interruptions. This transmission mode requires the DPI Pixel FIFO to have the capacity to store a full line of active pixel data. For optimal performance, there must be a relatively big difference between the pixel required bandwidth and the DSI link bandwidth. This allows the MIPI DSI controller to quickly dispatch an entire active video line in a single data burst and then return to low-power mode.

52.2.2.4.2 Non-Burst Mode

In this mode, the processor uses the partitioning properties of the MIPI DSI controller to divide the video line transmission into several DSI packets, so that the pixel required bandwidth matches with the DSI link bandwidth. By doing so, the controller does not require a full line of pixel data to be stored inside the DPI Pixel FIFO, but instead it requires only one video packet content to be stored.

52.2.2.4.3 Guidelines for Selecting the Burst or Non-Burst Mode

Selecting the Burst or Non-Burst mode depends on the system configuration and the device requirements. Choose the video transmission mode that suits the application scenario. The Burst mode is beneficial because it increases the probability of the link spending more time in the low-power mode, thereby decreasing power consumption. However, the following conditions must be met for availing the maximum benefits from the Burst mode of operation:

- The MIPI DSI controller must have sufficient memory to store a pixel line without causing FIFOs overflow.
- The display device must be able to receive a pixel line in a single burst without causing a reception buffer overflow.
- The DSI output bandwidth must be higher than the DPI system interface input bandwidth, in such a way, that enables the link to go to low-power once per line.

In Burst mode, if the system does not meet the above requirements, then it can result in the display device malfunctioning due to pixel data loss. This error possibility depends on the system capability to store temporary pixel data.

If all the conditions for using the Burst mode cannot be met, use the Non-Burst mode to avoid the errors caused by the Burst mode. The Non-Burst mode provides a better matching of rates for pixel transmission, enabling:

- Only a certain amount of pixels to be stored in the memory and not an entire pixel line (lesser DPI RAM requirements in the MIPI DSI controller)
- Operation with devices that support only a small amount of pixel buffering (less than a pixel line)

The DSI Non-Burst mode must be configured in such way that the DSI's output pixel ratio matches the DPI's input pixel ratio. Thereby, reducing the memory requirements on both host and device sides, or on either host or device side. Dividing a pixel line into several chunks of pixels and optionally interleaving them with null packets helps in achieving this.

In the Non-Burst mode, to match the DSI's output pixel ratio (left hand side of the "=" sign) and DPI's input pixel ratio (right hand side of the "=" sign), program the transmission parameters of the MIPI DSI controller as follows:

- On enabling the null packets, program the transmission parameters as follows:

```
(lanebyteclkperiod * vid_num_chunks (vid_pkt_size * bytes_per_pixel + 12 + vid_null_size))/
number_of_lanes = pixels_per_line * dpipclkperiod
```

- On disabling the null packets, program the transmission parameters as follows:

```
(lanebyteclkperiod * vid_num_chunks (vid_pkt_size * bytes_per_pixel + 6))/number_of_lanes =
pixels_per_line * dpipclkperiod
```

52.2.2.5 Updating DPI Configuration Dynamically

Using shadow registers, it is possible to update DPI configuration dynamically without impacting the current frame. The VID_SHADOW_CTRL register controls this feature.

The controller uses the new configuration, only when the system requests it.

To update DPI configuration during the transmission of a video frame, the configuration of that frame must be stored in auxiliary registers. This allows the new frame configurations to be set through the APB interface without corrupting the current frame.

By default, this feature is disabled. To enable this feature, set the vid_shadow_en field of the VID_SHADOW_CTRL register to 1. On enabling this feature, the system provides the configuration in the auxiliary registers.

Figure 251 illustrates the necessary steps to update the DPI configuration.

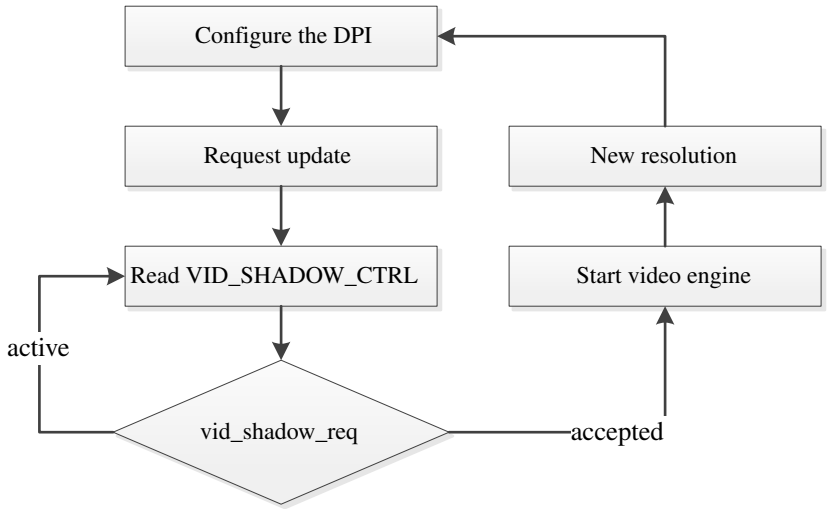


Figure 251. Flow to Update the DPI Configuration using Shadow Registers

52.2.2.5.1 Immediate Update

On enabling the shadow register feature, the auxiliary registers require DPI to be configured before the video engine starts. This means that, after a reset, vid_shadow_req is immediately granted.

To immediately update the active registers without the reset, set the vid_shadow_en and vid_shadow_req bits of the VID_SHADOW_CTRL register to 0.

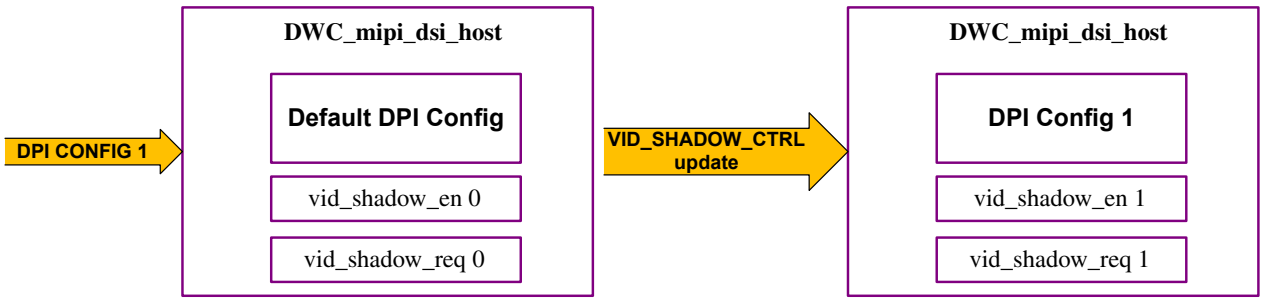


Figure 252. Immediate Update Procedure

52.2.2.5.2 Updating the Configuration During the Transmission of a Frame Using APB

To update the DPI configuration, follow these steps:

1. Set the `vid_shadow_en` bit of the `VID_SHADOW_CTRL` register to 1.
2. Set the `vid_shadow_req` bit of `VID_SHADOW_CTRL` register to 1.
3. Monitor the `vid_shadow_req` bit. Update completion sets this bit to 0.

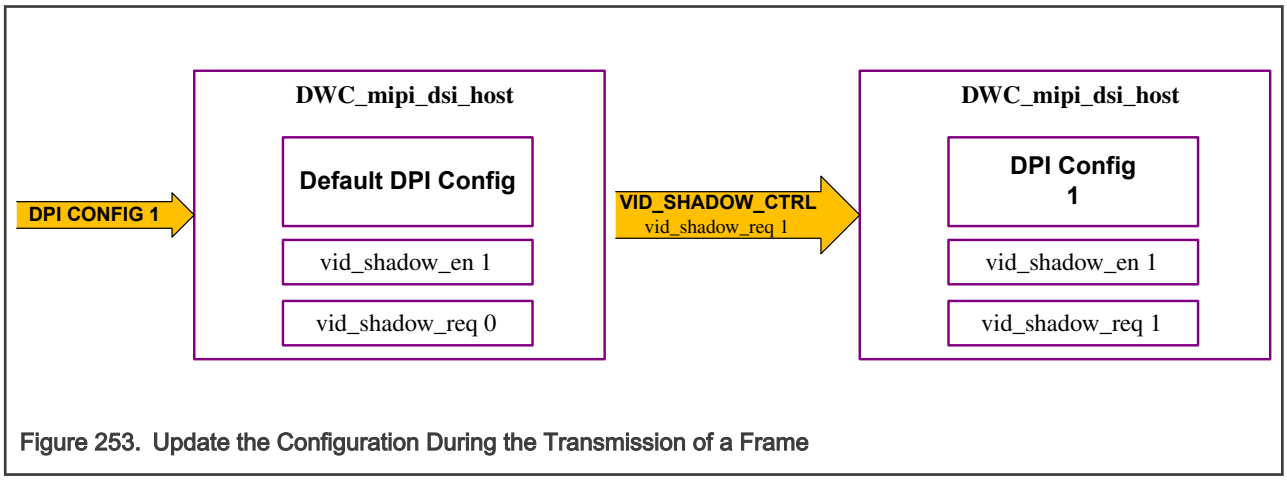


Figure 253. Update the Configuration During the Transmission of a Frame

52.2.2.5.3 Updating the Configuration During the Transmission of a Frame Using the External Signal

If the video engine's resolution change request is not possible through APB, use the `dpiupdatecfg` pin to request it. To monitor this pin, set the `vid_shadow_pin_req` bit of the `VID_SHADOW_CTRL` register to 1.

At the rising edge of the `dpiupdatecfg` signal, the controller generates an update request.

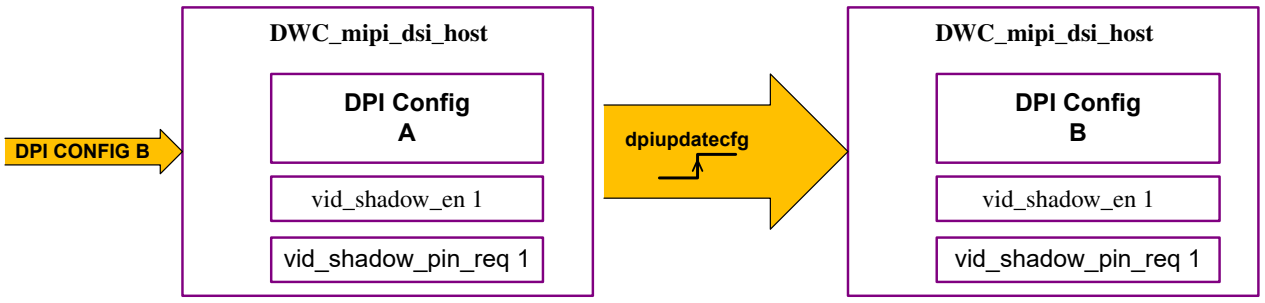


Figure 254. Update Configuration Using the External Signal

52.2.2.5.4 Requesting a Configuration Update

It is possible to request for the DPI configuration update at any part of the frame, because the MIPI DSI controller waits until the end of the frame to update it. However, avoid sending the update request during the first line of the frame because the data must propagate between clock domains.

52.2.2.6 Registers Related to DPI

Registers related to DPI are as follows:

- DPI_VCID
- DPI_COLOR_CODING
- DPI_CFG_POL
- DPI_LP_CMD_TIM

52.2.3 APB Slave Generic Interface

52.2.3.1 Overview of APB Slave Generic Interface

The APB Slave interface allows the transmission of generic information in Command mode.

52.2.3.2 Description of APB Slave Generic Interface

The MIPI DSI controller supports the transmission of write and read command mode packets, as described in the *MIPI DSI Specification*. The controller uses the APB register access to build these packets.

Based on the operation, the functions of the GEN_PLD_DATA register are as follows:

- Writing into this register, sends the data as payload when sending a Command mode packet.
- Reading this register, returns the payload of a read operation.

The GEN_HDR register contains the Command mode packet header type and header data. Writing into this register triggers the packet transmission implying that for a long Command mode packet, the packet's payload needs to be written in advance into the GEN_PLD_DATA register.

The valid packets available to be transmitted through the generic interface are as follows:

- Generic Write Short Packet 0 Parameters
- Generic Write Short Packet 1 Parameters
- Generic Write Short Packet 2 Parameters

- Generic Read Short Packet 0 Parameters
- Generic Read Short Packet 1 Parameters
- Generic Read Short Packet 2 Parameters
- Maximum Read Packet Configuration
- Generic Long Write Packet
- DCS Write Short Packet 0 Parameters
- DCS Write Short Packet 1 Parameters
- DCS Read Short Packet 0 Parameters
- DCS Write Long Packet

A set of bits in the CMD_PKT_STATUS register report the status of the FIFOs associated with APB interface support.

The controller transports the generic interface packets using one of the DSI transmission modes: Video mode or Command mode. If neither of these modes is selected, the controller does not transmit the packets through the link, and the related FIFOs eventually overflows.

52.2.3.3 Packet Transmission Using the Generic Interface

The packet transfer through the APB bus is based on the following conditions:

- The APB protocol defines that the write and read procedure takes two clock cycles each to be executed. This means that the maximum input data rate through the APB interface is always half the speed of the APB clock.
- The data input bus has a maximum width of 32 bits. This allows for a relation to be defined between the input APB clock frequency and the maximum bit rate achievable by the APB interface.
- When using only APB, the DSI link pixel bit rate is:

  ```
  DSI link pixel bit rate = (APB clock frequency) * 16 Mbps
  ```

- When using only the APB interface, the theoretical DSI link maximum bit rate is:

  ```
  DSI link maximum bit rate = APB clock frequency (in MHz)* 32 / 2 Mbps
  ```

  In the above equation, the number 32 represents the APB data bus width. The division by two is because each APB write procedure takes two clock cycles to be executed.
- The available bandwidth depends on the APB clock frequency, and increases with the clock frequency.

To drive the APB interface to achieve high bandwidth Command mode traffic transported by the DSI link, the MIPI DSI controller must operate only in the Command mode, and the APB interface must be the only data source currently in use. Thus, the APB interface has the entire bandwidth of the DSI link and does not share it with any another input interface source.

The memory write commands require maximum throughput from the APB interface, because they contain the most amount of data conveyed by the DSI link. While writing the packet information, first write the payload of a given packet into the payload FIFO using the GEN_PLD_DATA register. For the payload data of the command parameters, place the first byte to be transmitted in the least significant byte position of the APB data bus. For more information, see GEN_PLD_DATA register in the “Register Descriptions”.

After writing the payload, write the packet header into the command FIFO, so that it is correctly stored inside the Command FIFO. For more information about the packet header organization on the 32-bit APB data bus, see GEN_HDR register in the “Register Descriptions”.

The payload data for a memory write command contains pixel information and it must follow the pixel to byte conversion organization referred in the *MIPI DCS Specification Annex A*.

The Figure 255 to Figure 259 illustrates how the pixel data must be organized in the APB data write bus. The DCS long packets conveys the memory write commands, and these packets are encapsulated in a DSI packet. The DSI specifies that the DCS command must be present in the first payload byte of the packet, as shown in the diagrams. In Figure 255 to Figure 259, the Write Memory Command can be replaced by the DCS command Write Memory Start and Write Memory Continue.

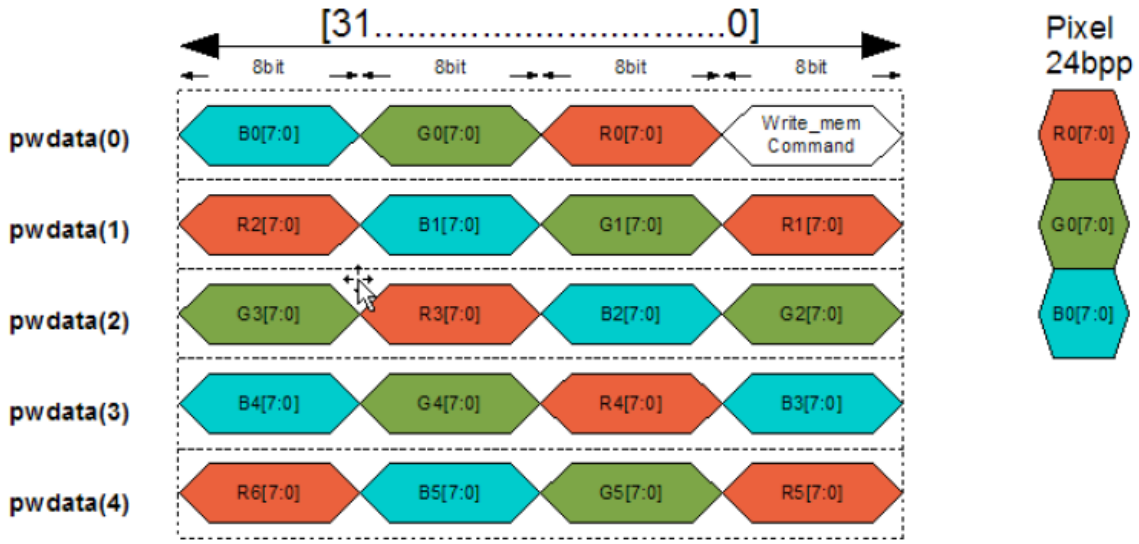


Figure 255. 24 bpp APB Pixel to Byte Organization

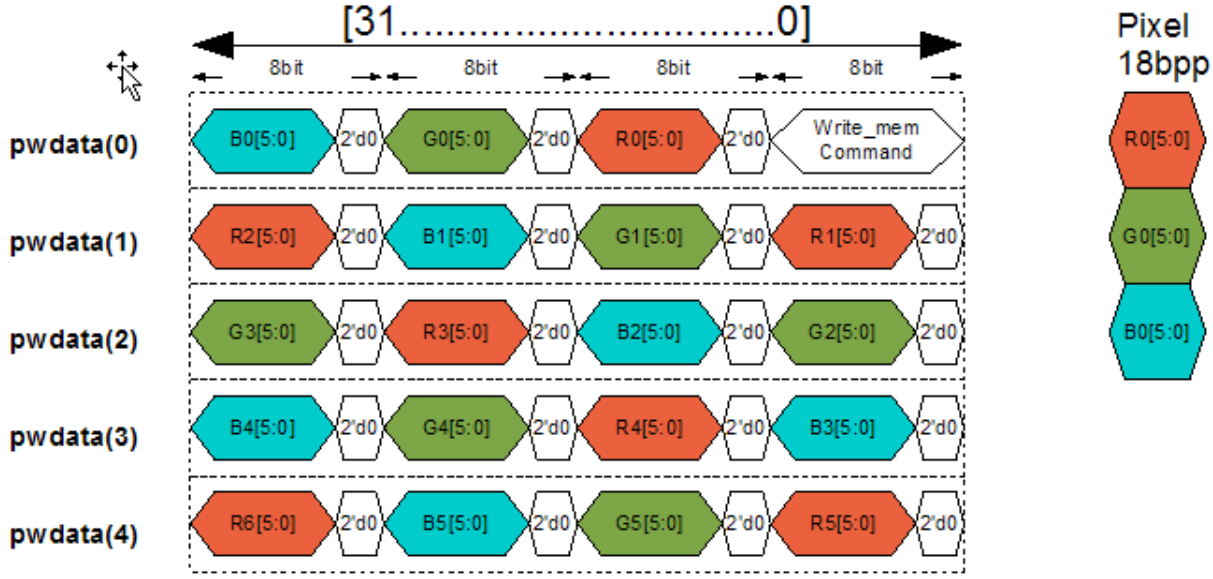


Figure 256. 18 bpp APB Pixel to Byte Organization

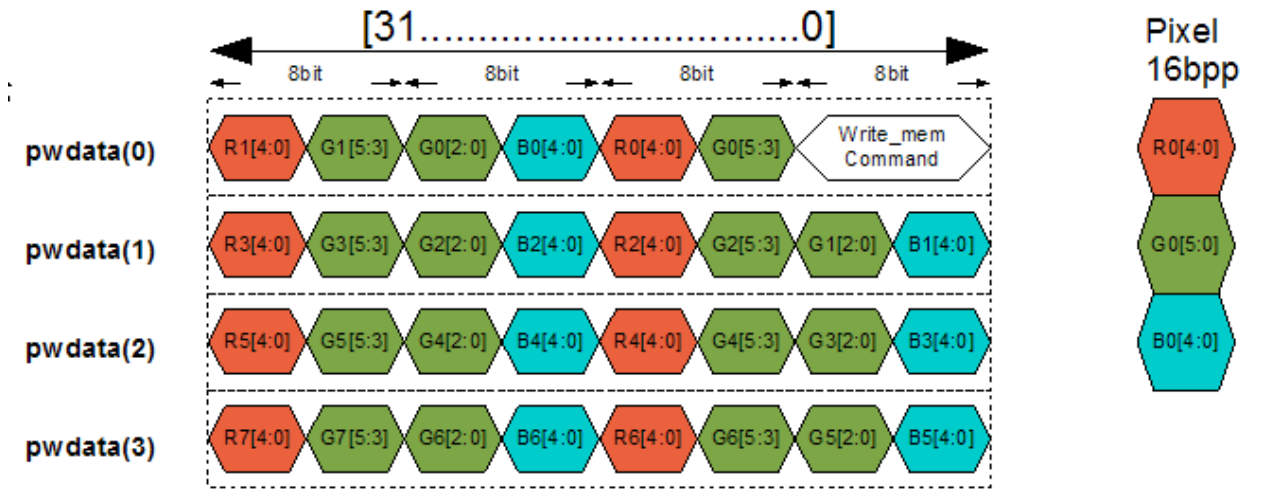


Figure 257. 16 bpp APB Pixel to Byte Organization

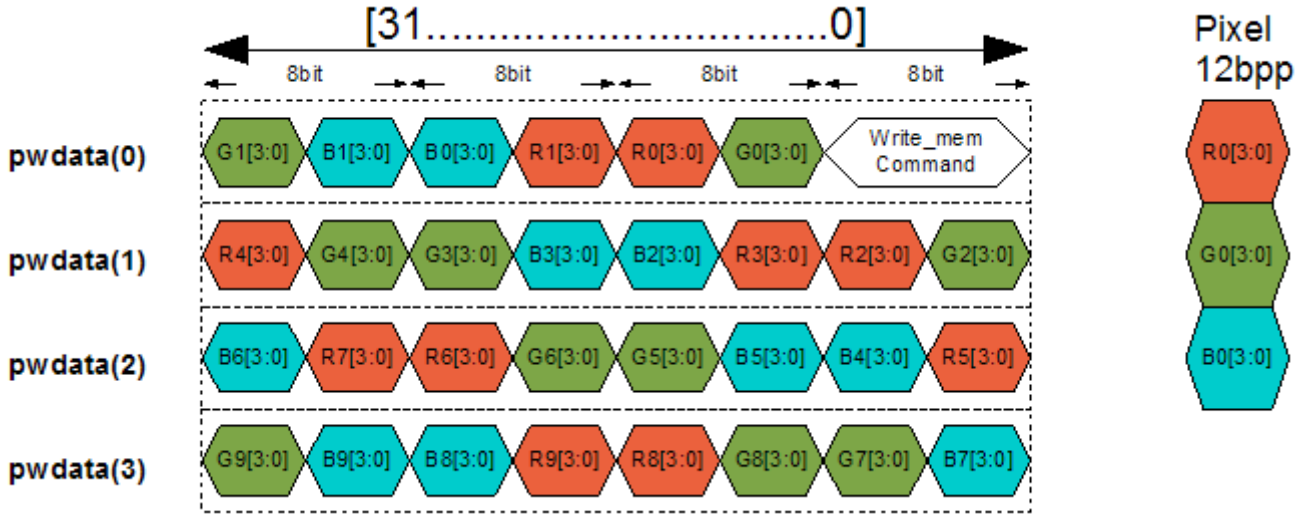


Figure 258. 12 bpp APB Pixel to Byte Organization

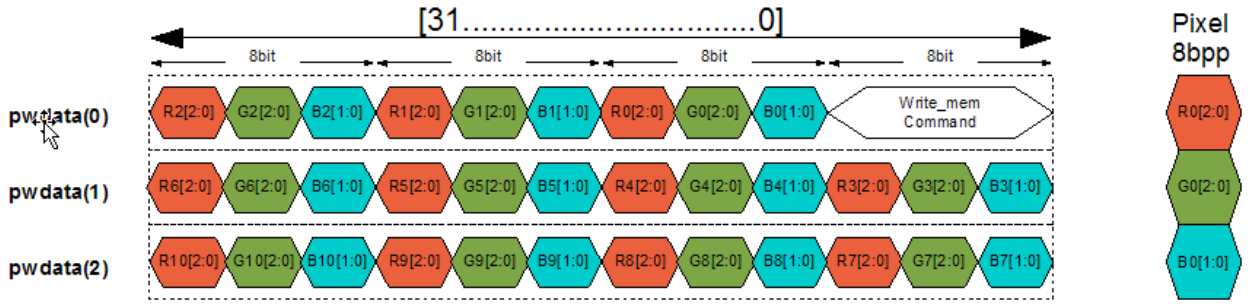


Figure 259. 8 bpp APB Pixel to Byte Organization

52.2.4 Transmission of Commands

52.2.4.1 Transmission of Commands in Video Mode

In the Video mode, the MIPI DSI controller supports the transmission of commands in both high-speed and low-power. The MIPI DSI controller uses Blanking or Low-power (BLLP) periods to transmit the commands inserted through the APB Generic interface. The shaded areas of Figure 260 illustrates these blanking periods.

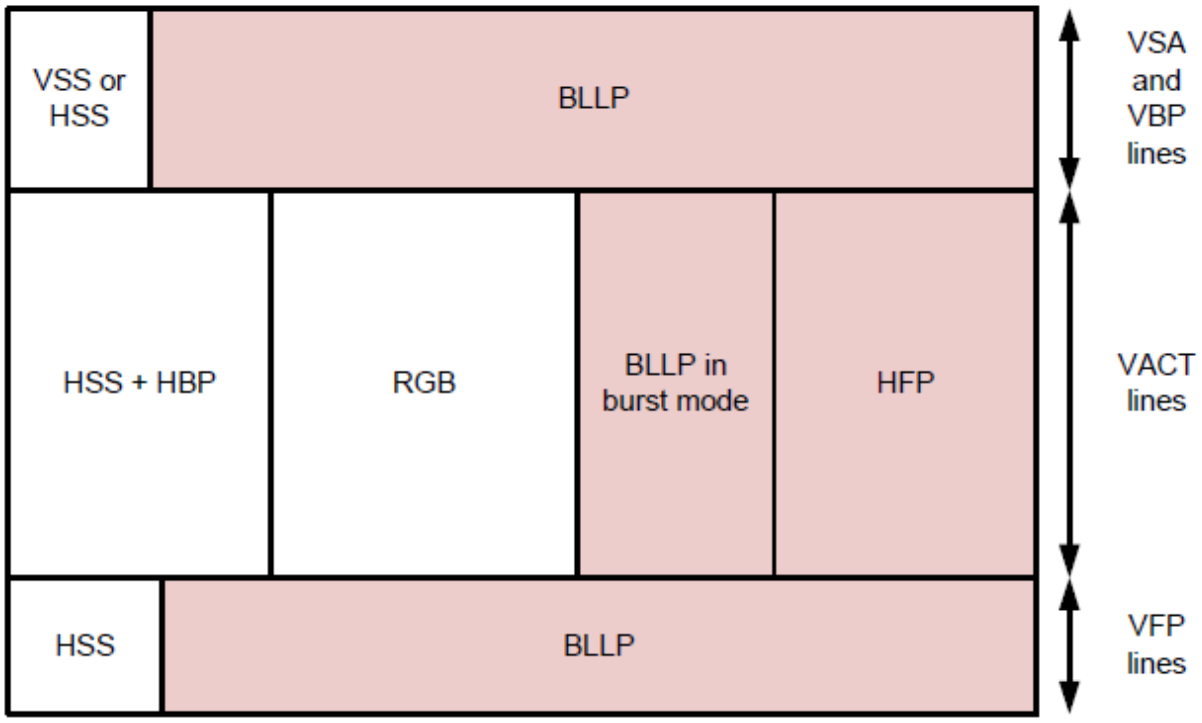


Figure 260. Command Transmission Periods within the Image Area

The controller transmits the commands in the blanking periods after the following packets or states:

- Vertical Sync Start (VSS) packets, if the Video Sync pulses are not enabled
- Horizontal Sync End (HSE) packets, in the VSA, VBP, and VFP regions
- Horizontal Sync Start (HSS) packets, if the Video Sync pulses are not enabled in the VSA, VBP, and VFP regions
- Horizontal Active (HACT) state

Besides the areas corresponding to BLLP, it is possible to send large commands during the last line of a frame. But this violates the line time for the Video mode and sets the `edpihalt` signal to request the DPI video timing signals to remain inactive. If a command does not fit into any BLLP area, it is postponed to the last line, causing the violation of the line time for the Video mode, as illustrated in Figure 261.

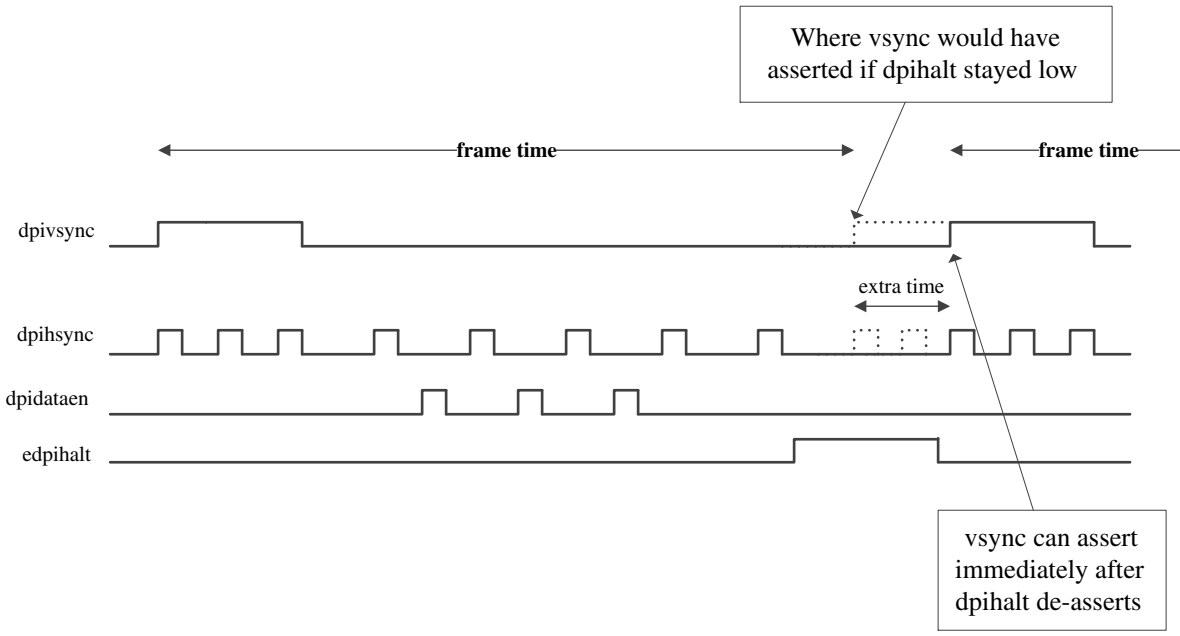


Figure 261. Transmission of Commands on the Last Line of a Frame

The controller transmits only one command per line, even in the case of the last line of a frame but one command is possible for each line.

In low-power, the controller sends only one command per line. In high-speed, the MIPI DSI controller sends as many commands as the controller determines to fit in the available time.

The MIPI DSI controller avoids sending commands in the last line because the last line might be shorter than the others. For instance, the line time (tL) can be half a cycle longer than the tL on the DPI, that is, each line in the frame takes half a cycle from time for the last line. This results in the last line being (½ cycle) x (number of lines -1) shorter than tL.

On transmitting a command, the controller asserts the `edpihalt` signal in the last line. The `dpivsync_edpiwms` signal asserts immediately after `edpihalt` de-asserts.

The `dpicolorm` and `dpishutdn` input signals also triggers the sending of command packets. The commands are DSI data types Color Mode On, Color Mode Off, Shut Down Peripheral, and Turn on Peripheral. These commands must not be sent in the VACT region. On setting the `lp_cmd_en` bit of the `VID_MODE_CFG` register to 1, the controller sends these commands in the low-power mode. In low-power mode, the `outvact_lpcmd_time` field of the `DPI_LP_CMD_TIM` register determines if it is possible to transmit these commands. It is assumed that `outvact_lpcmd_time` is greater than or equal to four bytes (number of bytes in a short packet), because the MIPI DSI controller does not transmit these commands on the last line.

If the frame_bta_ack_en field of the VID_MODE_CFG register is set, the MIPI DSI controller generates a BTA after the last line of a frame. This might coincide with a write command or a read command. In either case, the controller asserts the edpihalt signal, until an acknowledge is received (control of the DSI bus is returned to the host).

52.2.4.2 Transmission of Commands in Low-Power Mode

It is possible to configure the MIPI DSI controller to send low-power commands during the high-speed video mode transmission.

To enable this feature, set the lp_cmd_en bit of the VID_MODE_CFG register to 1.

In this case, it is necessary to calculate the time available in bytes, to transmit a command in low-power mode in the Horizontal Front Porch (HFP), Vertical Sync Active (VSA), Vertical Back Porch (VBP), and Vertical Front Porch (VFP) regions.

Bits 8 to 13 of the VID_MODE_CFG register indicates if it is possible for the MIPI DSI controller to go to low-power when idle. If the lp_cmd_en bit is set (1'b1) and non-video packets are in queue, the MIPI DSI controller ignores the low-power configuration and transmits low-power commands, even if it is not allowed to enter low-power in a specific region. After the low-power commands transmission, the MIPI DSI controller remains in low-power until a sync event occurs.

For example, consider VFP selected as high-speed region (lp_vfp_en = 1'b0) with lp_cmd_en set as a command to transmit in low-power in the VPF region. Then the controller transmits this command in low-power, and the line stays in low-power until a new HSS arrives.

52.2.4.2.1 Calculating the Time to Transmit Commands in Low-Power Mode in the VSA, VBP, and VFP Regions

The outvact_lpcmd_time field of the DPI_LP_CMD_TIM register indicates the time available (in bytes) to transmit a command in low-power mode (based on the escape clock) on a line during the VSA, VBP, and the VFP regions.

Calculation of outvact_lpcmd_time depends on the Video mode that you use. Figure 262 illustrates the timing intervals for the Non-Burst Video mode with sync pulses. Figure 263 illustrates the timing intervals for the Burst Video mode and Non-Burst Video mode with sync events.

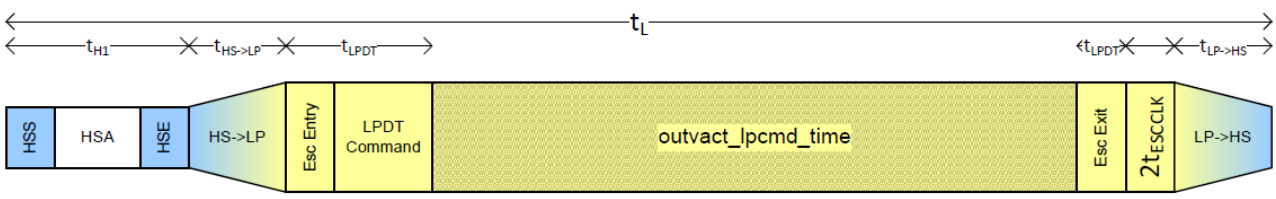


Figure 262. outvact_lpcmd_time for Non-Burst with Sync Pulses

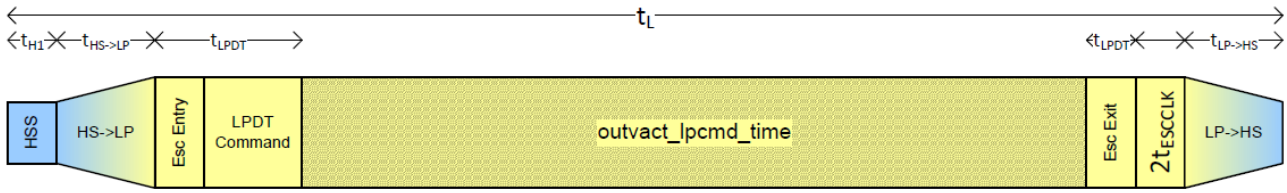


Figure 263. outvact_lpcmd_time for Burst or Non-Burst with Sync Events

This time calculation, is as follows:

```
outvact_lpcmd_time = (tL - (tH1 + tHS->LP + tLP->HS + tLPDT + 2tESCCLK))/(2 × 8 × tESCCLK)
```

Where,

- tL = Line time

- tH1 = Time of the HSA pulse for sync pulses mode (Figure 262) or the time to send the HSS packet, including EoTp (Figure 263)
- tHS->LP = Time to enter the low-power mode
- tLP->HS = Time to leave the low-power mode
- tLPDT = PHY timing related with Escape Mode Entry, LPDT command, and Escape Exit. According to the PHY specification, this value is always 11 bits in low-power (or 22 TX Escape clock cycles).
- tESCCLK = Escape clock period as programmed in the `tx_esc_clk_division` field of the `CLKMGR_CFG` register
- 2xtESCCLK = Delay imposed by the controller implementation

In the above equation, division by eight is done to convert the time available to bytes. Division by two is done because one bit is transmitted once in every two escape clock cycles. The `outvact_lpcmd_time` field can be compared directly with the size of the command to be transmitted to determine if there is enough time to transmit the command. This field limits the maximum size of a command that can be transmitted in low-power mode to 255 bytes. Program this register to a value greater than or equal to 4 bytes for the transmission of the DCTRL commands, such as the shutdown and color in low-power mode.

For example, consider a frame with 12.4 µs per line and assume an escape clock frequency of 20 MHz and a lane bit rate of 800 Mbits. In this case, it is possible to send 124 bits in escape mode (that is, 124 bit = 12.4 µs * 20 MHz/2). Still, you have to consider the PHY protocol and PHY timings. The assumptions are as follows:

- Lane byte clock period is 10 ns (800 Mbits per Lane)
- Escape clock period is 50 ns (the `tx_esc_clk_division` field of the `CLKMGR_CFG` register set to 5)
- Video is transmitted in Non-Burst mode with sync pulses bounded by HSS and HSE packets
- DSI is configured for two lanes
- PHY takes 180 ns to transit from low-power to high-speed modes (the `phy_lp2hs_time` field of the `PHY_TMR_CFG` register set to 18)
- PHY takes 200 ns to transit from high-speed to low-power modes (the `phy_hs2lp_time` field of the `PHY_TMR_CFG` register set to 20)
- tHSA = 420 ns

In this example, a 13-byte command can be transmitted as follows:

```
outvact_lpcmd_time= (12.4 µs - (420 ns + 180 ns +200 ns +(22×50 ns + 2×50 ns)))/(2×8×50 ns) = 13 bytes
```

52.2.4.2.2 Calculating the Time to Transmit the Commands in Low-Power Mode in the HFP Region

The `invact_lpcmd_time` field of the `DPI_LP_CMD_TIM` register indicates the time available (in bytes) to transmit a command in low-power mode (based on the escape clock) in the Vertical Active (VACT) region. To calculate the value of `invact_lpcmd_time`, consider the Video mode that you use. Figure 264 illustrates the timing intervals for Non-Burst Video mode with sync pulses. Figure 265 illustrates the timing intervals for Non-Burst Video mode with sync events. Figure 266 illustrates the Burst Video mode.

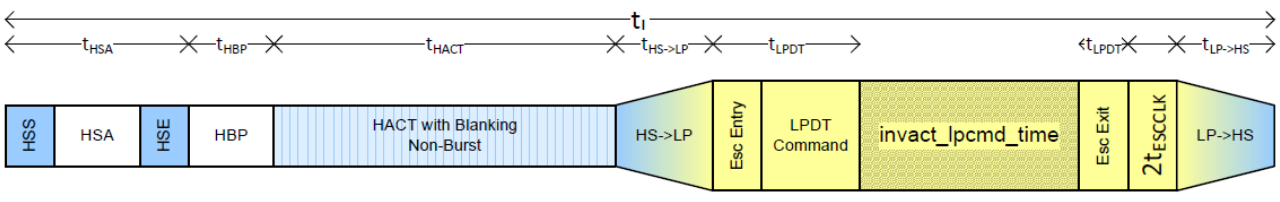


Figure 264. invact_lpcmd_time for Non-Burst with Sync Pulses

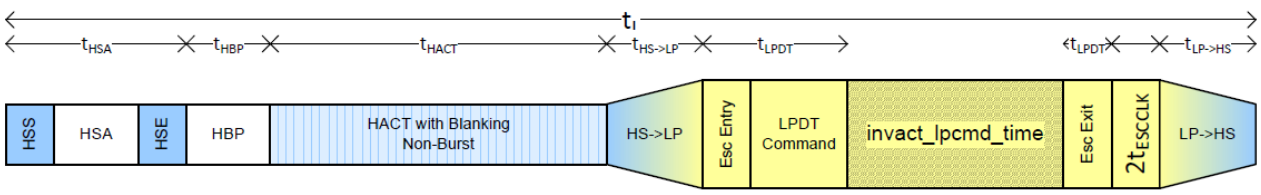


Figure 265. invact_lpcmd_time for Non-Burst with Sync Events

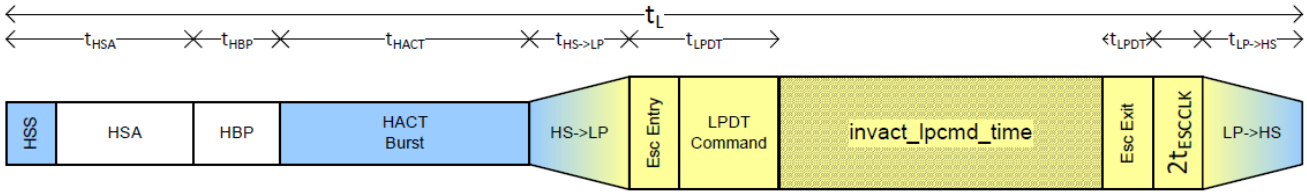


Figure 266. invact_lpcmd_time for Burst Mode

This time calculation, is as follows:

```
invact_lpcmd_time = (tL - (tHSA+tHBP+tHACT+ tHS->LP + tLP->HS + tLPDT + 2tESCCLK))/(2×8×tESCCLK)
```

Where

- tL = Line time
- tHSA = Time of the HSA pulse (value configured in the `VID_HSA_TIME` register)
- tHBP = Time of Horizontal back porch (value configured in the `VID_HBP_TIME` register)
- tHACT = Time of Video active. For Burst mode, the Video active is time compressed and is calculated as follows:

  ```
  tHACT = vid_pkt_size * Bytes_per_Pixel /Number_Lanes * tLane_byte_clk
  ```

- tESCCLK = Escape clock period programmed in the `tx_esc_clk_division` field of the `CLKMGR_CFG` register

The `invact_lpcmd_time` field can be compared directly with the size of the command to be transmitted to determine if there is time to transmit the command.

For example, consider a frame with 16.4 µs per line and assume an escape clock frequency of 20 MHz and a Lane bit rate of 800 Mbits. In this case, it is possible to send 420 bits in escape mode (that is, 164 bits = 16.4 µs * 20 MHz/2). Because this is the Vertical Active region of the frame, consider the HSA, HBP, and HACT timings apart from the D-PHY protocol and PHY timings. The assumptions, are as follows:

- PHY: D-PHY
- Number of active lanes is 4
- Lane byte clock period (lanebyteclkperiod) is 10 ns (800 Mbits per Lane)
- Escape clock period is 50 ns (the `tx_esc_clk_division` field of the `CLKMGR_CFG` register set to 5)
- PHY takes 180 ns to transit from low-power to high-speed modes (the `phy_lp2hs_time` field of the `PHY_TMR_CFG` register set to 18)
- PHY takes 200 ns to transit from high-speed to low-power modes (the `phy_hs2lp_time` field of the `PHY_TMR_CFG` register set to 20)
- tHSA = 420 ns
- tHBP = 800 ns

- tHACT = 12800 ns to send 1280 pixel at 24 bpp
- Video is transmitted in Non-Burst mode
- MIPI DSI controller is configured for four lanes

In this example, consider sending the video in Non-Burst mode. The `invact_lpcmd_time` is calculated as:

```
invact_lpcmd_time = (16.4 us -(420 ns + 800 ns + 12.8 us + 180 ns +200 ns +(22×50 ns + 2×50 ns))/(2×8×50 ns)
= 1 byte
```

Therefore, it is possible to transmit only one byte in this period. A short packet (for example, generic short write) requires a minimum of four bytes. Therefore, in this example, commands are not sent in the VACT region.

On enabling the Burst mode, more time is available to transmit the commands in the VACT region, because HACT is time compressed.

```
invact_lpcmd_time = (16.4 us - (420 ns + 800 ns + (1280×3/4×10 ns) + 180 ns + 200 ns + (22×50 ns + 2×50
ns)/(2×8×50 ns) = 5 bytes
```

Thus for Burst mode, the `invact_lpcmd_time` is 5 bytes and you can effectively send a 4-byte short packet.

52.2.4.2.3 Transmission of Commands in Different Periods

The `outvact_lpcmd_time` and `invact_lpcmd_time` fields allow a simple comparison to determine if it is possible to transmit a command in any of the BLLP periods.

Figure 267 illustrates the meaning of `invact_lpcmd_time` and `outvact_lpcmd_time`, matching them with the shaded areas and the VACT region.

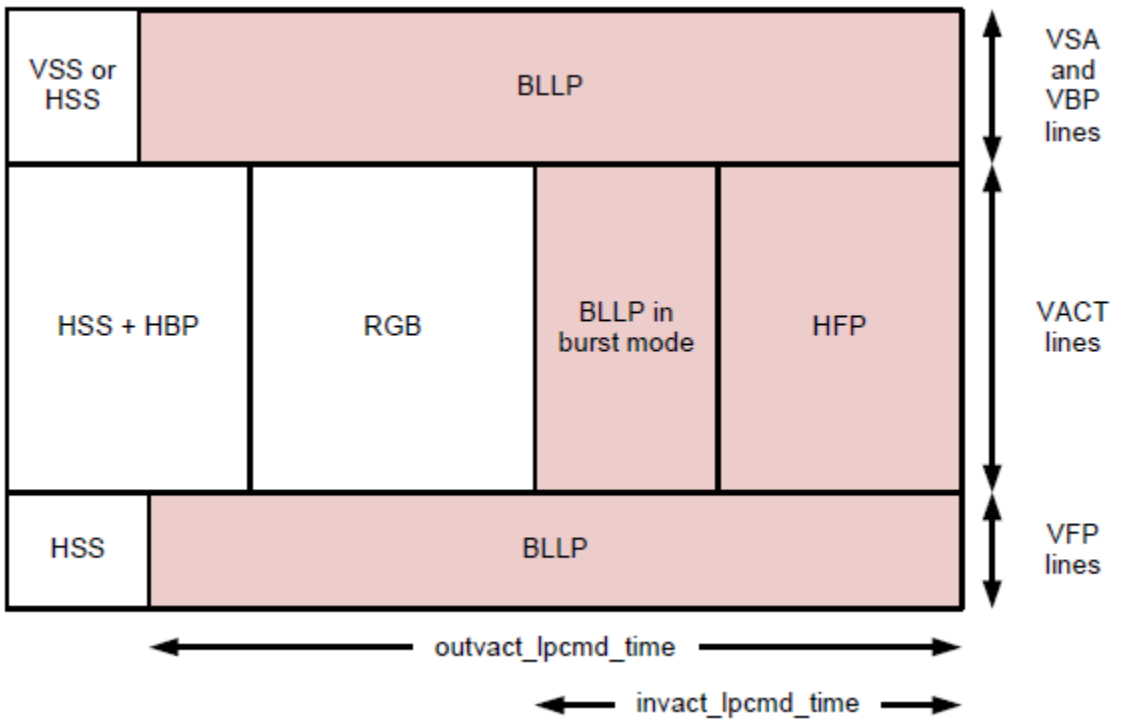


Figure 267. Location of outvact_lpcmd_time and invact_lpcmd_time in the Image Area

52.2.4.3 Transmission of Commands in High-Speed Mode

The controller sends the commands in high-speed in Video mode, if the `lp_cmd_en` bit of the `VID_MODE_CFG` register is set to 0. In this case, the MIPI DSI controller, automatically determines the area where each command can be sent and requires no programming or calculation.

52.2.4.4 Read Command Transmission

The `max_rd_time` field of the `PHY_TMR_RD_CFG` register configures the maximum amount of time required to perform a read command in lane byte clock cycles. It is calculated as follows:

```
max_rd_time = Time to transmit the read command in low-power mode + Time to enter and leave low-power mode
+ Time to return the read data packet from the peripheral device
```

The time to return the read data packet from the peripheral depends on the number of bytes read and the escape clock frequency of the peripheral, not the escape clock of the host. In both the high-speed and low-power mode, the `max_rd_time` field determines if there is time to complete a read command in a BLLP period.

In high-speed mode (`lp_cmd_en` register field set to 0), `max_rd_time` is calculated as follows:

```
max_rd_time = (tHS->LP + tLP->HS + tread + 2x tBTA) / lanebyteclkperiod
```

In low-power mode (`lp_cmd_en` register field set to 1), `max_rd_time` is calculated as follows:

```
max_rd_time = (tHS->LP + tLP->HS + tLPDT + tlprd + tread + 2x tBTA) / lanebyteclkperiod
```

Where,

- tHS->LP = Time to enter the low-power mode
- tLP->HS = Time to leave the low-power mode
- tLPDT = PHY timing related to Escape mode entry, LPDT command, and Escape mode exit. According to the PHY specification, this value is always 11 bits in low-power (or 22 TX escape clock cycles).
- tlprd = Read command time in low-power mode (64 * TX esc clock)
- tread = Time to return the read data packet from the peripheral
- tBTA = time to perform a bus turnaround (PHY dependent)

It is recommended to keep the maximum number of bytes read from the peripheral to a minimum, to have sufficient time available to issue the read commands in a line time.

To avoid sending read commands on the last line of a frame, ensure the following:

```
(max_rd_time x Lane byte clock period) < (outvact_lpcmd_time x 16 x escape clock period of the host)
```

If it is necessary to read a large number of parameters (greater than 16), increase the `max_rd_time` on executing the read command. On completion of the read operation, decrease the `max_rd_time` value.

Issuing a read command on the last line of a frame, asserts the `edpihalt` signal, and remains asserted until the read command is in progress, during which the video transmission must be stopped.

52.2.4.4.1 Clock Lane in Low-Power Mode (D-PHY)

To reduce D-PHY power consumption, the MIPI DSI controller, when not transmitting in the high-speed mode, allows clock lane transition to the low-power mode. The controller automatically handles the transition of the clock lane from high-speed (clock lane active sending clock) to low-power state without direct intervention by the software. To enable this feature, configure the `phy_txrequestclkhs` and the `auto_clklane_ctrl` bits of the `LPCLK_CTRL` register.

52.2.4.4.2 Clock Lane in Low-Power Mode for Command Mode (D-PHY)

In the Command mode, it is possible for the MIPI DSI controller, to place the clock lane in the low-power mode, when it does not have any high-speed packets to transmit.

52.2.4.4.3 Clock Lane in Low-Power Mode for Video Mode (D-PHY)

In the Video mode (DPI), the MIPI DSI controller uses its internal video and PHY timing configurations to determine if there is time available for the clock line to enter the low-power mode, without compromising video data transmission of pixel data and sync events.

Along with a correct configuration of the Video mode (for more information about DPI, see DPI), the MIPI DSI controller also needs to know the time required by the clock lane to go from high-speed to low-power and from low-power to high-speed.

Program the `PHY_TMR_LPCLK_CFG` register with the following values:

- `phy_clkhs2lp = Time from high-speed to low-power in clock lane / Byte clock period in high-speed (lanebyteclk)`
- `phy_clklp2hs = Time from low-power to high-speed in clock lane / Byte clock period in high-speed (lanebyteclk)`

Based on the programmed values, the MIPI DSI controller, calculates if there is enough time for the clock lane to enter the low-power mode, during inactive regions of the video frame.

There is an exception where the clock lane is activated even when there is no high-speed packet to be transmitted. If a command is not allowed to be transmitted in any of the available blanking periods, it is transmitted during the last line of the frame. This situation asserts the `edpihalt` signal, indicating to the video engine to halt the video generation. Upon `edpihalt` signal assertion, the controller requests the clock lane to start transmission of high-speed clock. This is to avoid delay in the transmission of the next frame vertical synchronism event, by the clock lane low-power to high-speed transition delay. For more information, see Figure 261.

The MIPI DSI controller decides the best approach, for power saving, from the following possible scenarios:

- There is no sufficient time to go to the low-power mode. Therefore, blanking period is added as shown in Figure 268.

Figure 268. Clock Lane and Data Lanes in High-Speed

- There is sufficient time for the data lanes to go to the low-power mode but not enough time for the clock lane to enter the low-power mode. Figure 269 illustrates this.

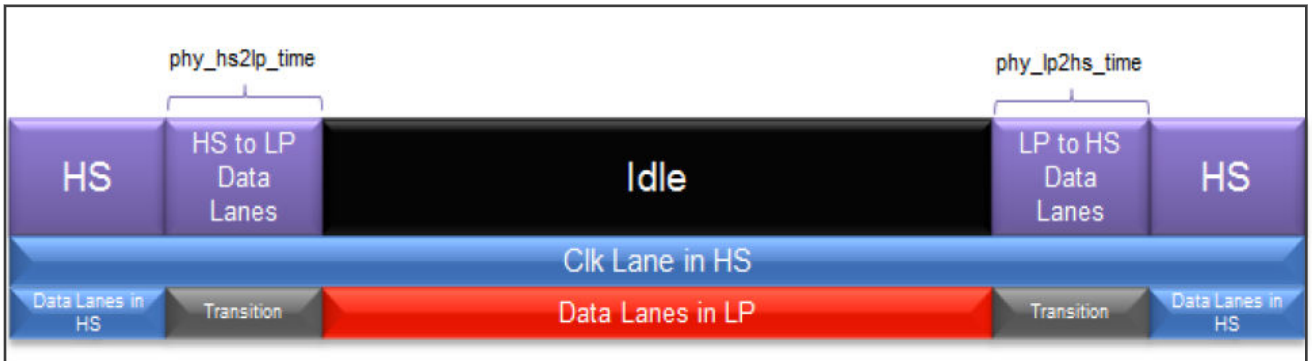


Figure 269. Clock Lane in High-Speed and Data Lanes in Low-Power

- There is sufficient time for both data lanes and the clock lane to go to the low-power mode. Figure 270 illustrates this.

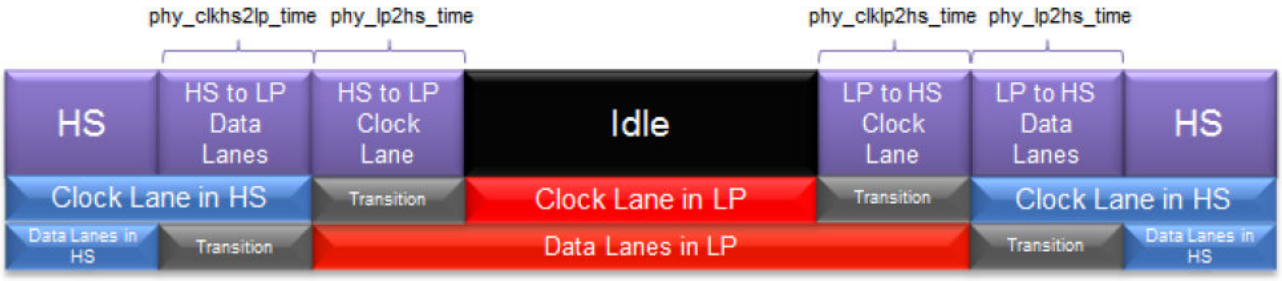


Figure 270. Clock Lane and Data Lanes in Low-Power

52.2.5 Virtual Channels

52.2.5.1 Overview of Virtual Channels

The MIPI DSI controller supports a specific virtual channel for each interface, enabling the system to address multiple displays simultaneously.

On configuring the DPI for a particular Virtual Channel, it is possible to use the APB Slave Generic interface to issue commands, while the video stream is being transmitted. By doing this, it is possible to send the commands through the ongoing video stream, addressing different virtual channels and thus enable the interface with multiple displays.

During the Video mode, the video stream transmission has the maximum priority. Therefore, the controller transmits sideband packets such as the ones from the generic interface, only when there is time available within the video stream transmission.

The MIPI DSI controller identifies the available time periods and uses them to transport the generic interface packets. Figure 271 illustrates where the MIPI DSI controller inserts the packets from the APB Generic interface within the video stream transmitted by the DPI.

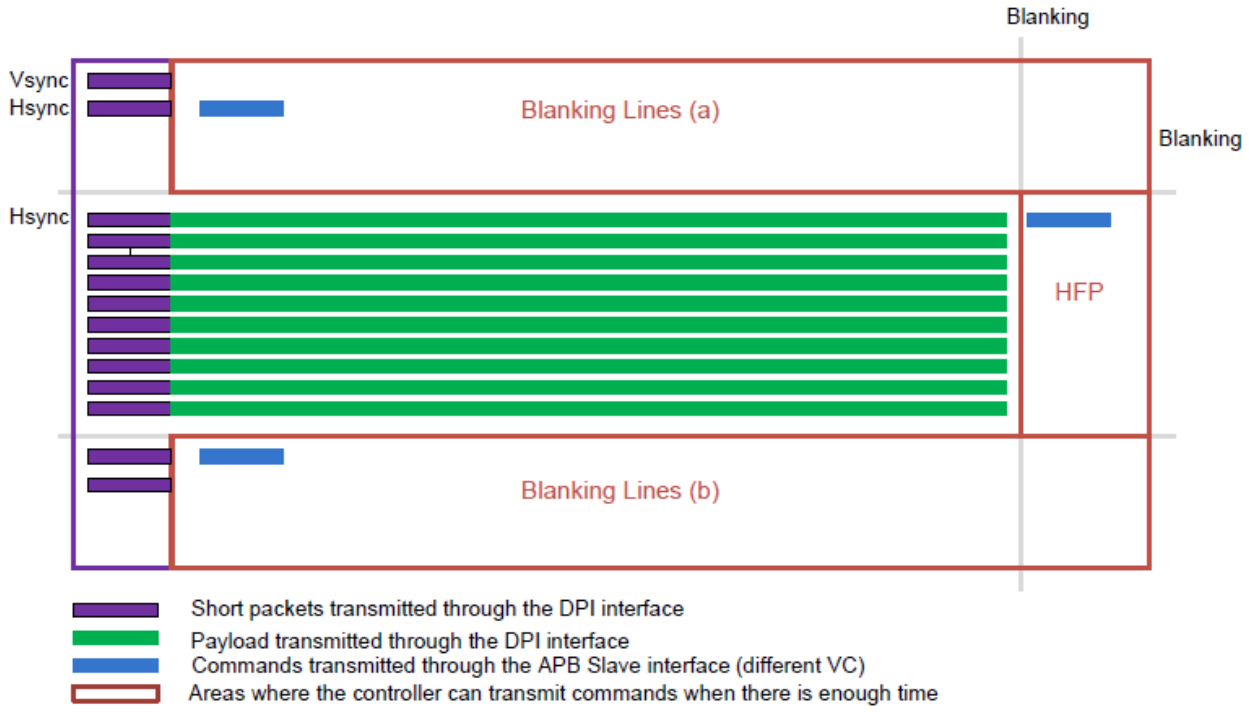


Figure 271. Command Transmission Through Generic Interface When DPI is the Configured System Interface

It is also possible to address the multiple displays with only the generic interface using different Virtual Channels. The configuration does not restrict the generic interface to any particular virtual channel, and therefore, it is possible to issue packets with different virtual channels. This enables the interface to time multiplex the packets to be provided to the displays with different virtual channels.

52.2.5.2 Registers Related to Virtual Channels

Use the following registers to configure the virtual channel ID associated with transmissions over the configured system interface, and APB Slave Generic interfaces:

- `DPI_VCID`: The `dpi_vcid` field of `DPI_VCID` register, configures the DPI virtual channel ID indexed to the Video mode packets.
- `GEN_HDR`: This register configures the packet header (which includes the virtual channel ID to be used) for transmissions using APB Slave Generic interface.

52.2.6 Video Mode Pattern Generator

52.2.6.1 Overview of Video Mode Pattern Generator

The Video Mode Pattern Generator allows the transmission of horizontal or vertical color bar and PHY BER testing pattern without any stimuli.

Table 500 lists the frame requirements that must be defined in the video registers.

Table 500. Frame Requirement Configuration Registers

Register Name	Description
VID_MODE_CFG	Video mode configuration
VID_PKT_SIZE	Video packet size
VID_NUM_CHUNKS	Number of chunks
VID_NULL_SIZE	Null packet size
VID_HSA_TIME	Horizontal sync active time
VID_HBP_TIME	Horizontal back porch time
VID_HLINE_TIME	Line time
VID_VSA_LINES	Vertical sync active period
VID_VBP_LINES	Vertical back porch period
VID_VFP_LINES	Vertical front porch period
VID_VACTIVE_LINES	Vertical resolution

52.2.6.2 Registers Related to Video Mode Pattern Generator

For more information, see the following registers related to video mode pattern generator:

- VID_MODE_CFG
- VID_PKT_SIZE
- VID_NUM_CHUNKS
- VID_NULL_SIZE
- VID_HSA_TIME
- VID_HBP_TIME
- VID_HLINE_TIME
- VID_VSA_LINES
- VID_VBP_LINES
- VID_VFP_LINES
- VID_VACTIVE_LINES

52.2.6.3 Color Bar Pattern

The color bar pattern comprises of eight bars of white, yellow, cyan, green, magenta, red, blue, and black colors. To calculate the color width, divide the line pixel size (vertical pattern) or the number of lines (horizontal pattern) by eight.

In the vertical color bar mode, each single color bar has a width of the number of pixels in a line divided by eight. In case, the number of pixels in a line is not divisible by eight, the last color (black) contains the remaining pixels.

In the horizontal color bar mode, each color line has a color width of the number of lines in a frame divided by eight. In case, the number of lines in a frame is not divisible by eight, the last color (black) contains the remaining lines.

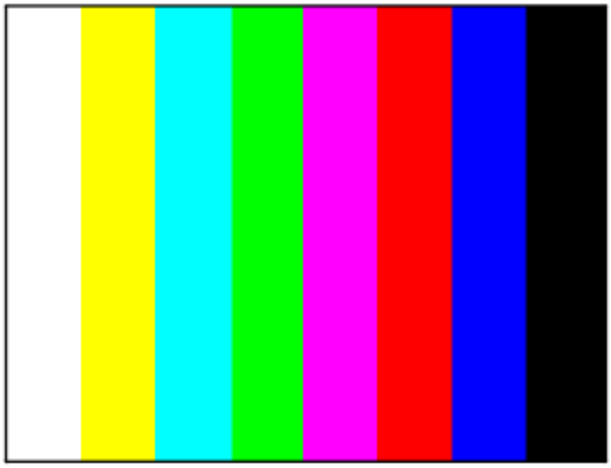


Figure 272. Vertical Color Bar Mode

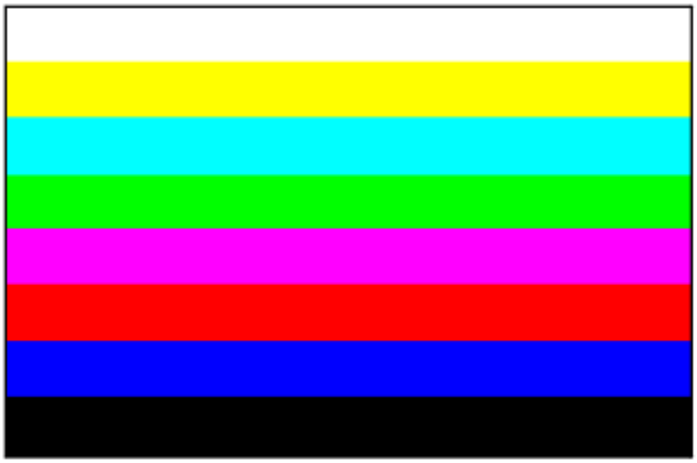


Figure 273. Horizontal Color Bar Mode

52.2.6.4 Color Coding

52.2.6.4.1 RGB Components

Table 501 lists the RGB components.

Table 501. RGB Components

	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
R	High	High	Low	Low	High	High	Low	Low
G	High	High	High	High	Low	Low	Low	Low
B	High	Low	High	Low	High	Low	High	Low

52.2.6.4.2 YCbCr Components

To avoid YCbCr negative components, YCbCr color coding is coded with the following offsets:

- Y has a range of 219 and an offset of +16
- CB and CR have a range of ±112 and offset of +128

Table 502 lists the YCbCr 8-Bit components.

Table 502. YCbCr 8-Bit Components

	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
Y [7:0]	235	210	170	145	106	81	41	16
Cr [7:0]	128	16	166	54	202	90	240	128
Cb [7:0]	128	146	16	34	222	240	110	128

Table 503 lists the YCbCr 10-Bit components. Each component of the 20-Bit YCbCr 4:2:2 loosely packed is calculated by multiplying the respective component of the 16-Bit YCbCr 4:2:2 by 4.

Table 503. YCbCr 10-Bit Components

	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
Y [9:0]	940	840	680	580	424	324	164	64
Cr [9:0]	512	64	664	216	808	360	960	512
Cb [9:0]	512	584	64	136	888	960	440	512

Table 504 lists the YCbCr 12-Bit components. Each component of the 24-Bit YCbCr 4:2:2 loosely packed is calculated by multiplying the respective component of the 16-Bit YCbCr 4:2:2 by 16.

Table 504. YCbCr 12-Bit Components

	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
Y [11:0]	3760	3360	2720	2320	1696	1296	656	256
Cr [11:0]	2048	256	2656	864	3232	1440	3840	2048
Cb [11:0]	2048	2336	256	544	3552	3840	1760	2048

52.2.6.5 BER Testing Pattern

The BER testing pattern simplifies the conformance testing. This pattern tests the RX PHY capability to receive the data correctly. The following data patterns are required:

- X bytes of 0xAA (high-frequency pattern, inverted)
- X bytes of 0x33 (mid-frequency pattern)
- X bytes of 0xF0 (low-frequency pattern, inverted)
- X bytes of 0x7F (lone 0 pattern)
- X bytes of 0x55 (high-frequency pattern)

- X bytes of 0xCC (mid-frequency pattern, inverted)
- X bytes of 0x0F (low-frequency pattern)
- Y bytes of 0x80 (lone 1 pattern)

In most cases, Y is equal to X. However, depending on the line length and the color coding used, Y might be a different value than X.

With RGB888 color coding and horizontal resolution in multiples of eight, the pattern as shown in Figure 274, appears on the DSI display.

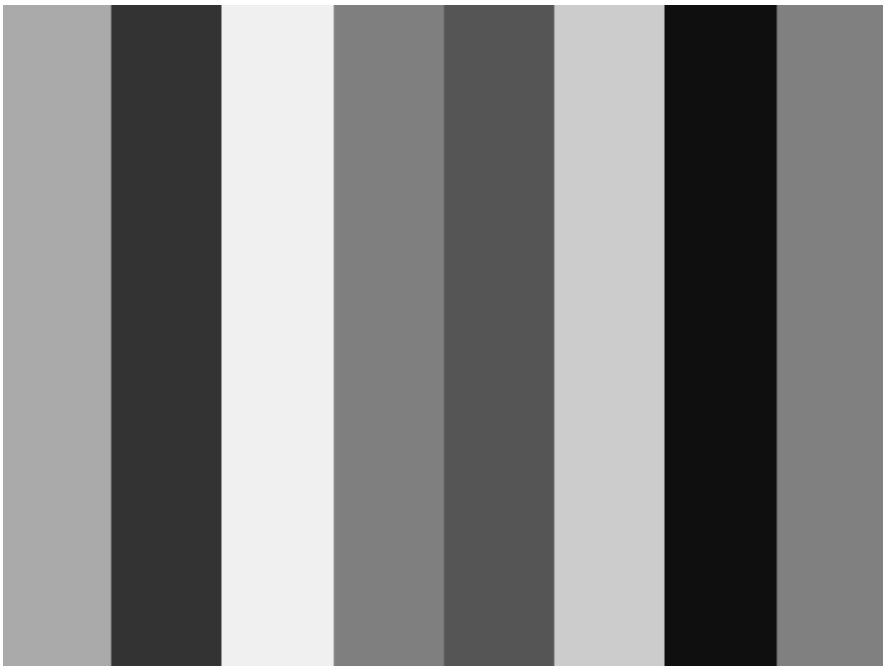


Figure 274. RGB888 BER Testing Pattern

52.2.6.6 Video Mode Pattern Generator Resolution

The minimum supported resolution for the video pattern generator is the resolution where `VID_PKT_SIZE` is greater than 32-bit.

52.2.6.6.1 Vertical Pattern

The division of horizontal resolution (pixels) for the eight test pattern colors, determines the width of each color. If the horizontal resolution is not divisible by eight, the last color (black) is extended to fill the resolution. In the example in Figure 275, the horizontal resolution is 103.

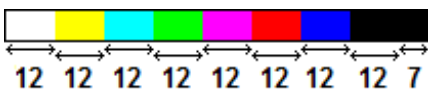


Figure 275. Vertical Pattern (103*15)

52.2.6.6.2 Horizontal Pattern

The division of the number of vertical resolution (lines) for eight test pattern colors, determines the width of each color bar. If the vertical resolution is not divisible by eight, the last color (Black) is extended to fill the resolution.

Figure 276. Horizontal Pattern (103*15)

52.2.7 Supported 2-Port RAM Types

The MIPI DSI controller uses 2-port RAMs to implement FIFOs for data storage and synchronization. The 2-port RAMs can be categorized on the basis of the way in which data is accessed for read operations, and the MIPI DSI controller works with these 2-port RAM categories, yielding the same performance.

Synchronous 2-port RAM

The synchronous 2-port RAM requires a read clock to get the addressed read data. The data retrieval is in synchronous with `rd_clk`, but has an extra clock cycle of delay. The synchronous 2-port RAM returns data at the interface one clock cycle after the read address is given. The Figure 277 illustrates a synchronous 2-port RAM.

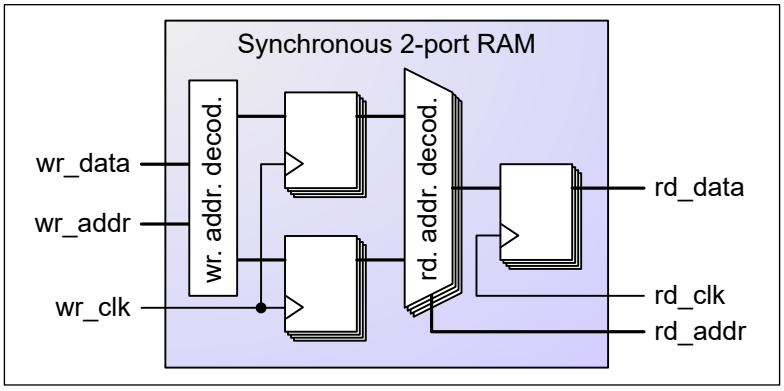


Figure 277. Synchronous 2-port RAM

52.2.8 Timeout Timers

52.2.8.1 Peripheral Response Timeout

A peripheral might not immediately respond correctly to some of the received packets. For example, consider a peripheral receiving a read request, but due to its architecture, cannot access the RAM for a while. This may be because the panel is being

refreshed, and therefore the peripheral takes some time to respond. In this case, set a timeout to ensure that the host waits long enough, so that the device processes the previous data before receiving the new data or responds correctly to new requests.

Table 505 lists the events belonging to various categories having an associated timeout for peripheral response.

Table 505. List of Events of Different Categories of the PRESP_TO Counter

Category	Event
Items implying a BTA PRESP_TO	Bus Turn-Around
READ requests indicating a PRESP_TO (replicated for high-speed and low-power)	• (0x04) Generic read, no parameters short • (0x14) Generic read, 1 parameter short • (0x24) Generic read, 2 parameters short • (0x06) DCS read, no parameters short
WRITE requests indicating a PRESP_TO (replicated for high-speed and low-power)	• (0x03) Generic short write, no parameters short • (0x13) Generic short write, 1 parameter short • (0x23) Generic short write, 2 parameters short • (0x29) Generic long write long • (0x05) DCS short write, no parameters short • (0x15) DCS short write, 1 parameter short • (0x39) DCS long write/write_LUT Command packet long • (0x37) Set maximum return packet size

The MIPI DSI controller ensures that, on sending an event that triggers a timeout, the PHY switches to the Stop state and a counter starts running until it reaches the value of that timeout. The link remains in the low-power-11 state and unused until the timeout ends, even if there are other events ready to be transmitted.

Figure 278, Figure 279, and Figure 280 illustrates the counting flow in the PRESP_TO counter for the three categories listed in Table 505.

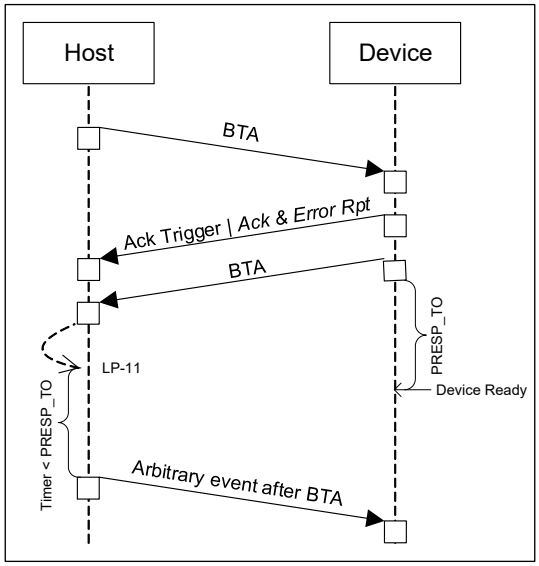


Figure 278. Timing of PRESP_TO after a Bus Turn-Around

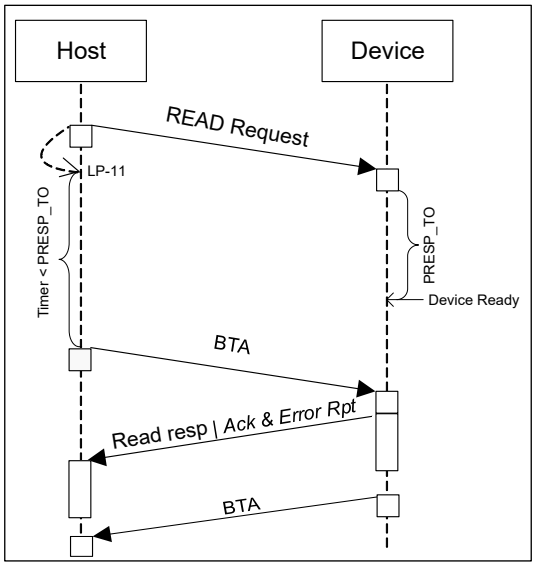


Figure 279. Timing of PRESP_TO after a Read Request (high-speed or low-power)

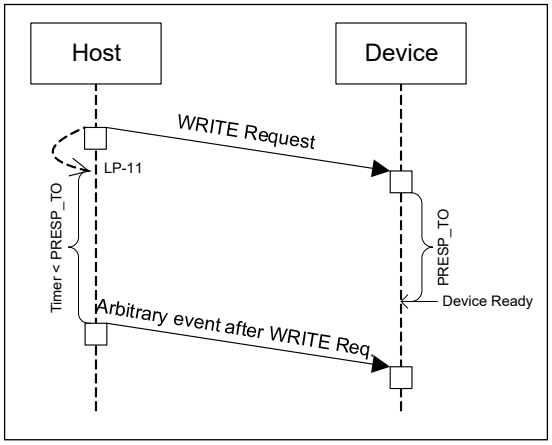


Figure 280. Timing of PRESP_TO after a Write Request (high-speed or low-power)

The `BTA_TO_CNT`, `HS_RD_TO_CNT`, `LP_RD_TO_CNT`, `HS_WR_TO_CNT`, and `LP_WR_TO_CNT` registers, configures the PRESP_TO counter. The values in these registers are measured in number of cycles of the `lanebyteclk` clock. Use these registers only in the Command mode.

Do not use these registers in the Video mode, because in Video mode for proper refreshing of the display, a rigid timing schedule must be met and it must not be broken by these or any other timeouts.

Setting a given timeout to 0, disables going into low-power-11 state, and timeout for events of that category.

The read and the write requests in high-speed mode are distinct from the read and the write requests in low-power mode. For example, on setting `HS_RD_TO_CNT` register to zero, and `LP_RD_TO_CNT` register to a non-zero value, then a generic read with no parameters does not activate the PRESP_TO counter in high-speed, but it activates the PRESP_TO counter in low-power.

52.2.8.2 Errors Raised by Timeouts

The MIPI DSI controller implements a set of timers and conditions to notify the errors. The MIPI DSI controller has a set of registers to control the timers used to determine if a timeout has occurred. It also contains a set of interruption status registers that are cleared upon a read operation. Optionally, these registers also trigger an interrupt signal when an error occurs within the DSI connection.

When a high-speed transmission timeout occurs, the `to_hs_tx` field of the `INT_ST1` register is set, rising an interrupt pin. The timeout is configured in the `hstx_to_cnt` field of the `TO_CNT_CFG` register. A 16-bit counter measures the time during which the high-speed mode is active. If that counter reaches the value defined by the `hstx_to_cnt` field, the `to_hs_tx` field is asserted and an internal soft reset is generated to the MIPI DSI controller.

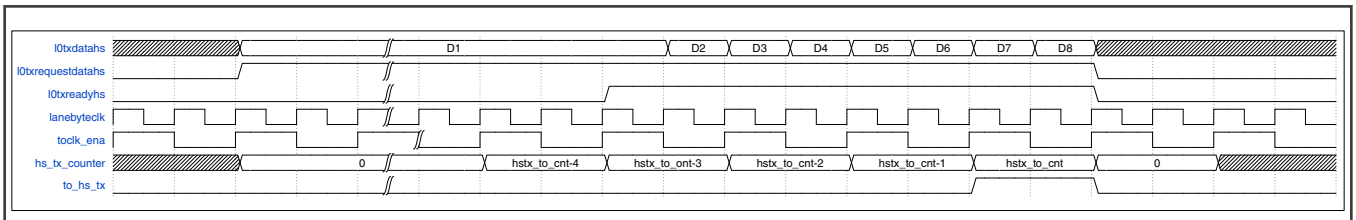


Figure 281. High-Speed Transmission Errors Raised by Timeouts

The `to_lp_rx` field of the `INT_ST1` register is set to flag the contention detection when a low-power reception timeout occurs, rising an interrupt pin. The timeout is configured in the `lprx_to_cnt` field of the `TO_CNT_CFG` register. A 16-bit counter measures the time during which the low-power reception is active (that is the rising edge of `phyturnrequest` signal until the falling edge of the `phydirection` signal).

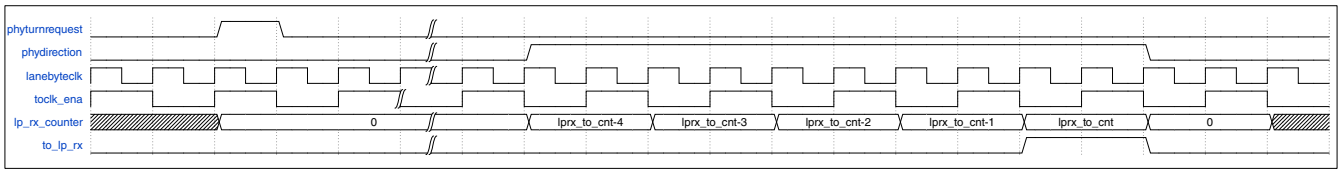


Figure 282. Low-Power Reception Errors Raised by Timeouts

If that counter reaches the value defined by the `lprx_to_cnt` field, the `to_lp_rx` field is set and an internal soft reset is generated to the MIPI DSI controller.

The value configured in the `to_clk_division` field of the `CLKMGR_CFG` register defines the time units for these 16-bit counters. The configured `to_clk_division` register field value defines the time unit for the timeout limits using the Lane byte clock (`lanebyteclk`) as input. This mechanism increases the range to define these limits.

52.2.9 Error Control

The `INT_ST0` and `INT_ST1` registers report error conditions, and trigger an interrupt pin to inform the system about the occurrence of errors.

The MIPI DSI controller has one interrupt pin, that is set high when an error occurs in either the `INT_ST0` or the `INT_ST1` register.

Program the `INT_MSK0` and `INT_MSK1` mask registers, to mask the triggering of the interrupt pin. By default, all errors are masked. Setting any bit of these registers to 1, enables the interrupt for a specific error. The error bit is always set in the respective INT_STn register. The `INT_ST0` and `INT_ST1` registers are always cleared after a read operation. A read operation of all registers that caused the interrupt, clears the interrupt pin.

The interrupt force registers (`INT_FORCE0` and `INT_FORCE1`) are used for test purposes. These registers trigger the interrupt events individually without the need to activate the conditions that trigger the interrupt sources; this is because it is extremely complex to generate the stimuli for that purpose.

This feature also facilitates the development and testing of the software associated with the interrupt events. Setting any bit of these registers to 1 triggers the corresponding interrupt.

Figure 283 illustrates the location of some of the errors. For more information about the causes of the errors and the action to be taken, see Error Handling

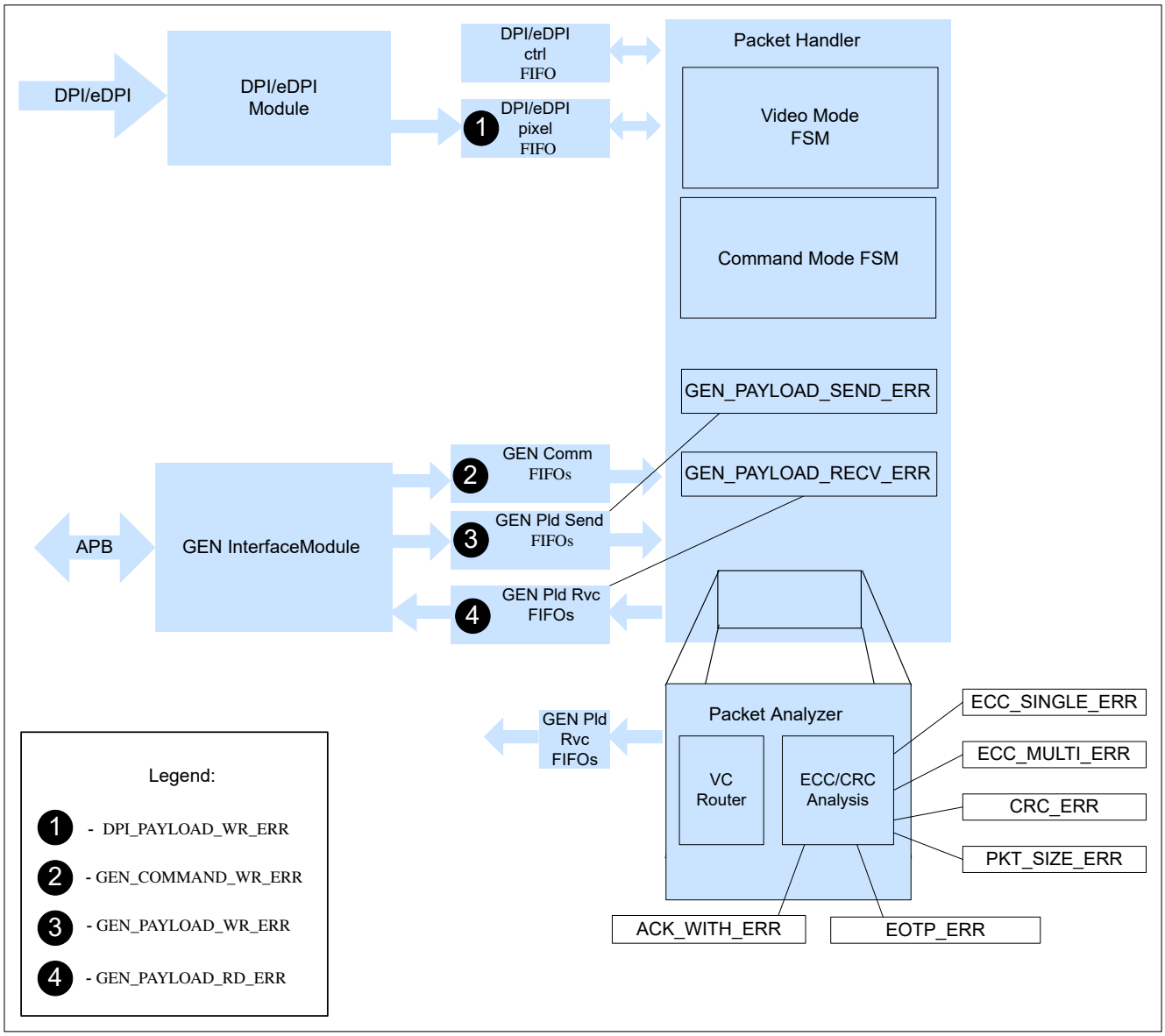


Figure 283. Error Sources

52.2.10 Error Handling

This appendix explains the conditions that trigger the interrupts and provides suggestions to recover from the error states.

52.2.10.1 Causes of Errors and Recovery

The interrupts triggered by the `INT_ST0` or `INT_ST1` register bits are error conditions that can occur only at the debug stage. Error Handling explains the reasons that set off these interrupts and also explains how to recover from these interrupts.

NOTE

- Though most of the interrupts are only expected to occur during the debug process and must be analyzed and corrected during the design stage, some of the interrupts might occur if the link is exposed to external factors that induce noise in the link.
- In addition to the causes and suggestions for recovery listed in Table 506 there can be other causes and methods of handling each interrupt.

Table 506. Error Cause and Recovery

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
INT_ST0 Register Bits			
20	dphy_errors_4	The D-PHY reports the LP1 contention error. The D-PHY host detects the contention while trying to drive the line high.	Recover the D-PHYs from contention. Reset the MIPI DSI controller and transmit the packets again. If this error is recurrent, analyze the connectivity between the Host and the Device carefully.
19	dphy_errors_3	D-PHY reports the LP0 contention error. The D-PHY Host detects the contention while trying to drive the line low.	Recover the D-PHYs from contention. Reset the MIPI DSI controller and transmit the packets again. If this error is recurrent, analyze the connectivity between the Host and the Device carefully.
18	dphy_errors_2	The D-PHY reports the False Control error. The D-PHY detects an incorrect line state sequence in lane 0 lines.	Device does not behave as expected. Communication with the Device is not properly established. This is an unrecoverable error. Reset the MIPI DSI controller and the D-PHY. If this error is recurrent, analyze the behavior of the Device.
17	dphy_errors_1	The D-PHY reports the LPDT Error. The D-PHY detects that the LPDT did not match a multiple of 8 bits.	The data reception is not reliable. The D-PHY recovers but the received data from the Device might not be reliable. It is recommended to reset the MIPI DSI controller and repeat the RX transmission.
16	dphy_errors_0	The D-PHY reports the Escape Entry Error. The D-PHY does not recognize the received Escape Entry Code.	The D-PHY Host does not recognize the Escape Entry Code. The Transmission is ignored. The PHY Host recovers but the system must repeat the RX reception.
15	ack_with_err_15	This error is directly retrieved from Acknowledge with Error packet. The Device detected a protocol violation in the reception.	See the display documentation. When this error is active, the Device should have another read-back command that reports additional information about this error. Read the additional information and take appropriate actions.
14	ack_with_err_14	The Acknowledge with Error packet contains this error. The Device chooses to use this bit for error report.	See the Device documentation regarding possible reasons for this error and take appropriate actions.
13	ack_with_err_13	The Acknowledge with Error packet contains this error. The Device reports that the transmission length does not match the packet length.	Possible reason for this error is the presence of multiple errors in the packet header (more than two), therefore, the error detection fails and the Device does not discard the packet. In this case, the packet header is corrupt and can cause decoding mismatches.

Table continues on the next page...

Table 506. Error Cause and Recovery (continued)

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
			Transmit the packets again. If this error is recurrent, analyze the connectivity between the Host and the Device carefully.
12	ack_with_err_12	The Acknowledge with Error packet contains this error. The Device does not recognize the VC ID in at least one of the received packets.	Check the Device capabilities and configure the Host to properly address the Device VC ID. Repeat the transmission.
11	ack_with_err_11	The Acknowledge with Error packet contains this error. The Device does not recognize the data type of at least one of the received packets.	Check the Device capabilities. It is possible that there are some packets that are not supported by the Device. Repeat the transmission.
10	ack_with_err_10	The Acknowledge with Error packet contains this error. The Device detects the CRC errors in at least one of the received packets.	Some of the long packets, transmitted after the last Acknowledge request, might contain the CRC errors in the payload. If the payload content is critical, transmit the packets again. If this error is recurrent, analyze the connectivity between the Host and the Device carefully.
9	ack_with_err_9	The Acknowledge with Error packet contains this error. The Device detects multi-bit ECC errors in at least one of the received packets.	The Device does not interpret the packets transmitted after the last Acknowledge request. If the packets are critical, transmit the packets again. If this error is recurrent, analyze the connectivity between the Host and the Device carefully.
8	ack_with_err_8	The Acknowledge with Error packet contains this error. The Device detects and corrects the one bit ECC error in at least one of the received packets.	No action is required. The Device acknowledges the packet. If this error is recurrent, analyze the signal integrity or the noise conditions of the link.
7	ack_with_err_7	The Acknowledge with Error packet contains this error. The Device detects the Line Contention through LP0/LP1 detection.	This error might corrupt the low-power data reception and transmission. Ignore the packets and transmit them again. The Device recovers automatically. If this error is recurrent, check the Device capabilities and the connectivity between the Host and Device. For more information, see section 7.2.1 of the *MIPI DSI Specification 1.1*.
6	ack_with_err_6	The Acknowledge with Error packet contains this error. The Device detects the False Control Error.	The device detects one of the following: • The low-power-10 (low-power request) is not followed by the remainder of a valid escape or turnaround sequence. • The low-power-01 (high-speed request) is not followed by a bridge state (low-power-00).

Table continues on the next page...

Table 506. Error Cause and Recovery (continued)

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
			The PHY communications are corrupted. This error is unrecoverable. Reset the MIPI DSI controller and the PHY. For more information, see section 7.1.6 of the *MIPI DSI Specification 1.1*.
5	ack_with_err_5	The Acknowledge with Error packet contains this error. The display timeout counters for a high-speed reception and low-power transmission expire.	It is possible that the Host and Device timeout counters are not correctly configured. The Device HS_TX timeout must be shorter than the Host HS_RX timeout. Host LP_RX timeout must be longer than the Device LP_TX timeout. Check and confirm that the Host configuration is consistent with the Device specifications. This error is automatically recovered, although there is no guarantee that all the packets in the transmission or reception are complete. For more information about this error, see section 7.2.2 of the *MIPI DSI Specification 1.1*.
4	ack_with_err_4	The Acknowledge with Error packet contains this error. The Device reports that the LPDT is not aligned in an 8-bit boundary.	There is no guarantee that the Device properly receives the packets. Transmit the packets again. For more information about this error, see section 7.1.5 of the *MIPI DSI Specification*.
3	ack_with_err_3	The Acknowledge with Error packet contains this error. The Device does not recognize the Escape Mode Entry command.	The Device does not recognize the Escape Mode Entry code. Check the Device capability. For more information about this error, see section 7.1.4 of the *MIPI DSI Specification*. Repeat the transmission to the Device.
2	ack_with_err_2	The Acknowledge with Error packet contains this error. The Device detects the high-speed transmission did not end in an 8-bit boundary when the EoT sequence is detected.	There is no guarantee that the Device properly received the packets. Re-transmission should be performed. Transmit the packets again. For more information about this error, see section 7.1.3 of the *MIPI DSI Specification 1.1*.
1	ack_with_err_1	The Acknowledge with Error packet contains this error. The Device detects that the SoT leader sequence is corrupted.	The Device discards the incoming transmission. Re-transmission should be performed by the Host. For more information about this error, see section 7.1.2 of the *MIPI DSI Specification 1.1*.
0	ack_with_err_0	The Acknowledge with Error packet contains this error. The Device reports that the SoT sequence is received with errors but synchronization can still be achieved.	The Device is tolerant to single bit and some multi-bit errors in the SoT sequence. Yet, the packet correctness is compromised. If the packet content is important, transmit the packets again. For additional information about this error, see section 7.1.1 of the *MIPI DSI Specification 1.1*.
INT_ST1 Register Bits			

Table continues on the next page...

Table 506. Error Cause and Recovery (continued)

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
12	gen_pld_recv_err	An overflow occurs in the Generic read FIFO.	The Read FIFO size is not correctly dimensioned for the maximum read-back packet size. Configure the Device to return the read data with a suitable size for the Host dimensioned FIFO. This error corrupts the data stored in the FIFOs. Reset the MIPI DSI controller and repeat the read procedure.
11	gen_pld_rd_err	An underflow occurs in the Generic read FIFO.	System does not wait for the read procedure to end and starts retrieving the data from the FIFO. The read data is requested before it is fully received. This error corrupts the data. Reset the MIPI DSI controller and repeat the read procedure. Check that the read procedure is completed before reading the data through the APB interface.
10	gen_pld_send_err	An underflow occurs in the Generic write payload FIFO.	The system writes the packet header before the respective packet payload is completely loaded into the payload FIFO. This error is unrecoverable, the transmitted packet is corrupted. Reset the MIPI DSI controller and repeat the write procedure.
9	gen_pld_wr_err	An overflow occurs in the Generic write payload FIFO.	The payload FIFO size is not correctly dimensioned to store the total payload of a long packet. Data stored in the FIFOs is corrupted. Reset the MIPI DSI controller and repeat the write procedure.
8	gen_cmd_wr_err	An overflow occurs in the Generic command FIFO.	The command FIFO size is not correctly dimensioned to store the total headers of a burst of packets, as per the FIFO dimensioning recommendations. Data stored in the FIFOs is corrupted. Reset the MIPI DSI controller and repeat the write procedure.
7	dpi_pld_wr_err	An overflow occurs in the DPI pixel payload FIFO.	The controller FIFO dimensions are not correctly set up for the operating resolution. Check the Video mode configuration registers. They must be consistent with the DPI video resolution. This error corrupts the pixel data sequence. Reset the MIPI DSI controller and re-initiate the Video transmission.
6	eopt_err	Host receives a transmission that does not end with an End of Transmission packet.	This error is not critical for the data integrity of the received packets. Check whether the Device supports the transmission of EoTp packets.
5	pkt_size_err	Host receives a transmission that does not end in the expected by boundaries.	The integrity of the received data cannot be guaranteed. Reset the MIPI DSI controller and repeat the read procedure.
4	crc_err	Host reports that a received long packet has a CRC error in its payload.	The received payload data is corrupted. Reset the MIPI DSI controller and repeat the read procedure. If

Table continues on the next page...

Table 506. Error Cause and Recovery (continued)

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
			this error is recurrent, check the DSI connectivity link for the noise levels.
3	ecc_multi_err	Host reports that a received packet contains multiple ECC errors.	The received packet is corrupted. The MIPI DSI controller ignores all the following packets. The MIPI DSI controller must repeat the read procedure.
2	ecc_single_err	Host reports that a received packet contains a single bit error.	This error is not critical because the MIPI DSI controller can correct the error and properly decode the packet. If this error is recurrent, check the DSI connectivity link for signal integrity and noise levels.
1	to_lp_rx	Host reports that the configured timeout counter for the low-power reception has expired.	Once the configured timeout counter ends, the MIPI DSI controller automatically resets and recovers to normal operation. Packet transmissions happening during this event are lost. If this error is recurrent, check the timer configuration for any issue. This timer must be greater than the maximum low-power transmission generated by the Device.
0	to_hs_tx	Host reports that the configured timeout counter for the high-speed transmission has expired.	Once the configured timeout counter ends, the MIPI DSI controller automatically resets and recovers to normal operation. Packet transmissions happening during this event are lost. If this error is recurrent, check the timer configuration for any issue. This timer must be greater than the maximum high-speed transmission bursts generated by the Host.

52.2.11 Clocking

The MIPI DSI host controller includes the following clock signals:

Table 507. MIPI DSI Host Controller Clocks

Clock Name	Description
pclk	APB Bus Clock
dpipclk	Pixel clock of DPI interface
scanclk	Scan clock
cfg_clk	PHY configuration clock used for the initialization of the PHY. Also used for exiting ULPS state.
scanbyteclk	PHY scan clock used for lanebyteclk.
scantxclkesc	PHY scan clock used for txclkesc.
scanrxclkesc	PHY scan clock used for rxclkesc.
refclk	PHY reference clock

Table continues on the next page...

Table 507. MIPI DSI Host Controller Clocks (continued)

Clock Name	Description
clkp	Positive D-PHY Differential Clock Line
clkn	Negative D-PHY Differential Clock Line
pll_scanclk	PLL scan clock

52.2.12 Reset

The MIPI DSI controller includes the following reset signal:

Table 508. MIPI DSI controller reset signals

Reset signal	Description
presetn	APB Asynchronous Reset Signal. This signal acts as a global reset.

52.2.13 Interrupts

Table 509. MIPI DSI controller interrupts

Name	Description
interrupt	Interruptions Signal. Synchronous to pclk.

52.3 External signals

This module has no external signals.

52.4 Memory Map and register definition

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

This section includes the MIPI DSI module memory map and detailed descriptions of all registers.

52.4.1 Static and Dynamic Registers

The MIPI DSI controller has the following two kinds of registers:

- **Static Registers**: Static registers can be configured only when the MIPI DSI controller is in reset state.
- **Dynamic Registers**: Dynamic registers can be configured when needed, and do not require the MIPI DSI controller to be idle or in reset state.

Table 510 lists the general registers of the MIPI DSI controller.

Table 510. General Registers

Register	Address	Type
VERSION	0x0	Static
PWR_UP	0x4	Static
CLKMGR_CFG	0x8	Static

Table continues on the next page...

Table 510. General Registers (continued)

Register	Address	Type
PCKHDL_CFG	0x2C	Static
MODE_CFG	0x34	Dynamic
CMD_MODE_CFG	0x68	Static
CMD_PKT_STATUS	0x74	Dynamic
TO_CNT_CFG	0x78	Static
BTA_TO_CNT	0x8C	Static
LPCLK_CTRL	0x94	Static
DSC_PARAMETER	0xF0	Static
HS_RD_TO_CNT	0x7C	Static
HS_WR_TO_CNT	0x84	Static
LP_RD_TO_CNT	0x80	Static
LP_WR_TO_CNT	0x88	Static

Table 511 lists the registers related to PHY.

Table 511. Register Related to PHY

Register	Address	Type
PHY_RSTZ	0xA0	Dynamic
PHY_IF_CFG	0xA4	Static
PHY_ULPS_CTRL	0xA8	Dynamic
PHY_TX_TRIGGERS	0xAC	Dynamic
PHY_STATUS	0xB0	Dynamic
PHY_TST_CTRL0	0xB4	Dynamic
PHY_TST_CTRL1	0xB8	Dynamic
PHY_CAL	0xCC	Dynamic
PHY_MODE	0xFC	Static

Table 512 lists the registers related to interrupt.

Table 512. Registers Related to Interrupt

INT_ST0	0xBC	Dynamic
INT_ST1	0xC0	Dynamic
INT_MSK0	0xC4	Dynamic
INT_MSK1	0xC8	Dynamic
INT_FORCE0	0xD8	Dynamic
INT_FORCE1	0xDC	Dynamic

Table 513 lists the registers related to Generic interface.

Table 513. Registers Related to Generic Interface

Register	Address	Type
GEN_VCID	0x30	Static
GEN_HDR	0x6c	Dynamic
GEN_PLD_DATA	0x70	Dynamic

Table 514 lists the registers related to DPI.

Table 514. Registers Related to DPI

Register	Address	Type
DPI_VCID	0xC	Static
DPI_COLOR_CODING	0x10	Static
DPI_CFG_POL	0x14	Static
DPI_LP_CMD_TIM	0x18	Static
SDF_3D	0x90	Dynamic
PHY_TMR_LPCLK_CFG	0x98	Static
PHY_TMR_CFG	0x9C	Static
PHY_TMR_RD_CFG	0xF4	Static
VID_SHADOW_CTRL	0x100	Dynamic
DPI_VCID_ACT	0x10C	Dynamic
DPI_COLOR_CODING_ACT	0x110	Static

Table continues on the next page...

Table 514. Registers Related to DPI (continued)

Register	Address	Type
DPI_LP_CMD_TIM_ACT	0x118	Static
VID_PKT_STATUS	0x168	Dynamic
SDF_3D_ACT	0x190	Dynamic

Table 515 lists the registers related to Video mode pattern generator.

Table 515. Registers Related to Video Mode Pattern Generator

Register	Address	Type
VID_MODE_CFG	0x38	Static
VID_PKT_SIZE	0x3C	Static
VID_NUM_CHUNKS	0x40	Static
VID_NULL_SIZE	0x44	Static
VID_HSA_TIME	0x48	Static
VID_HBP_TIME	0x4C	Static
VID_HLINE_TIME	0x50	Static
VID_VSA_LINES	0x54	Static
VID_VBP_LINES	0x58	Static
VID_VFP_LINES	0x5C	Static
VID_VACTIVE_LINES	0x60	Static
VID_MODE_CFG_ACT	0x138	Dynamic
VID_PKT_SIZE_ACT	0x13C	Dynamic
VID_NUM_CHUNKS_ACT	0x140	Dynamic
VID_NULL_SIZE_ACT	0x144	Static
VID_HSA_TIME_ACT	0x148	Dynamic
VID_HBP_TIME_ACT	0x14c	Dynamic
VID_HLINE_TIME_ACT	0x150	Dynamic
VID_VSA_LINES_ACT	0x154	Static

Table continues on the next page...

Table 515. Registers Related to Video Mode Pattern Generator (continued)

Register	Address	Type
VID_VBP_LINES_ACT	0x15C	Dynamic
VID_VFP_LINES_ACT	0x158	Dynamic
VID_VACTIVE_LINES_ACT	0x160	Dynamic

52.4.2 MIPI_DSI register descriptions

52.4.2.1 MIPI_DSI memory map

MIPI DSI Registers

MIPI_DSI base address: 4AE1_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Core Version (VERSION)	32	R	3134_312A
4	Power Up (PWR_UP)	32	RW	0000_0000
8	Clock Divider Configuration (CLKMGR_CFG)	32	RW	0000_0000
C	DPI Virtual Channel ID (DPI_VCID)	32	RW	0000_0000
10	DPI color coding (DPI_COLOR_CODING)	32	RW	0000_0000
14	DPI Signals Polarity Configuration (DPI_CFG_POL)	32	RW	0000_0000
18	DPI Low-power Commands Timing (DPI_LP_CMD_TIM)	32	RW	0000_0000
2C	PCKHDL Configuration (PCKHDL_CFG)	32	RW	0000_0000
30	Generic Interface Virtual Channel ID (GEN_VCID)	32	RW	0000_0000
34	Mode Configuration (MODE_CFG)	32	RW	0000_0001
38	Video Mode Configuration (VID_MODE_CFG)	32	RW	0000_0000
3C	Video Packet Size (VID_PKT_SIZE)	32	RW	0000_0000
40	Video Number of Chunks (VID_NUM_CHUNKS)	32	RW	0000_0000
44	Video Null Packets Size (VID_NULL_SIZE)	32	RW	0000_0000
48	Video HSA Time (VID_HSA_TIME)	32	RW	0000_0000
4C	Video HBP Time (VID_HBP_TIME)	32	RW	0000_0000
50	Video Line Time (VID_HLINE_TIME)	32	RW	0000_0000
54	VSA Period (VID_VSA_LINES)	32	RW	0000_0000
58	VBP Period (VID_VBP_LINES)	32	RW	0000_0000
5C	VFP Period (VID_VFP_LINES)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
60	Vertical Resolution of Video (VID_VACTIVE_LINES)	32	RW	0000_0000
68	Command Mode Configuration (CMD_MODE_CFG)	32	RW	0000_0000
6C	Generic Interface Header Packet Configuration (GEN_HDR)	32	RW	0000_0000
70	Generic Interface Payload Data (GEN_PLD_DATA)	32	RW	0000_0000
74	Command Packet Status (CMD_PKT_STATUS)	32	R	0005_0015
78	Timeout Counter Configuration (TO_CNT_CFG)	32	RW	0000_0000
7C	Peripheral Response Timeout after High-speed Read Operations (HS_RD_TO_CNT)	32	RW	0000_0000
80	Peripheral Response Timeout after Low-power Read Operations (LP_RD_TO_CNT)	32	RW	0000_0000
84	Peripheral Response Timeout after High-speed Write Operations (HS_WR_TO_CNT)	32	RW	0000_0000
88	Peripheral Response Timeout after Low-power Write Operations (LP_WR_TO_CNT)	32	RW	0000_0000
8C	Peripheral Response Timeout after Bus Turnaround Completion (BTA_TO_CNT)	32	RW	0000_0000
90	3D Control Information for VSS Packets in Video Mode (SDF_3D)	32	RW	0000_0000
94	LP Clock Control (LPCLK_CTRL)	32	RW	0000_0000
98	D-PHY Clock Lanes LP2HS and HS2LP Time Configuration (PHY_TMR_LPCLK_CFG)	32	RW	0000_0000
9C	D-PHY Data Lanes LP2HS and HS2LP Time Configuration (PHY_TMR_CFG)	32	RW	0000_0000
A0	D-PHY Reset Control (PHY_RSTZ)	32	RW	0000_0000
A4	PHY Interface Configuration (PHY_IF_CFG)	32	RW	0000_0003
A8	D-PHY ULPS Control (PHY_ULPS_CTRL)	32	RW	0000_0000
AC	D-PHY Trigger Transmission Control (PHY_TX_TRIGGERS)	32	RW	0000_0000
B0	D-PHY Status (PHY_STATUS)	32	R	0000_1540
B4	D-PHY Test Control 0 (PHY_TST_CTRL0)	32	RW	0000_0001
B8	D-PHY Test Control 1 (PHY_TST_CTRL1)	32	RW	0000_0000
BC	Interrupt Status 0 (INT_ST0)	32	R	0000_0000
C0	Interrupt Status 1 (INT_ST1)	32	R	0000_0000
C4	Interrupt Mask 0 (INT_MSK0)	32	RW	0000_0000
C8	Interrupt Mask 1 (INT_MSK1)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
CC	D-PHY Skew Calibration (PHY_CAL)	32	RW	0000_0000
D8	Interrupt Force 0 (INT_FORCE0)	32	RW	0000_0000
DC	Interrupt Force Register 1 (INT_FORCE1)	32	RW	0000_0000
F0	Display Stream Compression Configuration (DSC_PARAMETER)	32	RW	0000_0000
F4	PHY Read Command Time Configuration (PHY_TMR_RD_CFG)	32	RW	0000_0000
100	Video Shadow Control (VID_SHADOW_CTRL)	32	RW	0000_0000
10C	DPI Virtual Channel ID Value (DPI_VCID_ACT)	32	R	0000_0000
110	DPI Color Coding Value (DPI_COLOR_CODING_ACT)	32	R	0000_0000
118	DPI Low-power Commands Timing Value (DPI_LP_CMD_TIM_ACT)	32	R	0000_0000
138	Video Mode Configuration Value (VID_MODE_CFG_ACT)	32	R	0000_0000
13C	Video Packet Size Value (VID_PKT_SIZE_ACT)	32	R	0000_0000
140	Video Number Chunks Value (VID_NUM_CHUNKS_ACT)	32	R	0000_0000
144	Video Null Packets Size Value (VID_NULL_SIZE_ACT)	32	R	0000_0000
148	Video HSA Time Value (VID_HSA_TIME_ACT)	32	R	0000_0000
14C	Video HBP Time Value (VID_HBP_TIME_ACT)	32	R	0000_0000
150	Video Line Time Value (VID_HLINE_TIME_ACT)	32	R	0000_0000
154	VSA Period Value (VID_VSA_LINES_ACT)	32	R	0000_0000
158	VBP Period Value (VID_VBP_LINES_ACT)	32	R	0000_0000
15C	VFP Period Value (VID_VFP_LINES_ACT)	32	R	0000_0000
160	Vertical Resolution of Video Value (VID_VACTIVE_LINES_ACT)	32	R	0000_0000
168	Video Packet Status (VID_PKT_STATUS)	32	R	0001_0005
190	3D Control Information Value (SDF_3D_ACT)	32	R	0000_0000

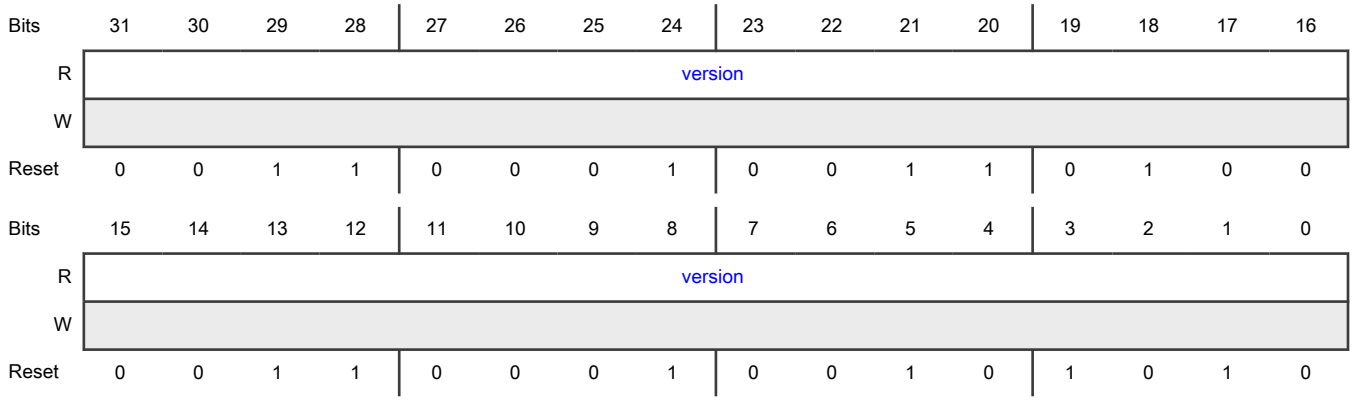
52.4.2.2 Core Version (VERSION)

Contains the version of the DSI host controller.

Offset

Register	Offset
VERSION	0h

Diagram



Fields

Field	Description
31-0 version	Version Indicates the version of the MIPI DSI host.

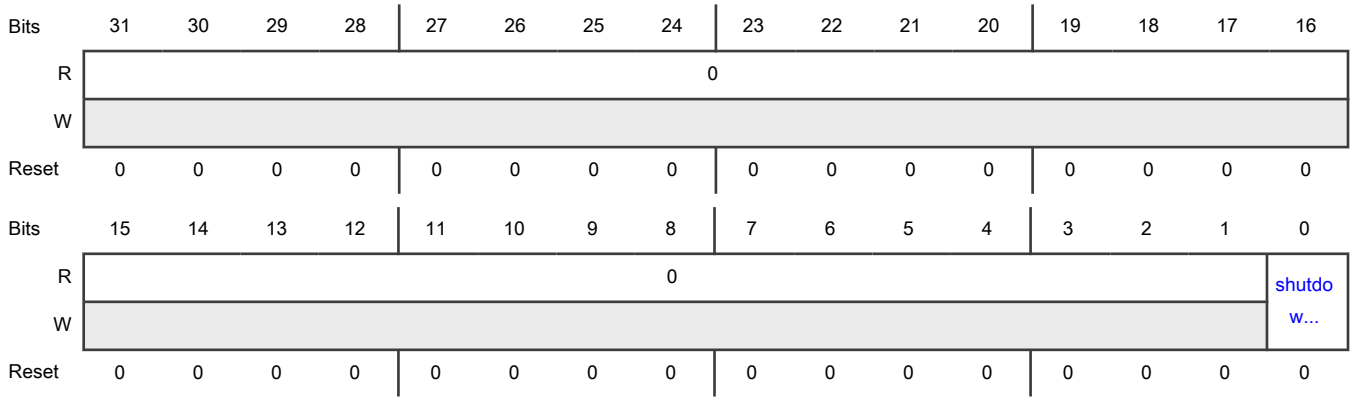
52.4.2.3 Power Up (PWR_UP)

Controls the power up of the controller.

Offset

Register	Offset
PWR_UP	4h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 shutdownz	Reset or Power up Configures the controller either to power up or to reset. 0 - Reset the controller 1 - Power up the controller

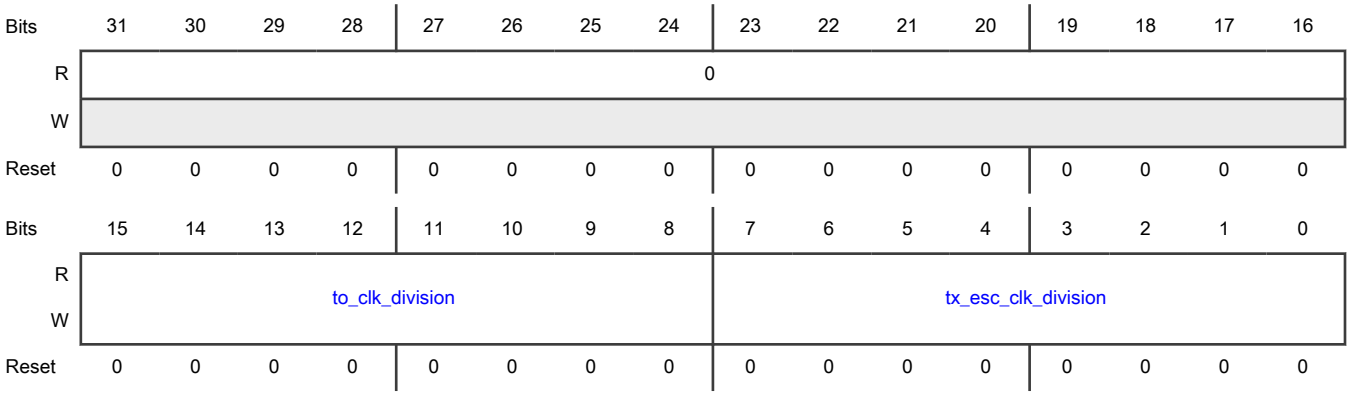
52.4.2.4 Clock Divider Configuration (CLKMGR_CFG)

Configures the factor for internal dividers to divide lanebyteclk for timeout purposes.

Offset

Register	Offset
CLKMGR_CFG	8h

Diagram



Fields

Field	Description
31-16 —	Reserved and read as zero.
15-8 to_clk_division	Time Out Clock Division Indicates the division factor for the Time Out clock used as the timing unit in the configuration of high-speed to low-power and low-power to high-speed transition error.
7-0	Transmit Escape Clock Division

Table continues on the next page...

Table continued from the previous page...

Field	Description
tx_esc_clk_division	This field indicates the division factor for the TX Escape clock source (lanebyteclk). The values 0 and 1 stop the TX_ESC clock generation.

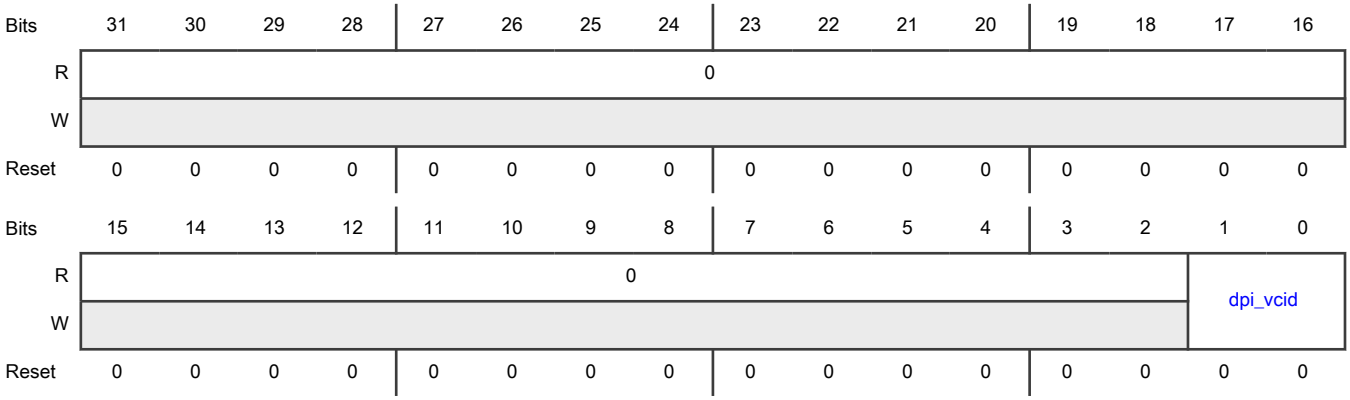
52.4.2.5 DPI Virtual Channel ID (DPI_VCID)

Configures the Virtual Channel ID for DPI traffic.

Offset

Register	Offset
DPI_VCID	Ch

Diagram



Fields

Field	Description
31-2 —	Reserved and read as zero.
1-0 dpi_vcid	DPI Virtual Channel ID Configures the DPI virtual channel ID that is indexed to the Video mode packets.

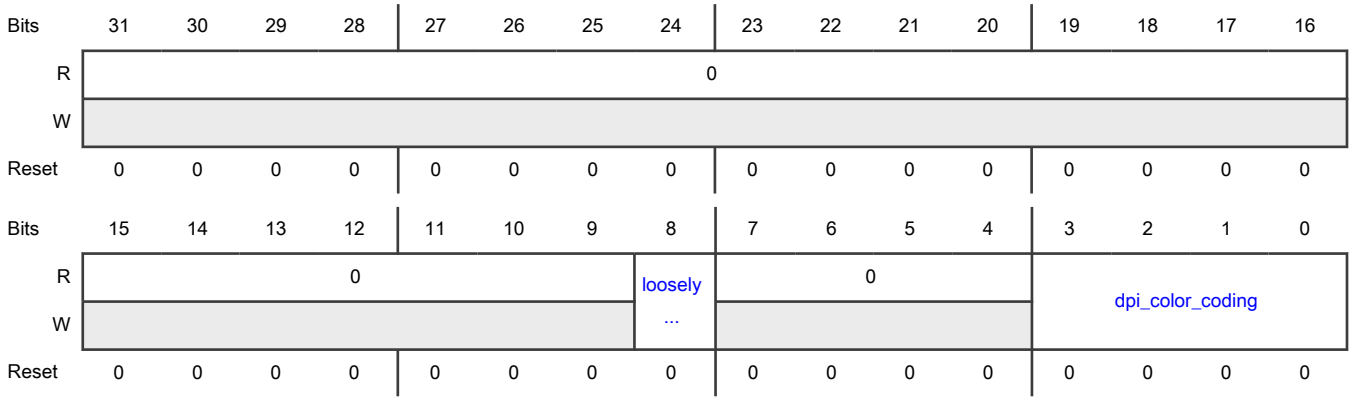
52.4.2.6 DPI color coding (DPI_COLOR_CODING)

Configures DPI color coding.

Offset

Register	Offset
DPI_COLOR_CODING	10h

Diagram



Fields

Field	Description
31-9 —	Reserved and read as zero.
8 loosely18_en	Loosely 18 Enable Activates loosely packed variant to 18-bit configurations. 0 - Disable 1 - Enable
7-4 —	Reserved and read as zero.
3-0 dpi_color_coding	DPI Color Coding Configures the DPI color for Video Mode/eDPI Command Mode coding as follows: 0000 - 16-bit configuration 1 0001 - 16-bit configuration 2 0010 - 16-bit configuration 3 0011 - 18-bit configuration 1 0100 - 18-bit configuration 2 0101 - 24-bit 0110 - 20-bit YCbCr 4:2:2 loosely packed / Reserved for eDPI Command Mode 0111 - 24-bit YCbCr 4:2:2 / Reserved for eDPI Command Mode 1000 - 16-bit YCbCr 4:2:2 / Reserved for eDPI Command Mode 1001 - 30-bit - DSC_ENC 10bit / Reserved for eDPI Command Mode 1010 - 36-bit / Reserved for eDPI Command Mode 1011 - 12-bit YCbCr 4:2:0 / Reserved for eDPI Command Mode

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1100 - DSC24 compressed Data 1101-1111 - Reserved

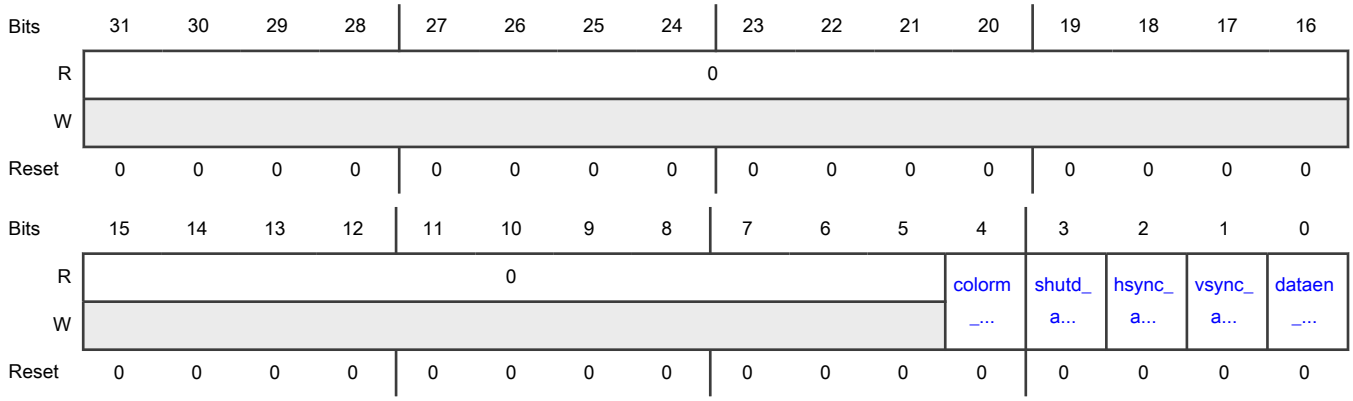
52.4.2.7 DPI Signals Polarity Configuration (DPI_CFG_POL)

Configures the polarity of DPI signals.

Offset

Register	Offset
DPI_CFG_POL	14h

Diagram



Fields

Field	Description
31-5 —	Reserved and read as zero.
4 colorm_active_low	Color Mode Pin as Active Low Configures the color mode pin (dpicolorm) as active low. 0 - Disable 1 - Enable
3 shutd_active_low	Shutdown Pin as Active Low Configures the shutdown pin (dpishutdn) as active low. 0 - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable
2 hsync_active_low	HSYNC Enable Pin as Active Low Configures the horizontal synchronization pin (dpihsync) as active low. 0 - Disable 1 - Enable
1 vsync_active_low	VSYNC Enable Pin as Active Low Configures the vertical synchronization pin (dpivsync) as active low. 0 - Disable 1 - Enable
0 dataen_active_low	Data Enable Pin as Active Low Configures the data enable pin (dpidataen) as active low. 0 - Disable 1 - Enable

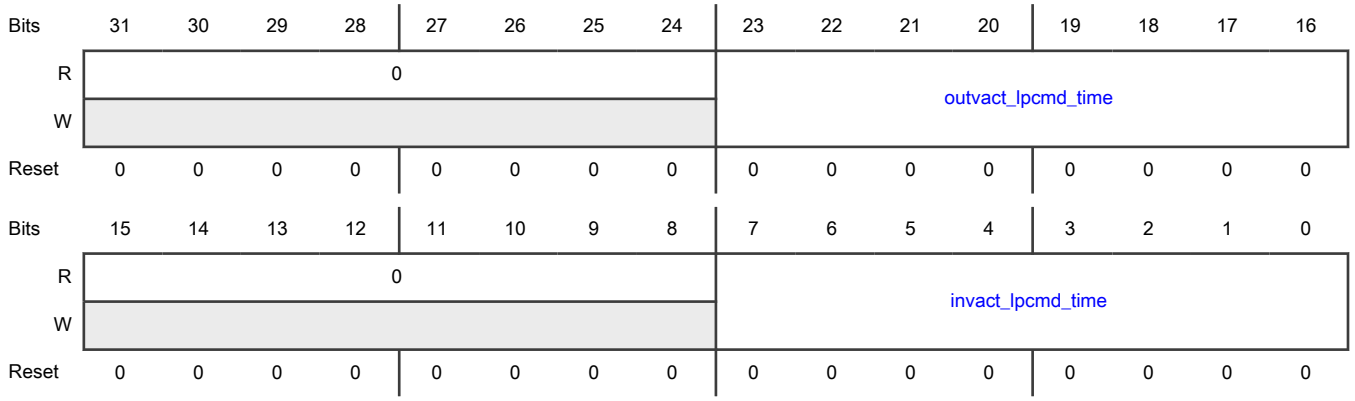
52.4.2.8 DPI Low-power Commands Timing (DPI_LP_CMD_TIM)

Configures the timing for Low-power commands sent while in video mode.

Offset

Register	Offset
DPI_LP_CMD_TIM	18h

Diagram



Fields

Field	Description
31-24 —	Reserved and read as zero.
23-16 outvact_lpcmd_time	Out VACT Low-power Command Time Use for the transmission of commands in low-power mode. Defines the size, in bytes, of the largest packet that can fit in a line during the VSA, VBP, and VFP regions.
15-8 —	Reserved and read as zero.
7-0 invact_lpcmd_time	In VACT Low-power Command Time Use for the transmission of commands in low-power mode. Defines the size, in bytes, of the largest packet that can fit in a line during the VACT region.

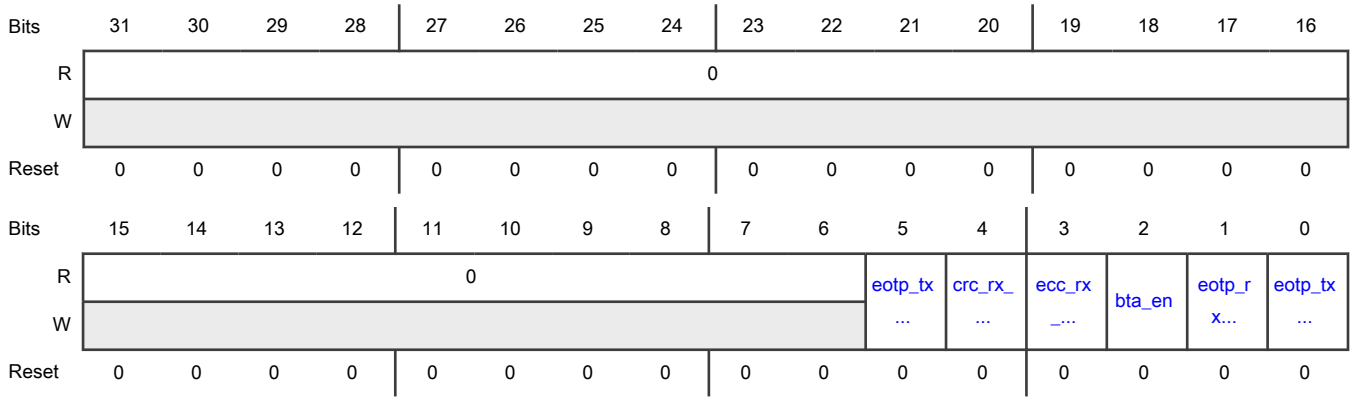
52.4.2.9 PCKHDL Configuration (PCKHDL_CFG)

Configures how EoTp, BTA, CRC and ECC are to be used, to meet peripherals characteristics

Offset

Register	Offset
PCKHDL_CFG	2Ch

Diagram



Fields

Field	Description
31-6	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
5 eotp_tx_lp_en	EoTp Transmit Low-power Enable Enables the EoTp transmission in low-power. 0 - Disable 1 - Enable
4 crc_rx_en	CRC Receive Enable Enables the CRC reception and error reporting. 0 - Disable 1 - Enable
3 ecc_rx_en	ECC Receive Enable Enables the ECC reception, error correction, and reporting. 0 - Disable 1 - Enable
2 bta_en	BTA Enable Enables the Bus Turn-Around (BTA) request. 0 - Disable 1 - Enable
1 eotp_rx_en	EoTp Receive Enable Enables the EoTp reception. 0 - Disable 1 - Enable
0 eotp_tx_en	EoTp Transmit Enable Enables the EoTp transmission in high-speed. 0 - Disable 1 - Enable

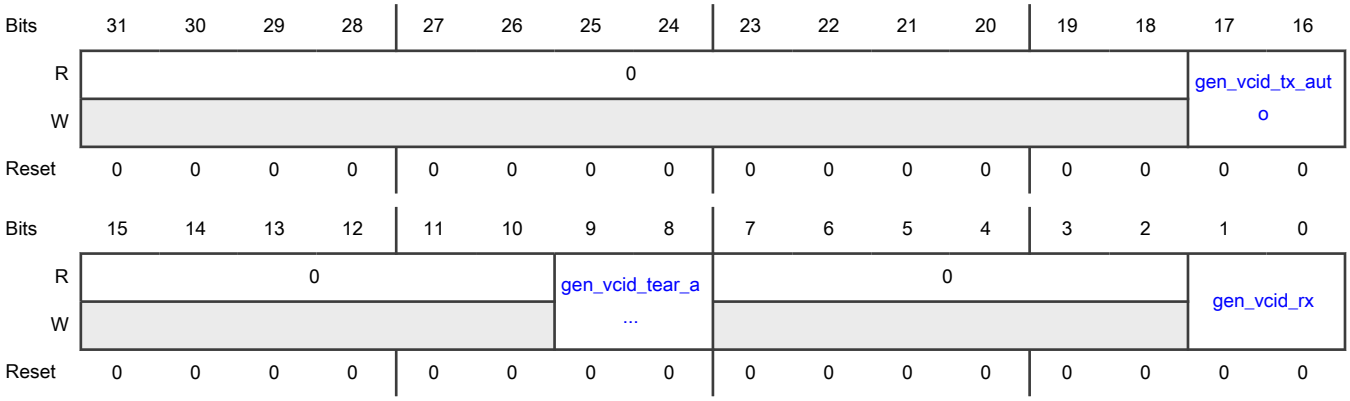
52.4.2.10 Generic Interface Virtual Channel ID (GEN_VCID)

Configures the Virtual Channel ID of READ responses to store and return to Generic interface.

Offset

Register	Offset
GEN_VCID	30h

Diagram



Fields

Field	Description
31-18 —	Reserved and read as zero.
17-16 gen_vcid_tx_auto	Generic Virtual Channel ID Transmit Indicates the Generic interface virtual channel identification where generic packet is automatically generated and transmitted.
15-10 —	Reserved and read as zero.
9-8 gen_vcid_tear_auto	Generic Virtual Channel ID Tear Effect Indicates the virtual channel identification for tear effect by hardware
7-2 —	Reserved and read as zero.
1-0 gen_vcid_rx	Generic Virtual Channel ID Receive Indicates the Generic interface read-back virtual channel identification.

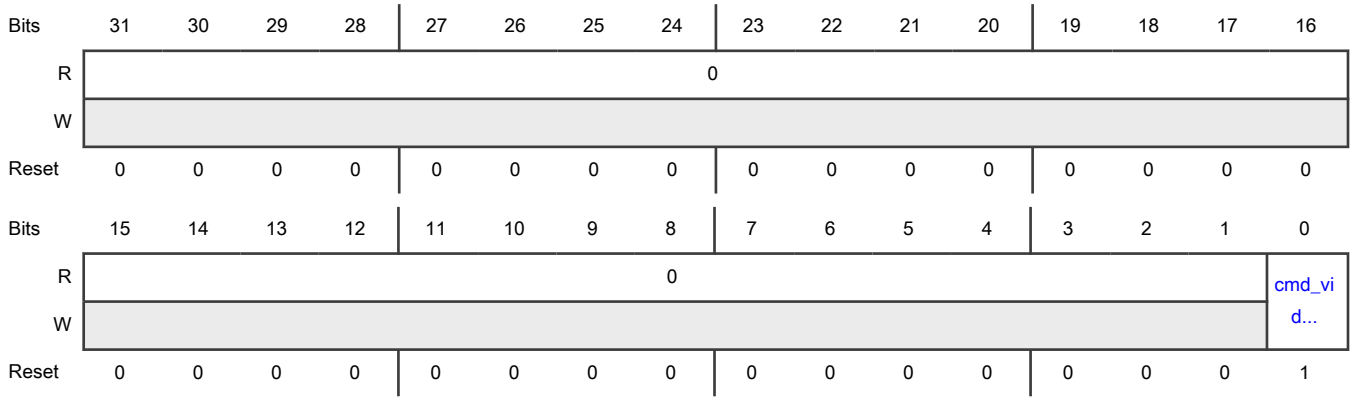
52.4.2.11 Mode Configuration (MODE_CFG)

Configures the mode of operation between Video or Command Mode. (Commands can still be sent while in video mode.)

Offset

Register	Offset
MODE_CFG	34h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 cmd_video_mode	Command/Video Mode Configures the operation mode: 0 - Video mode 1 - Command mode

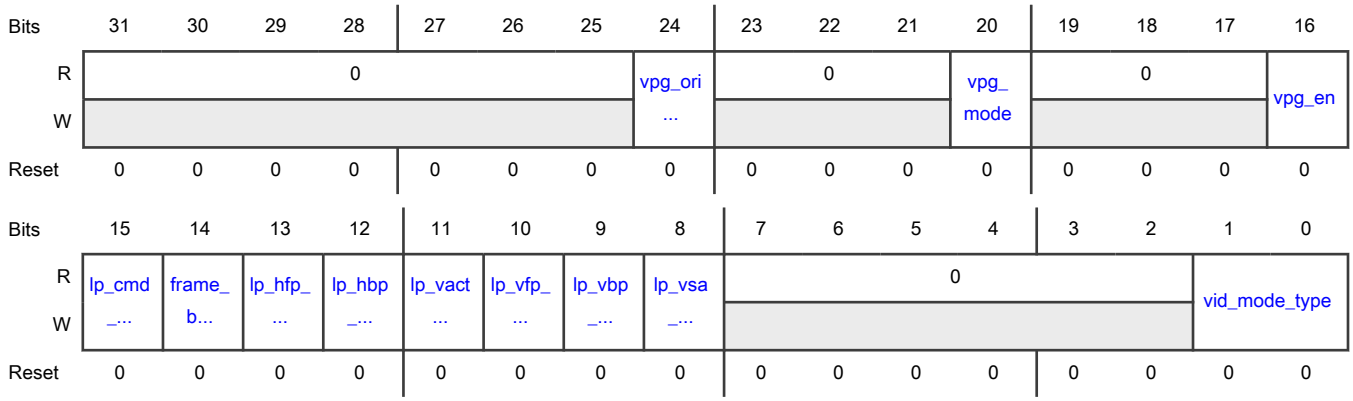
52.4.2.12 Video Mode Configuration (VID_MODE_CFG)

Configures several aspects of Video mode operation, the transmission mode, switching to low-power in the middle of a frame, enabling acknowledge and whether to send commands in low-power.

Offset

Register	Offset
VID_MODE_CFG	38h

Diagram



Fields

Field	Description
31-25 —	Reserved and read as zero.
24 vpg_orientation	VPG Orientation Indicates the color bar orientation as follows: 0 - Vertical mode 1 - Horizontal mode
23-21 —	Reserved and read as zero.
20 vpg_mode	VPG Mode Selects the pattern: 0 - Horizontal or vertical 1 - Vertical only
19-17 —	Reserved and read as zero.
16 vpg_en	VPG Enable Enables the video mode pattern generator. 0 - Disable 1 - Enable
15 lp_cmd_en	Low-power Command Enable Enables the command transmission only in low-power mode. 0 - Disable 1 - Enable
14 frame_bta_ack_en	Frame BTA Acknowledge Enable Enables the request for an acknowledge response at the end of a frame. 0 - Disable 1 - Enable
13 lp_hfp_en	Low-power HFP Enable Enables the return to low-power inside the HFP period when timing allows. 0 - Disable 1 - Enable
12	Low-power HBP Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
lp_hbp_en	Enables the return to low-power inside the HBP period when timing allows. 0 - Disable 1 - Enable
11 lp_vact_en	Low-power VACT Enable Enables the return to low-power inside the VACT period when timing allows. 0 - Disable 1 - Enable
10 lp_vfp_en	Low-power VFP Enable Enables the return to low-power inside the VFP period when timing allows. 0 - Disable 1 - Enable
9 lp_vbp_en	Low-power VBP Enable Enables the return to low-power inside the VBP period when timing allows. 0 - Disable 1 - Enable
8 lp_vsa_en	Low-power VSA Enable Enables the return to low-power inside the VSA period when timing allows. 0 - Disable 1 - Enable
7-2 —	Reserved and read as zero.
1-0 vid_mode_type	Video Mode Type Indicates the video mode transmission type as follows: 00 - Non-burst with sync pulses 01 - Non-burst with sync events 10 - Burst mode 11 - Burst mode

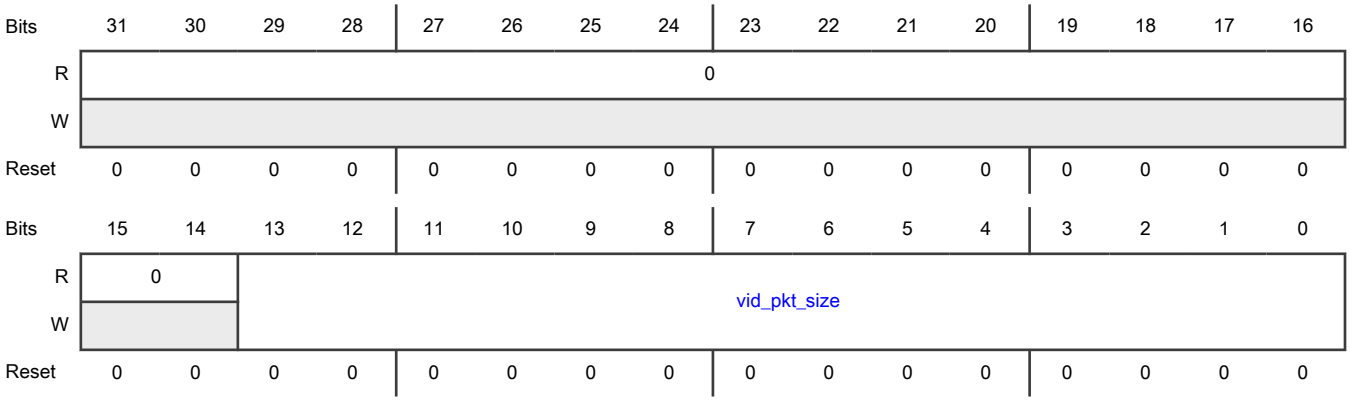
52.4.2.13 Video Packet Size (VID_PKT_SIZE)

Configures the video packet size.

Offset

Register	Offset
VID_PKT_SIZE	3Ch

Diagram



Fields

Field	Description
31-14 —	Reserved and read as zero.
13-0 vid_pkt_size	Video Packet Size Configures the number of pixels in a single video packet. - For 18-bit not loosely packed data types, this number must be a multiple of 4. - For YCbCr data types, it must be a multiple of 2, as described in the DSI specification. - For DSC24, this field configures the number of Bytes. Note that when using slices (i.e chunks) the minimum size packet size must be 3.

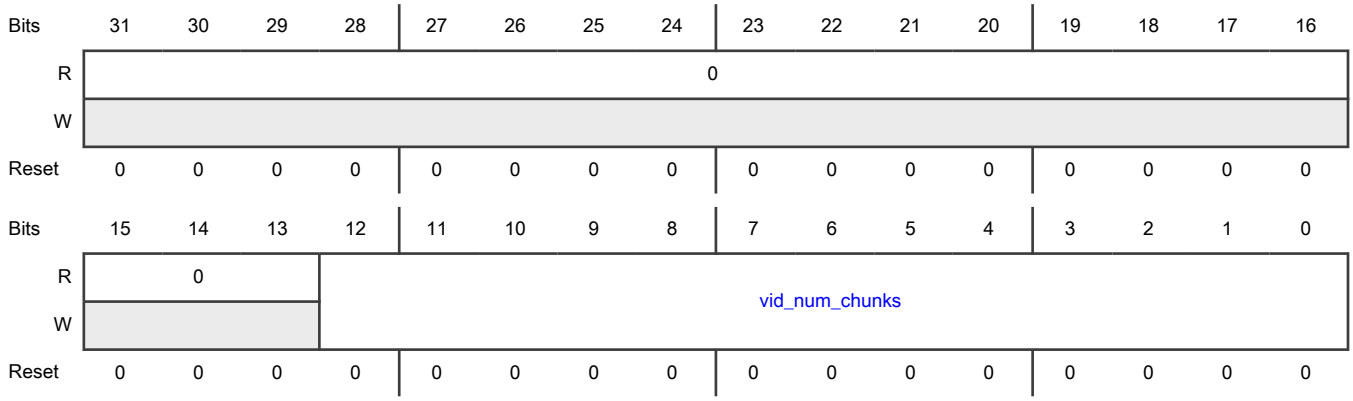
52.4.2.14 Video Number of Chunks (VID_NUM_CHUNKS)

Configures the number of chunks to use. The data in each chunk has the size provided by VID_PKT_SIZE.

Offset

Register	Offset
VID_NUM_CHUNKS	40h

Diagram



Fields

Field	Description
31-13 —	Reserved and read as zero.
12-0 vid_num_chunks	Video Number Chunks Configures the number of chunks to be transmitted during a Line period (a chunk is pair made of a video packet and a null packet). If set to 0 or 1, video line is still transmitted in a single packet. If set to 1 that packet is part of a chunk, meaning that a null packet follows it (if vid_null_size>0). Otherwise, multiple chunks are used to transmit each video line.

52.4.2.15 Video Null Packets Size (VID_NULL_SIZE)

Configures the size of null packets.

Offset

Register	Offset
VID_NULL_SIZE	44h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				vid_null_size											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-13 —	Reserved and read as zero.
12-0 vid_null_size	Video Null Size Configures the number of bytes inside a null packet. Setting to 0 disables null packets.

52.4.2.16 Video HSA Time (VID_HSA_TIME)

Configures the video HSA time.

Offset

Register	Offset
VID_HSA_TIME	48h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				vid_hsa_time											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-12 —	Reserved and read as zero.
11-0 vid_hsa_time	Video HSA Time Configures the Horizontal Synchronization Active period in lane byte clock cycles.

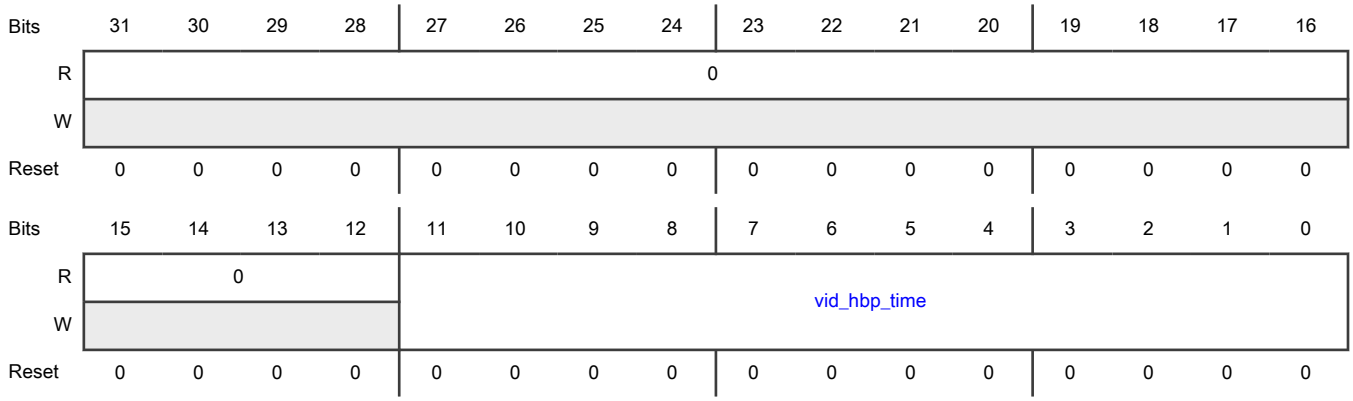
52.4.2.17 Video HBP Time (VID_HBP_TIME)

Configures the video HBP time.

Offset

Register	Offset
VID_HBP_TIME	4Ch

Diagram



Fields

Field	Description
31-12 —	Reserved and read as zero.
11-0 vid_hbp_time	Video HBP Time Configures the Horizontal Back Porch period in lane byte clock cycles.

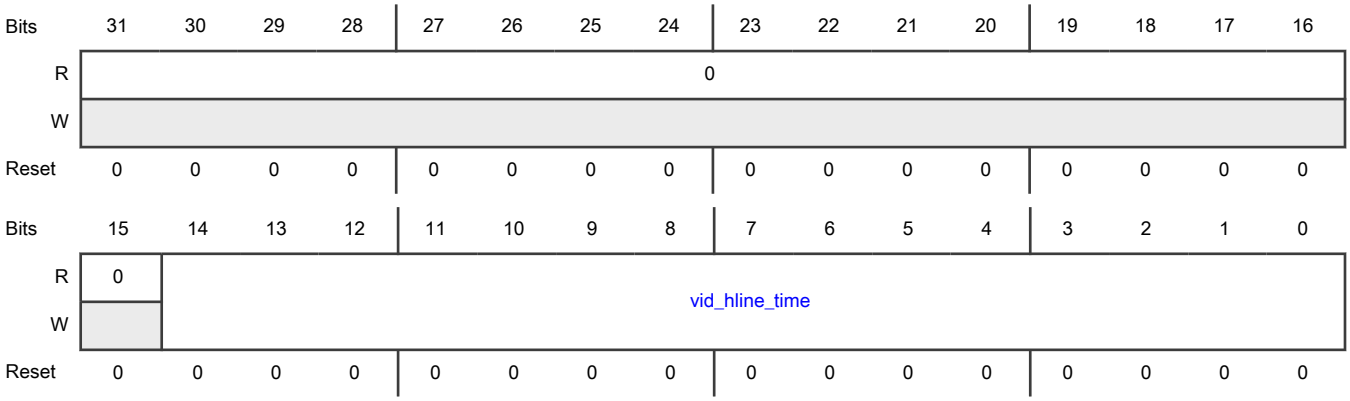
52.4.2.18 Video Line Time (VID_HLINE_TIME)

Configures the overall time for each video line.

Offset

Register	Offset
VID_HLINE_TIME	50h

Diagram



Fields

Field	Description
31-15 —	Reserved and read as zero.
14-0 vid_hline_time	Video Total Line Time Configures the size of the total line time (HSA+HBP+HACT+HFP) counted in lane byte clock cycles.

52.4.2.19 VSA Period (VID_VSA_LINES)

Configures the VSA period.

Offset

Register	Offset
VID_VSA_LINES	54h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						vsa_lines									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-10 —	Reserved and read as zero.
9-0 vsa_lines	VSA Lines Configures the Vertical Synchronization Active period measured in number of horizontal lines.

52.4.2.20 VBP Period (VID_VBP_LINES)

Configures the VBP period.

Offset

Register	Offset
VID_VBP_LINES	58h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						vbp_lines									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-10 —	Reserved and read as zero.
9-0 vbp_lines	VBP Lines Configures the Vertical Back Porch period measured in number of horizontal lines.

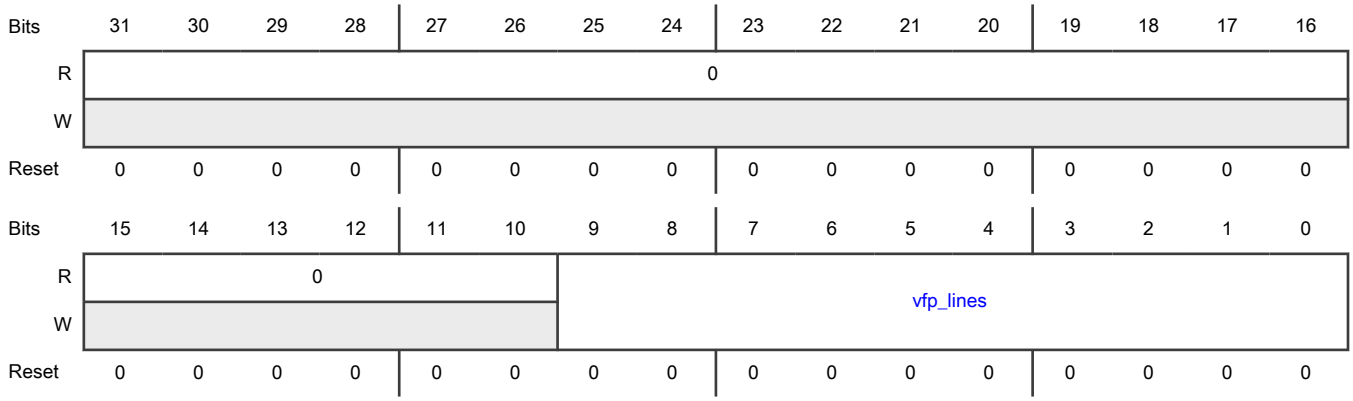
52.4.2.21 VFP Period (VID_VFP_LINES)

Configures the VFP period.

Offset

Register	Offset
VID_VFP_LINES	5Ch

Diagram



Fields

Field	Description
31-10 —	Reserved and read as zero.
9-0 vfp_lines	VFP Lines Configures the Vertical Front Porch period measured in number of horizontal lines.

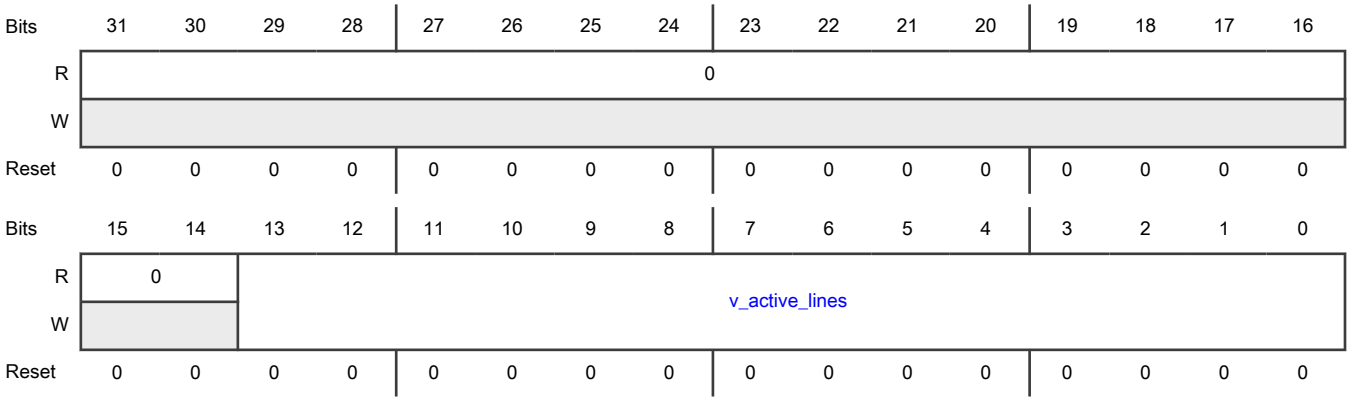
52.4.2.22 Vertical Resolution of Video (VID_VACTIVE_LINES)

Configures the vertical resolution of video.

Offset

Register	Offset
VID_VACTIVE_LINES	60h

Diagram



Fields

Field	Description
31-14 —	Reserved and read as zero.
13-0 v_active_lines	Vertical Active Lines Configures the Vertical Active period measured in number of horizontal lines.

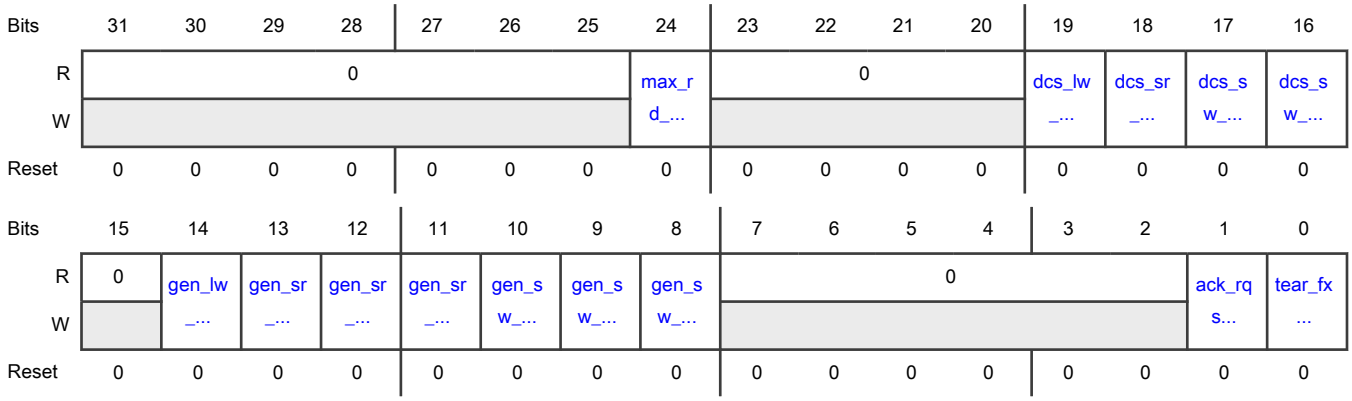
52.4.2.23 Command Mode Configuration (CMD_MODE_CFG)

Configures several aspect of command mode operation, tearing effect, acknowledge for each packet and the speed mode to transmit each Data Type related to commands.

Offset

Register	Offset
CMD_MODE_CFG	68h

Diagram



Fields

Field	Description
31-25 —	Reserved and read as zero.
24 max_rd_pkt_size	Maximum Read Packet Size Command Transmit Configures the maximum read packet size command transmission type: 0 - Transition type is high-speed 1 - Transition type is low-power
23-20 —	Reserved and read as zero.
19 dcs_lw_tx	DCS Long Write Command Transmit Configures the DCS long write packet command transmission type: 0 - Transition type is high-speed 1 - Transition type is low-power
18 dcs_sr_0p_tx	DCS Short Read Zero Parameter Command Transmit Configures the DCS short read packet with zero parameter command transmission type: 0 - Transition type is high-speed 1 - Transition type is low-power
17 dcs_sw_1p_tx	DCS Short Write One Parameter Command Transmit Configures the DCS short write packet with one parameter command transmission type: 0 - Transition type is high-speed 1 - Transition type is low-power
16	DCS Short Write Zero Parameter Command Transmit

Table continues on the next page...

Table continued from the previous page...

Field	Description
dcs_sw_0p_tx	Configures the DCS short write packet with zero parameter command transmission type: 0 - Transition type is high-speed 1 - Transition type is low-power
15 —	Reserved and read as zero.
14 gen_lw_tx	Generic Long Write Packet Command Transmit Configures the Generic long write packet command transmission type: 0 - Transition type is high-speed 1 - Transition type is low-power
13 gen_sr_2p_tx	Generic Short Read Two Parameter Command Transmit Configures the Generic short read packet with two parameters command transmission type: 0 - Transition type is high-speed 1 - Transition type is low-power
12 gen_sr_1p_tx	Generic Short Read One Parameter Command Transmit Configures the Generic short read packet with one parameter command transmission type: 0 - Transition type is high-speed 1 - Transition type is low-power
11 gen_sr_0p_tx	Generic Short Read Zero Parameter Command Transmit Configures the Generic short read packet with zero parameter command transmission type: 0 - Transition type is high-speed 1 - Transition type is low-power
10 gen_sw_2p_tx	Generic Short Write Two Parameter Command Transmit Configures the Generic short write packet with two parameters command transmission type: 0 - Transition type is high-speed 1 - Transition type is low-power
9 gen_sw_1p_tx	Generic Short Write One Parameter Command Transmit Configures the Generic short write packet with one parameter command transmission type: 0 - Transition type is high-speed 1 - Transition type is low-power
8 gen_sw_0p_tx	Generic Short Write Zero Parameter Command Transmit Configures the Generic short write packet with zero parameter command transmission type:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Transition type is high-speed 1 - Transition type is low-power
7-2 —	Reserved and read as zero.
1 ack_rqst_en	Acknowledge Request Enable Enables the acknowledge request after each packet transmission. 0 - Disable 1 - Enable
0 tear_fx_en	Tearing Effect Acknowledge Enable Enables the tearing effect acknowledge request. 0 - Disable 1 - Enable

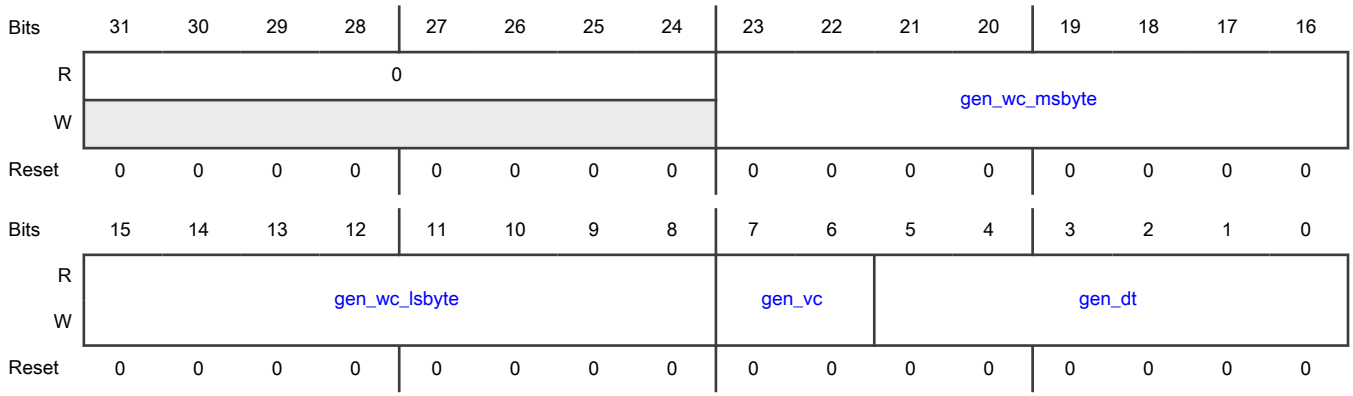
52.4.2.24 Generic Interface Header Packet Configuration (GEN_HDR)

Sets the header for new packets sent using the Generic interface.

Offset

Register	Offset
GEN_HDR	6Ch

Diagram



Fields

Field	Description
31-24 —	Reserved and read as zero.
23-16 gen_wc_msbyte	Generic Word Count MSB Configures the most significant byte of the header packet's word count for long packets or data 1 for short packets.
15-8 gen_wc_lsbyte	Generic Word Count LSB Configures the least significant byte of the header packet's Word count for long packets or data 0 for short packets.
7-6 gen_vc	Generic Virtual Channel ID Configures the Virtual Channel ID of the header packet.
5-0 gen_dt	Generic Data Type Configures the packet Data Type of the header packet.

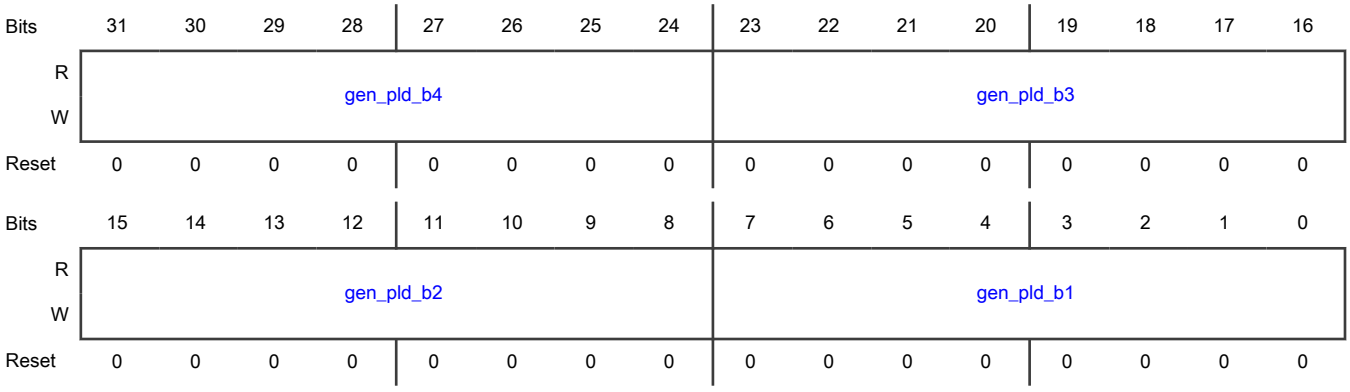
52.4.2.25 Generic Interface Payload Data (GEN_PLD_DATA)

Sets the payload for packets sent using the Generic interface and, when read returns the contents of READ responses from the peripheral.

Offset

Register	Offset
GEN_PLD_DATA	70h

Diagram



Fields

Field	Description
31-24 gen_pld_b4	Generic Payload Byte 4 Indicates byte 4 of the packet payload.
23-16 gen_pld_b3	Generic Payload Byte 3 Indicates byte 3 of the packet payload.
15-8 gen_pld_b2	Generic Payload Byte 2 Indicates byte 2 of the packet payload.
7-0 gen_pld_b1	Generic Payload Byte 1 Indicates byte 1 of the packet payload.

52.4.2.26 Command Packet Status (CMD_PKT_STATUS)

Contains information about the status of FIFOs related to the Generic interface.

Offset

Register	Offset
CMD_PKT_STATUS	74h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				0	0	0	0	0				gen_buf...	gen_buf...	gen_buf...	gen_buf...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	gen_rd_...	gen_pld...	gen_pld...	gen_pld...	gen_pld...	gen_cmd...	gen_cmd...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1

Fields

Field	Description
31-28 —	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
27 —	Reserved and read as zero.
26 —	Reserved and read as zero.
25 —	Reserved and read as zero.
24 —	Reserved and read as zero.
23-20 —	Reserved and read as zero.
19 gen_buff_pld_full	Generic Buffer for Payload Full Indicates the full status of the generic payload internal buffer. 0 - Not Full 1 - Full
18 gen_buff_pld_empty	Generic Buffer for Payload Empty Indicates the empty status of the generic payload internal buffer. 0 - Not Empty 1 - Empty
17 gen_buff_cmd_full	Generic Buffer for Command Full Indicates the full status of the generic command internal buffer. 0 - Not Full 1 - Full
16 gen_buff_cmd_empty	Generic Buffer for Command Empty Indicates the empty status of the generic command internal buffer. 0 - Not Empty 1 - Empty
15 —	Reserved and read as zero.
14 —	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
13 —	Reserved and read as zero.
12 —	Reserved and read as zero.
11 —	Reserved and read as zero.
10 —	Reserved and read as zero.
9 —	Reserved and read as zero.
8 —	Reserved and read as zero.
7 —	Reserved and read as zero.
6 gen_rd_cmd_busy	Generic Read Command Busy Sets when a read command is issued and cleared when the entire response is stored in the FIFO for GENERIC interface. 0 - Not Busy 1 - Busy
5 gen_pld_r_full	Generic Payload Read Full Indicates the full status of the generic read payload FIFO. 0 - Not Full 1 - Full
4 gen_pld_r_empty	Generic Payload Read Empty Indicates the empty status of the generic read payload FIFO. 0 - Not Empty 1 - Empty
3 gen_pld_w_full	Generic Payload Write Full Indicates the full status of the generic write payload FIFO. 0 - Not Full

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Full
2 gen_pld_w_empty	Generic Payload Write Empty Indicates the empty status of the generic write payload FIFO. 0 - Not Empty 1 - Empty
1 gen_cmd_full	Generic Command Full Indicates the full status of the generic command FIFO. 0 - Not Full 1 - Full
0 gen_cmd_empty	Generic Command Empty Indicates the empty status of the generic command FIFO. 0 - Not Empty 1 - Empty

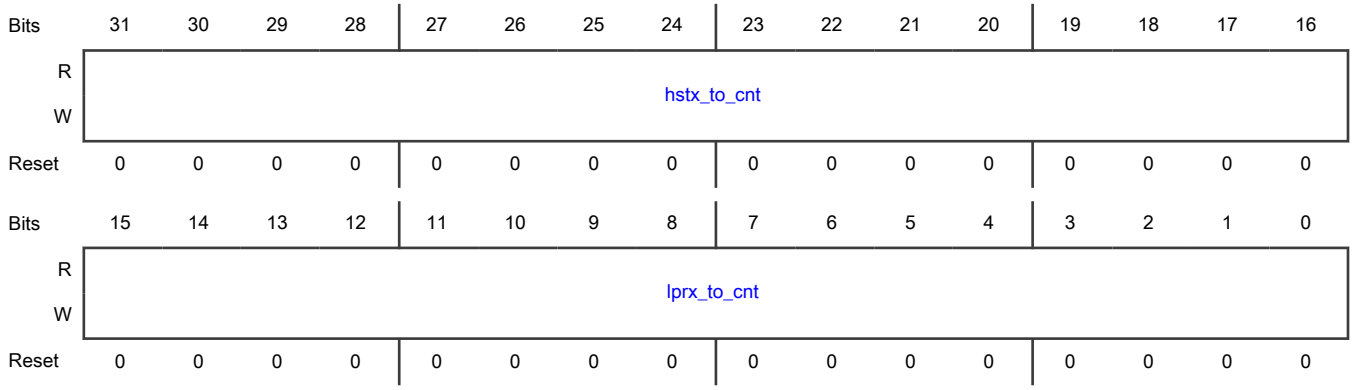
52.4.2.27 Timeout Counter Configuration (TO_CNT_CFG)

Configures counters that trigger timeout errors. Used to warn the system of a failure through an interrupt and restart the controller in case of unexpected situations that cause deadlock conditions.

Offset

Register	Offset
TO_CNT_CFG	78h

Diagram



Fields

Field	Description
31-16 hstx_to_cnt	High-speed Timeout Count Configures the timeout counter that triggers a high-speed transmission timeout contention detection (measured in TO_CLK_DIVISION cycles). If using non-burst mode and there is not sufficient time to switch from high-speed to low-power and back in the period which is from one line data finishing to the next line sync start, the DSI link returns low-power state once per frame, then you should configure the TO_CLK_DIVISION and hstx_to_cnt to satisfy the following formula: hstx_to_cnt * lanebyteclkperiod * TO_CLK_DIVISION >= the time of one FRAME data transmission * (1 + 10%) In burst mode, RGB pixel packets are time-compressed, leaving more time during a scan line. So if in burst mode and there is sufficient time to switch from high-speed to low-power and back in the period of time from one line data finishing to the next line sync start, the DSI link can return low-power mode and back in this time interval to save power. If you choose so, you should configure the TO_CLK_DIVISION and hstx_to_cnt to satisfy the following formula: hstx_to_cnt * lanebyteclkperiod * TO_CLK_DIVISION >= the time of one LINE data transmission * (1 + 10%)
15-0 lprx_to_cnt	Low-power Timeout Count Configures the timeout counter that triggers a low-power reception timeout contention detection (measured in TO_CLK_DIVISION cycles).

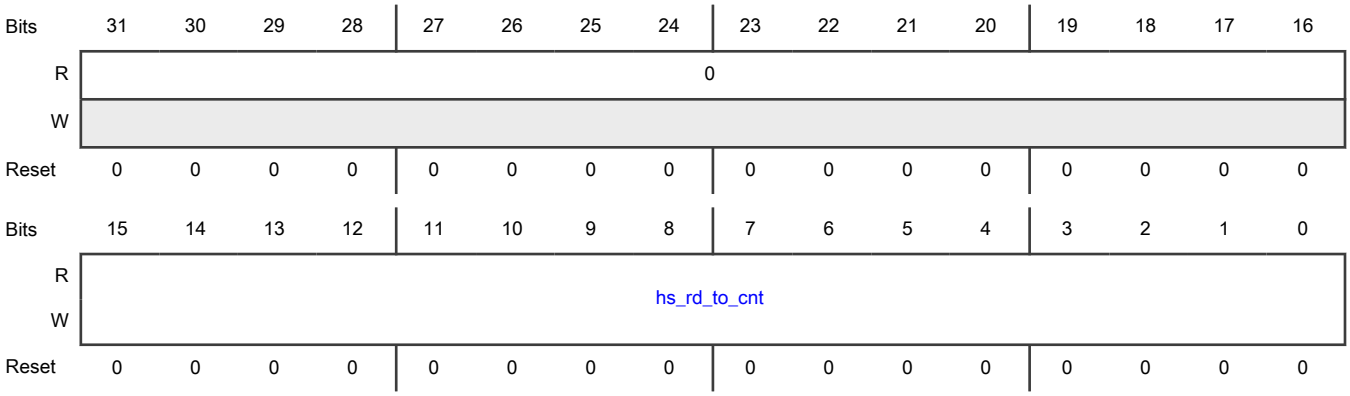
52.4.2.28 Peripheral Response Timeout after High-speed Read Operations (HS_RD_TO_CNT)

Configures the Peripheral Response timeout after high-speed Read operations.

Offset

Register	Offset
HS_RD_TO_CNT	7Ch

Diagram



Fields

Field	Description
31-16 —	Reserved and read as zero.
15-0 hs_rd_to_cnt	High-speed Read Timeout Count Sets a period for which MIPI DSI host keeps the link still, after sending a high-speed Read operation. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.

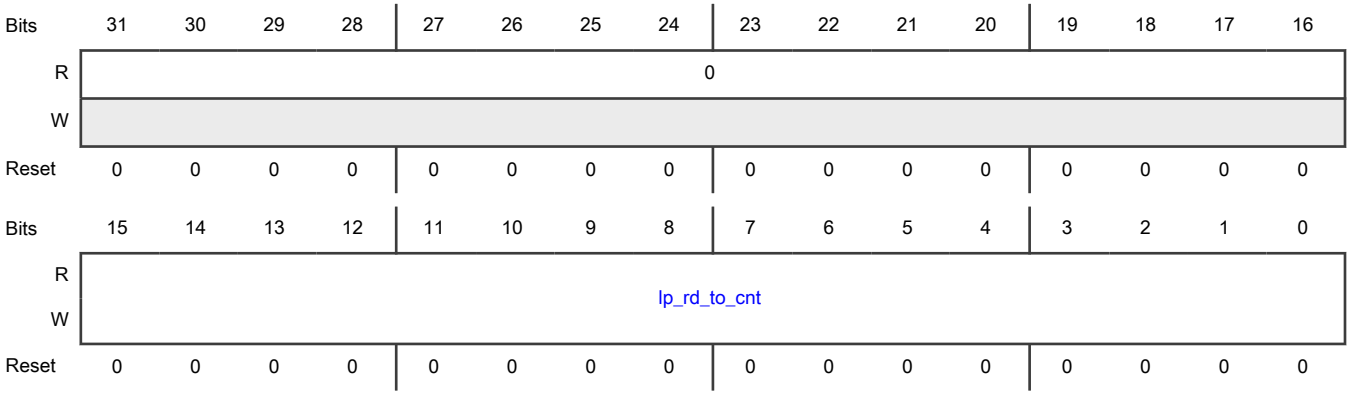
52.4.2.29 Peripheral Response Timeout after Low-power Read Operations (LP_RD_TO_CNT)

Configures the Peripheral Response timeout after low-power Read operations.

Offset

Register	Offset
LP_RD_TO_CNT	80h

Diagram



Fields

Field	Description
31-16 —	Reserved and read as zero.
15-0 lp_rd_to_cnt	Low-power Read Timeout Count Sets a period for which MIPI DSI host keeps the link still, after sending a low-power Read operation. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.

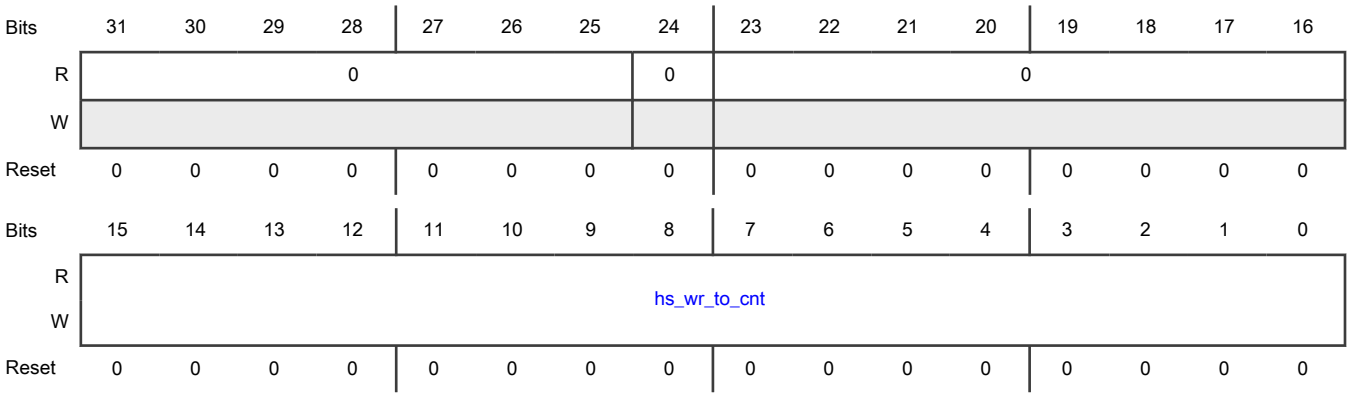
52.4.2.30 Peripheral Response Timeout after High-speed Write Operations (HS_WR_TO_CNT)

Configures the Peripheral Response timeout after high-speed Write operations.

Offset

Register	Offset
HS_WR_TO_CNT	84h

Diagram



Fields

Field	Description
31-25 —	Reserved and read as zero.
24 —	Reserved and read as zero.
23-16 —	Reserved and read as zero.
15-0 hs_wr_to_cnt	High-speed Write Timeout Count Sets a period for which MIPI DSI host keeps the link still, after sending a high-speed Write operation. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.

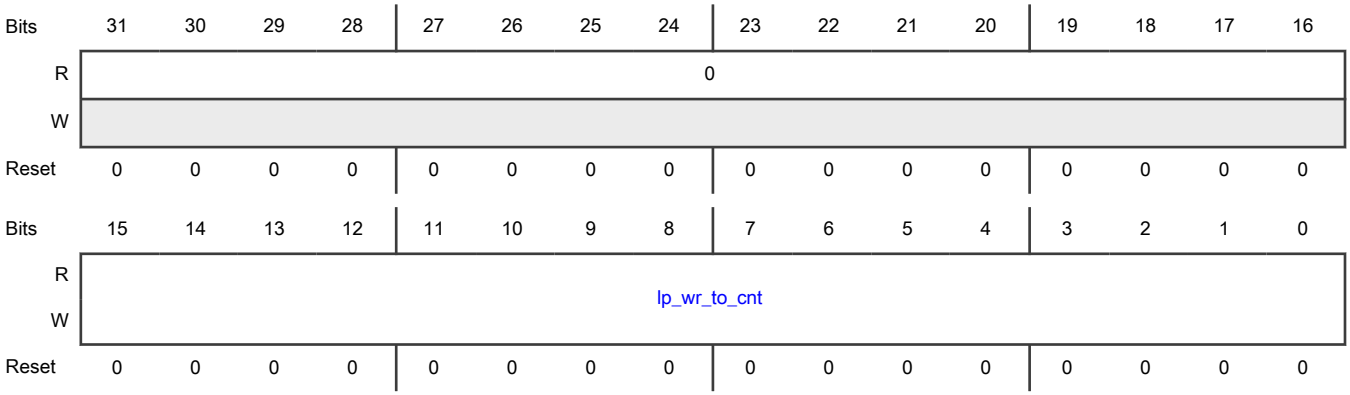
52.4.2.31 Peripheral Response Timeout after Low-power Write Operations (LP_WR_TO_CNT)

Configures the Peripheral Response timeout after low-power Write operations.

Offset

Register	Offset
LP_WR_TO_CNT	88h

Diagram



Fields

Field	Description
31-16 —	Reserved and read as zero.
15-0 lp_wr_to_cnt	Low-power Write Timeout Count Sets a period for which MIPI DSI host keeps the link still, after sending a low-power Write operation. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.

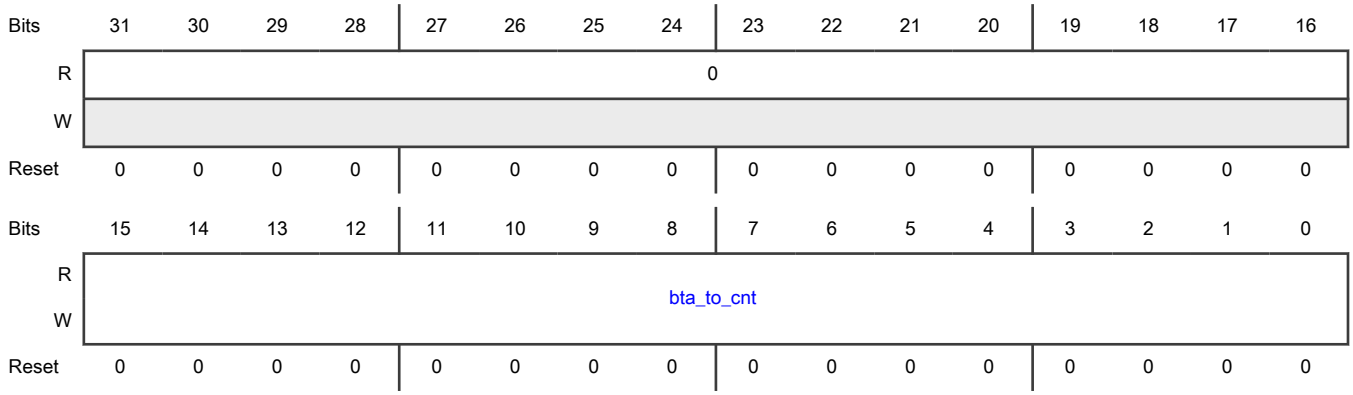
52.4.2.32 Peripheral Response Timeout after Bus Turnaround Completion (BTA_TO_CNT)

Configures the Peripheral Response timeout after Bus Turnaround completion.

Offset

Register	Offset
BTA_TO_CNT	8Ch

Diagram



Fields

Field	Description
31-16 —	Reserved and read as zero.
15-0 bta_to_cnt	BTA Timeout Count Sets a period for which MIPI DSI host keeps the link still, after completing a Bus Turnaround. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.

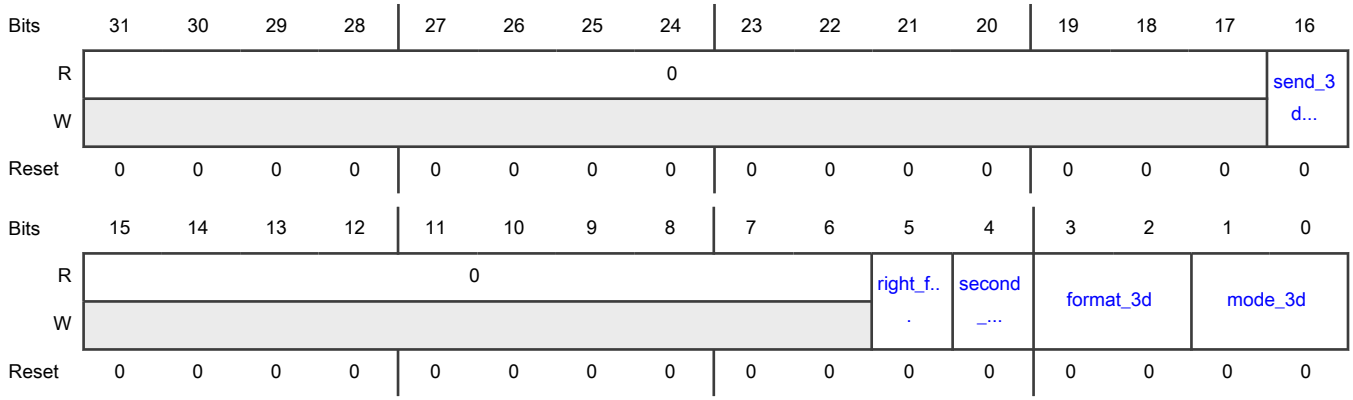
52.4.2.33 3D Control Information for VSS Packets in Video Mode (SDF_3D)

Stores 3D control information for VSS packets in video mode.

Offset

Register	Offset
SDF_3D	90h

Diagram



Fields

Field	Description
31-17 —	Reserved and read as zero.
16 send_3d_cfg	Send 3D Control Payload Causes the next VSS packet to include 3D control payload in every VSS packet. 0 - Disable 1 - Enable
15-6 —	Reserved and read as zero.
5 right_first	Right First Defines the left/right order: 0 - Left eye is sent first, then right eye 1 - Right eye data is sent first, then left eye
4 second_vsync	Second VSYNC Defines whether there is a second VSYNC pulse between Left and Right Images, when 3D Image Format is Frame-based: 0 - No sync pulses between left and right data 1 - Sync pulse HSYNC, VSYNC, blanking between left and right data
3-2 format_3d	Format 3D Defines 3D Image Format: 00 - Alternating lines of left and right data 01 - Alternating frames of left and right data 10 - Alternating pixels of left and right data 11 - Reserved
1-0 mode_3d	Mode 3D Defines 3D Mode On/Off and Display Orientation: 00 - 3D Mode Off , 2D Mode On 01 - 3D Mode On, Portrait Orientation 10 - 3D Mode On, Landscape Orientation 11 - Reserved

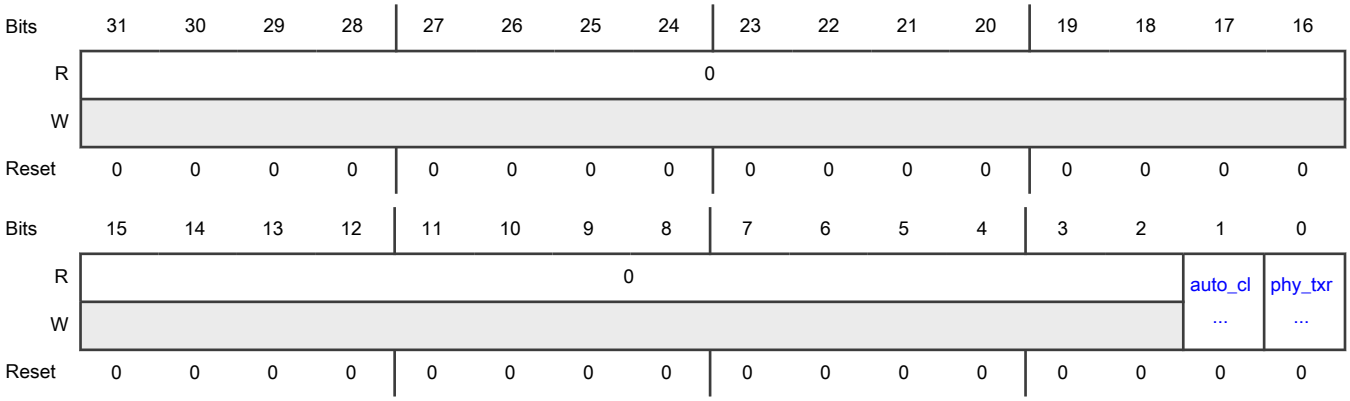
52.4.2.34 LP Clock Control (LPCLK_CTRL)

Configures the possibility for using non continuous clock in the clock lane.

Offset

Register	Offset
LPCLK_CTRL	94h

Diagram



Fields

Field	Description
31-2 —	Reserved and read as zero.
1 auto_clklane_ctrl	Automatic Clock Lane Control Enables the automatic mechanism to stop providing clock in the clock lane when time allows. 0 - Disable 1 - Enable
0 phy_txrequestclkhs	D-PHY txrequestclkhs Control Controls the D-PHY PPI txrequestclkhs signal. 0 - Disable 1 - Enable

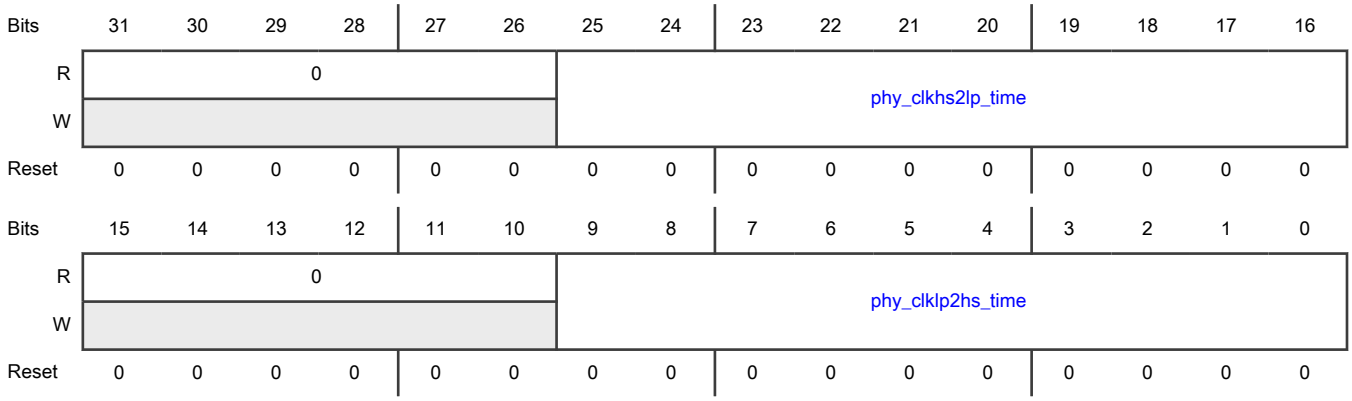
52.4.2.35 D-PHY Clock Lanes LP2HS and HS2LP Time Configuration (PHY_TMR_LPCLK_CFG)

Sets the time that MIPI DSI host assumes in calculations for the clock lane to switch between high-speed and low-power.

Offset

Register	Offset
PHY_TMR_LPCLK_CFG	98h

Diagram



Fields

Field	Description
31-26 —	Reserved and read as zero.
25-16 phy_clkhs2lp_time	D-PHY Clock High Power to Low Power Time Configures the maximum time that the D-PHY clock lane takes to go from high-speed to low-power transmission measured in lane byte clock cycles.
15-10 —	Reserved and read as zero.
9-0 phy_clklp2hs_time	D-PHY Clock Low Power to High Power Time Configures the maximum time that the D-PHY clock lane takes to go from low-power to high-speed transmission measured in lane byte clock cycles.

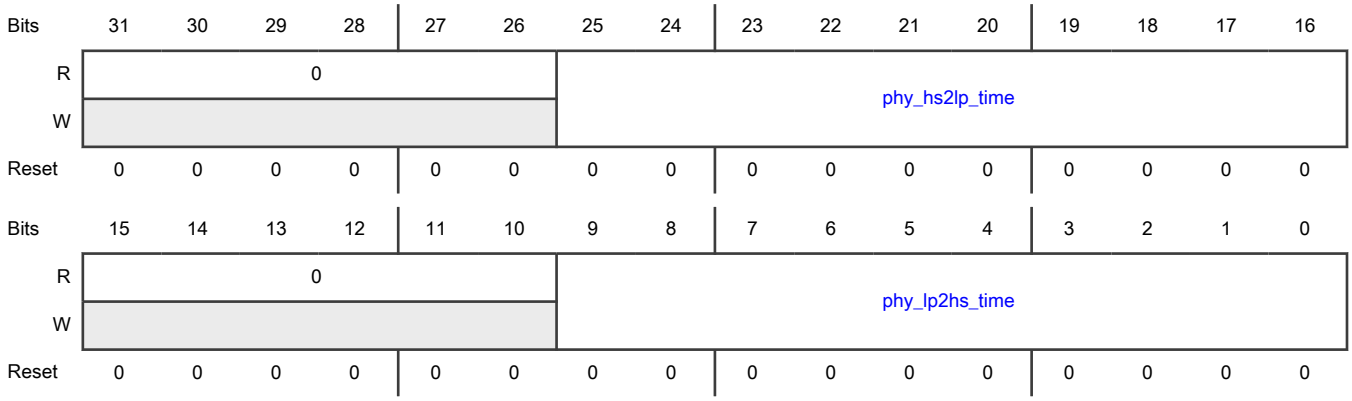
52.4.2.36 D-PHY Data Lanes LP2HS and HS2LP Time Configuration (PHY_TMR_CFG)

Sets the time that MIPI DSI host assumes in calculations for the data lanes to switch between high-speed and low-power.

Offset

Register	Offset
PHY_TMR_CFG	9Ch

Diagram



Fields

Field	Description
31-26 —	Reserved and read as zero.
25-16 phy_hs2lp_time	D-PHY High-speed to Low-power Time Configures the maximum time that the D-PHY data lanes take to go from high-speed to low-power transmission measured in lane byte clock cycles.
15-10 —	Reserved and read as zero.
9-0 phy_lp2hs_time	D-PHY Low-power to High-speed Time Configures the maximum time that the D-PHY data lanes take to go from low-power to high-speed transmission measured in lane byte clock cycles.

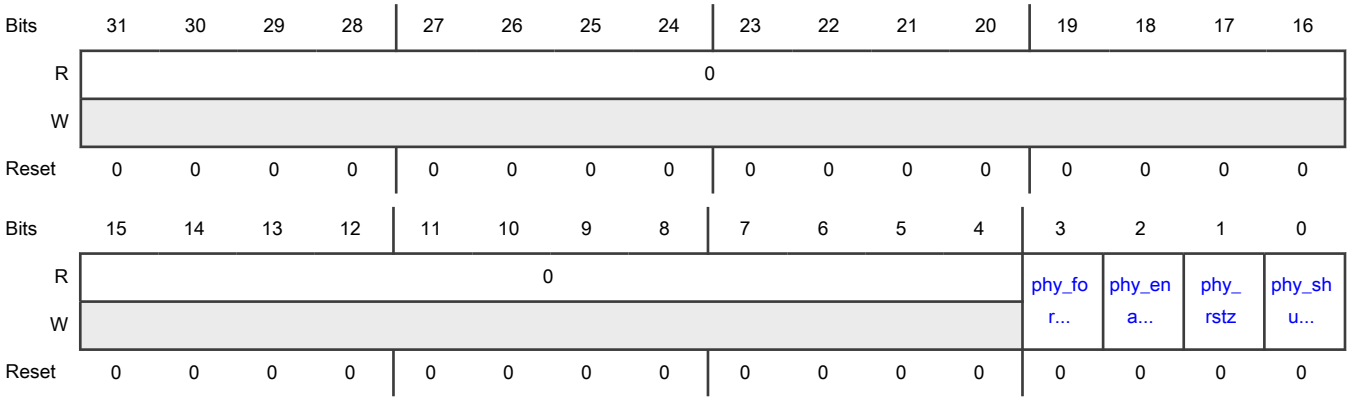
52.4.2.37 D-PHY Reset Control (PHY_RSTZ)

Controls resets and the PLL of the D-PHY.

Offset

Register	Offset
PHY_RSTZ	A0h

Diagram



Fields

Field	Description
31-4 —	Reserved and read as zero.
3 phy_forcepll	D-PHY Force PLL When the D-PHY is in ULPS, this bit enables the D-PHY PLL. 0 - Disable 1 - Enable
2 phy_enableclk	D-PHY Enable Clock Enables the D-PHY Clock Lane Module. 0 - Disable 1 - Enable
1 phy_rstz	D-PHY Reset State When set to 0, this bit places the digital section of the D-PHY in the reset state.
0 phy_shutdownz	D-PHY Power-down State When set to 0, this bit places the complete D-PHY macro in power-down state.

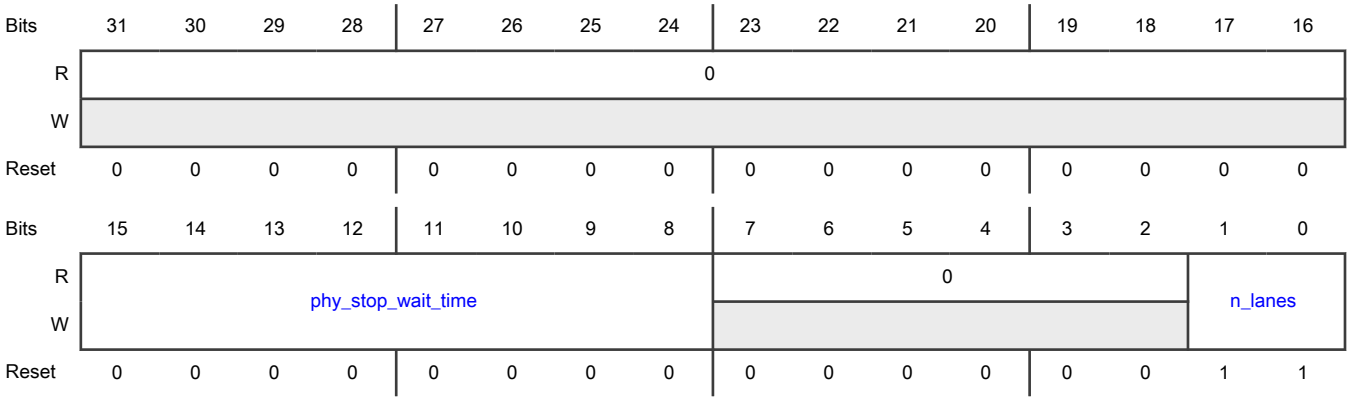
52.4.2.38 PHY Interface Configuration (PHY_IF_CFG)

Configures the number of active lanes and the minimum time to remain in stop state.

Offset

Register	Offset
PHY_IF_CFG	A4h

Diagram



Fields

Field	Description
31-16 —	Reserved and read as zero.
15-8 phy_stop_wait_time	PHY Stop Wait Time Configures the minimum time PHY needs to stay in StopState before requesting an high-speed transmission
7-2 —	Reserved and read as zero.
1-0 n_lanes	Number of Lanes Configures the number of active data lanes: 00 - lane 0 01 - lanes 0 and 1 10 - lanes 0, 1, and 2 11 - lanes 0, 1, 2, and 3

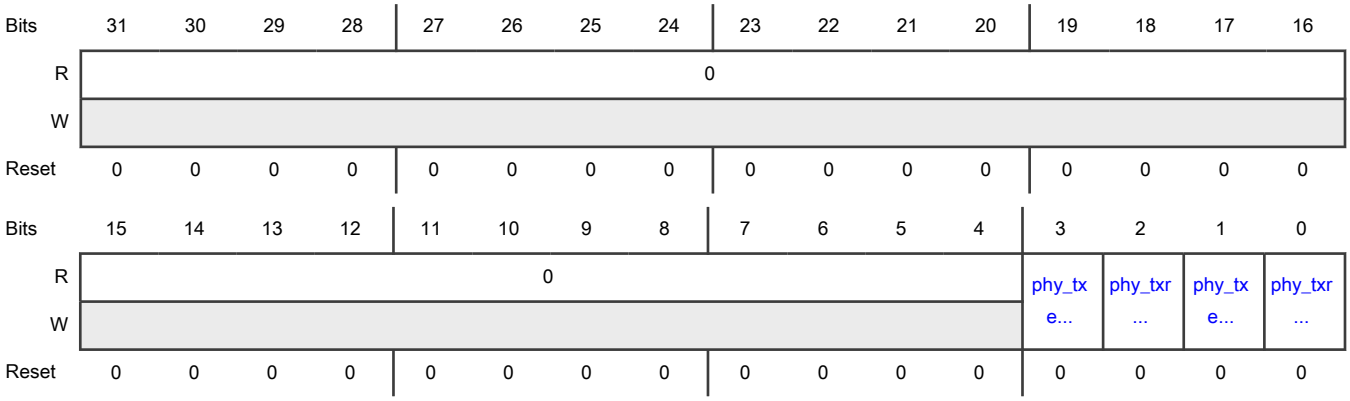
52.4.2.39 D-PHY ULPS Control (PHY_ULPS_CTRL)

Configures entering and leaving ULPS in the D-PHY.

Offset

Register	Offset
PHY_ULPS_CTRL	A8h

Diagram



Fields

Field	Description
31-4 —	Reserved and read as zero.
3 phy_txexitulpslan	ULPS Mode Exit on All Active Data Lanes
2 phy_txrequlpslan	ULPS Mode Request on All Active Data Lanes
1 phy_txexitulpsclk	ULPS Mode Exit on Clock Lane
0 phy_txrequlpsclk	ULPS Mode Request on Clock Lane

52.4.2.40 D-PHY Trigger Transmission Control (PHY_TX_TRIGGERS)

Configures the pins that activate triggers in the D-PHY.

Offset

Register	Offset
PHY_TX_TRIGGERS	ACh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												phy_tx_triggers			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved and read as zero.
3-0 phy_tx_triggers	Controls the Trigger Transmissions.

52.4.2.41 D-PHY Status (PHY_STATUS)

Contains information about the status of the D-PHY.

Offset

Register	Offset
PHY_STATUS	B0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			phy_up...	phy_sto...	phy_up...	phy_sto...	phy_up...	phy_sto...	phy_rxu...	phy_up...	phy_sto...	phy_up...	phy_sto...	phy_dir...	phy_lock
W																
Reset	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0

Fields

Field	Description
31-13 —	Reserved and read as zero.
12 phy_ulpsactivenot3lane	D-PHY ulpsactivenot3lane Indicates the status of ulpsactivenot3lane D-PHY signal.
11 phy_stopstate3lane	D-PHY phystopstate3lane Indicates the status of phystopstate3lane D-PHY signal.
10 phy_ulpsactivenot2lane	D-PHY ulpsactivenot2lane Indicates the status of ulpsactivenot2lane D-PHY signal.
9 phy_stopstate2lane	D-PHY phystopstate2lane Indicates the status of phystopstate2lane D-PHY signal.
8 phy_ulpsactivenot1lane	D-PHY ulpsactivenot1lane Indicates the status of ulpsactivenot1lane D-PHY signal.
7 phy_stopstate1lane	D-PHY phystopstate1lane Indicates the status of phystopstate1lane D-PHY signal.
6 phy_rxulpsesc0lane	D-PHY rxulpsesc0lane Indicates the status of rxulpsesc0lane D-PHY signal.
5 phy_ulpsactivenot0lane	D-PHY ulpsactivenot0lane Indicates the status of ulpsactivenot0lane D-PHY signal.
4 phy_stopstate0lane	D-PHY phystopstate0lane Indicates the status of phystopstate0lane D-PHY signal.
3 phy_ulpsactivenotclk	D-PHY phy_ulpsactivenotclk Indicates the status of phyulpsactivenotclk D-PHY signal.
2	D-PHY stopstateclklane

Table continues on the next page...

Table continued from the previous page...

Field	Description
phy_stopstateclklane	Indicates the status of phystopstateclklane D-PHY signal.
1 phy_direction	D-PHY Direction Indicates the status of phydirection D-PHY signal.
0 phy_lock	D-PHY Lock Indicates the status of phylock D-PHY signal.

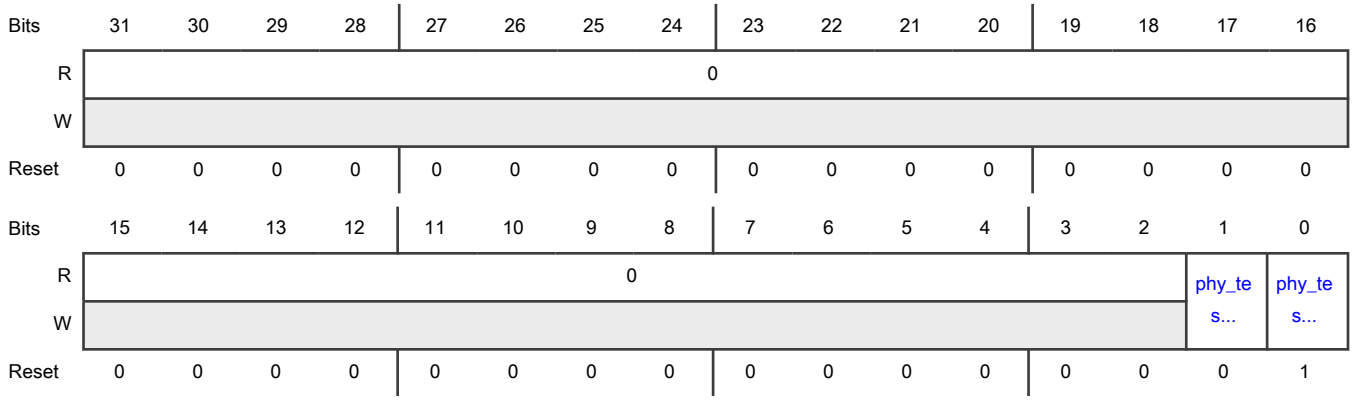
52.4.2.42 D-PHY Test Control 0 (PHY_TST_CTRL0)

Controls clock and clear pins of the D-PHY vendor specific interface.

Offset

Register	Offset
PHY_TST_CTRL0	B4h

Diagram



Fields

Field	Description
31-2 —	Reserved and read as zero.
1 phy_testclk	D-PHY Test Clock This bit is used to clock the TESTDIN bus into the D-PHY.
0 phy_testclr	D-PHY Test Interface Clear (active high).

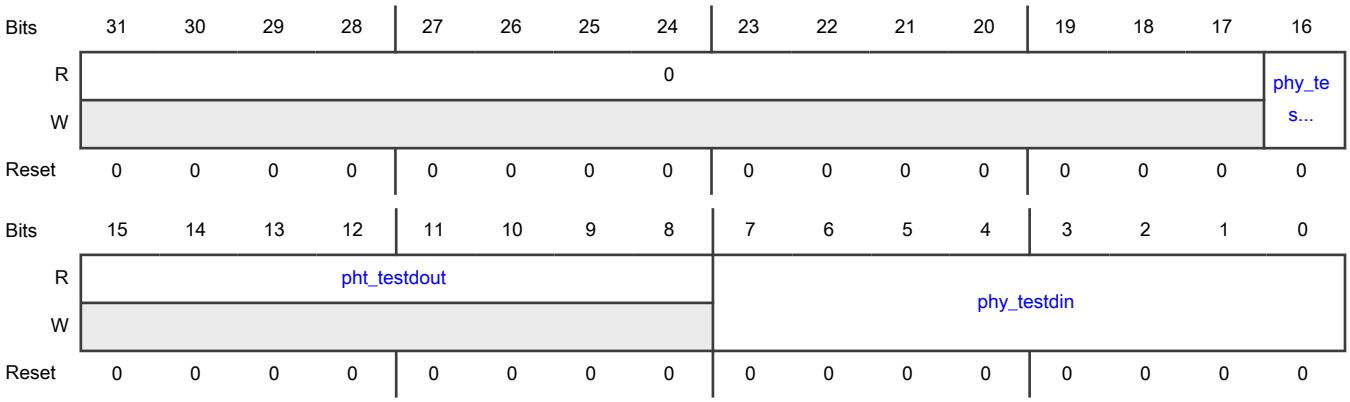
52.4.2.43 D-PHY Test Control 1 (PHY_TST_CTRL1)

Controls data and enable pins of the D-PHY vendor specific interface.

Offset

Register	Offset
PHY_TST_CTRL1	B8h

Diagram



Fields

Field	Description
31-17 —	Reserved and read as zero.
16 phy_testen	PHY Test Enable PHY test interface operation selector: 0 - Data write operation is set on the rising edge of the testclk signal. 1 - Address write operation is set on the falling edge of the testclk signal.
15-8 pht_testdout	PHY Test Data Output PHY output 8-bit data bus for read-back and internal probing functionalities.
7-0 phy_testdin	PHY Test Data Input PHY test interface input 8-bit data bus for internal register programming and test functionalities access.

52.4.2.44 Interrupt Status 0 (INT_ST0)

Contains the status of interrupt sources from acknowledge reports and the D-PHY.

Offset

Register	Offset
INT_ST0	BCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											dphy_er...	dphy_er...	dphy_er...	dphy_er...	dphy_er...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ack_wit...	ack_wit...	ack_wit...	ack_wit...	ack_wit...	ack_wit...	ack_wit...	ack_wit...	ack_wit...	ack_wit...	ack_wit...	ack_wit...	ack_wit...	ack_wit...	ack_wit...	ack_wit...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-21 —	Reserved and read as zero.
20 dphy_errors_4	D-PHY Errors 4 Indicates LP1 contention error ErrContentionLP1 from Lane 0. 0 - Disabled 1 - Enabled
19 dphy_errors_3	D-PHY Errors 3 Indicates LP0 contention error ErrContentionLP0 from Lane 0. 0 - Disabled 1 - Enabled
18 dphy_errors_2	D-PHY Errors 2 Indicates control error ErrControl from Lane 0. 0 - Disabled 1 - Enabled
17 dphy_errors_1	D-PHY Errors 1 Indicates ErrSyncEsc low-power data transmission synchronization error from Lane 0. 0 - Disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enabled
16 dphy_errors_0	D-PHY Error 0 Indicates ErrEsc escape entry error from Lane 0. 0 - Disabled 1 - Enabled
15 ack_with_err_15	Acknowledge with Error 15 Retrieves the DSI protocol violation from the Acknowledge error report. 0 - Disabled 1 - Enabled
14 ack_with_err_14	Acknowledge with Error 14 Retrieves the reserved (specific to device) from the Acknowledge error report. 0 - Disabled 1 - Enabled
13 ack_with_err_13	Acknowledge with Error 13 Retrieves the invalid transmission length from the Acknowledge error report. 0 - Disabled 1 - Enabled
12 ack_with_err_12	Acknowledge with Error 12 Retrieves the DSI VC ID Invalid from the Acknowledge error report. 0 - Disabled 1 - Enabled
11 ack_with_err_11	Acknowledge with Error 11 Retrieves the not recognized DSI data type from the Acknowledge error report. 0 - Disabled 1 - Enabled
10 ack_with_err_10	Acknowledge with Error 10 Retrieves the checksum error (long packet only) from the Acknowledge error report. 0 - Disabled 1 - Enabled
9 ack_with_err_9	Acknowledge with Error 9 Retrieves the ECC error, multi-bit (detected, not corrected) from the Acknowledge error report.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Disabled 1 - Enabled
8 ack_with_err_8	Acknowledge with Error 8 Retrieves the ECC error, single-bit (detected and corrected) from the Acknowledge error report. 0 - Disabled 1 - Enabled
7 ack_with_err_7	Acknowledge with Error 7 Retrieves the reserved (specific to device) from the acknowledge error report. 0 - Disabled 1 - Enabled
6 ack_with_err_6	Acknowledge with Error 6 Retrieves the False Control error from the Acknowledge error report. 0 - Disabled 1 - Enabled
5 ack_with_err_5	Acknowledge with Error 5 Retrieves the Peripheral Timeout error from the Acknowledge error report. 0 - Disabled 1 - Enabled
4 ack_with_err_4	Acknowledge with Error 4 Retrieves the low-power Transmit Sync error from the Acknowledge error report. 0 - Disabled 1 - Enabled
3 ack_with_err_3	Acknowledge with Error 3 Retrieves the Escape Mode Entry Command error from the Acknowledge error report. 0 - Disabled 1 - Enabled
2 ack_with_err_2	Acknowledge with Error 2 Retrieves the EoT Sync error from the Acknowledge error report. 0 - Disabled 1 - Enabled
1	Acknowledge with Error 1

Table continues on the next page...

Table continued from the previous page...

Field	Description
ack_with_err_1	Retrieves the SoT Sync error from the Acknowledge error report. 0 - Disabled 1 - Enabled
0 ack_with_err_0	Acknowledge with Error 0 Retrieves the SoT error from the Acknowledge error report. 0 - Disabled 1 - Enabled

52.4.2.45 Interrupt Status 1 (INT_ST1)

Contains the status of interrupt sources related to timeouts, ECC, CRC, packet size, EoTp, and Generic interfaces.

Offset

Register	Offset
INT_ST1	C0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											0	dpi_buf...	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	gen_pld...	gen_pld...	gen_pld...	gen_pld...	gen_cmd...	dpi_pld...	eopt_err	pkt_siz...	crc_err	ecc_mil...	ecc_sin...	to_lp_rx	to_hs_tx
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-21 —	Reserved and read as zero.
20 —	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
19 dpi_buff_pld_under	DPI Buffer Payload Underflow Indicates that an underflow has occurred when reading payload to build DSI packet for video mode. 0 - Disabled 1 - Enabled
18 —	Reserved and read as zero.
17 —	Reserved and read as zero.
16 —	Reserved and read as zero.
15 —	Reserved and read as zero.
14 —	Reserved and read as zero.
13 —	Reserved and read as zero.
12 gen_pld_recev_err	Generic Payload Receive Error Indicates that during a generic interface packet read back, the payload FIFO becomes full and the received data is corrupted. 0 - Disabled 1 - Enabled
11 gen_pld_rd_err	Generic Payload Read Error Indicates that during a DCS read data, the payload FIFO becomes empty and the data sent to the interface is corrupted. 0 - Disabled 1 - Enabled
10 gen_pld_send_err	DPI Payload Send Error Indicates that during a Generic interface packet build, the payload FIFO becomes empty and corrupt data is sent. 0 - Disabled 1 - Enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 gen_pld_wr_err	Generic Payload Write Error Indicates that the system tried to write a payload data through the Generic interface and the FIFO is full. Therefore, the payload is not written. 0 - Disabled 1 - Enabled
8 gen_cmd_wr_err	Generic Command Write Error Indicates that the system tried to write a command through the Generic interface and the FIFO is full. Therefore, the command is not written. 0 - Disabled 1 - Enabled
7 dpi_pld_wr_err	DPI Payload Write Error Indicates that during a DPI pixel line storage, the payload FIFO becomes full and the data stored is corrupted. 0 - Disabled 1 - Enabled
6 eopt_err	EoTp Error Indicates that the EoTp packet has not been received at the end of the incoming peripheral transmission. 0 - Disabled 1 - Enabled
5 pkt_size_err	Packet Size Error Indicates that the packet size error has been detected during the packet reception. 0 - Disabled 1 - Enabled
4 crc_err	CRC Error Indicates that the CRC error has been detected in the received packet payload. 0 - Disabled 1 - Enabled
3 ecc_milti_err	ECC Multiple Error Indicates that the ECC multiple error has been detected in a received packet.
2 ecc_single_err	ECC Single Error Indicates that the ECC single error has been detected and corrected in a received packet. 0 - Disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enabled
1 to_lp_rx	Timeout Low-power Receive Indicates that the low-power reception timeout counter reached the end and contention has been detected. 0 - Disabled 1 - Enabled
0 to_hs_tx	Timeout High-speed Transmit Indicates that the high-speed transmission timeout counter reached the end and contention has been detected. 0 - Disabled 1 - Enabled

52.4.2.46 Interrupt Mask 0 (INT_MSK0)

Configures masks for the sources of interrupts that affect the INT_ST0 register. Write 1 to un-mask each error report.

Offset

Register	Offset
INT_MSK0	C4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											mask_dp...	mask_dp...	mask_dp...	mask_dp...	mask_dp...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	mask_ac...	mask_ac...	mask_ac...	mask_ac...	mask_ac...	mask_ac...	mask_ac...	mask_ac...	mask_ac...	mask_ac...	mask_ac...	mask_ac...	mask_ac...	mask_ac...	mask_ac...	mask_ac...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-21	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
20 mask_dphy_errors_4	Mask for dphy_errors_4
19 mask_dphy_errors_3	Mask for dphy_errors_3
18 mask_dphy_errors_2	Mask for dphy_errors_2
17 mask_dphy_errors_1	Mask for dphy_errors_1
16 mask_dphy_errors_0	Mask for dphy_errors_0
15 mask_ack_with_err_15	Mask for ack_with_err_15
14 mask_ack_with_err_14	Mask for ack_with_err_14
13 mask_ack_with_err_13	Mask for ack_with_err_13
12 mask_ack_with_err_12	Mask for ack_with_err_12
11 mask_ack_with_err_11	Mask for ack_with_err_11
10	Mask for ack_with_err_10

Table continues on the next page...

Table continued from the previous page...

Field	Description
mask_ack_with_err_10	
9 mask_ack_with_err_9	Mask for ack_with_err_9
8 mask_ack_with_err_8	Mask for ack_with_err_8
7 mask_ack_with_err_7	Mask for ack_with_err_7
6 mask_ack_with_err_6	Mask for ack_with_err_6
5 mask_ack_with_err_5	Mask for ack_with_err_5
4 mask_ack_with_err_4	Mask for ack_with_err_4
3 mask_ack_with_err_3	Mask for ack_with_err_3
2 mask_ack_with_err_2	Mask for ack_with_err_2
1 mask_ack_with_err_1	Mask for ack_with_err_1
0 mask_ack_with_err_0	Mask for ack_with_err_0

52.4.2.47 Interrupt Mask 1 (INT_MSK1)

Configures masks for the sources of interrupts that affect the INT_ST1 register.

Offset

Register	Offset
INT_MSK1	C8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											0	mask_dp...	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	mask_ge...	mask_ge...	mask_ge...	mask_ge...	mask_ge...	mask_dp...	mask_eo...	mask_pk...	mask_cr...	mask_ec...	mask_ec...	mask_to...	mask_to...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-21 —	Reserved and read as zero.
20 —	Reserved and read as zero.
19 mask_dpi_buff_pld_under	Mask for dpi_buff_pld_under
18 —	Reserved and read as zero.
17 —	Reserved and read as zero.
16 —	Reserved and read as zero.
15 —	Reserved and read as zero.
14	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
13 —	Reserved and read as zero.
12 mask_gen_pld_recev_err	Mask for gen_pld_recev_err
11 mask_gen_pld_rd_err	Mask for gen_pld_rd_err
10 mask_gen_pld_send_err	Mask for gen_pld_send_err
9 mask_gen_pld_wr_err	Mask for gen_pld_wr_err
8 mask_gen_cmd_wr_err	Mask for gen_cmd_wr_err
7 mask_dpi_pld_wr_err	Mask for dpi_pld_wr_err
6 mask_eopt_err	Mask for eopt_err
5 mask_pkt_size_err	Mask for pkt_size_err
4 mask_crc_err	Mask for crc_err
3 mask_ecc_milti_err	Mask for ecc_milti_err
2	Mask for ecc_single_err

Table continues on the next page...

Table continued from the previous page...

Field	Description
mask_ecc_single_err	
1 mask_to_lp_rx	Mask for to_lp_rx
0 mask_to_hs_tx	Mask for to_hs_tx

52.4.2.48 D-PHY Skew Calibration (PHY_CAL)

Controls the skew calibration of D-PHY.

Offset

Register	Offset
PHY_CAL	CCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															txskewc...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-1 —	Reserved and read as zero.
0 txskewcalhs	High-speed skew calibration is started when txskewcalhs is set high (assuming that PHY is in Stop state).

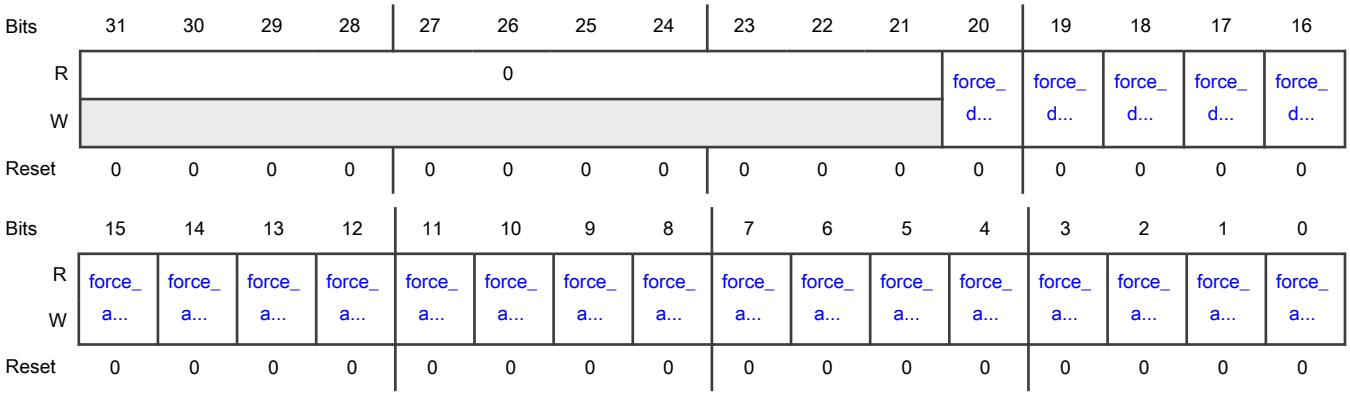
52.4.2.49 Interrupt Force 0 (INT_FORCE0)

Forces errors that affect INT_ST0.

Offset

Register	Offset
INT_FORCE0	D8h

Diagram



Fields

Field	Description
31-21 —	Reserved and read as zero.
20 force_dphy_errors_4	Force dphy_errors_4
19 force_dphy_errors_3	Force dphy_errors_3
18 force_dphy_errors_2	Force dphy_errors_2
17 force_dphy_errors_1	Force dphy_errors_1
16 force_dphy_errors_0	Force dphy_errors_0
15	Force ack_with_err_15

Table continues on the next page...

Table continued from the previous page...

Field	Description
force_ack_with_err_15	
14 force_ack_with_err_14	Force ack_with_err_14
13 force_ack_with_err_13	Force ack_with_err_13
12 force_ack_with_err_12	Force ack_with_err_12
11 force_ack_with_err_11	Force ack_with_err_11
10 force_ack_with_err_10	Force ack_with_err_10
9 force_ack_with_err_9	Force ack_with_err_9
8 force_ack_with_err_8	Force ack_with_err_8
7 force_ack_with_err_7	Force ack_with_err_7
6 force_ack_with_err_6	Force ack_with_err_6
5 force_ack_with_err_5	Force ack_with_err_5
4	Force ack_with_err_4

Table continues on the next page...

Table continued from the previous page...

Field	Description
force_ack_with_err_4	
3 force_ack_with_err_3	Force ack_with_err_3
2 force_ack_with_err_2	Force ack_with_err_2
1 force_ack_with_err_1	Force ack_with_err_1
0 force_ack_with_err_0	Force ack_with_err_0

52.4.2.50 Interrupt Force Register 1 (INT_FORCE1)

Forces interrupts that affect INT_ST1.

Offset

Register	Offset
INT_FORCE1	DCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											0	force_d...	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	force_g...	force_g...	force_g...	force_g...	force_g...	force_d...	force_e...	force_p...	force_c...	force_e...	force_e...	force_t...	force_t...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-21 —	Reserved and read as zero.
20 —	Reserved and read as zero.
19 force_dpi_buff_pld_under	Force for dpi_buff_pld_under.
18 —	Reserved and read as zero.
17 —	Reserved and read as zero.
16 —	Reserved and read as zero.
15 —	Reserved and read as zero.
14 —	Reserved and read as zero.
13 —	Reserved and read as zero.
12 force_gen_pld_recev_err	Force gen_pld_recev_err
11 force_gen_pld_rd_err	Force gen_pld_rd_err
10 force_gen_pld_send_err	Force gen_pld_send_err
9 force_gen_pld_wr_err	Force gen_pld_wr_err

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 force_gen_cmd_wr_err	Force gen_cmd_wr_err
7 force_dpi_pld_wr_err	Force dpi_pld_wr_err
6 force_eopt_err	Force eopt_err
5 force_pkt_size_err	Force pkt_size_err
4 force_crc_err	Force crc_err
3 force_ecc_milti_err	Force ecc_milti_err
2 force_ecc_single_err	Force ecc_single_err
1 force_to_lp_rx	Force to_lp_rx
0 force_to_hs_tx	Force to_hs_tx

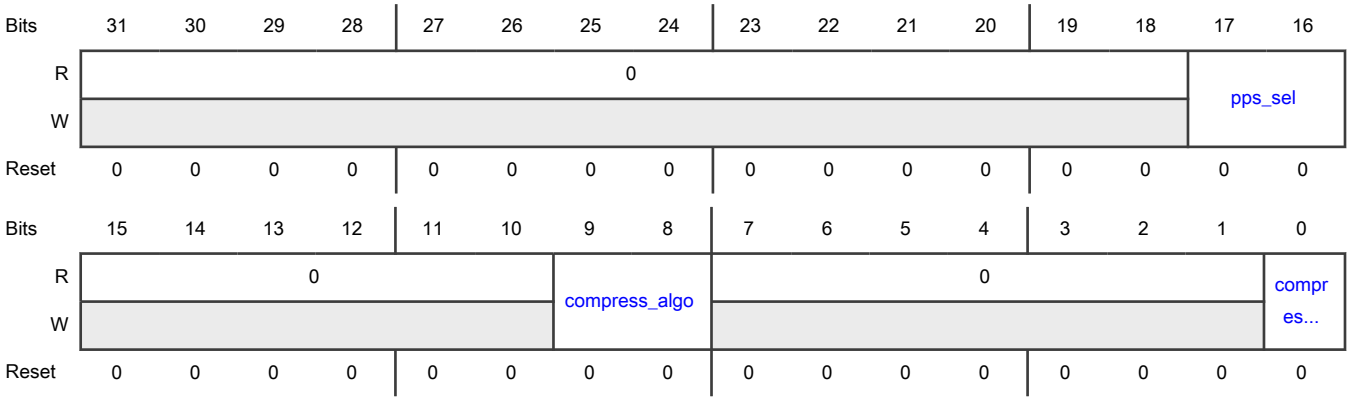
52.4.2.51 Display Stream Compression Configuration (DSC_PARAMETER)

Configures Display Stream Compression

Offset

Register	Offset
DSC_PARAMETER	F0h

Diagram



Fields

Field	Description
31-18 —	Reserved and read as zero.
17-16 pps_sel	PPS Select Indicates the PPS selector: 00 - PPS Table 1 01 - PPS Table 2 10 - PPS Table 3 11 - PPS Table 4
15-10 —	Reserved and read as zero.
9-8 compress_algo	Compression Algorithm Indicates the algorithm identifier: 00 - VESA DSC Standard 1.1 01,10 - Reserved 11 - Vendor-specific algorithm
7-1 —	Reserved and read as zero.
0 compression_mode	Compression Mode Enables the Compression mode. 0 - Disable 1 - Enable

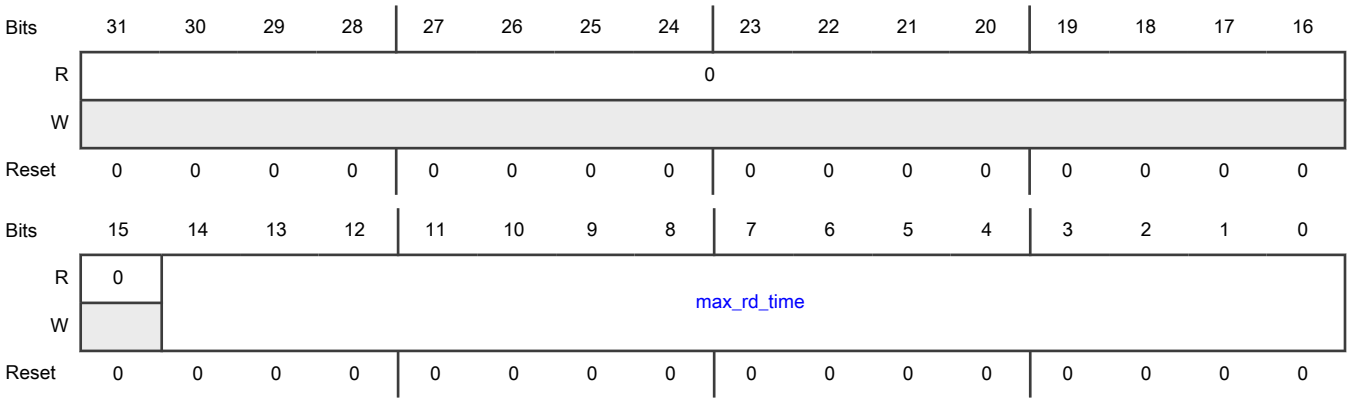
52.4.2.52 PHY Read Command Time Configuration (PHY_TMR_RD_CFG)

Configures times related to PHY to perform some operations in lane byte clock cycles.

Offset

Register	Offset
PHY_TMR_RD_CFG	F4h

Diagram



Fields

Field	Description
31-15 —	Reserved and read as zero.
14-0 max_rd_time	Maximum Read Time Configures the maximum time required to perform a read command in lane byte clock cycles. Can only be modified when no read command is in progress.

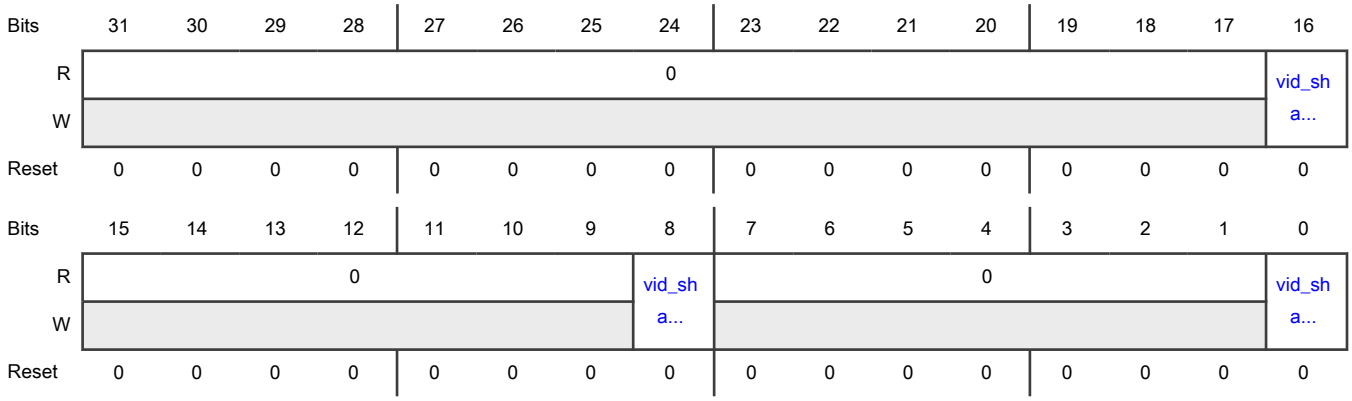
52.4.2.53 Video Shadow Control (VID_SHADOW_CTRL)

Controls DPI shadow feature.

Offset

Register	Offset
VID_SHADOW_CTRL	100h

Diagram



Fields

Field	Description
31-17 —	Reserved and read as zero.
16 vid_shadow_pin_req	Video Shadow Pin Request When set to 1, the video request is done by external pin. In this mode vid_shadow_req is ignored. 0 - Disable 1 - Enable
15-9 —	Reserved and read as zero.
8 vid_shadow_req	Video Shadow Request Requests that the DPI registers from regbank are copied to the auxiliary registers. When the request is completed this bit is auto clear. 0 - Disable 1 - Enable
7-1 —	Reserved and read as zero.
0 vid_shadow_en	Video Shadow Enable When set to 1, DPI receives the active configuration from the auxiliary registers. When the feature is set at the same time than vid_shadow_req the auxiliary registers are automatically updated. 0 - Disabled 1 - Enabled

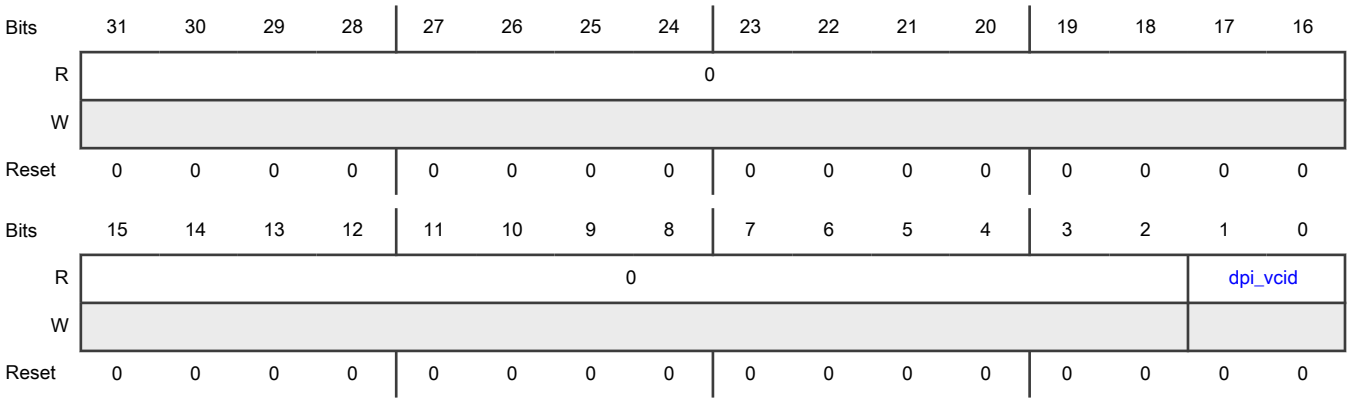
52.4.2.54 DPI Virtual Channel ID Value (DPI_VCID_ACT)

Holds the value that the controller is using for DPI_VCID.

Offset

Register	Offset
DPI_VCID_ACT	10Ch

Diagram



Fields

Field	Description
31-2 —	Reserved and read as zero.
1-0 dpi_vcid	DPI Virtual Channel ID Specifies the DPI virtual channel ID that is indexed to the Video mode packets.

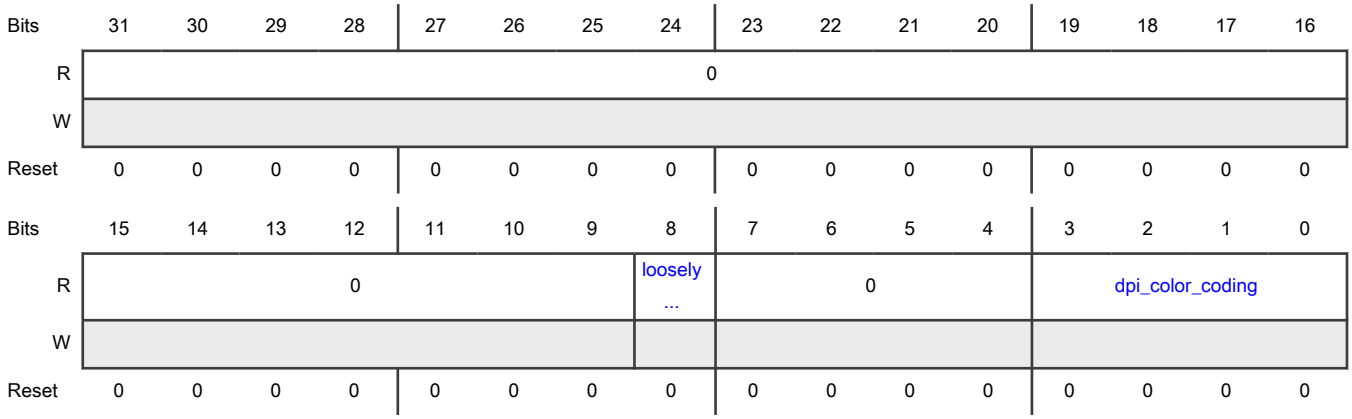
52.4.2.55 DPI Color Coding Value (DPI_COLOR_CODING_ACT)

Holds the value that the controller is using for DPI_COLOR_CODING.

Offset

Register	Offset
DPI_COLOR_CODING_ACT	110h

Diagram



Fields

Field	Description
31-9 —	Reserved and read as zero.
8 loosely18_en	Loosely 18 Bit Enabled Activates loosely packed variant to 18-bit configurations. 0 - Disabled 1 - Enabled
7-4 —	Reserved and read as zero.
3-0 dpi_color_coding	DPI Color Coding Configures the DPI color for Video Mode/eDPI Command Mode coding as follows: 0000 - 16-bit configuration 1 0001 - 16-bit configuration 2 0010 - 16-bit configuration 3 0011 - 18-bit configuration 1 0100 - 18-bit configuration 2 0101 - 24-bit 0110 - 20-bit YCbCr 4:2:2 loosely packed / Reserved for eDPI Command Mode 0111 - 24-bit YCbCr 4:2:2 / Reserved for eDPI Command Mode 1000 - 16-bit YCbCr 4:2:2 / Reserved for eDPI Command Mode 1001 - 30-bit - DSC_ENC 10bit / Reserved for eDPI Command Mode 1010 - 36-bit / Reserved for eDPI Command Mode 1011 - 12-bit YCbCr 4:2:0 / Reserved for eDPI Command Mode

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1100 - DSC24 compressed Data 1101-1111 - Reserved

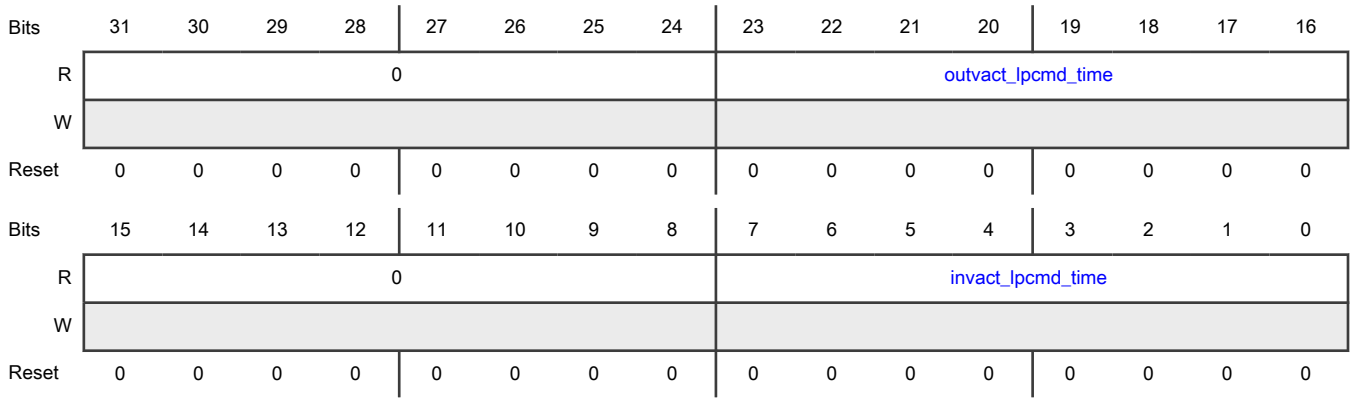
52.4.2.56 DPI Low-power Commands Timing Value (DPI_LP_CMD_TIM_ACT)

Holds the value that controller is using for DPI_LP_CMD_TIM.

Offset

Register	Offset
DPI_LP_CMD_TIM_ACT	118h

Diagram



Fields

Field	Description
31-24 —	Reserved and read as zero.
23-16 outvact_lpcmd_time	Out VACT Low-power Command Size Use for the transmission of commands in low-power mode. Specifies the size, in bytes, of the largest packet that can fit in a line during the VSA, VBP, and VFP regions.
15-8 —	Reserved and read as zero.
7-0 invact_lpcmd_time	In VACT Low-power Command Size Use for the transmission of commands in low-power mode. Specifies the size, in bytes, of the largest packet that can fit in a line during the VACT region.

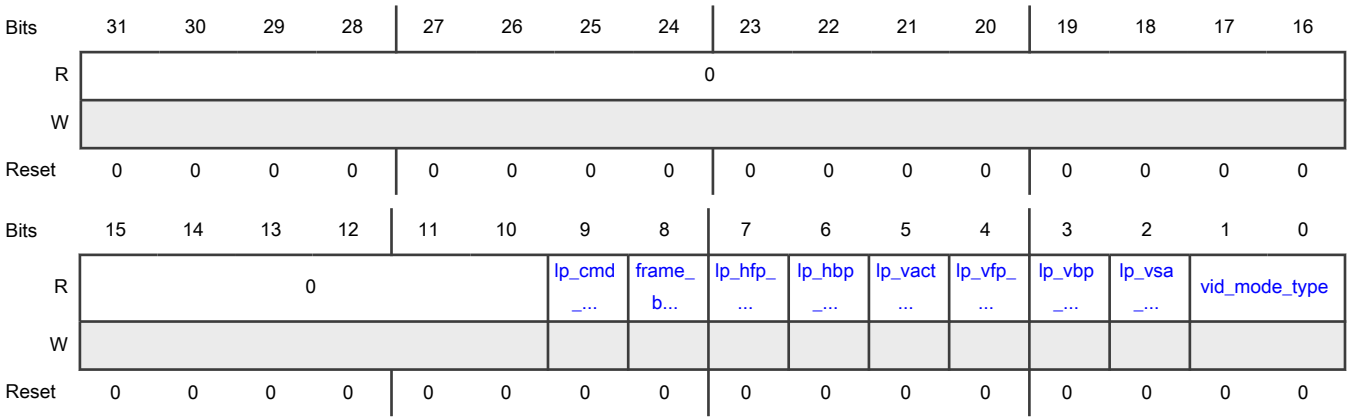
52.4.2.57 Video Mode Configuration Value (VID_MODE_CFG_ACT)

Holds the value that the controller is using for VID_MODE_CFG.

Offset

Register	Offset
VID_MODE_CFG_ACT	138h

Diagram



Fields

Field	Description
31-10 —	Reserved and read as zero.
9 lp_cmd_en	Low-power Command Enable Enables the command transmission only in low-power mode. 0 - Disabled 1 - Enabled
8 frame_bta_ack_en	Frame BTA Acknowledge Enable Enables the request for an acknowledge response at the end of a frame. 0 - Disabled 1 - Enabled
7 lp_hfp_en	Low-power HFP Enable Enables the return to low-power inside the HFP period when timing allows. 0 - Disabled 1 - Enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
6 lp_hbp_en	Low-power HBP Enable Enables the return to low-power inside the HBP period when timing allows. 0 - Disabled 1 - Enabled
5 lp_vact_en	Low-power VFP Enable Enables the return to low-power inside the VACT period when timing allows. 0 - Disabled 1 - Enabled
4 lp_vfp_en	Low-power VFP Enable Enables the return to low-power inside the VFP period when timing allows. 0 - Disabled 1 - Enabled
3 lp_vbp_en	Low-power VBP Enable Enables the return to low-power inside the VBP period when timing allows. 0 - Disabled 1 - Enabled
2 lp_vsa_en	Low-power VSA Enable Enables the return to low-power inside the VSA period when timing allows. 0 - Disabled 1 - Enabled
1-0 vid_mode_type	Video Mode Type Specifies the video mode transmission type as follows: 00 - Non-burst with sync pulses 01 - Non-burst with sync events 10 - Burst mode 11 - Burst mode

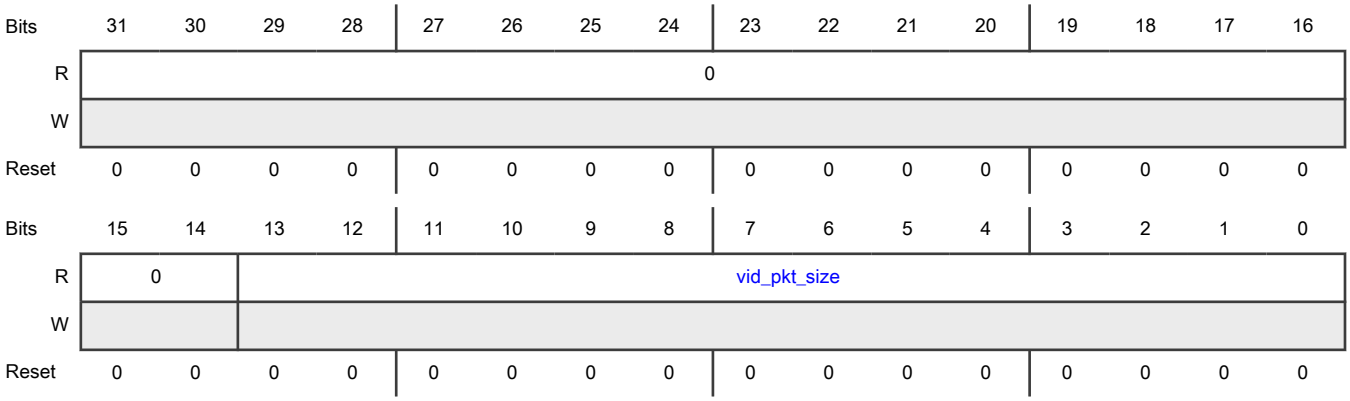
52.4.2.58 Video Packet Size Value (VID_PKT_SIZE_ACT)

Holds the value that the controller is using for VID_PKT_SIZE.

Offset

Register	Offset
VID_PKT_SIZE_ACT	13Ch

Diagram



Fields

Field	Description
31-14 —	Reserved and read as zero.
13-0 vid_pkt_size	Video Packet Size Specifies the number of pixels in a single video packet. For 18-bit not loosely packed this number must be a multiple of 4 and YCbCr Data Types, it must be a multiple of 2, as described in DSI specification.

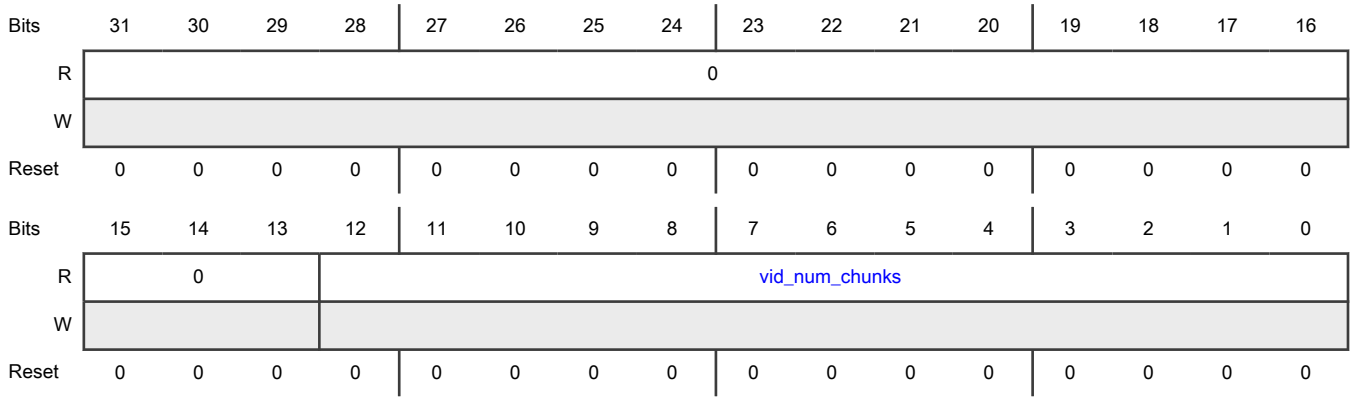
52.4.2.59 Video Number Chunks Value (VID_NUM_CHUNKS_ACT)

Holds the value that the controller is using for VID_NUM_CHUNKS.

Offset

Register	Offset
VID_NUM_CHUNKS_ACT	140h

Diagram



Fields

Field	Description
31-13 —	Reserved and read as zero.
12-0 vid_num_chunks	Video Number Chunks Specifies the number of chunks to be transmitted during a Line period (a chunk is pair made of a video packet and a null packet). If set to 0 or 1, video line is still transmitted in a single packet. If set to 1 that packet is part of a chunk, meaning that a null packet follows it (if vid_null_size>0). Otherwise, multiple chunks are used to transmit each video line.

52.4.2.60 Video Null Packets Size Value (VID_NULL_SIZE_ACT)

Holds the value that the controller is using for VID_NULL_SIZE.

Offset

Register	Offset
VID_NULL_SIZE_ACT	144h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				vid_null_size											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-13 —	Reserved and read as zero.
12-0 vid_null_size	Video Null Size Specifies the number of bytes inside a null packet. Setting to 0 disables null packets.

52.4.2.61 Video HSA Time Value (VID_HSA_TIME_ACT)

Holds the value that the controller is using for VID_HSA_TIME.

Offset

Register	Offset
VID_HSA_TIME_ACT	148h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				vid_hsa_time											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-12 —	Reserved and read as zero.
11-0 vid_hsa_time	Video HSA Time Specifies the Horizontal Synchronization Active period in lane byte clock cycles.

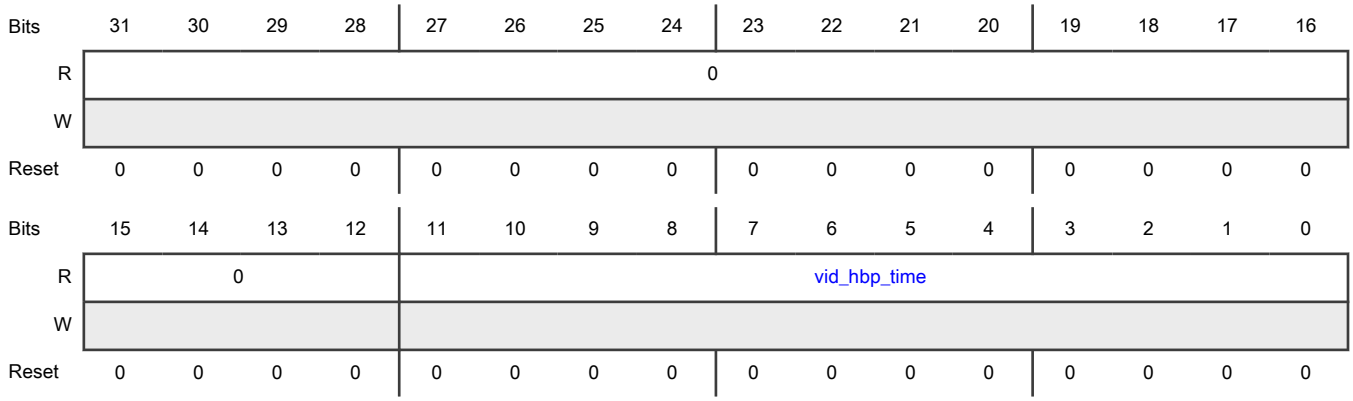
52.4.2.62 Video HBP Time Value (VID_HBP_TIME_ACT)

Holds the value that the controller is using for VID_HBP_TIME.

Offset

Register	Offset
VID_HBP_TIME_ACT	14Ch

Diagram



Fields

Field	Description
31-12 —	Reserved and read as zero.
11-0 vid_hbp_time	Video HBP Time Specifies the Horizontal Back Porch period in lane byte clock cycles.

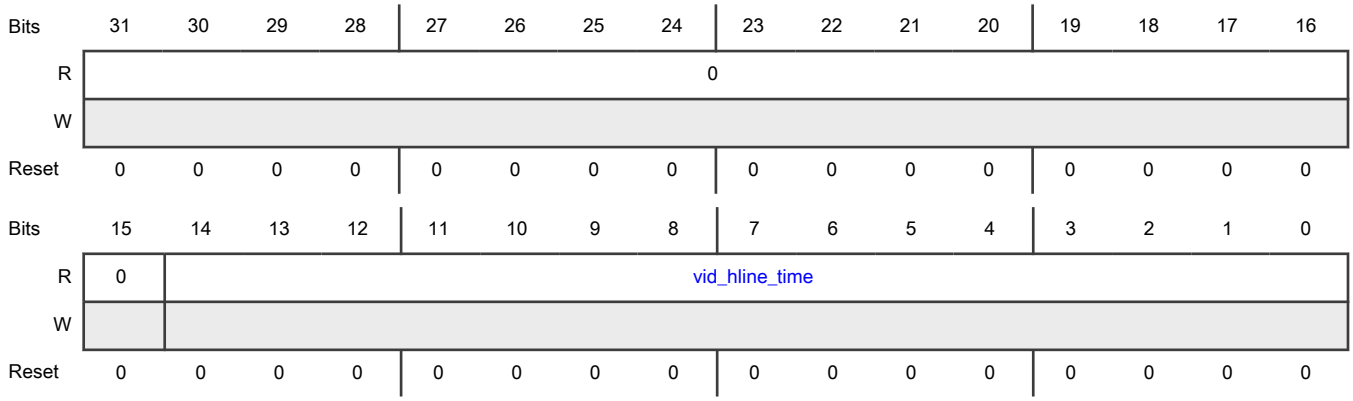
52.4.2.63 Video Line Time Value (VID_HLINE_TIME_ACT)

Holds the value that the controller is using for VID_HLINE_TIME.

Offset

Register	Offset
VID_HLINE_TIME_ACT	150h

Diagram



Fields

Field	Description
31-15 —	Reserved and read as zero.
14-0 vid_hline_time	Video HLINE Time Specifies the size of the total line time (HSA+HBP+HACT+HFP) counted in lane byte clock cycles.

52.4.2.64 VSA Period Value (VID_VSA_LINES_ACT)

Holds the value that the controller is using for VID_VSA_LINES.

Offset

Register	Offset
VID_VSA_LINES_ACT	154h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						vsa_lines									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-10 —	Reserved and read as zero.
9-0 vsa_lines	VSA Lines Specifies the Vertical Synchronization Active period measured in number of horizontal lines.

52.4.2.65 VBP Period Value (VID_VBP_LINES_ACT)

Holds the value that the controller is using for VID_VBP_LINES.

Offset

Register	Offset
VID_VBP_LINES_ACT	158h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						vbp_lines									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-10 —	Reserved and read as zero.
9-0 vbp_lines	VBP Lines Specifies the Vertical Back Porch period measured in number of horizontal lines.

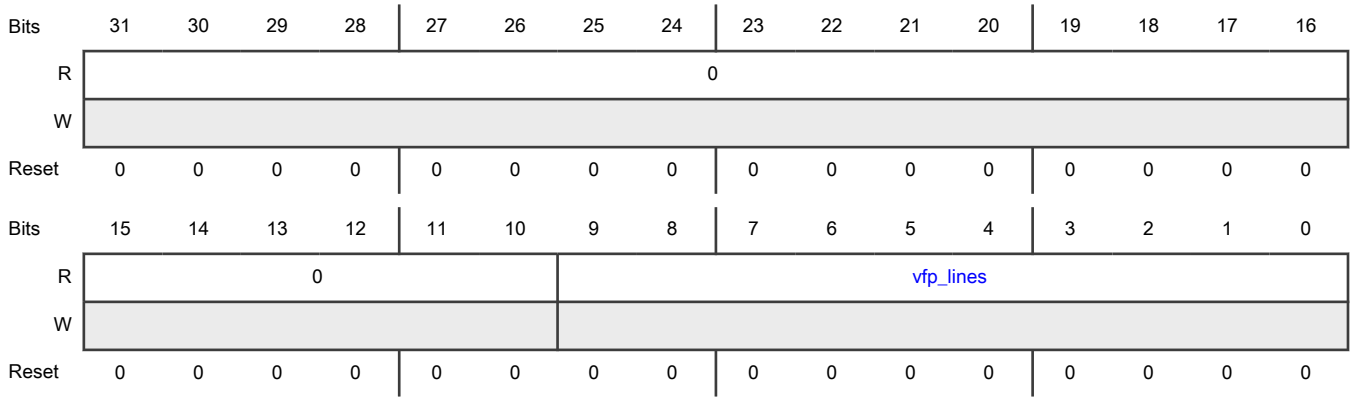
52.4.2.66 VFP Period Value (VID_VFP_LINES_ACT)

Holds the value that the controller is using for VID_VFP_LINES.

Offset

Register	Offset
VID_VFP_LINES_ACT	15Ch

Diagram



Fields

Field	Description
31-10 —	Reserved and read as zero.
9-0 vfp_lines	VFP Lines Specifies the Vertical Front Porch period measured in number of horizontal lines.

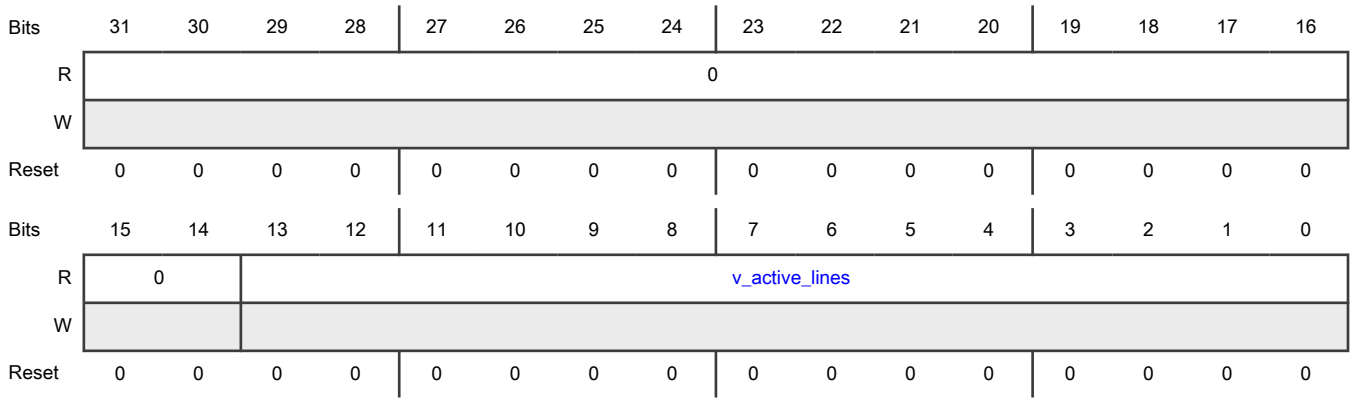
52.4.2.67 Vertical Resolution of Video Value (VID_VACTIVE_LINES_ACT)

Holds the value that the controller is using for VID_VACTIVE_LINES.

Offset

Register	Offset
VID_VACTIVE_LINES_ACT	160h

Diagram



Fields

Field	Description
31-14 —	Reserved and read as zero.
13-0 v_active_lines	Vertical Active Lines Specifies the Vertical Active period measured in number of horizontal lines.

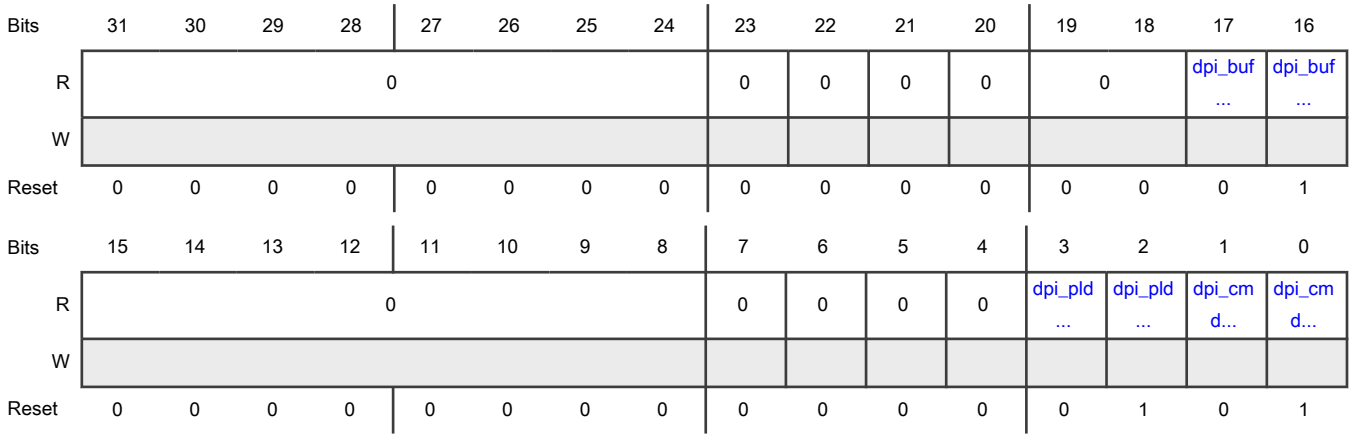
52.4.2.68 Video Packet Status (VID_PKT_STATUS)

Contains information about the status of FIFOs related to DPI and eDPI interfaces.

Offset

Register	Offset
VID_PKT_STATUS	168h

Diagram



Fields

Field	Description
31-24 —	Reserved and read as zero.
23 —	Reserved and read as zero.
22 —	Reserved and read as zero.
21 —	Reserved and read as zero.
20 —	Reserved and read as zero.
19-18 —	Reserved and read as zero.
17 dpi_buff_pld_full	DPI Buffer Payload Full Indicates the full status of the payload internal buffer for video Mode. This bit is set to 0 for command Mode.
16 dpi_buff_pld_empty	DPI Buffer Payload Empty Indicates the empty status of the payload internal buffer for video Mode. This bit is set to 0 for command Mode.
15-8 —	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 —	Reserved and read as zero.
6 —	Reserved and read as zero.
5 —	Reserved and read as zero.
4 —	Reserved and read as zero.
3 dpi_pld_w_full	DPI Payload Write Full Indicates the full status of write payload FIFO for video Mode. This bit is set to 0 for command Mode.
2 dpi_pld_w_empty	DPI Payload Write Empty Indicates the empty status of write payload FIFO for video Mode. This bit is set to 0 for command Mode.
1 dpi_cmd_w_full	DPI Command Write Full Indicates the full status of write command FIFO for video Mode. This bit is set to 0 for command Mode.
0 dpi_cmd_w_empty	DPI Command Write Empty Indicates the empty status of write command FIFO for video Mode. This bit is set to 0 for command Mode.

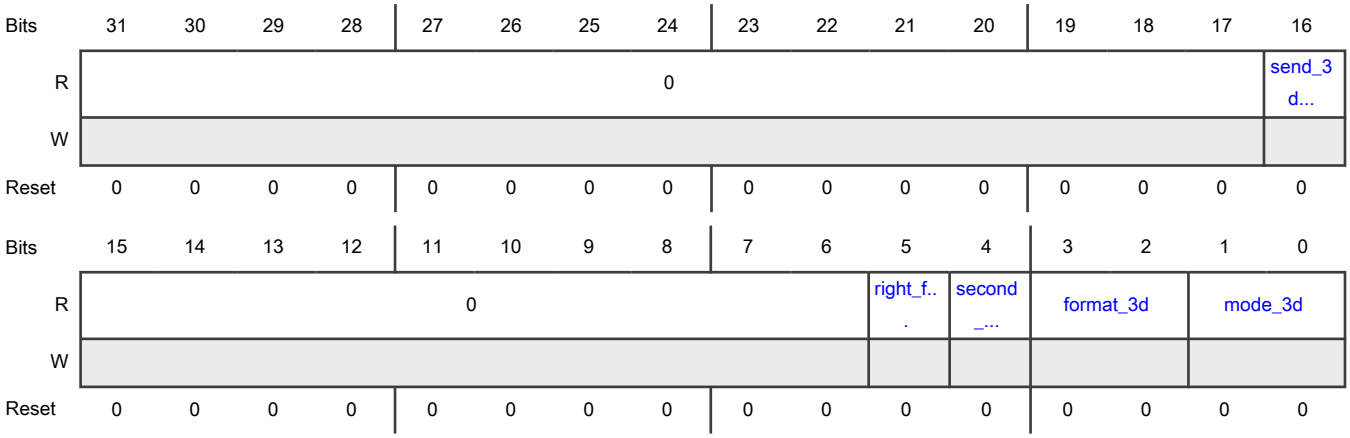
52.4.2.69 3D Control Information Value (SDF_3D_ACT)

Holds the value that the controller is using for SDF_3D.

Offset

Register	Offset
SDF_3D_ACT	190h

Diagram



Fields

Field	Description
31-17 —	Reserved and read as zero.
16 send_3d_cfg	Send 3D Configuration When set, causes the next VSS packet to include 3D control payload in every VSS packet.
15-6 —	Reserved and read as zero.
5 right_first	Right First Specifies the left/right order: 0 - Left eye is sent first, then right eye 1 - Right eye data is sent first, then left eye
4 second_vsync	Second VSYNC Specifies whether there is a second VSYNC pulse between Left and Right Images, when 3D Image Format is Frame-based: 0 - No sync pulses between left and right data. 1 - Sync pulse (HSYNC, VSYNC, blanking) between left and right data.
3-2 format_3d	Format 3D Specifies 3D Image Format: 00 - Alternating lines of left and right data 01 - Alternating frames of left and right data 10 - Alternating pixels of left and right data 11 - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
1-0 mode_3d	Mode 3D Specifies 3D Mode On/Off and Display Orientation: 00 - 3D Mode Off, 2D Mode On 01 - 3D Mode On, Portrait Orientation 10 - 3D Mode On, Landscape Orientation 11 - Reserved

Chapter 53
MIPI-DSI DPHY

53.1 Overview

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

The MIPI 4-lane Tx D-PHY module implements the physical layer for the MIPI D-PHY interface. It is targeted for high-speed data digital transmission between a host processor and display drivers or camera interfaces. The module includes four data lanes and one clock lane.

53.1.1 Block diagram

The block diagram of the MIPI Tx D-PHY is given below.

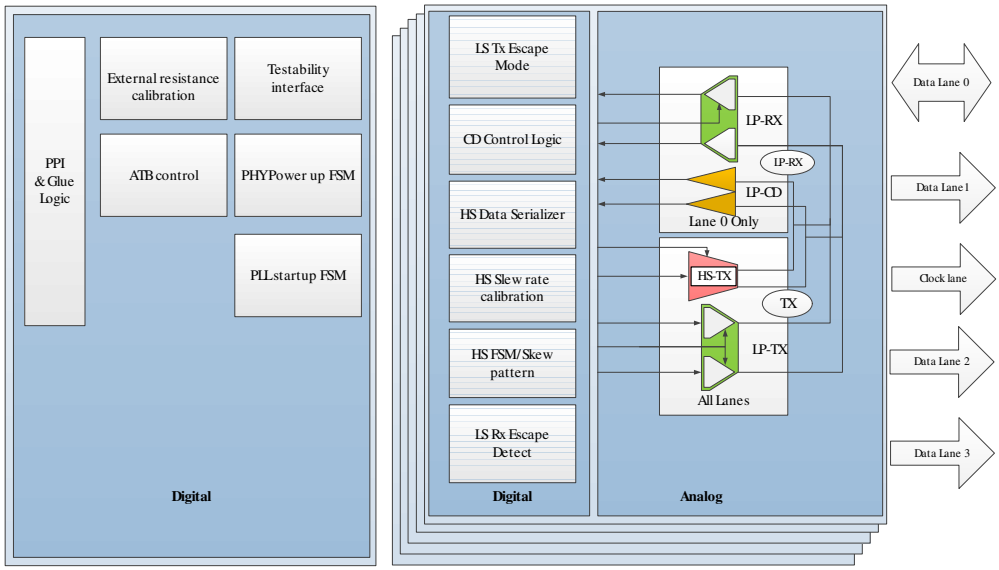


Figure 284. MIPI Tx D-PHY block diagram

- HS Driver for each lane to implement high-speed Tx functionality
- LP Driver for each lane to the low-power Tx functionality
- LP Receiver only on data lane 0
- PLL to generate high-speed clocks required in High Speed operation mode
- Logic related to analog to digital interface, high-speed data path, calibration algorithms, data alignment, and PPI logic
- Logic responsible for PHY power up, resistor termination calibration, and tester interface

53.1.2 Features

The MIPI Tx D-PHY includes the following features:

- Compliant to MIPI D-PHY interface specification, revision 1.2
- Lane operation ranging from 80 Mbps to 1.5 Gbps in forward direction
- Aggregate throughput up to 6 Gbps with four data lanes
- Maximum LP data rate supported of 10 Mbps

- PHY-Protocol Interface (PPI) for clock and data lanes
- Low-power Escape modes and Ultra Low Power state
- Optional usage of external reference resistor (200 ohm)

53.2 Functional description

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

The following sections describe functional details of the MIPI Tx D-PHY module.

53.2.1 External resistor calibration

The MIPI TX D-PHY supports the use of an optional external reference resistor. The calibration machine relies on external resistor Rext (200 Ω ±1%). The tx_cb_cal_repl[3:0] bits are progressively enabled and disabled resulting in an overall internal replica resistance value higher or lower than Rext. The results “higher than” and “lower than” are the output of internal comparator compared to Rext.

53.2.2 Power up and ULPS control FSM

The PHY supports a Finite State Machine (FSM) that defines the power up states and controls for ULPS. The power up flow diagram is provided below.

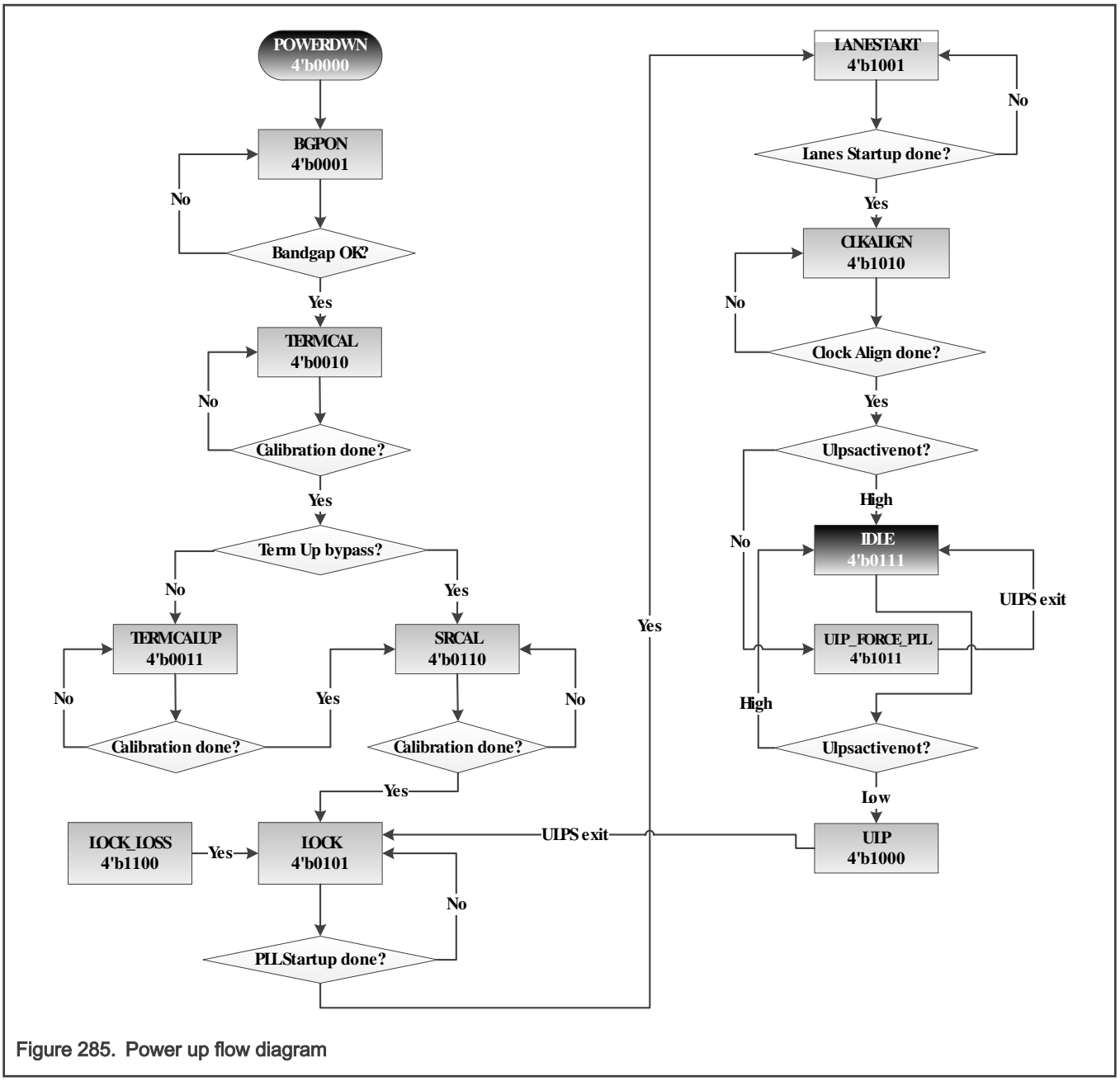


Figure 285. Power up flow diagram

53.2.3 Operating Modes

This section describes the various operating modes of the PHY.

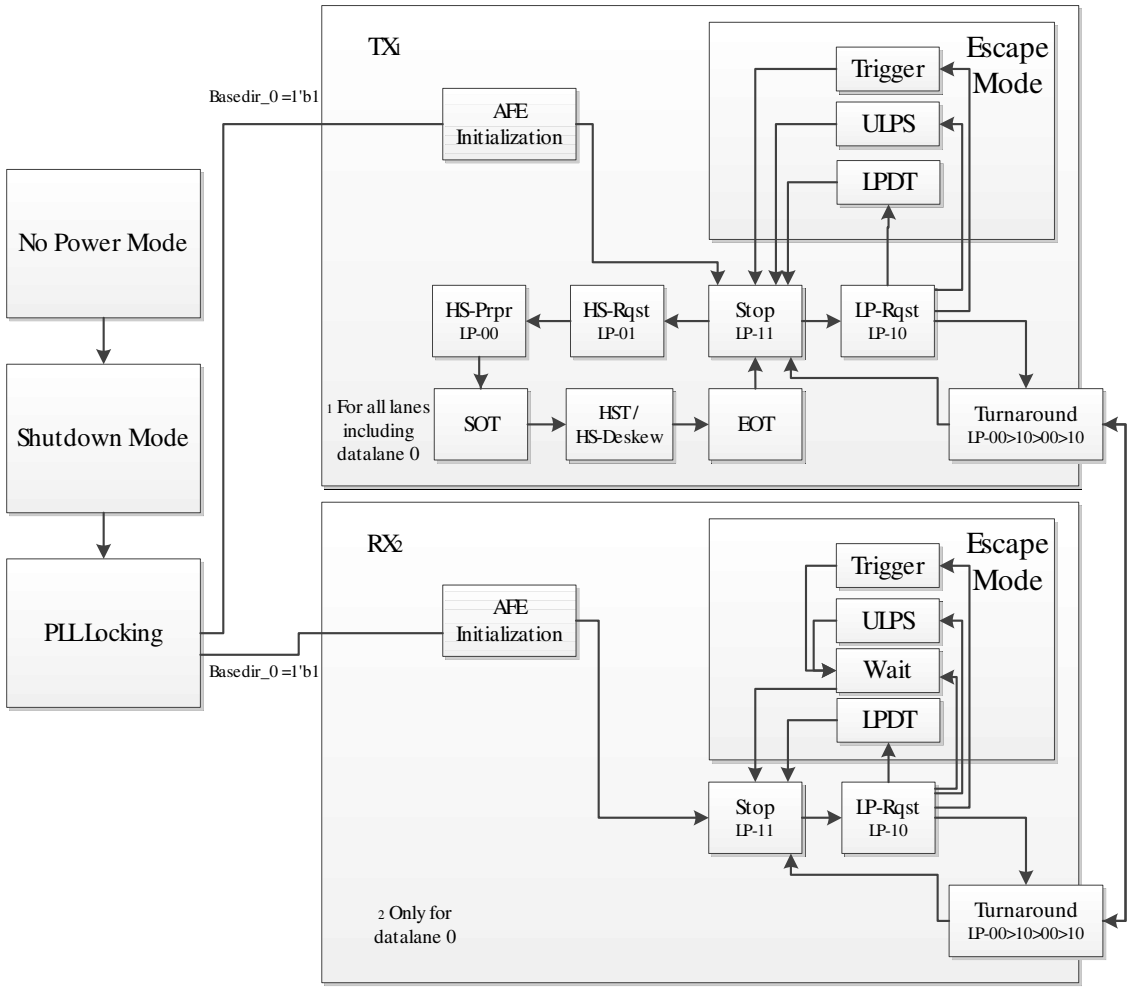


Figure 286. Rx D-PHY Operating Modes

53.2.3.1 Initialization

53.2.3.1.1 No Power Mode

The No Power mode is characterized by the non-existence of any supply voltage applied to the PHY. In order to get to the powered modes, proper voltages should be applied to the PHY.

53.2.3.1.2 Shutdown Mode

This mode is the lowest power consumption mode, where all analog blocks are disabled, and digital logic is reset. The shutdown mode is entered when rstz and shutdownz are in the low state. In this mode, the differential lines datap/datan and clkp/clkn are high impedance (Hi-Z).

All necessary MIPI D-PHY configurations to set the desired operation should be made with the MIPI D-PHY on this mode, before power-up.

53.2.3.2 Active modes

53.2.3.2.1 Control Mode

Control mode is the default operating mode. After initialization is completed, the D-PHY remains in this default mode until a request is made either by the protocol layer, or directly through the sequence of low power signals in the lanes in case of RX. While in control mode, the transmitter side sets the LP-11 state in the lines. The LP-11 state is also called the Stop state.

The receiver side remains in control mode while receiving LP-11 in the lines. Any request must start from and end in Stop state. Following a request, a lane can leave control mode for either high-speed data transfer mode, Escape mode, or Ultra Low Power state.

53.2.3.2.2 High-Speed Data Reception

Once the initialization sequence is completed, the D-PHY remains in control mode, which is the default operating mode, until some request appears. High-speed is one of the possible requests at this point. High-speed data transfer occurs in bursts. Only during these bursts that the lane is in high-speed mode. A high-speed burst must start from and return to a Stop state (control mode). A high-speed burst allows for the transmission of payload data by the data lanes.

A valid DDR clock in the clock lane is active during data reception.

53.2.3.2.3 Escape Mode

Escape mode is a special mode of operation that uses the data lanes to communicate using the low-power states at low-speed. The PHY supports this mode in both directions for data lane 0 and only in forward direction for the other lanes. A data lane enters the Escape mode through an Escape mode entry procedure (LP-11, LP-10, LP-00, LP-01, LP-00). If an LP-11 is detected before reaching LP-00 state, the entry is aborted and the receiver returns to the Stop state. Once the sequence is correctly completed, the transmitter sends an 8-bit command to indicate a requested action.

Table 516 shows the Escape mode supported actions. If the entry command is not valid, it is ignored, ERRESC error flag goes high, and the receiver waits until the transmitter returns to the Stop state. The PHY applies Spaced-One-Hot encoding (a Mark state is interleaved with a Space state) on commands and data.

Each symbol consists of the following two parts:

- One-Hot phase
- Space state to transmit one bit, a Mark-1 should be sent followed by the Space state. In the case of a zero bit, a Mark-0 should be sent followed by Space state.

Table 516. Escape Mode Sequences for Data Lanes

Escape Mode Action	Entry Command Pattern (First Bit to Last Bit to be Transmitted)	Command Type
Low-Power Data Transmission	8'b11100001	mode
Ultra- Lower Power State	8'b00011110	mode
Reset Trigger	8'b01100010	trigger
Unknown - 3	8'b01011101	trigger
Unknown - 4	8'b00100001	trigger
Unknown - 5	8'b10100000	trigger

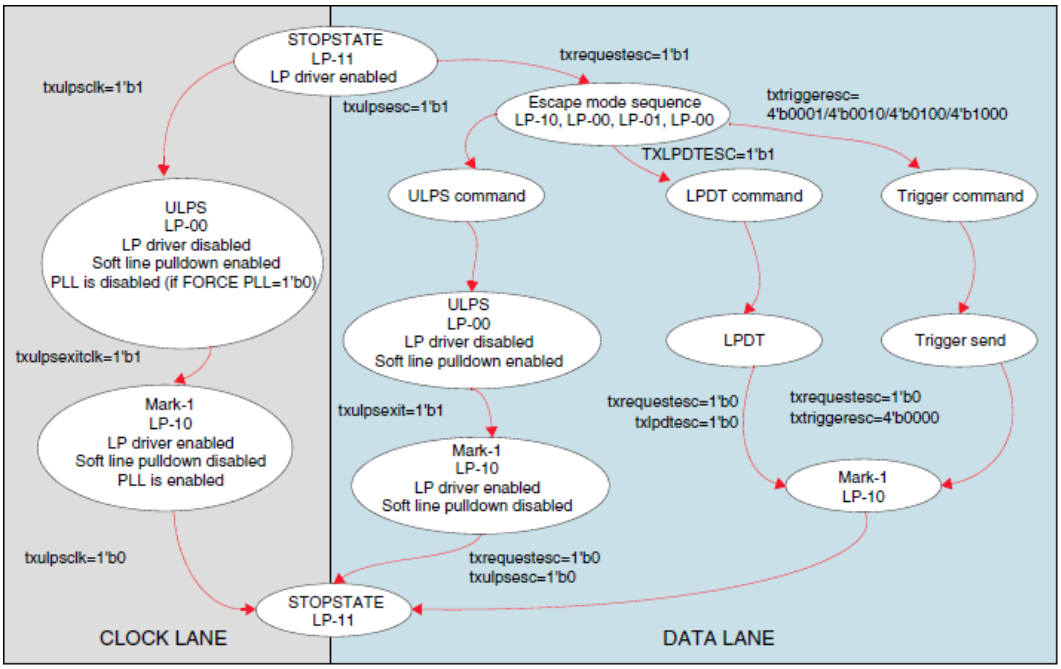


Figure 287. Escape Mode Sequences State Diagram

53.2.3.2.3.1 Low-Power Data Transmission (LPDT)

In LPDT mode, the data can be transmitted by the protocol at low speed in Low-Power mode—high-speed receivers are off and low-power receivers are on. During LPDT, the protocol can pause by maintaining a Space state on the lines. The last state before exiting Escape mode is Mark-1—not considered as 1 bit— because it is followed by the Stop state.

Data shall be encoded on the lines with the same Spaced-One-Hot code as used for the Entry Commands. This encoding allows data to be self-clocked and does not rely on the clock lane.

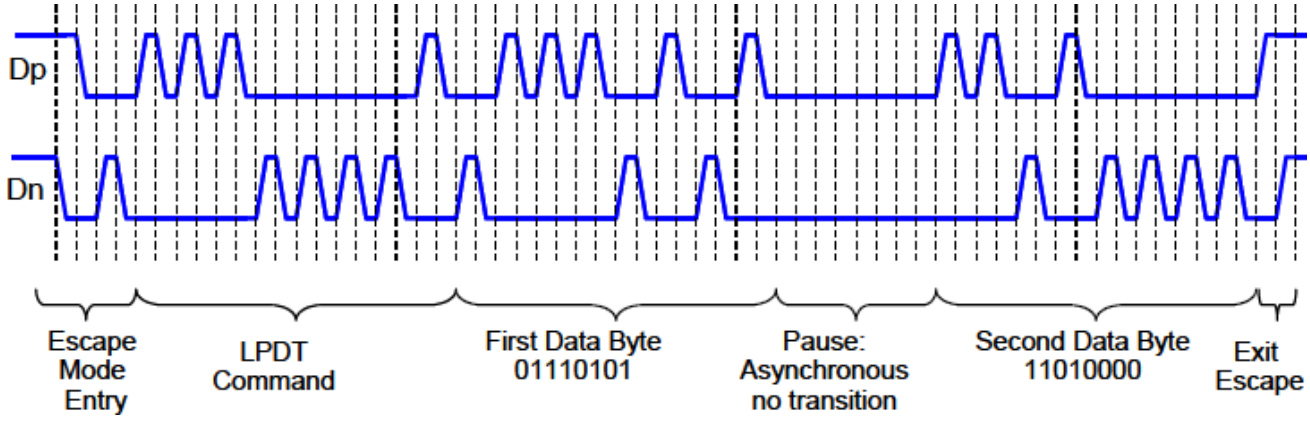


Figure 288. LPDT Example

53.2.3.2.3.2 Remote Trigger

The Remote Trigger mode allows the protocol to send a flag to the receiving side, on request of the transmitting side. If the Entry Command pattern matches the Reset-Trigger Command a Trigger is flagged to the protocol at the receive side via the logical PPI. Any bit received after a Trigger Command but before the Lines go to Stop state shall be ignored. Therefore, dummy bytes can be concatenated in order to provide clock information to the receive side.

53.2.3.2.3.3 Ultra Low Power State (ULPS)

The ULPS mode has the lowest power consumption, excluding the Shutdown mode. For data lanes, this mode is entered by sending an Ultra Lower Power State entry command, after the Escape mode entry command. During this mode, the lines are in the Space state (LP-00). Although the clock lane does not support regular Escape mode, the clock lane supports ULPS.

To exit ULPS, the transmitter side drives a Mark-1 followed by a Stop state.

53.2.3.2.4 Turnaround

The PHY allows the transmission direction of the data lane 0 to be swapped by means of a turnaround request. The sequence of a turnaround operation occurs as follows:

- Transmitter side
  - Starts when turnrequest_0 input is set high on transmitter side, assuming the PHY is in Stop state
  - Transmitter sends specific low-power sequence—LP-11, LP-10, LP-00, L-10, L-00
- Receiver side
  - After receiving complete low-power sequence, receiver enables low-power driver
  - Transmits LP-00, indicating change in direction_0 output low
  - Overlapping LP-00 transmitted by initial transmitter
  - Initial receiver side transmits LP-10, followed by LP-11, returning to Control mode

The turnaround procedure is the same for a Forward-to-Reverse direction or Reverse-to-Forward direction. However, in the Reverse-to-Forward direction, lane0 only supports Escape Mode. Figure 289 shows the state diagram for a turnaround sequence.

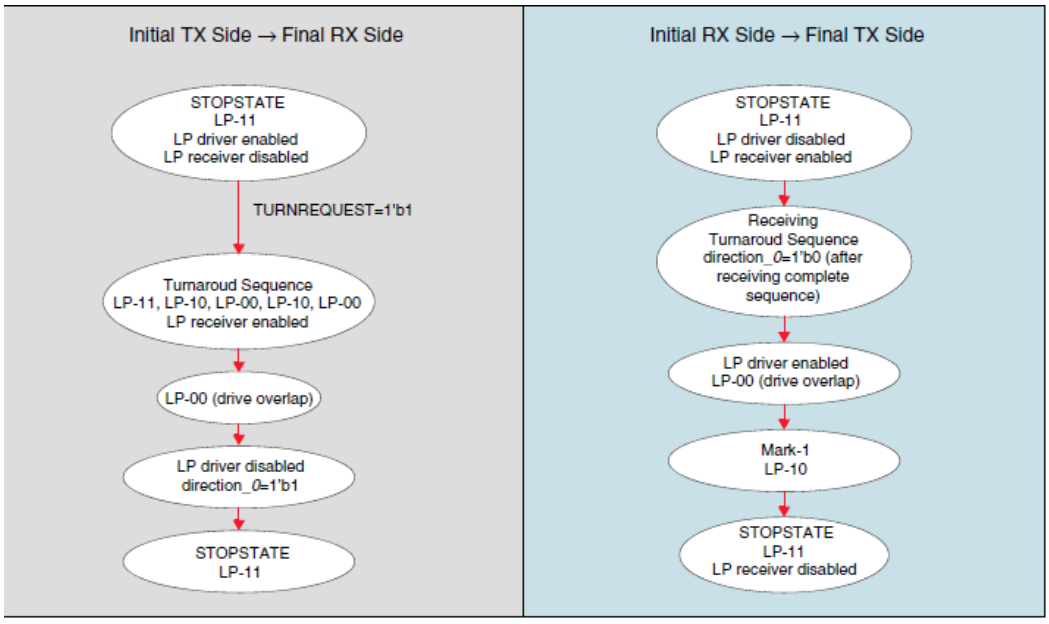


Figure 289. Turnaround Sequence State Diagram

53.3 Initialization

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

53.3.1 Start-up sequence

The D-PHY operating frequency and input reference clock frequency need to be configured through hsfreqrange[6:0], and cfgclkfreqrange[5:0].

hsfreqrange can only be configured before the D-PHY power-up. It cannot be changed during active operation.

cfgclkfreqrange[5:0] needs to be configured according to the following formula: cfgclkfreqrange[5:0] = round[(Fcfg_clk(MHz) - 17) * 4].

Chapter 54
MIPI-CSI2

54.1 Overview

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

The MIPI_CSI2 controller implements the CSI-2 protocol on the host side. The MIPI_CSI2 controller receives data from a CSI-2 compliant camera sensor.

54.1.1 Block Diagram

The block diagram for the MIPI_CSI2 controller is shown below.

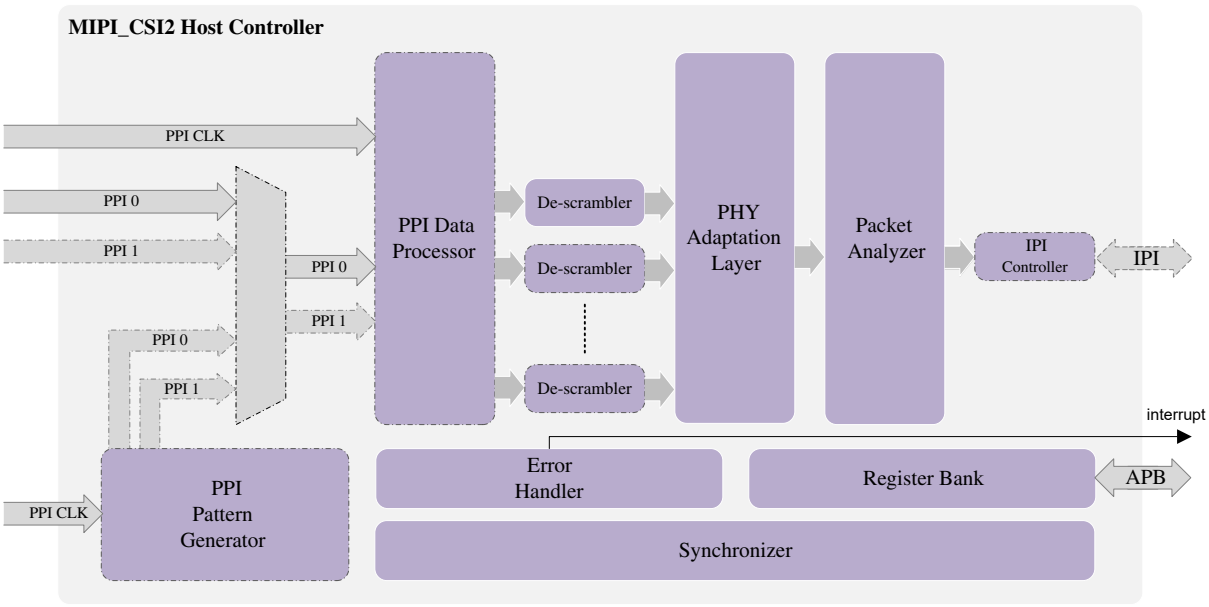


Figure 290. MIPI_CSI2 controller

The main blocks of MIPI_CSI2 are as follows:

- PPI Data Processor: This module assists in synchronizing the lanes and to monitor the timing requirements.
  - Pipeline: Registers the PPI high-speed related signals (`rxdatahs_*`, `rxvalidhs_*`, `errsoths_*`, and `errsotsynchs_*`) to avoid possible SoC timing constraints. Remaining signals are not registered because they are considered asynchronous inside the MIPI_CSI2 controller.
- PPI Pattern Generator: Through pattern injection, it provides an effortless debugging post-validation practice of PPI.
- De-scrambler: This module exploits the LFSR seeds, used by the transmitter, to convert PPI scrambled data to its original value. For more information see, De-scrambler.
- PHY Adaptation Layer: It is responsible for managing the PHY interface, including PHY error handling.
- Packet Analyzer: It implements data lane merging, along with header decoding, error detection and correction, frame size error detection, and CRC error detection.

- Image Pixel Interface (IPI) Controller: This module converts the received CSI-2 packets, from byte to pixel format, and produces a pixel data bus containing vertical and horizontal synchronism information. For more information see, Image Pixel Interface (IPI).
- Error Handler: This module notifies and monitors the error conditions on the CSI-2 link.
- Register Bank: Use the AMBA APB device interface to access the MIPI_CSI2 controller registers. Program the MIPI_CSI2 controller registers through the AMBA APB device interface to configure and control the behavior of the MIPI_CSI2 controller. The MIPI_CSI2 controller provides a fully programmable interrupt generator to notify the system about certain specific events.
- Synchronizer: This block contains all the synchronizers used to implement CDC functions.

The MIPI_CSI2 controller has the following interfaces:

- Image Pixel Interface (IPI) that provides:
  - 48-bit parallel bus operating at pixel clock rate and delivering one pixel per clock cycle
  - Vertical and Horizontal timing accurate video synchronization signals
- AMBA APB device bus for configuration purposes
- PHY-Protocol Interface (PPI)
  - D-PHY PPI interface as recommended in the *MIPI D-PHY Specification* using PPI-8

54.1.2 Features

The MIPI_CSI2 controller supports the following features:

- PHY-Protocol Interface (PPI) Pattern Generator with programmable packet-to-packet time
- Configurable pipeline interface between the PHY and MIPI_CSI2 controller
  - 1 pipeline stage
- Support for automatic D-PHY integration in non-automotive configurations
- Programmable value for the number of synchronization stages used for Clock Crossing Domain (CDC)
- Image Pixel Interface (IPI)
  - Two operating modes:
    - **Camera Timing** - The frame timing signals, and the vertical or horizontal synchronism are generated based on the synchronization of Short Packets received from the sensor.
    - **Controller Timing** - The frame timing signals are generated based on the IPI registers.
  - Generates pixel stream in two different modes:
    - 48-Bit
    - 16-Bit
  - Supports several data formats:
    - RGB
    - YUV
    - RAW
    - User defined
    - Embedded data (when operating in Camera Timing mode and only with RAW image data)
  - Data decoding based on configurable data type
  - Additional pins that provide useful information:
    - End-of-Line indication

- Number of valid pixels/bytes transmitted per clock cycle
- First and Last Data Valid Indications
- End-of-Frame indication

— Possibility to flush IPI memory (automatically or manually)

— Possibility to ignore Frame Start as a synchronization event

— Possibility to select Packets used for IPI Synchronism Events

— Possibility to reduce memory requirements, down to the minimum FIFO depth of 32

— Back-pressure mechanism

54.2 Functional Description

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

54.2.1 PPI Data Processor

If data lanes get out of sync with the `rxbyteclkhs` or `rxwordclkhs` clock, there is a skew between them. This can be caused by using multiple PHYs to emulate more lanes or due to place-and-route issues. The MIPI_CSI2 controller syncs the lanes and removes the skew.

54.2.1.1 Pipeline Stage on the PPI

When you select a pipeline stage on the PPI interface, the controller adds a buffer with a configurable length to a subset of the PPI signals. The controller buffers the following inputs:

- `rxdatahs`
- `rxvalidhs`
- `errsoths`
- `errsotsynchs`

The intent of this pipeline is to decrease the timing requirements for the interface between PHY and controller.

54.2.2 De-scrambler

Data Scrambling is used to mitigate the effects of EMI and RF self-interference by spreading the information transmission energy of the link over a possibly large frequency band, using a data randomization technique. Transmitted data is scrambled with a Pseudo-Random-Bit-Stream (PRBS) to reduce the likelihood of a repetitive pattern on a lane basis.

The PRBS is generated using the Galois form of a Linear Feedback Shift Register (LSFR) that implements the generator polynomial:

$G(x) = x^{16} + x^5 + x^4 + x^3 + 1$

Each data lane has an instance of the de-scrambling block. For flexibility, each lane can be configured to have a different 16-bit LSFR seed by the `SCRAMBLING_SEEDX` registers where X is the number of the lanes.

The recommended initial D-PHY seed values to initialize the scrambler LFSR in lanes 1 through 2 are specified in the *MIPI CSI-2 Specification*.

Data is de-scrambled when `rxvalidhs` is high and the LFSR seed is reset before its rising edge as shown in Figure 291.

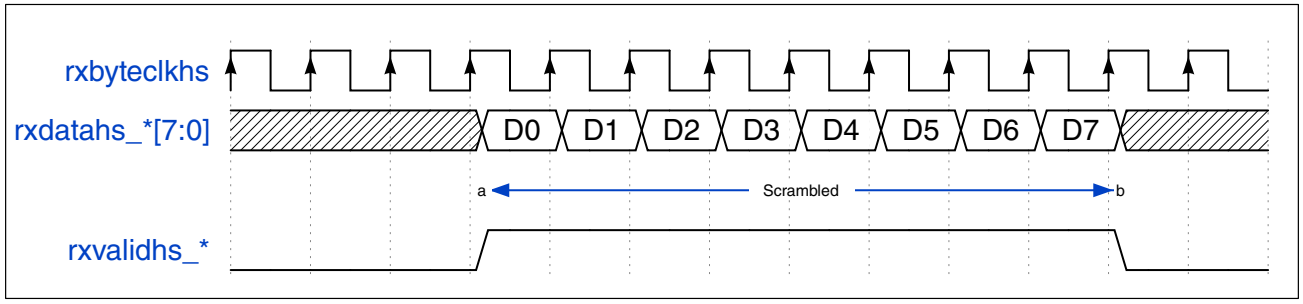


Figure 291. Scrambling and Seed Re-initialization Timing

54.2.3 Configuring the De-scrambler

The `scramble_enable` field of the `SCRAMBLING` register enables this feature. Use the `SCRAMBLING_SEED1` to `SCRAMBLING_SEED2` registers to configure the seed used by de-scrambler.

54.2.4 Memory

The 2-port RAMs are used by the MIPI_CSI2 controller to implement FIFOs for data storage and synchronization.

Figure 292 shows the timing diagram of the MIPI_CSI2 controller memory access.

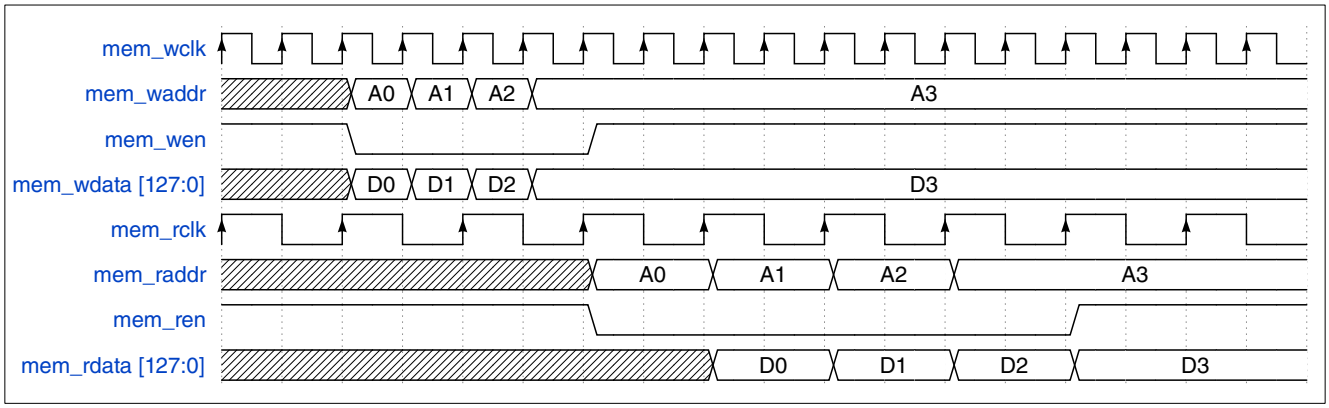


Figure 292. Timing Representation of Memory Access

54.2.5 Error Detection

The MIPI_CSI2 controller analyzes the received packets and determines if there are protocol errors. It is possible to monitor the following errors:

- PHY errors: Such as synchronization pattern mismatch.
- Packet errors:
  - Such as Payload CRC and ECC (D-PHY)
- Frame errors: Such as incorrect frame sequence, reception of a CRC error in the most recent frame, and the mismatch between Frame Start and Frame End.
- IPI-level errors: Errors that occur on the IPI.

Table 517. Error list and handling

Register	Field	Description	Error Frequency	Image Pixel Interface (IPI)	
				Action/ Consequence	Recommended Reset
INT_ST_PHY_FATAL	phy_errsotsynchs_*	Start of transmission error on data lane * (no synchronization achieved)	Single	Follow PHY recommendation	_
			Multiple	Follow PHY recommendation	csi2_resetn_rw dphy_rstz
INT_ST_PKT_FATAL	err_ecc_double	D-PHY: Header ECC contains at least 2 errors, unrecoverable.	Single	Reset is recommended if ipi_auto_flush is disabled. Depending on the packet, full frame can be compromised. If ipi_auto_flush is enabled, recovers on next frame.	ipi_softrstn
			Multiple	Reset is recommended. May indicate ecosystem issue.	csi2_resetn_rw dphy_rstz
INT_ST_BNDRY_FRAME_FATAL	err_vc*	Error matching Frame Start with Frame End for virtual channel *.	Single	Reset is recommended.	ipi_softrstn
			Multiple	Reset is recommended. May indicate ecosystem issue.	csi2_resetn_rw dphy_rstz
INT_ST_SEQ_FRAME_FATAL	err_f_seq_vc*	Incorrect Frame sequence detected in virtual channel *.	Single	Reset is recommended if ipi_auto_flush is disabled. Depending on the packet, full frame can be compromised. If ipi_auto_flush is enabled, recovers on next frame.	ipi_softrstn

Table continues on the next page...

Table 517. Error list and handling (continued)

Register	Field	Description	Error Frequency	Image Pixel Interface (IPI)	
				Action/ Consequence	Recommended Reset
			Multiple	Reset is recommended. May indicate ecosystem issue.	csi2_resetn_rw dphy_rstz
INT_ST_CRC_FRAME_FATAL	err_frame_data_vc*	Last received Frame in virtual channel *, had at least one CRC error.	Single	Reset is recommended if ipi_auto_flush is disabled. Depending on the packet, full frame can be compromised. If ipi_auto_flush is enabled, recovers on next frame.	ipi_softrstn
			Multiple	Reset is recommended. May indicate ecosystem issue.	csi2_resetn_rw dphy_rstz
INT_ST_PLD_CRC_FATAL	err_crc_vc*	Payload Checksum error detected on virtual channel *.	Single	Reset is recommended if ipi_auto_flush is disabled. Depending on the packet, full frame can be compromised. If ipi_auto_flush is enabled, recovers on next frame.	ipi_softrstn
			Multiple	Reset is recommended. May indicate ecosystem issue.	csi2_resetn_rw dphy_rstz
INT_ST_IPI_FATAL	int_pulse_delay_overflow	Reporting internal fifo overflow on pulse delay block.	Single	Reset is recommended if ipi_auto_flush is disabled. Review IPI Internal Pulse	ipi_softrstn

Table continues on the next page...

Table 517. Error list and handling (continued)

Register	Field	Description	Error Frequency	Image Pixel Interface (IPI)	
				Action/ Consequence	Recommended Reset
				Delay Configuration. Data could be corrupted. If ipi_auto_flush is enabled, recovers on next frame.	
			Multiple	Reset is recommended. May indicate ecosystem issue. Reviewing TX and RX configurations is needed.	csi2_resetn_rw
INT_ST_IPI_FATAL	int_event_fifo_overflow	Reporting internal fifo overflow.	Single	Reset is recommended if ipi_auto_flush is disabled. Review IPI Internal FIFO Configuration. Data could be corrupted. If ipi_auto_flush is enabled recovers on next frame.	ipi_softrstn
			Multiple	Reset is recommended. May indicate ecosystem issue. Reviewing TX and RX configurations is needed.	csi2_resetn_rw
INT_ST_IPI_FATAL	pixel_if_fifo_nempty_fs	Camera timing mode: The FIFO of pixel interface is not	Single	Reset is recommended if ipi_auto_flush is disabled.	ipi_softrstn

Table continues on the next page...

Table 517. Error list and handling (continued)

Register	Field	Description	Error Frequency	Image Pixel Interface (IPI)	
				Action/ Consequence	Recommended Reset
		empty at the start of a new frame. There are some scenarios where synchronization events of new frame can be lost on IPI interface.		Review IPI configuration. If ipi_auto_flush is enabled recovers on next frame.	
			Multiple	Reset is recommended. May indicate ecosystem issue. Reviewing TX and RX configurations is needed.	csi2_resetn_rw
INT_ST_IPI_FATAL	pixel_if_fifo_overflow	The FIFO of pixel interface has lost information because some data arrived and FIFO is already full.	Single	Reset is recommended if ipi_auto_flush is disabled. Review IPI configuration. If ipi_auto_flush is enabled, recovers on next frame.	ipi_softrstn
			Multiple	Reset is recommended. May indicate ecosystem issue.	csi2_resetn_rw
INT_ST_IPI_FATAL	pixel_if_fifo_underflow	The FIFO has become empty before the expected number of pixels (calculated from the packet header) could be extracted to the pixel interface.	Single	Reset is recommended if ipi_auto_flush is disabled. Review IPI and PHY configuration. If ipi_auto_flush is enabled recovers on next frame.	ipi_softrstn
			Multiple	Reset is recommended. May indicate ecosystem issue.	csi2_resetn_rw

Table continues on the next page...

Table 517. Error list and handling (continued)

Register	Field	Description	Error Frequency	Image Pixel Interface (IPI)	
				Action/ Consequence	Recommended Reset
INT_ST_PHY	phy_erresc*	Escape Entry Error on data lane *.	Single	Follow PHY recommendation.	-
			Multiple	Reset is recommended.	csi2_resetn_rw dphy_rstz
INT_ST_PHY	phy_errsoths_*	Start of transmission error on data lane * (synchronization can still be achieved).	Single	Follow PHY recommendation.	-
			Multiple	Reset is recommended.	csi2_resetn_rw dphy_rstz
INT_ST_DATA_ID	err_id_vc*	Unrecognized or unimplemented data type detected in virtual channel*.	Single	Packet discarded, no impact.	-
			Multiple	Check configurations is recommended. May indicate ecosystem issue.	-
INT_ST_ECC_CORRECTED	err_ecc_corrected*	D-PHY: Header error detected and corrected on virtual channel*.	Single	No impact on the controller.	-
			Multiple	No impact on the controller. May indicate link degradation.	-

54.2.6 System Data Interfaces

The MIPI_CSI2 controller implements the following data interface:

- Image Pixel Interface (IPI): Provides timing information and pixel data.

54.2.6.1 Image Pixel Interface (IPI)

IPI is a interface to deliver:

- Timing Information
- Pixel Data

The IPI stores the data received from IDI controller in memory. It reorganizes the data retrieved from memory into a pixel stream format. The controller conveys this stream to the interface through a pixel data bus and additional frame timing signals containing vertical and horizontal synchronism information.

The Image Pixel Interface requires an external pixel clock. The two operating modes of IPI for the generation of frames are: Camera Timing and Controller Timing.

IPI interface contains a FIFO architecture which requires a 2-port RAM instantiation. This RAM allows the MIPI_CSI2 controller to buffer the incoming data until it is necessary to process pixels to the output.

The IPI monitors packets that match the virtual channel, programmed in the `IPI_VCID` register. All other Virtual Channel identifier packets are ignored. This interface interprets Frame Start, Frame End, Line Start, Line End, and blanking packets to gather timing information.

Based on `IPI_DATA_TYPE` register configuration, the IPI accepts only one of the following image data types:

- RGB
- YUV
- RAW
- User defined
- Embedded data (when operating in Camera Timing mode and only with RAW image data)

54.2.6.1.1 Operating Modes

The IPI has the following operating modes for the generation of frame timing signals:

- Camera Timing: In this mode, the frame timing signals and the vertical or horizontal synchronism are generated based on the synchronization of short packets received from the sensor.
- Controller Timing: In this mode, the frame timing signals are generated based on the IPI registers.

These modes can be configured through the `IPI_MODE[0]` register.

54.2.6.1.1.1 Camera Timing

In the Camera Timing mode, the frame timing signals, and vertical or horizontal synchronism, are generated based on the packets such as Frame Start, Line Start, and blanking packets. This mode provides a data stream in a pixel-based format.

54.2.6.1.1.1.1 Vertical Timing Information

According to the *MIPI CSI-2 Specification*, the presence of vertical timing information in the CSI-2 link is mandatory because camera devices must transmit the respective vertical frame synchronism packets on every frame - Frame Start (FS) and Frame End (FE) are mandatory while Line Start (LS) and Line End (LE) are optional.

Frame Start (FS) is the first event of the frame with timing relevance. As shown in Figure 293, by default `IPI_ADV_FEATURES[24]` register field is 0, so IPI does not interpret this packet as a synchronization event. As shown in Figure 294, when the `IPI_ADV_FEATURES[24]` register field is 1, Frame Start triggers a Vertical Synchronization event on IPI.

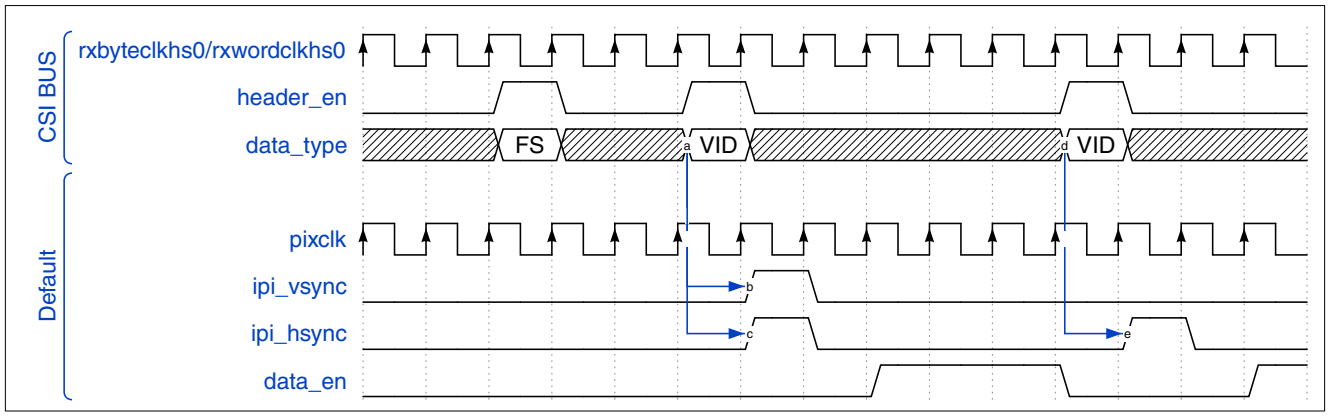


Figure 293. IPI_ADV_FEATURES[24] Set to 0 (Default)

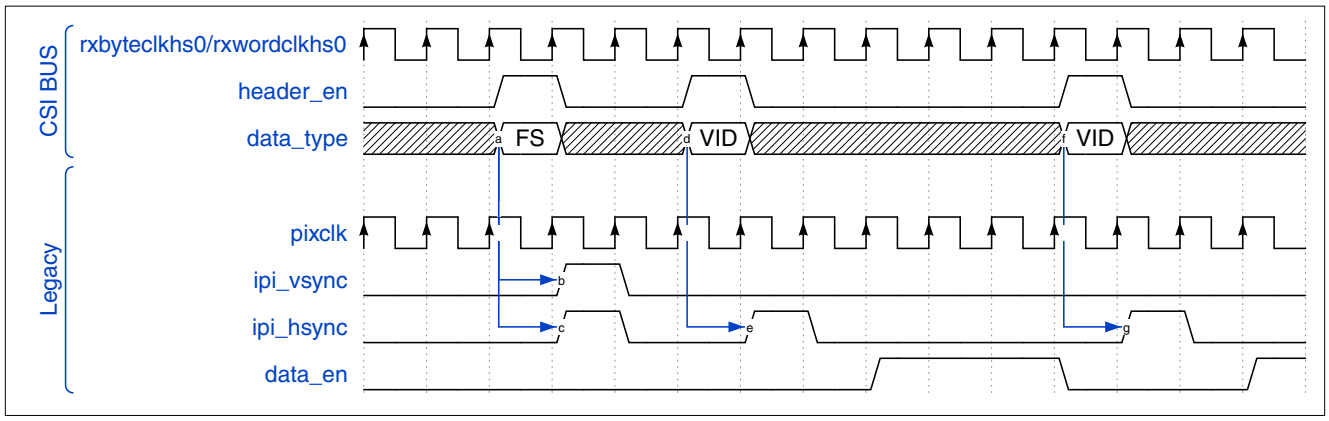


Figure 294. IPI_ADV_FEATURES[24] Set to 1 (Legacy)

When the `IPI_ADV_FEATURES[24]` register field is 0, IPI chooses the first packet received after Frame Start as a reference to create synchronism signals for the first line of the current frame. The chosen CSI packet can be any one of the following:

- Line Start
- Video
- Null
- Blanking
- Embedded

Synchronism signals for the remaining lines of the frame are created based on the programmed value in the `IPI_ADV_FEATURES[16]` register field. Based on the value of `IPI_ADV_FEATURES[16]` register field, the types of selection are as follows:

- Auto Selection: When `IPI_ADV_FEATURES[16]` register field is 0, the controller uses the first packet received after Frame Start to create the synchronism signals at IPI. In this mode, the controller assumes that a line pattern exists. For example, include both the Line Start (LS) and Line End (LE) synchronization packets in each line of the frame, or include the blanking packets in the beginning of each line of the frame. That is, for a specific frame, each line starts with the same packet data type.
- Manual Selection: When `IPI_ADV_FEATURES[16]` register field is 1, IPI synchronism signals are generated based on the value specified in the `IPI_ADV_FEATURES[17:21]` register fields. Table 518 describes the mapping between `IPI_ADV_FEATURES[17:21]` and the packets used for the IPI synchronism signals.

Table 518. Packets for IPI Synchronism Signals and IPI_ADV_FEATURES[17:21] Mapping

IPI_ADV_FEATURES[17:21]	Packets used for IPI Synchronism Events
17	Video
18	Line Start
19	Null
20	Blanking
21	Embedded

As illustrated in Figure 295, it is possible to choose more than one event. When the `IPI_ADV_FEATURES[16]` register field is 0, that is, the Line Event selection is disabled, then the events selected in `IPI_ADV_FEATURES[17:21]` register fields are ignored.

Figure 295 to Figure 298 illustrate some of the possible scenarios.

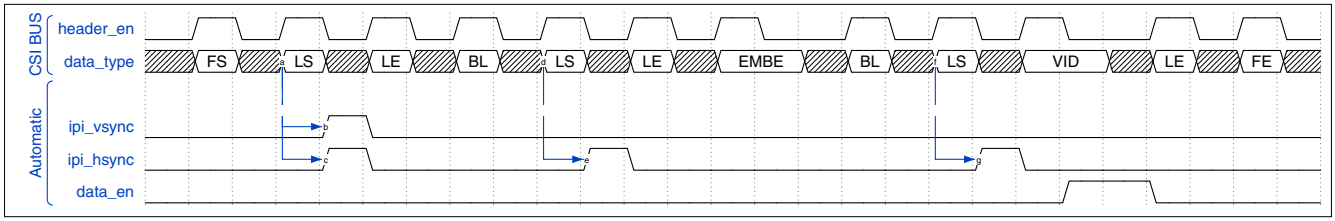


Figure 295. line_event_selection =0 or line_event_selection =1 and en_line_start =1

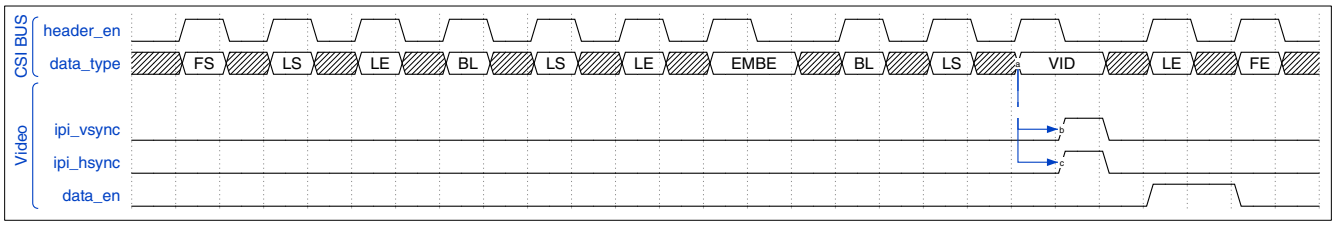


Figure 296. line_event_selection =1, en_video=1

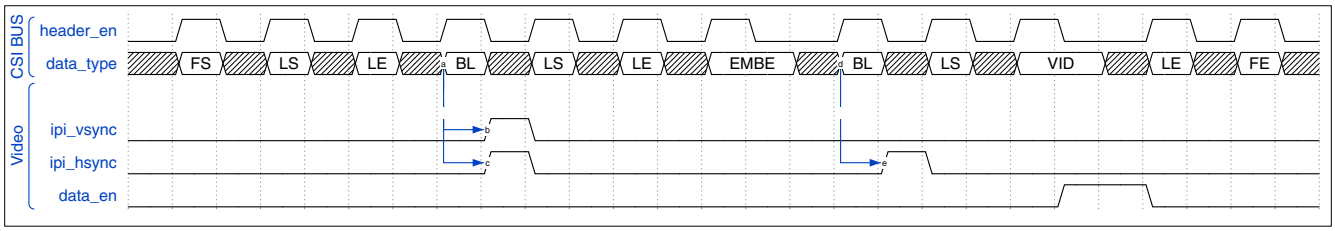


Figure 297. line_event_selection =1, en_blanking =1

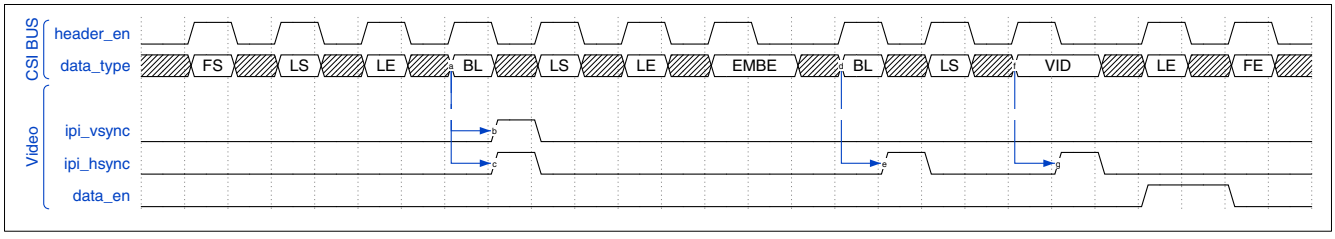


Figure 298. line_event_selection =1, en_blanking =1, en_video=1

54.2.6.1.1.1.2 Horizontal Timing Information

According to the *MIPI CSI-2 specification*, the presence of horizontal timing information in a CSI-2 link is optional. When horizontal timing information is not received, IPI reconstructs the horizontal timing signals through the blanking or/and data packets received.

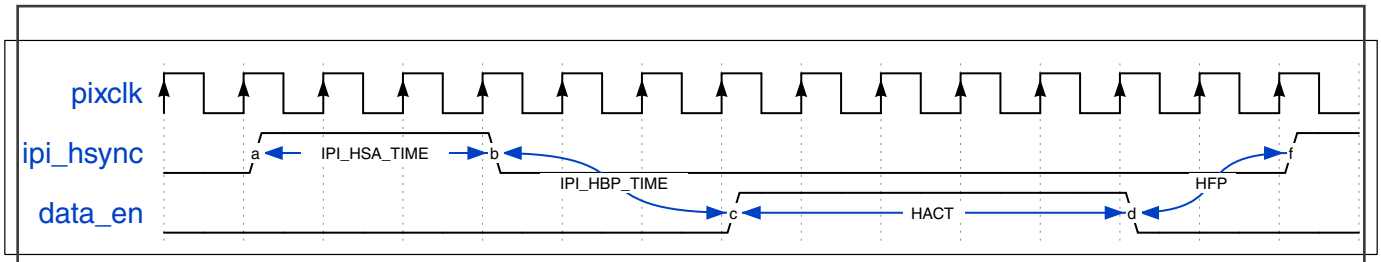


Figure 299. Horizontal Timing Information

In the Camera Timing mode, configure the following registers to build the horizontal frame:

- `IPI_HSA_TIME`: Configures the HSYNC duration.
- `IPI_HBP_TIME`: Configures the minimum HBP duration. Even if the pixel or embedded data is already available, the transmission is stalled to ensure a minimum distance between HSYNC and HACT. The HBP period is automatically extended if the data is not available at the time of the video.

HACT is the time spent in transmitting the pixel data. HFP is the distance between the end of the video and the next packet. HFP does not need to be configured. In the Camera Timing mode, the HFP period is defined by the sensor.

54.2.6.1.1.1.3 IPI Cut-Through

The IPI Cut-Through mode allows the pixel stream to be non-continuous. If the pixel interface FIFO becomes empty during pixel transmission, the operation is suspended. As shown below, transmission is resumed as soon as the memory has valid pixels.

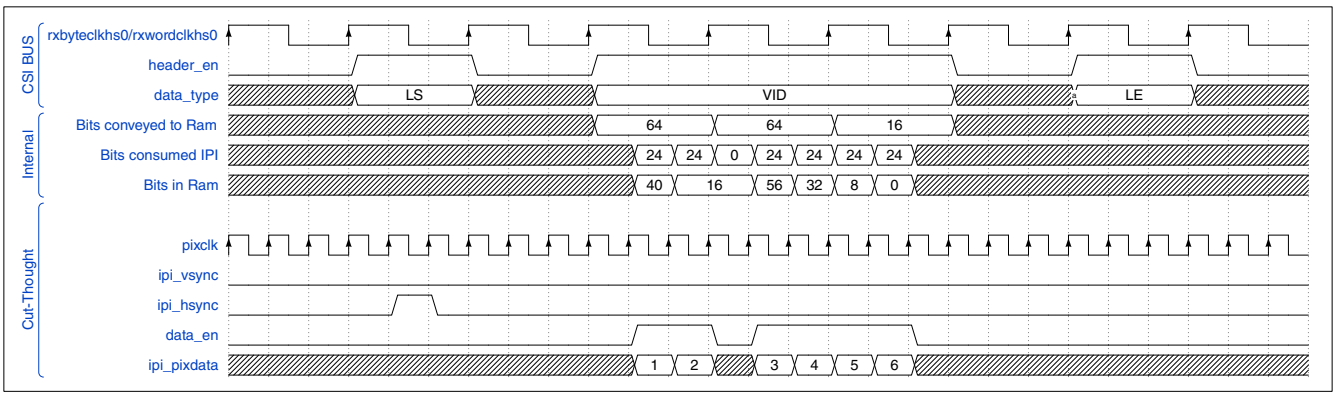


Figure 300. IPI Cut-through

Configuring the IPI_MODE[16]- ipi_cut_through, activates the IPI Cut-Through feature in any IPI interface. In camera mode, IPI Cut-Through feature is useful only when IPI throughput is higher than the PPI throughput.

54.2.6.1.1.1.4 Halt Functionality

The `ipi_halt` pin enforces the back pressure mechanism. The MIPI_CSI2 controller stops transmitting (sync events and payload) in the IPI interface on asserting the `ipi_halt` pin. The data transmission resumes after the de-assertion of the `ipi_halt` pin.

If a new line starts and the data from the previous line is not transmitted, the MIPI_CSI2 controller delays the sync events and payload. It asserts an error (`pixel_if_hline_err`) indicating that the frame generated by the IPI interface is changed. Upon this, the IPI transmits the programmed frame.

The `ipi_halt` pin can be asserted at any period of the frame. The following examples demonstrate the behavior of the IPI interface in different scenarios.

ipi_halt Asserts During ipi_pixen

The ipi_halt signal does not affect IPI_HSA_TIME, IPI_HBP_TIME, and IPI_HLINE_TIME. However, the HFP duration is smaller than expected.

The IPI still transmits valid ipi_pixdata information during HALT_PIXENZ pixclk cycles.

When ipi_halt de-asserts, the IPI needs HALTZ_PIXEN pixclk cycles to resume the transmission.

Figure 301 illustrates this process.

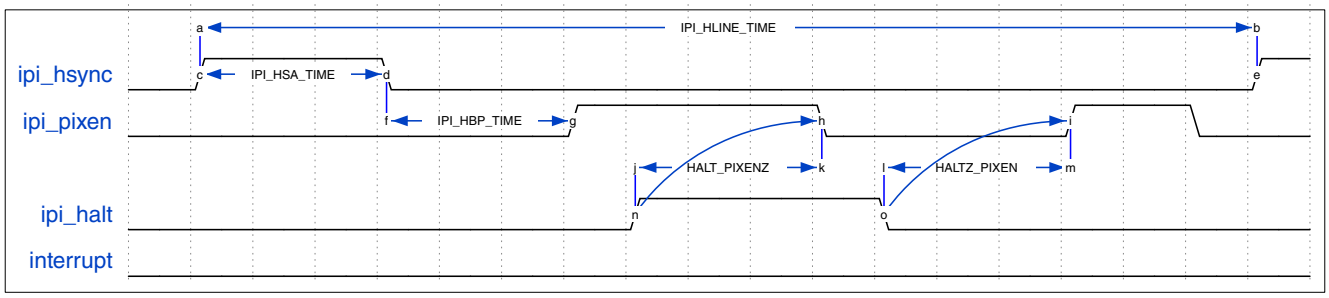


Figure 301. ipi_halt During ipi_pixen Assertion

ipi_halt Asserts During HSYNC and delay ipi_pixen

When ipi_halt asserts during ipi_hsync signal, no update is required on this signal. In Figure 13, ipi_halt remains high when the ipi_pixen is supposed to be asserted. In this situation, the IPI waits for ipi_halt de-assertion + HALTZ_PIXEN_S pixclk cycles before asserting the ipi_pixen signal.

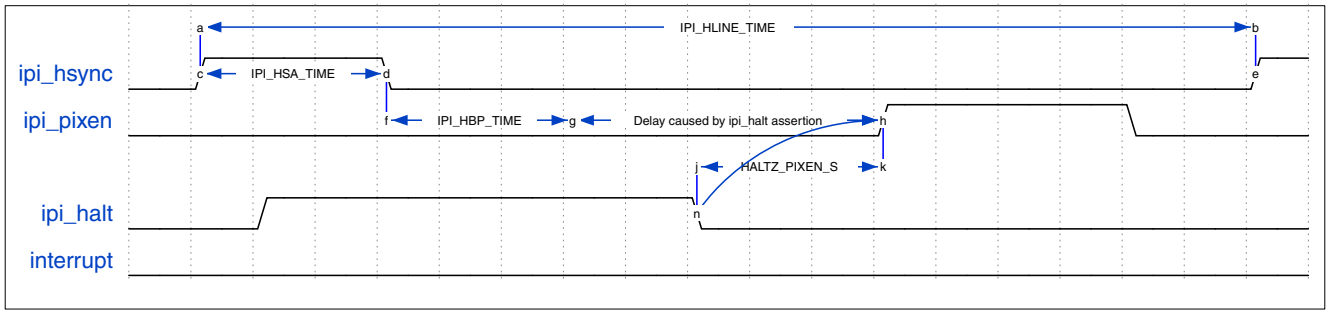


Figure 302. ipi_halt Signal Asserts During all IPI_HSA_TIME and IPI_HBP_TIME

ipi_halt Asserts During Horizontal Front Porch and delay ipi_pixen HSYNC

The assertion of ipi_halt during the Horizontal Front Porch delays the next ipi_hsync signal assertion. If in the Controller Timing mode, it triggers the pixel_if_hline_err interrupt that indicates the horizontal line time error.

Upon de-assertion of ipi_halt, the IPI resumes the frame transmission after HALTZ_HSYNC pixclk cycles.

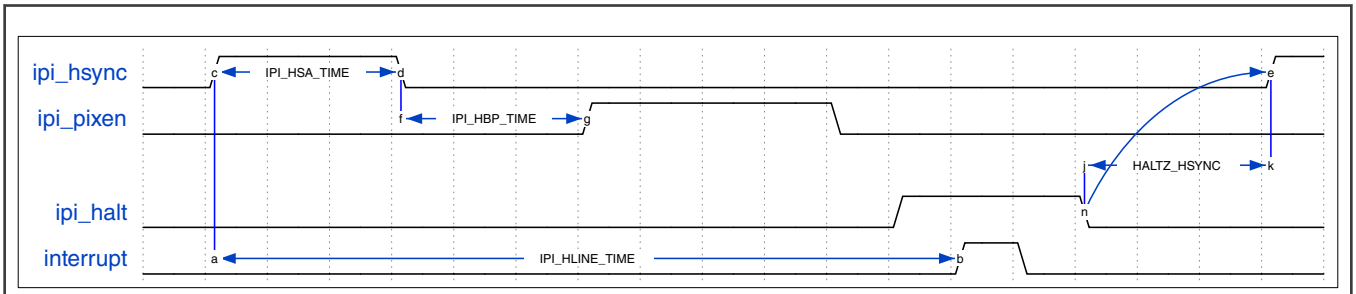


Figure 303. IPI_HLINE_TIME Affected by ipi_halt Assertion

54.2.6.1.1.1.5 Frame Requirements

To ensure the correct generation of events from the received packets, the following criteria has to be met:

- Minimum time between packets
- Minimum back porch time
- Avoid FIFO underflow
- Avoid FIFO overflow

Minimum Time Between Packets

The `Pkt2Pkt_Time` represents the minimum time that occurs between two consecutive received packets.

$$pkt2pkt_time > (4 + 2*CSI2_HOST_DFLT_F_SYNC_TYPE)/Wlength.T_{PPI-HS} + (4 + CSI2_HOST_DFLT_F_SYNC_TYPE).T_{IPI}$$

Table 519 describes the variables in the above formula.

Table 519. Variable Description for Determining the Minimum Time Between Packets

Variable	Description
pkt2pkt_time	Time between Packets
Wlength	Bytes Received in each T_{PPI-HS} • In D-PHY PPI-8, it is 1
T_{PPI-HS}	PPI High-Speed Clock Period
T_{IPI}	IPI Clock Period

All the remaining variables represent the parameters with the same name.

Minimum Back Porch Time

To ensure that the pixel data is processed correctly, the back porch time must not be less than the time between the packets plus some margin.

$$(IPI_HSA_TIME + IPI_HBP_TIME).T_{IPI} > pkt2pkt_time + (2*short_pkt_ovh/N_LANES + 2)/Wlength.T_{PPI-HS} + (pkt2pkt_time + pixel_bytes/(N_LANES.Wlength).T_{PPI-HS}).(n_ipi_active - 1)$$

Table 520 describes the variables in the above formula.

Table 520. Variable Description for Determining the Minimum Back Porch Time

Variable	Description
pkt2pkt_time	Time between Packets
short_pkt_ovh	Short Packet Overhead • In D-PHY mode it is 4
Wlength	Bytes Received in each T_{PPI-HS} • In D-PHY PPI-8, it is 1
pixel_bytes	Pixel Bytes to Transmit (corresponding to HACT region)
n_ipi_active	Number of active IPIs (IPIx_MODE[24]). Only one IPI interface can be used.

Table continues on the next page...

Table 520. Variable Description for Determining the Minimum Back Porch Time (continued)

Variable	Description
T_{PPI-HS}	PPI High-Speed Clock Period
T_{IPI}	IPI Clock Period

All the remaining variables represent the registers or parameters with the same name.

Avoid FIFO Underflow

FIFO underflow occurs when the IPI has a higher throughput than the PHY, that is, data is consumed at a rate faster than that which the PHY can provide.

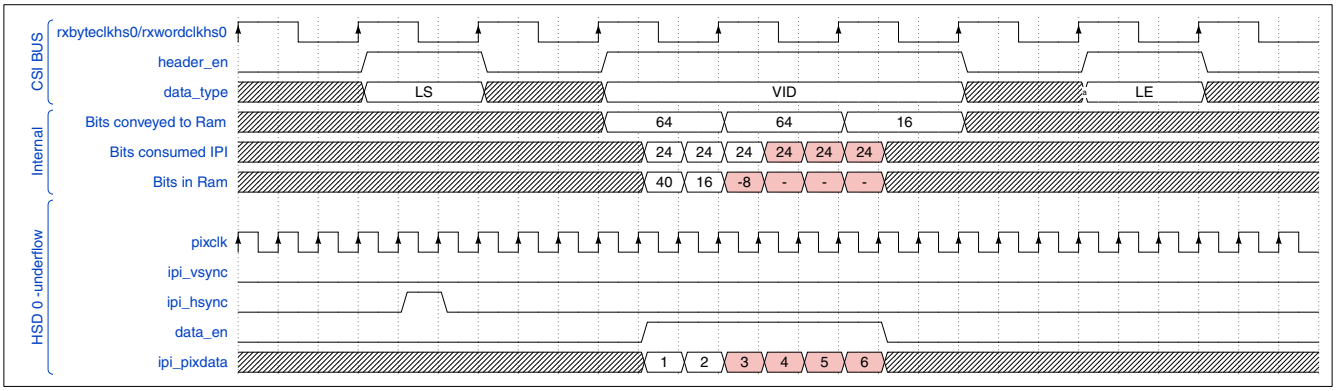


Figure 304. Underflow Scenario

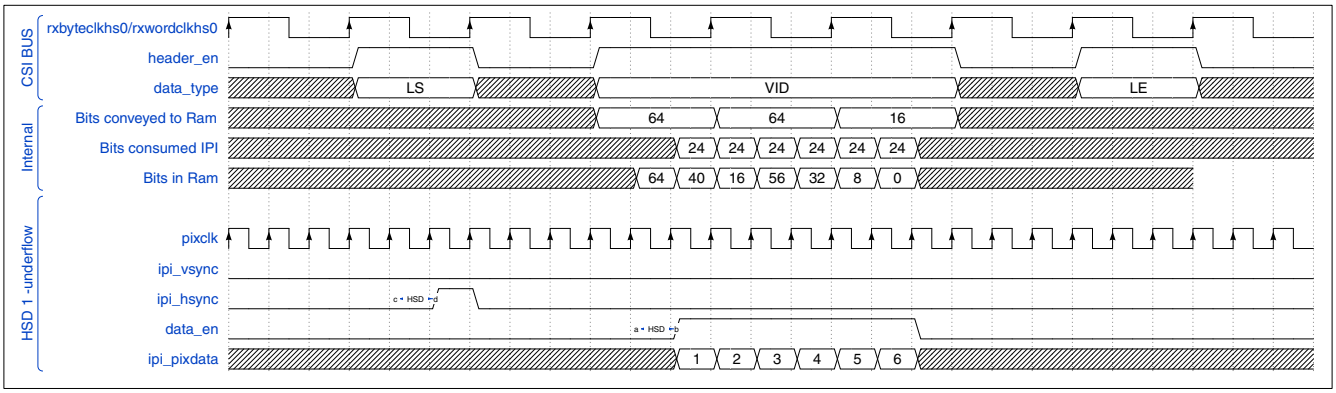


Figure 305. Underflow Scenario Avoided Using HSD

To avoid FIFO underflow, delay the transmission of the full frame to ensure that, at the transmission of the last pixel, it is already available in memory. Configure the Horizontal Sync Delay (HSD) to achieve this. The `IPI_HSD_TIME` register controls this delay.

In the Controller Timing mode, it is possible to add extra blanking lines which renders the use of HSD pointless. But using this technique increases the memory requirements.

Manipulating HSA and HBP such that their sum forces the video delay also avoids FIFO underflow. But HSD is the safer approach because the events from the sensor are not accurate.

To avoid the FIFO underflow, the IPI timings must be according to the following formula:

$IPI_HSD_TIME > (pkt2pkt_time + (long_pkt_ovh + pixel_bytes)/(N_LANES.Wlength).T_{PPI-HS}) - (IPI_HSA_TIME + IPI_HBP_TIME + IPI_HACT_TIME).T_{IPI}$

Table 521 describes the variables in the above formula.

Table 521. Variable Description for Determining how to Avoid FIFO Underflow

Variable	Description
pkt2pkt_time	Time between Packets
long_pkt_ovh	Long Packet Overhead • In D-PHY mode, it is 6.
Wlength	Bytes Received in each T_{PPI-HS} • In D-PHY PPI-8, it is 1
pixel_bytes	Pixel Bytes to Transmit (corresponding to HACT region)
T_{PPI-HS}	PPI High-Speed Clock Period
T_{IPI}	IPI Clock Period

All the remaining variables represent the registers or parameters with the same name.

Avoid FIFO Overflow

FIFO overflow occurs when the IPI has lower throughput than the PHY. To avoid FIFO overflow anticipate the transmission of the full frame to ensure that, at the transmission of the first pixel, the memory is not full.

To avoid FIFO overflow, the IPI timings must meet the following expression:

$(IPI_HSD_TIME + IPI_HSA_TIME + IPI_HBP_TIME + IPI_HACT_TIME).T_{IPI} \leq pkt2pkt_time + (long_pkt_ovh + pixel_bytes)/(N_LANES.Wlength).T_{PPI-HS} + (8 \times CSI2_HOST_IPI_FIFO_DEPTH)/(N_LANES.Wlength).T_{PPI-HS}$

Table 522 describes the variables in the above formula.

Table 522. Variable Description for Determining how to Avoid FIFO Overflow

Variable	Description
pkt2pkt_time	Time between Packets
long_pkt_ovh	Long Packet Overhead • In D-PHY Mode, it is 6
pixel_bytes	Pixel Bytes to Transmit (corresponding to HACT region)
Wlength	Bytes Received in each T_{PPI-HS} • In D-PHY PPI-8, it is 1
T_{PPI-HS}	PPI High-Speed Clock Period
T_{IPI}	IPI Clock Period

All the remaining variables represent the registers or parameters with the same name.

54.2.6.1.1.1.6 Memory Requirements

Memory is required for domain Crossing between CSI-2 link clock and pixel clock. The RAM size must accommodate the difference between the two clock domains and frame buffering requirements.

In the Controller Timing mode, minimum memory size is defined as the maximum HSD that the controller needs to avoid underflow scenario and the number of lines that are required to buffer. Adding artificial blanking lines in VSA or VFP to the generated frame, buffers the corresponding video lines.

When IPI Cut-Through is active, it reduces the MIPI_CSI2 controller memory requirements, down to the minimum FIFO depth of 32. This implies that the generated frame matches the one sent by the camera in line length and number of lines, and IPI have higher bandwidth.

Requirement 1:

$$MEM_req_1 = ((IPI_HSD_TIME + IPI_HSA_TIME + IPI_HBP_TIME) * T_{IPI} - pkt2pkt_time) / T_{PPI-HS}) * N_LANES * BytesPerHsClk$$

Requirement 2:

$$MEM_req_2 = ((IPI_HSD_TIME + IPI_HSA_TIME + IPI_HBP_TIME + nr_of_pixels_per_line) * T_{IPI} - pkt2pkt_time - HACT_PPI_period) / T_{PPI-HS}) * N_LANES * BytesPerHsClk$$

Table 523. Variable Description for Determining MIPI_CSI2 Memory Requirements

Variable	Description
pkt2pkt_time	Time between Packets
BytesPerHsClk	Bytes Transmitted per T_{PPI-HS}
nr_of_pixels_per_line	Number of Pixels to transmit per Line
HACT_PPI_period	Time it takes to transmit Pixel Data
T_{PPI-HS}	PPI High-Speed Clock Period
T_{IPI}	IPI Clock Period

Memory Requirement is as follows:

$$Min_Memory_depth = MAX\ (MEM_req_1, MEM_req_2) * 8 / Memory_Width$$

54.2.6.1.1.1.7 Example

The sensor sends RGB888 format: 3840 pixels * 2160 lines @60FPS.

The sensor sends 10 blanking or embedded data, that is, five before video and five after video.

Step 1: Required PHY Bandwidth (Gbps)

Consider an example of D-PHY 4 Lanes below.

NOTE

This chip supports up to 2 data lanes.

The required bandwidth can be calculated as:

$$Required\ BW\ (Gbps) = (8 * hact_byte_to_send) / (expected_PPI_hline_time - LS_to_video - video_to_LE)$$

Table 524 describes the variables in the above formula.

Table 524. Variable Description for Determining Required PHY Bandwidth

Variable	Description
hact_byte_to_send	Number of bytes to transmit pixel data
expected_PPI_hline_time	Total Line time (ns)
LS_to_video video_to_LE	Time to Line Start to Video (0 if LS/LE is disable) Time of Video to Line End (0 if LS/LE is disable)

LS_to_video and video_to_LE are defined as Packet to Packet time (ns). This time is defined by PHY parameters - HS2LP time and LP2HS time and is considered as 1000 ns for this example.

```
hact_byte_to_send = (video_pixels * bytes_per_pixel + long_packet_overhead) = 3840 * 3 + 6 = 11526 bytes
expected_PPI_hline_time = (10*10^8) /fps /total_lines = (10*10^8) /60/(5+2160+5) = 7680 ns
required_BW = (8* 11526)/(7680 - 1000 - 1000) = 16.23 Gbps
```

Step 2: Available PHY Bandwidth (Gbps)

```
Available_BW(Gbps)=(lanes*8)/PPI_HS_clk_period (ns)
```

Based on the above equation, the minimum PPI HS Clock period can be defined using the Required BW.

`PPI_HS_clk_period(ns)= (lanes*8)/required_BW(Gbps)` = (4*8) / 16.23 = 1.97 ns

To add a safety margin, the BW available must be above the minimum. For this example, PPI HS Clock is considered as 1.85 ns period.

```
Available_BW (Gbps) = 4*8/1.85 = 17.3 Gbps
```

This translates into a Line blanking period between the video and the next packet.

```
line_blanking = expected_PPI_hline_time - LS_to_video - video_to_LE - HACT_PPI_period
HACT_PPI_period = (hact_byte_to_send/wordlength/lanes) * PPI_HS_clk_period (ns)
HACT_PPI_period = (11526/1/4) * 1.85 = 5331 ns
line_blanking = 7680 - 1000 - 1000 - 5331 = 350 ns
```

This time is the time margin for any unaccounted event.

Step 3: IPI Configuration

The minimum pixel clock frequency can be calculated as:

IPI Minimum Frequency (MHz) `= (IPI_HACT_CYCLES+HSA+HBP+HFP)*1000/(HLINE TIME)`

Table 525 describes the variables in the above formula.

Table 525. Variable Description for Determining IPI Minimum Frequency

Variable	Description
IPI_HACT_CYCLES	Number of Pixels to transmit pixel data. If IPI 16 interface is used, multiply by the number of components for the respective data type.
HSA	Horizontal Sync Active cycles.

Table continues on the next page...

Table 525. Variable Description for Determining IPI Minimum Frequency (continued)

Variable	Description
HBP	Horizontal Back Porch cycles.
HFP	Horizontal Front Porch cycles. This is defined as the minimum to ensure margin for CDC or lack of clock precision.
HLINE TIME	Horizontal line time. To avoid complicated scenarios and unnecessary buffering, it is recommended to use the same value as in PHY calculations.

For this example, assume the following values:

- HSA – 100
- HBP – 200
- HFP – 800

Then, IPI Minimum frequency (MHz) = (3840 +100+200+800)*1000/7680 = 643 MHz

Step 4: Minimum Back Porch Time

Minimum Back Porch Time must be at least the equivalent as in PPI.

$(IPI_HSA_TIME + IPI_HBP_TIME).T_{IPI} > Pkt2PktTime + (2*ShortPktBytes/N_LANES + 2).T_{PPI-HS}/BytesPerHsClk$

$(100+100)*1.65 > 1000 +4*1.85 <=> 556.5 > 1007.4$

HBP need to be readjusted.

$HBP = 1007.4/T_{IPI} - HSA = 510.54 <=> HBP > 511$

Consider the following values for updating the frame:

- HSA – 100
- HBP – 544
- HFP – 456

If this update is not possible, decrease the IPI frequency until there is enough blanking time. Increasing the blanking time in step 3 accomplishes this.

Step 5: Avoid FIFO Underflow

FIFO underflow does not occur when Cut-Through mode is active. For other modes, configure HSD to avoid FIFO underflow.

$IPI_HSD_TIME > (LongPktBytes2Transmit/N_LANES.T_{PPI-HS}/BytesPerHsClk)/T_{IPI} - (IPI_HSA_TIME + IPI_HBP_TIME + IPI_HACT_TIME)$

$IPI_HSD_TIME > (11526/4*1.85)-1.555-(100+544+3840)$

$IPI_HSD_TIME > -1055.85$

This means that HSD compensation is not required, therefore, select the value as 1.

Step 6: Memory Calculation

Memory calculation requirements are defined in the following equations.

$MEM_req_1 = (((IPI_HSD_TIME + IPI_HSA_TIME + IPI_HBP_TIME)* T_{IPI} - Pkt2PktTime)/ T_{PPI-HS})* bytes_per_ppiclk$

$MEM_req_1 = (((1+ 100 + 544) * 1.555 - 1000)/ 1.85)* 4 =-0.5$

There is no requirement for the HSD impact.

MEM_req_2 = ((IPI_HSD_TIME + IPI_HSA_TIME + IPI_HBP_TIME + number_of_pixels_per_line)* T_{IPI} - Pkt2PktTime - HACT_PPI_period)/T_{PPI-HS})* N_LANES*BytesPerHsClk

MEM_req_2 = (((1+ 100 + 544 + 3840) * 1.555 – 1000 - 5331)/ 1.85)* 4 = 1343

Min_Memory_Depth = MAX (MEM_req_1, MEM_req_2)*8/ Memory_Width

Min_Memory_Depth = MAX (0,1343)*8/ 64= 172

54.2.6.1.1.1.8 Summary

To operate in the Camera Timing mode, configure the following registers:

- IPI_ADV_FEATURES
- IPI_HSA_TIME
- IPI_HBP_TIME
- IPI_HSD_TIME

54.2.6.1.1.2 Controller Timing

In this mode, the IPI generates frame timing signals based on the programed configuration.

This mode transforms the CSI stream in a video with the timing information.

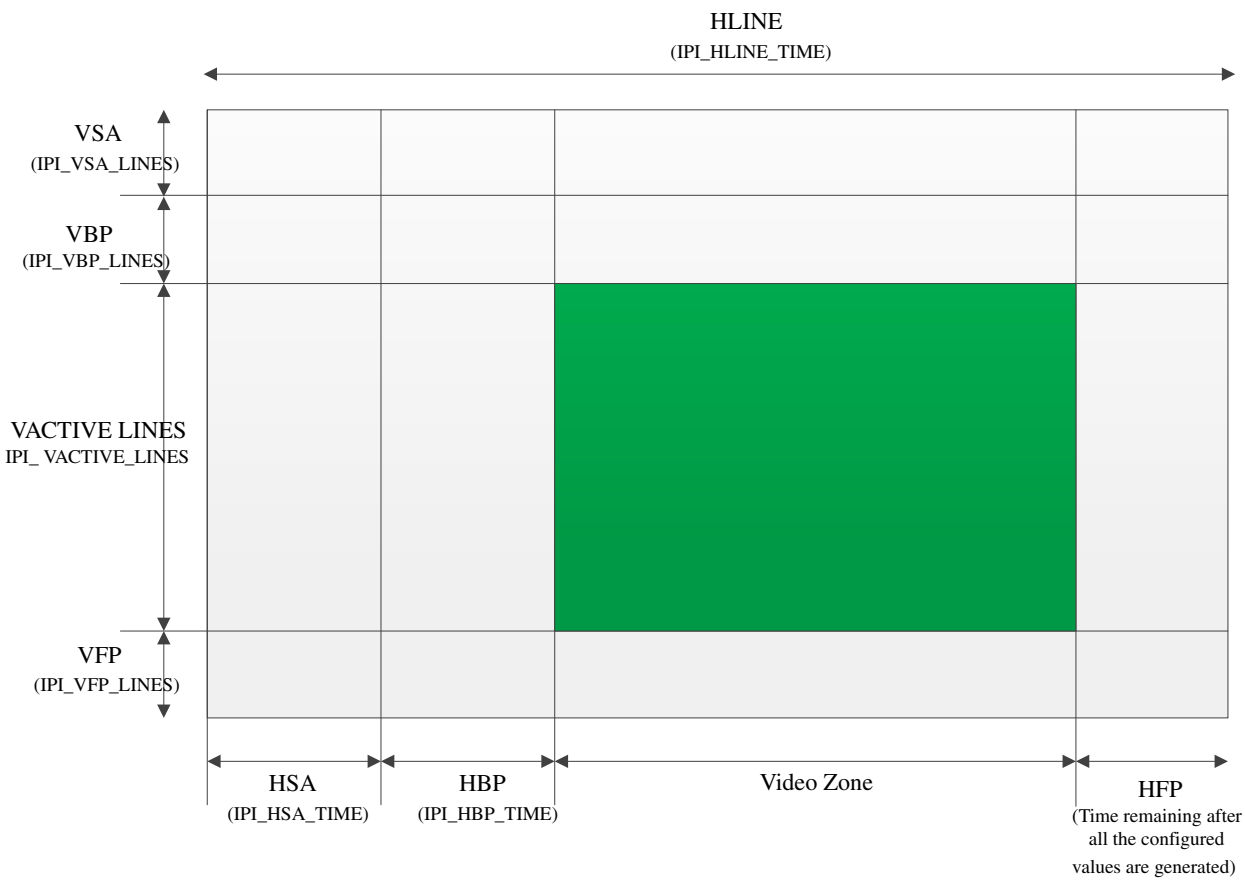


Figure 306. Frame Generated in Controller Timing

54.2.6.1.1.2.1 Vertical Timing

The IPI generates the frame and ignores all the other synchronism packets after a Frame Start is received.

The following registers define the number of lines in each zone:

- IPI_VSA_LINES
- IPI_VBP_LINES
- IPI_VACTIVE_LINES
- IPI_VFP_LINES

The duration of each line is defined by the horizontal timing.

54.2.6.1.1.2.2 Horizontal Timing

The horizontal timing is defined by the following parameters:

- IPI_HLINE_TIME: defines the duration of the line
- IPI_HSA_TIME: defines the time for which ipi_hsync is asserted
- IPI_HBP_TIME: defines the time for which the interface waits after the IPI_HSA_TIME until the video is transmitted

The remaining time of the line, excluding the video zone, corresponds to the HFP time.

All events of the line must fit the line. If any of the events does not fit the line, it affects the synchronization event of the subsequent lines.

54.2.6.1.1.2.3 IPI Cut-Through

In the Controller Timing mode, IPI Cut-Through feature is used in the same way as in the Camera Timing mode.

If a new line starts and the data from the previous line is not transmitted, The MIPI_CSI2 controller delays the sync events and payload. It asserts an error (pixel_if_hline_err) indicating that the frame generated by the IPI interface is changed. After this, the IPI transmits the programmed frame.

In the Controller Timing mode, IPI Cut-Through has an effect only when the IPI throughput is higher than the PPI throughput and the controller is not prebuffering a line. This feature has no effect in all other scenarios.

54.2.6.1.1.2.4 Halt Functionality

The ipi_halt pin enforces the back pressure mechanism. The MIPI_CSI2 controller stops transmitting (sync events and payload) in the IPI interface on asserting the ipi_halt pin. The data transmission resumes after the de-assertion of the ipi_halt pin.

If a new line starts and the data from the previous line is not transmitted, the MIPI_CSI2 controller delays the sync events and payload. It asserts an error (pixel_if_hline_err) indicating that the frame generated by the IPI interface is changed. Upon this, the IPI transmits the programmed frame.

The ipi_halt pin can be asserted at any period of the frame. The following examples demonstrate the behavior of the IPI interface in different scenarios.

ipi_halt Asserts During ipi_pixen

The ipi_halt signal does not affect IPI_HSA_TIME, IPI_HBP_TIME, and IPI_HLINE_TIME. However, the HFP duration is smaller than expected.

The IPI still transmits valid ipi_pixdata information during HALT_PIXENZ pixclk cycles.

When ipi_halt de-asserts, the IPI needs HALTZ_PIXEN pixclk cycles to resume the transmission.

Figure 307 illustrates this process.

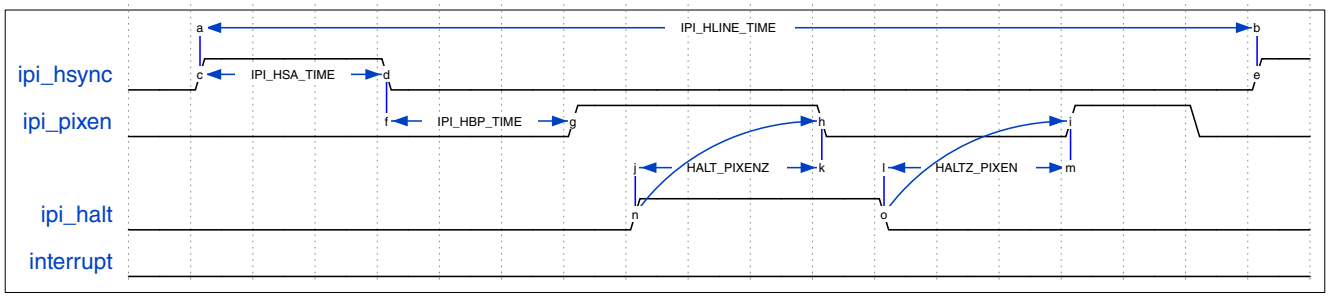


Figure 307. ipi_halt During ipi_pixen Assertion

ipi_halt Asserts During HSYNC and delay ipi_pixen

When `ipi_halt` asserts during `ipi_hsync` signal, no update is required on this signal. In Figure 13, `ipi_halt` remains high when the `ipi_pixen` is supposed to be asserted. In this situation, the IPI waits for ipi_halt de-assertion + HALTZ_PIXEN_S pixclk cycles before asserting the `ipi_pixen` signal.

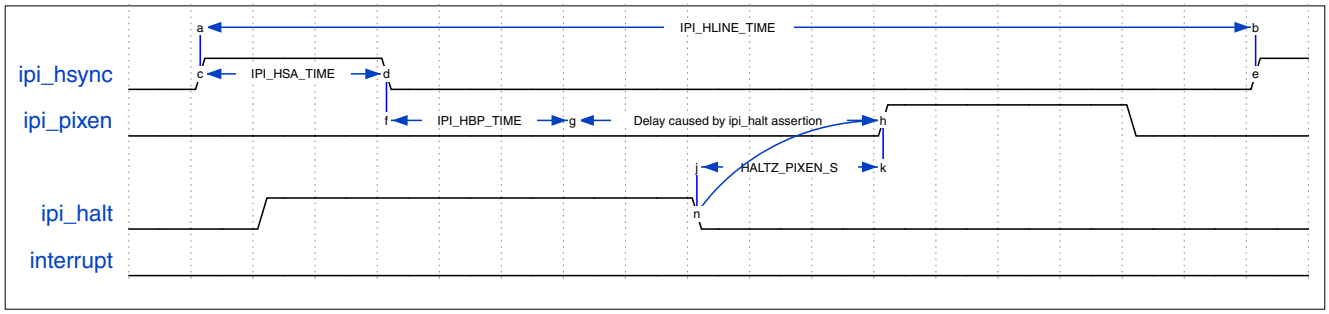


Figure 308. ipi_halt Signal Asserts During all IPI_HSA_TIME and IPI_HBP_TIME

ipi_halt Asserts During Horizontal Front Porch and delay ipi_pixen HSYNC

The assertion of `ipi_halt` during the Horizontal Front Porch delays the next `ipi_hsync` signal assertion. If in the Controller Timing mode, it triggers the `pixel_if_hline_err` interrupt that indicates the horizontal line time error.

Upon de-assertion of `ipi_halt`, the IPI resumes the frame transmission after `HALTZ_HSYNC` pixclk cycles.

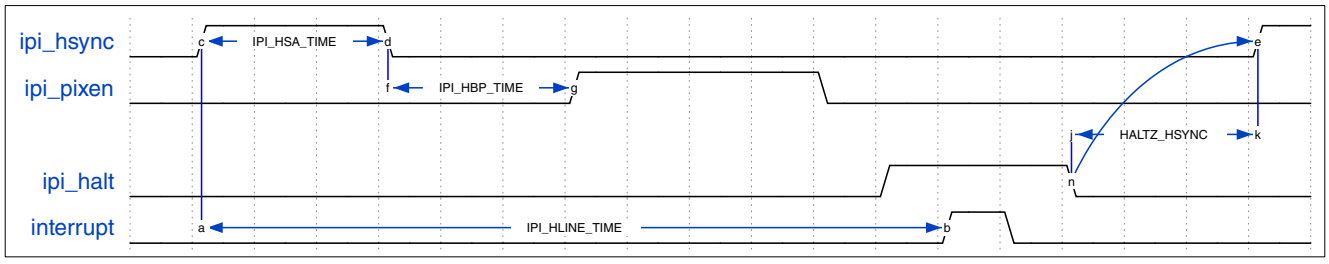


Figure 309. IPI_HLINE_TIME Affected by ipi_halt Assertion

54.2.6.1.1.2.5 Latency

Table 9 shows the latencies for IPI pixel stream modes in pixclk cycles.

Table 526. Latencies for IPI Pixel Stream Modes in pixclk Cycles

IPI Pixel Stream Mode	HALT_PIXENZ	HALTZ_PIXEN	HALTZ_PIXEN_S	HALTZ_HSYNC
16-bit	2	2	6	5
48-bit	3	3		

54.2.6.1.1.2.6 Frame Requirements

To ensure the correct generation of events from the received packets, the following criteria has to be met:

- Minimum time between packets
- Minimum back porch time
- Avoid FIFO underflow
- Avoid FIFO overflow

Minimum Time Between Packets

The `Pkt2Pkt_Time` represents the minimum time that occurs between two consecutive received packets.

$$pkt2pkt_time > (4 + 2*CSI2_HOST_DFLT_F_SYNC_TYPE)/Wlength.T_{PPI-HS} + (4 + CSI2_HOST_DFLT_F_SYNC_TYPE).T_{IPI}$$

Table 527 describes the variables in the above formula.

Table 527. Variable Description for Determining the Minimum Time Between Packets

Variable	Description
pkt2pkt_time	Time between Packets
Wlength	Bytes Received in each T_{PPI-HS} • In D-PHY PPI-8, it is 1
T_{PPI-HS}	PPI High-Speed Clock Period
T_{IPI}	IPI Clock Period

All the remaining variables represent the parameters with the same name.

Minimum Back Porch Time

To ensure that the pixel data is processed correctly, the back porch time must not be less than the time between the packets plus some margin.

$$(IPI_HSA_TIME + IPI_HBP_TIME).T_{IPI} > pkt2pkt_time + (2*short_pkt_ovh/N_LANES + 2)/Wlength.T_{PPI-HS} + (pkt2pkt_time + pixel_bytes/(N_LANES.Wlength).T_{PPI-HS}).(n_ipi_active - 1)$$

Table 528 describes the variables in the above formula.

Table 528. Variable Description for Determining the Minimum Back Porch Time

Variable	Description
pkt2pkt_time	Time between Packets
short_pkt_ovh	Short Packet Overhead

Table continues on the next page...

Table 528. Variable Description for Determining the Minimum Back Porch Time (continued)

Variable	Description
	• In D-PHY mode it is 4
Wlength	Bytes Received in each T_{PPI-HS} • In D-PHY PPI-8, it is 1
pixel_bytes	Pixel Bytes to Transmit (corresponding to HACT region)
n_ipi_active	Number of active IPIs (IPIx_MODE[24]). Only one IPI interface can be used.
T_{PPI-HS}	PPI High-Speed Clock Period
T_{IPI}	IPI Clock Period

All the remaining variables represent the registers or parameters with the same name.

Avoid FIFO Underflow

FIFO underflow occurs when the IPI has a higher throughput than the PHY, that is, data is consumed at a rate faster than that which the PHY can provide.

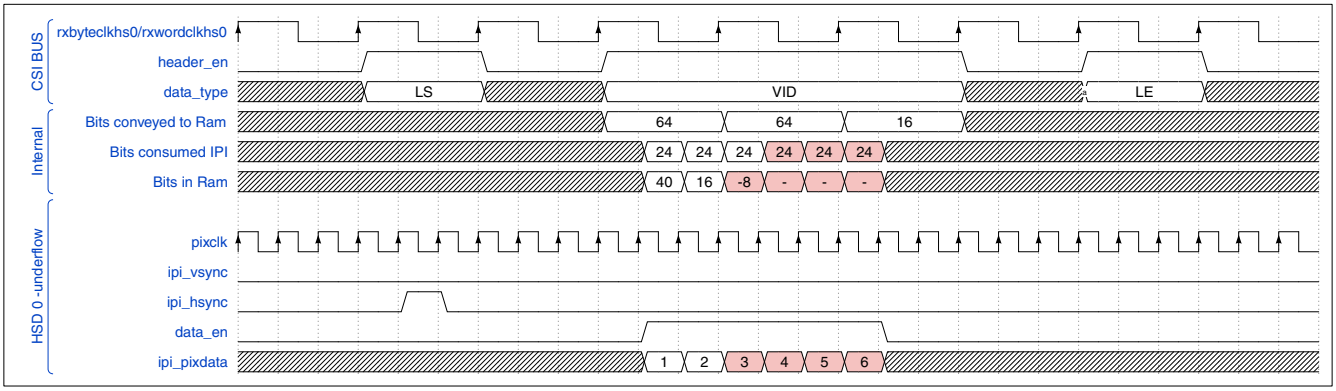


Figure 310. Underflow Scenario

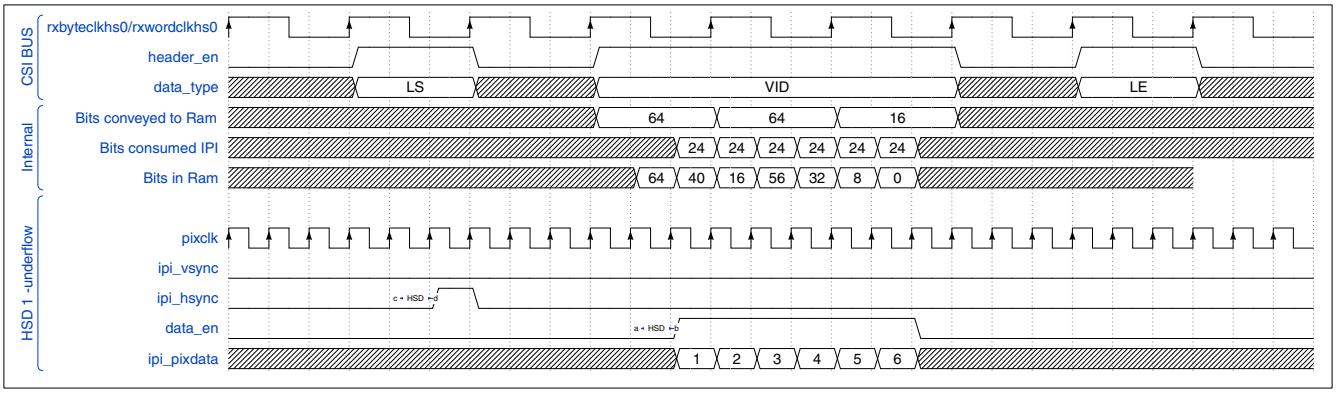


Figure 311. Underflow Scenario Avoided Using HSD

To avoid FIFO underflow, delay the transmission of the full frame to ensure that, at the transmission of the last pixel, it is already available in memory. Configure the Horizontal Sync Delay (HSD) to achieve this. The `IPI_HSD_TIME` register controls this delay.

In the Controller Timing mode, it is possible to add extra blanking lines which renders the use of HSD pointless. But using this technique increases the memory requirements.

Manipulating HSA and HBP such that their sum forces the video delay also avoids FIFO underflow. But HSD is the safer approach because the events from the sensor are not accurate.

To avoid the FIFO underflow, the IPI timings must be according to the following formula:

$$IPI_HSD_TIME > (pkt2pkt_time + (long_pkt_ovh + pixel_bytes)/(N_LANES.Wlength).T_{PPI\text{-}HS}) - (IPI_HSA_TIME + IPI_HBP_TIME + IPI_HACT_TIME).T_{IPI}$$

Table 529 describes the variables in the above formula.

Table 529. Variable Description for Determining how to Avoid FIFO Underflow

Variable	Description
pkt2pkt_time	Time between Packets
long_pkt_ovh	Long Packet Overhead • In D-PHY mode, it is 6.
Wlength	Bytes Received in each $T_{PPI\text{-}HS}$ • In D-PHY PPI-8, it is 1
pixel_bytes	Pixel Bytes to Transmit (corresponding to HACT region)
$T_{PPI\text{-}HS}$ T_{IPI}	PPI High-Speed Clock Period IPI Clock Period

All the remaining variables represent the registers or parameters with the same name.

Avoid FIFO Overflow

FIFO overflow occurs when the IPI has lower throughput than the PHY. To avoid FIFO overflow anticipate the transmission of the full frame to ensure that, at the transmission of the first pixel, the memory is not full.

To avoid FIFO overflow, the IPI timings must meet the following expression:

$$(IPI_HSD_TIME + IPI_HSA_TIME + IPI_HBP_TIME + IPI_HACT_TIME).T_{IPI} \leq pkt2pkt_time + (long_pkt_ovh + pixel_bytes)/(N_LANES.Wlength).T_{PPI\text{-}HS} + (8 \times CSI2_HOST_IPI_FIFO_DEPTH)/(N_LANES.Wlength).T_{PPI\text{-}HS}$$

Table 530 describes the variables in the above formula.

Table 530. Variable Description for Determining how to Avoid FIFO Overflow

Variable	Description
pkt2pkt_time	Time between Packets
long_pkt_ovh	Long Packet Overhead • In D-PHY Mode, it is 6
pixel_bytes	Pixel Bytes to Transmit (corresponding to HACT region)

Table continues on the next page...

Table 530. Variable Description for Determining how to Avoid FIFO Overflow (continued)

Variable	Description
Wlength	Bytes Received in each T_{PPI-HS} • In D-PHY PPI-8, it is 1
T_{PPI-HS}	PPI High-Speed Clock Period
T_{IPI}	IPI Clock Period

All the remaining variables represent the registers or parameters with the same name.

54.2.6.1.1.2.7 Memory Requirements

Memory is required for domain Crossing between CSI-2 link clock and pixel clock. The RAM size must accommodate the difference between the two clock domains and frame buffering requirements.

In the Controller Timing mode, minimum memory size is defined as the maximum HSD that the controller needs to avoid underflow scenario and the number of lines that are required to buffer. Adding artificial blanking lines in VSA or VFP to the generated frame, buffers the corresponding video lines.

When IPI Cut-Through is active, it reduces the MIPI_CSI2 controller memory requirements, down to the minimum FIFO depth of 32. This implies that the generated frame matches the one sent by the camera in line length and number of lines, and IPI have higher bandwidth.

Requirement 1:

$$MEM_req_1 = ((IPI_HSD_TIME + IPI_HSA_TIME + IPI_HBP_TIME) * T_{IPI} - pkt2pkt_time) / T_{PPI-HS}) * N_LANES * BytesPerHsClk$$

Requirement 2:

$$MEM_req_2 = ((IPI_HSD_TIME + IPI_HSA_TIME + IPI_HBP_TIME + nr_of_pixels_per_line) * T_{IPI} - pkt2pkt_time - HACT_PPI_period) / T_{PPI-HS}) * N_LANES * BytesPerHsClk$$

Table 531. Variable Description for Determining MIPI_CSI2 Memory Requirements

Variable	Description
pkt2pkt_time	Time between Packets
BytesPerHsClk	Bytes Transmitted per T_{PPI-HS}
nr_of_pixels_per_line	Number of Pixels to transmit per Line
HACT_PPI_period	Time it takes to transmit Pixel Data
T_{PPI-HS}	PPI High-Speed Clock Period
T_{IPI}	IPI Clock Period

Memory Requirement is as follows:

$$Min_Memory_depth = MAX\ (MEM_req_1, MEM_req_2) * 8 / Memory_Width$$

54.2.6.1.1.2.8 Example

Consider a scenario in which the sensor sends RGB888 format: 3840 pixels * 2160 lines @60FPS.

The sensor sends 10 blanking or embedded data, that is, five before video and five after video. The example described in section Example is valid for this scenario.

Consider another example where the scenario is same as the above, except that the sensor does not send any blanking before the video, that is, zero before video and ten after video. In this case also, the frame is the same as in first scenario.

Step 1 - Required PHY Bandwidth (Gbps)

This is same as Step 1 in section Example.

Step 2 - Available PHY Bandwidth (Gbps)

This is same as Step 2 in section Example.

Step 3 - IPI Configuration

For IPI configuration, configure the following registers:

- IPI_VSA_LINES = 1
- IPI_VBP_LINES = 4
- IPI_VACTIVE_LINES = 2160
- IPI_VFP_LINES = 5
- IPI_HLINE_TIME = EXPECTED PPI HLINE TIME/ T_{IPI} = 7680 / 1.55 = 4955
- IPI_HSA_TIME
- IPI_HBP_TIME

Step 4 - Minimum Back Porch Time

Minimum Back Porches are not required because one line is already buffered.

Step 5 - Avoid FIFO Underflow

FIFO Underflow does not occur because one line is already buffered.

Step 6 - Memory Calculation

MEM_req_1 is not required because one line is already buffered.

MEM_req_2 = ((IPI_HSD_TIME + IPI_HSA_TIME + IPI_HBP_TIME + nr_of_pixels_per_line)* T_{IPI} - Pkt2PktTime - HACT_PPI_period)/ T_{PPI-HS}* N_LANES*BytesPerHsClk

Mem_req_2 =(((1+ 100 + 544 + 3840) * 1.555 – 1000 - 5331)/ 1.85) * 4 = 1343

Extra_mem = Bytes2Transmit * number_of_lines_bufered= (3840*3)*5=57600

Min_Memory_depth = (MEM_req_2 + Extra_mem)*8/ Memory_Width

Min_Memory_depth = (57600 + 1343)*8/64 = 7368

54.2.6.1.1.2.9 Summary

To operate in the Controller Timing mode, configure the following registers:

- IPI_VSA_LINES
- IPI_VBP_LINES
- IPI_VACTIVE_LINES
- IPI_VFP_LINES
- IPI_HLINE_TIME
- IPI_HSA_TIME
- IPI_HBP_TIME

54.2.6.1.2 Interface Type

The IPI can generate pixel stream in two different modes: 48-bit or 16-bit interface. To select the mode, you can configure the `IPI_MODE[8]` register field.

54.2.6.1.2.1 IPI 48-Bit interface

Figure 312 illustrates the pixel color coding at the 48-bit interface. The different types of data output are as follows:

- User defined: A test mode to deliver any payload in 32-bit words.
- RAW data types and YUV 8-bit legacy: The interface transfers two or more pixels in parallel and reduces the number of clock cycles required to send the same image data. The output YUV data types are as follows:
  - For YUV422 data types, there is a distinction between even and odd pixels, because only Y component is received for odd pixels.
  - For RAW and YUV420 8-bit legacy, two pixels are present for each clock cycle and there is a distinction between odd and even lines.
  - For all other YUV 420 data types, there is a distinction between odd and even pixels and also between odd and even lines. These distinctions are appropriately flagged at the interface through the `ipi_odd_pixel` and `ipi_odd_line` signals.
  - Chroma-shifted versions of YUV 420 data types have the same sequence of component as the non-chroma-shifted ones, so they are presented only once.
- Other types: Deliver one pixel per clock cycle.

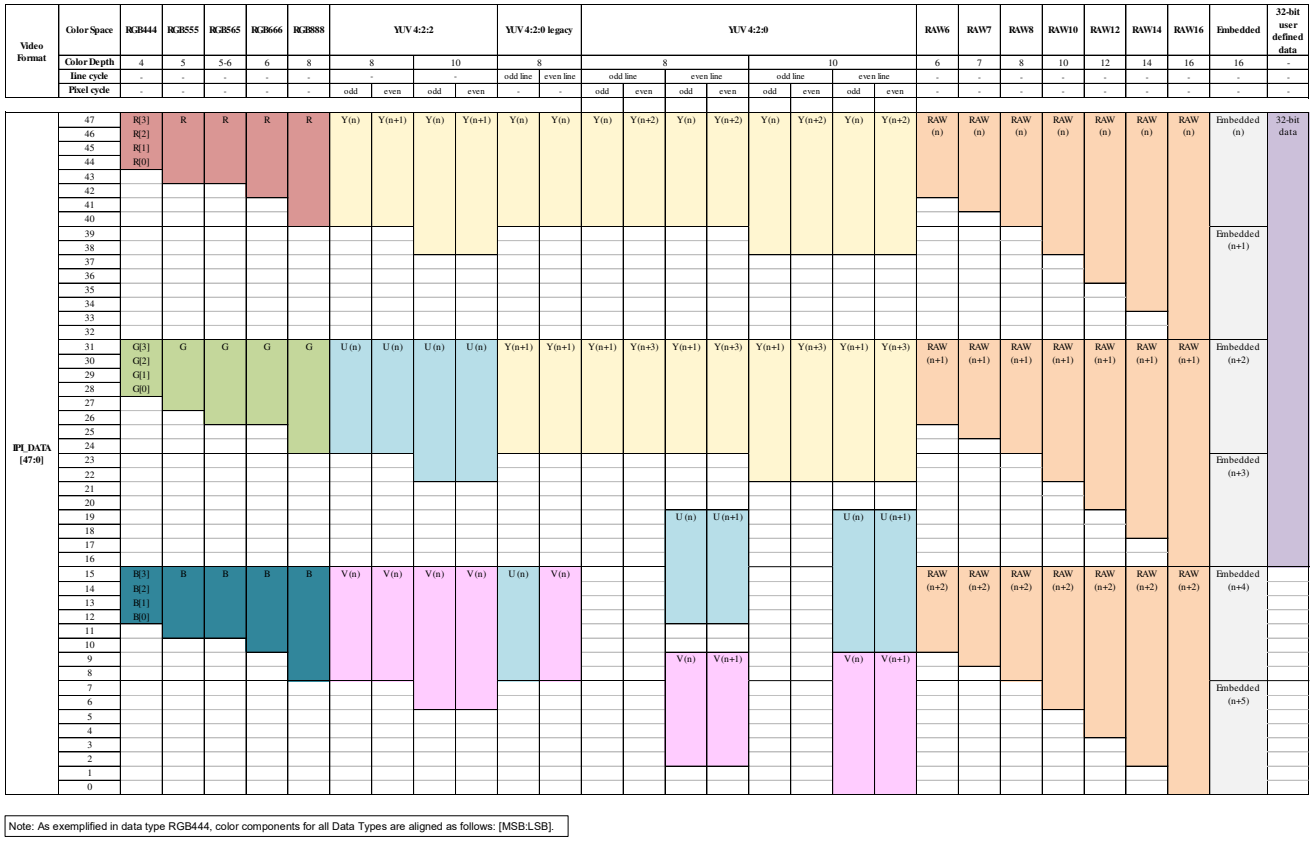


Figure 312. Location of Pixel Components at 48-bit IPI_Data for Each Data Type

54.2.6.1.2.2 IPI 16-bit Interface

The pixel color coding at the 16-bit interface allows IPI to deliver one color component at a time, instead of an entire pixel (or pair of pixels). This is configurable using the `IPI_MODE` register. When selected, the color components are delivered as shown in Figure 313. Each color component is present in a single location in a given clock cycle. Figure 313 describes the order in which each of them is transmitted.

For YUV420 data type, there is a distinction in the number of color components for odd or even lines, same as that in the 48-bit interface.

NOTE

The IPI 16-bit interface does not support RAW20 and RAW24 color codings.

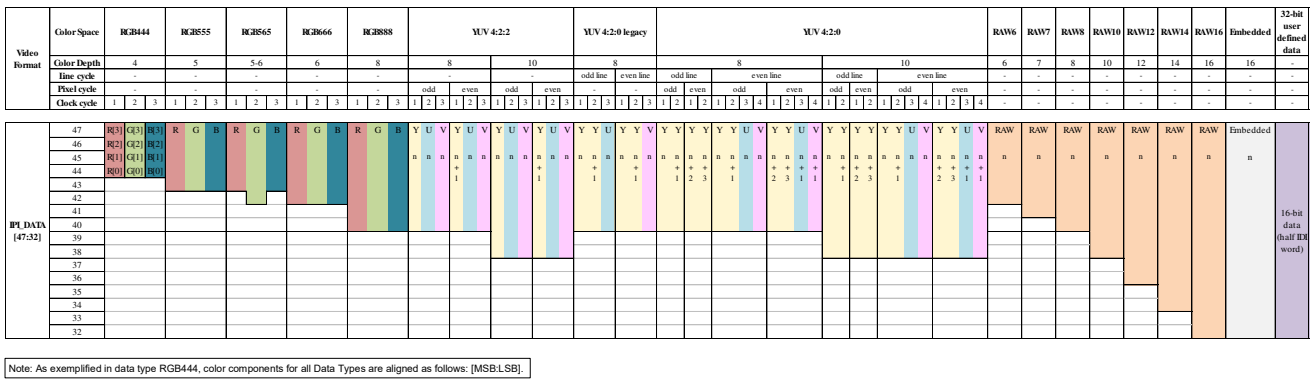


Figure 313. Location of Pixel Components at 16-bit IPI_Data for Each Data Type

54.2.6.1.3 IPI Additional Features

The pixel interface provides additional functionalities and added flexibility to the IPI behavior.

54.2.6.1.3.1 Data Type Overwrite

This feature changes how the IPI process receives the data. When active (bit 0 of the `IPI_ADV_FEATURES` register is set to 1), the data type configured in the `IPI_ADV_FEATURES` register decodes the IPI data instead of the received data type from the camera.

NOTE

Configure `IPI_DATA_TYPE` with the data type received from the camera.

To ensure a proper decoding, the configured data types must respect the relations present in Table 532.

Table 532. Supported Data Types for Data Type Overwrite Feature

Decode from IPI_DATA_TYPE	Decode to IPI_ADV_FEATURES Data type
Any data type except RGB444, YUV420 8-bit (legacy) and YUV422 8-bit	Any data type

54.2.6.1.3.2 End Frame Pin

The IPI interface provides the `ipi_end_frame` pin to indicate when the frame is transmitted. The IPI determines the end of the current frame and generates a pulse indicating that.

54.2.6.1.3.3 Support Embedded Data in IPI

According to the *MIPI CSI-2 Specification*, it is possible to embed extra lines containing additional information to the beginning and/or to the end of each picture frame.

In the Camera Timing mode, with this feature active, the MIPI_CSI2 controller is able to transmit the embedded data at the IPI interface. To differentiate pixel data from embedded data, use the output signal `ipi_embedded` on the IPI. This feature is active when you set `IPI_DATA_TYPE[8]` register to '1'.

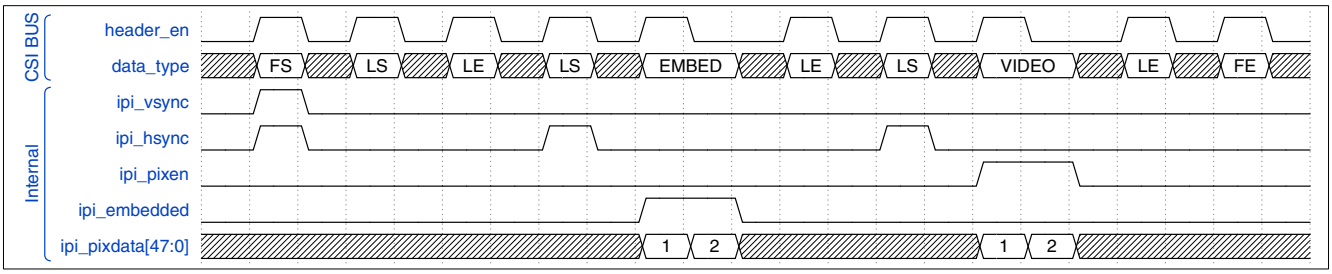


Figure 314. Example of Frame Generated when Embedded Data is Supported at the IPI Interface

Figure 314 shows an example where embedded data is present at the beginning of the frame. Table 533 provides the interpretation of output signal `ipi_pixen` and `ipi_embedded` values.

Table 533. Interpretation of Output Signal ipi_pixen and ipi_embedded Values

ipi_pixen	ipi_embedded	Interpretation
0	1	embedded data
1	0	video data

54.2.6.1.3.4 Memory Flush

The `IPI_MEM_FLUSH` register of the MIPI_CSI2 controller flush the memory, in case the memory is corrupted. Following are the ways to flush the memory:

- Manual: Set bit 0 of the register `IPI_MEM_FLUSH` to 1.
- Automatic: Set bit 8 of the `IPI_MEM_FLUSH` register to 1. This flushes the IPI memory when a `ipi_vsync` (frame start) is received. In the Automatic flush mechanism, it is not possible to overlap frames (that is, start buffering a new frame before the previous frame is transmitted).

The flush process must be concluded before the data can be stored in the memory. If the pixel clock is slower than the PHY clock the following conditions must be met:

- `time_to_flush_mem < time_frame_start_to_video`
- `time_to_flush_mem = (4 + 2 * CSI2_HOST_DFLT_F_SYNC_TYPE)/BytesPerHsClk.T`$_{PPI-HS}$ `+ (4 + CSI2_HOST_DFLT_F_SYNC_TYPE).T`$_{IPI}$

Table 534 describes the variables in the above formula.

Table 534. Variable Description for Determining When to Flush the Memory

Variable	Description
time_frame_start_to_video	Time between Frame Start and the first Video Packet
time_to_flush_mem	Time to Flush Memory
BytesPerHsClk	Bytes Received in each T_{PPI-HS}

Table continues on the next page...

Table 534. Variable Description for Determining When to Flush the Memory (continued)

Variable	Description
	• In D-PHY PPI-8, it is 1
T_{PPI-HS}	PPI High-Speed Clock Period
T_{IPI}	IPI Clock Period

All the remaining variables represent the parameters with the same name.

Figure 315 describes the relation between these parameters.

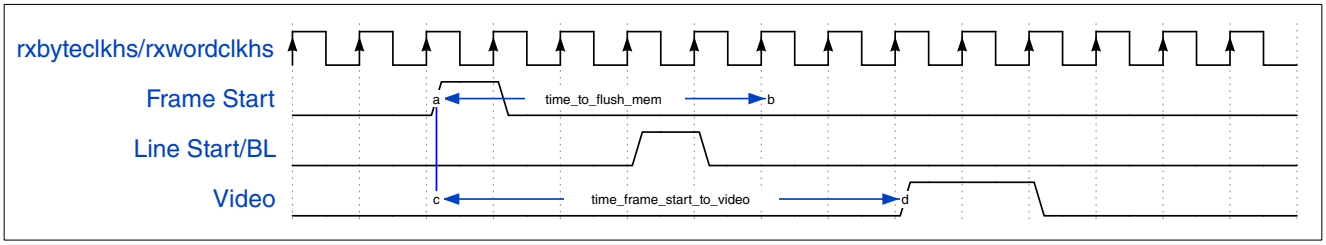


Figure 315. Auto Memory Flush Requirements

54.2.6.1.4 Advanced Image Pixel Interface

When enabled in the MIPI_CSI2 controller, Advanced Image Pixel Interface provides some useful information by adding extra pins on pixel interface:

- ipi_end_line
- ipi_data_valid
- ipi_data_begin and ipi_data_end

54.2.6.1.4.1 End-of-Line Indication

The IPI provides the `ipi_end_line` pin to indicate when a line is transmitted. The MIPI_CSI2 controller determines the end of the current line and generates a single pulse indicating that, as shown in Figure 316.

To enable line end indication, image frame must include Line Start (LS) synchronization packets and Line End (LE) synchronization packets in each line of the frame. If these packets are not received, ipi_end_line is not triggered.

This feature is only supported when:

- Frame Start does not trigger any sync event (`IPI_ADV_FEATURES[24] = 0`).
- When controller is operating in the Camera Timing (`IPI_MODE[0] = 0`).

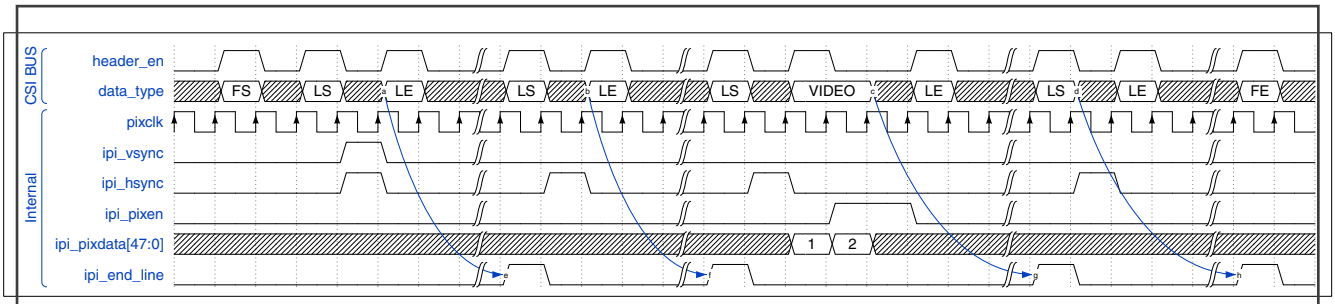


Figure 316. End-of-Line Indication

54.2.6.1.4.2 Number of Valid Pixels

IPI provides `ipi_data_valid[2:0]` bus to indicate the number of valid pixels/bytes transmitted per clock cycle. This is valid when `ipi_pixen` or `ipi_embedded` signals are active.

Figure 317 and Figure 318 illustrates examples of the `ipi_data_valid[2:0]` signal when IPI processes RAW and Embedded data types.

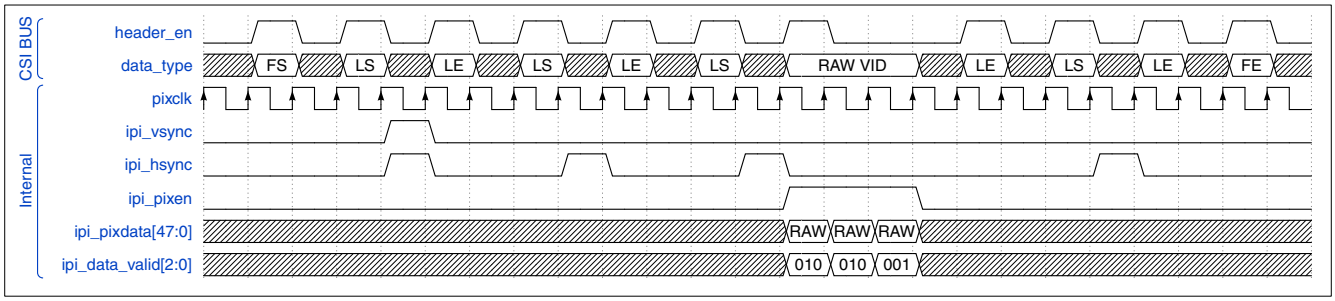


Figure 317. Number of Valid Pixels

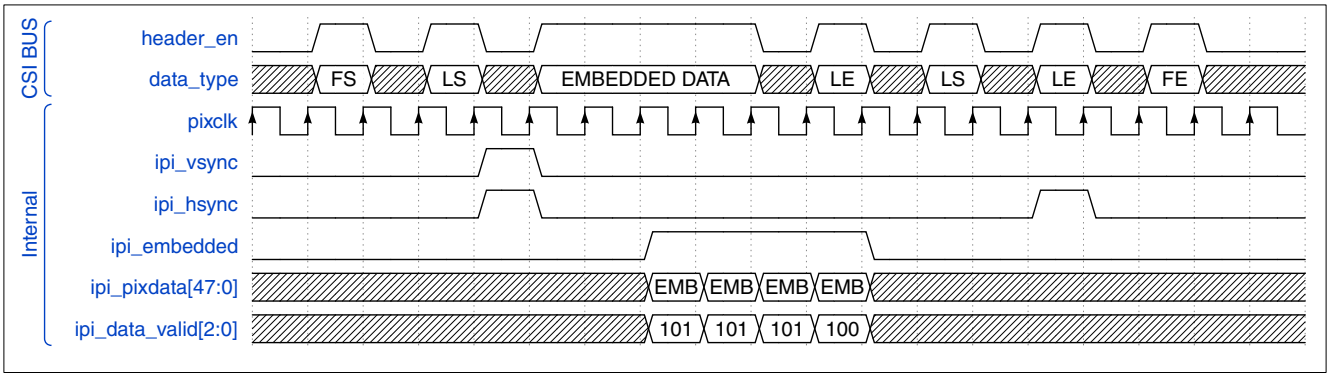


Figure 318. ipi_data_valid[2:0] When IPI Processes Embedded Data

54.2.6.1.4.3 First and Last Data Valid Indications

The IPI provides the `ipi_data_begin` and `ipi_data_end` pins to indicate transmission of the first and the last pixel of an entire line. This must be a single pulse, as shown in Figure 24.

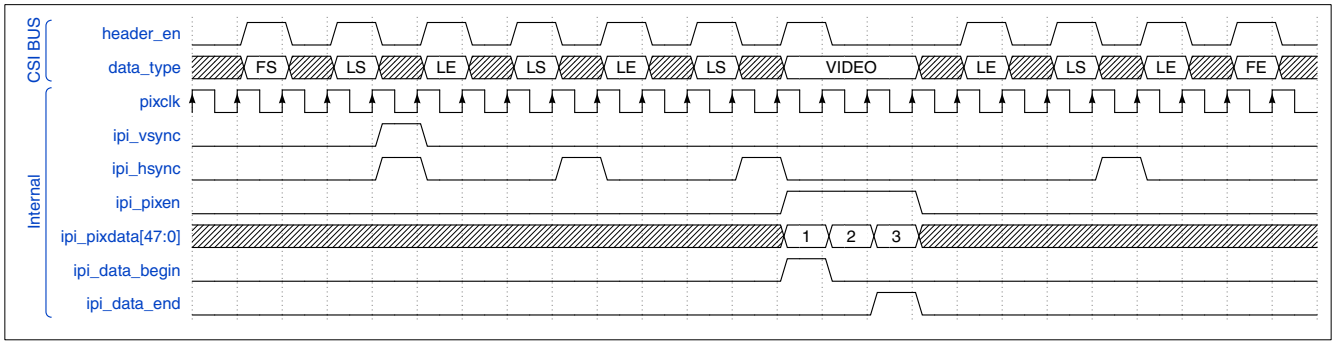


Figure 319. First and Last Data Valid Indications

54.2.7 Clocking

The MIPI_CSI2 host controller includes the following clock signals:

Table 535. MIPI_CSI2 Host Controller Clocks

Clock Name	Description	Direction
pclk	APB Bus Clock.The rising edge of pclk is used to time all transfers in the APB bus. The minimum required frequency is 20 MHz.	Input
fpclk	Free Running Version of APB Bus Clock pclk. In case the clock signal pclk is not always available during normal operation (for example, gating of pclk when not using the APB interface, in order to save power consumption), then a free running version clock is required.	Input
scanbyteclk	Scan Clock for D-PHY byte domain.	Input
clkn	Negative D-PHY Differential Clock Line	Input
clkp	Positive D-PHY Differential Clock Line	Input
scanclk	scan clk	Input
cfg_clk	D-PHY configuration clock used for the initialization of the PHY. It is also used for exiting the ULP state.	Input
pixclk	Image Pixel Interface (IPI) clock . There is no protocol limitation for this clock frequency.	Input

54.2.8 Reset

The MIPI_CSI2 host controller includes the following reset signal:

Table 536. MIPI_CSI2 host controller reset signals

Reset signal	Description
presetn	Peripheral reset. Global reset of the controller including all registers.

54.2.9 Interrupts

The interrupt structure in MIPI_CSI2 assumes:

- The interrupt registers are accessible through the AMBA APB bus.
- The interrupts are running in the AMBA APB clock domain (`pclk`).

Figure 320 shows the interrupt structure in the MIPI_CSI2 controller.

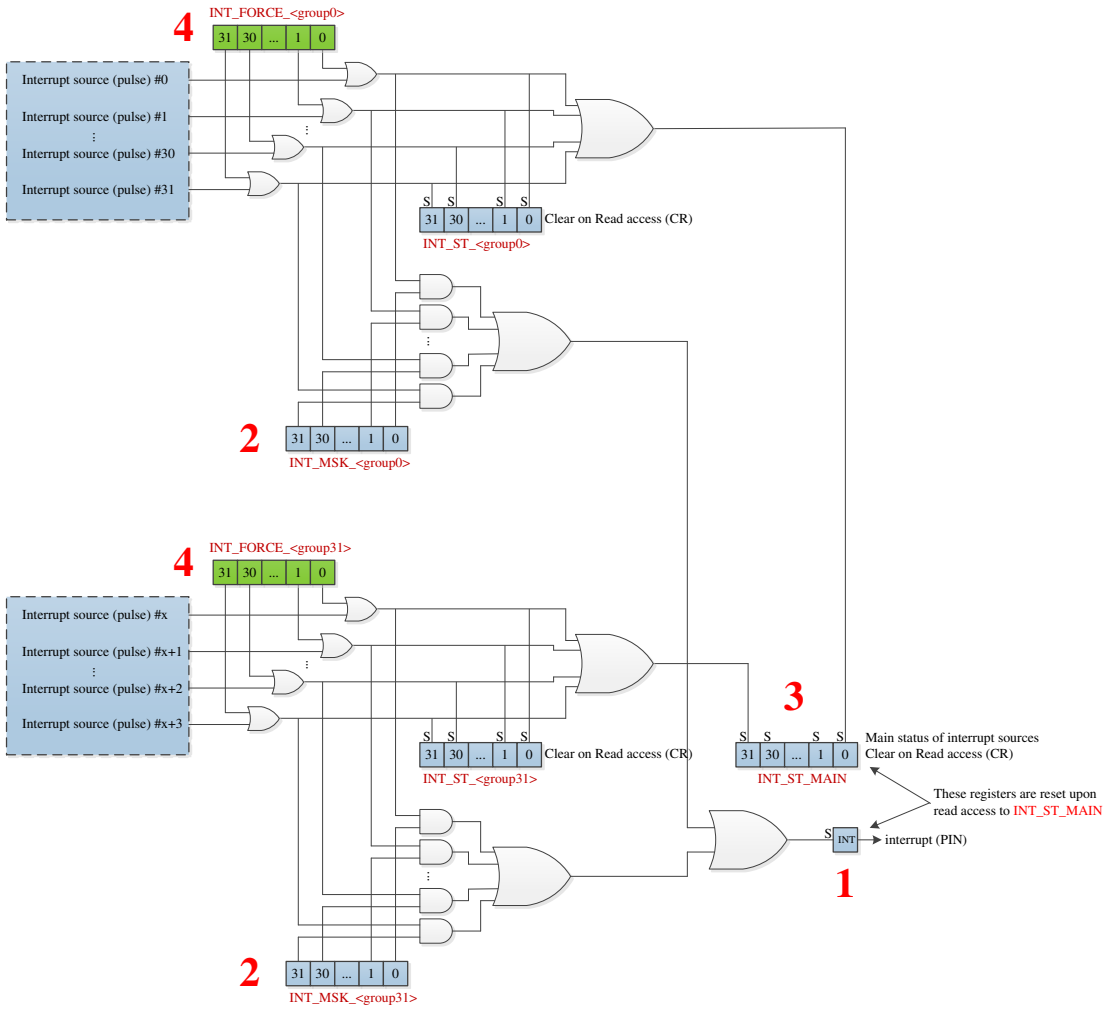


Figure 320. MIPI_CSI2 Interrupt Structure

Based on Figure 320 the MIPI_CSI2 controller interrupt structure can be explained as follows:

1. There is a single interrupt pin registered with the `pclk` signal, simplifying the system integration. The controller sets the interrupt pin and the applicable Interrupt Status registers at the same time.

2. The default value of the INT_MSK_<group> (Interrupt Mask) registers is 0, thus masking the interrupts. By default, the interrupt pin is disabled.
3. The `INT_ST_MAIN` register indicates the names of the INT_ST_<group> (Interrupt Status) registers which triggered the interrupt. The INT_ST_<group> (Interrupt Status) registers contain the status of the individual interrupt sources and is independent of the contents of the associated INT_MSK_<group> registers. So, it is possible to service the INT_ST_<group> (Interrupt Status) registers by polling. The INT_ST_<group> (Interrupt Status) registers are cleared when read. Reading the `INT_ST_MAIN` register also clears the interrupt pin.
4. The INT_FORCE_<group> (Interrupt Force) registers are used for testing purpose. These registers allow triggering of interrupt events individually, without activating the conditions that trigger the interrupt sources. This feature facilitates development and testing of the software associated with the interrupt events. The INT_FORCE_<group> registers are auto-cleared. The circuitry minimizes the gate count overhead because it avoids the need to have the Interrupt Force registers.

54.3 External signals

This module has no external signals.

54.4 Initialization

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

To initialize the MIPI_CSI2 host controller use this programming sequence.

1. Release PHY test codes from reset
2. Configure the PHY frequency range
3. Perform additional PHY test code configuration (optional)
4. Configure the number of active lanes
5. Release PHY from reset
6. Configure the data id values (optional)
7. Define errors to be masked
8. Release MIPI_CSI2 from reset
9. Check that data lanes are in Stop state

54.5 Memory Map and register definition

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

This section includes the MIPI_CSI2 module memory map and detailed descriptions of all registers.

54.5.1 MIPI_CSI2 register descriptions

54.5.1.1 mipi_csi2_host memory map

MIPI_CSI2 Registers

MIPI_CSI base address: 4AE0_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Core Version (VERSION)	32	R	3135_302A
4	Number of Lanes (N_LANES)	32	RW	0000_0001
8	Logic Reset (CSI2_RESETN)	32	RW	0000_0000
C	Main Interrupt Status (INT_ST_MAIN)	32	R	0000_0000
40	PHY Shutdown (PHY_SHUTDOWNZ)	32	RW	0000_0000
44	DPHY Reset (DPHY_RSTZ)	32	RW	0000_0000
48	RX PHY Status (PHY_RX)	32	R	0001_0000
4C	STOPSTATE PHY Status (PHY_STOPSTATE)	32	R	0000_0000
50	D-PHY Test and Control Interface 1 (PHY_TEST_CTRL0)	32	RW	0000_0001
54	D-PHY Test and Control Interface 2 (PHY_TEST_CTRL1)	32	RW	0000_0000
60	Pattern Generator Vertical Resolution (PPI_PG_PATTERN_VRES)	32	RW	0000_0000
64	Pattern Generator Horizontal Resolution (PPI_PG_PATTERN_HRES)	32	RW	0000_0000
68	Pattern Generator (PPI_PG_CONFIG)	32	RW	0000_0000
6C	Pattern Generator Enable (PPI_PG_ENABLE)	32	RW	0000_0000
70	Pattern Generator Status (PPI_PG_STATUS)	32	R	0000_0000
80	IPI Mode (IPI_MODE)	32	RW	0000_0000
84	IPI Virtual Channel (IPI_VCID)	32	RW	0000_0000
88	IPI Data Type (IPI_DATA_TYPE)	32	RW	0000_0000
8C	IPI Flush Memory (IPI_MEM_FLUSH)	32	RW	0000_0000
90	IPI HSA (IPI_HSA_TIME)	32	RW	0000_0000
94	IPI HBP (IPI_HBP_TIME)	32	RW	0000_0000
98	IPI HSD (IPI_HSD_TIME)	32	RW	0000_0000
9C	IPI HLINE (IPI_HLINE_TIME)	32	RW	0000_0000
A0	IPI Soft Reset (IPI_SOFTRSTN)	32	RW	0000_0001
AC	IPI Advanced Features (IPI_ADV_FEATURES)	32	RW	0000_0000
B0	IPI VSA (IPI_VSA_LINES)	32	RW	0000_0000
B4	IPI VBP (IPI_VBP_LINES)	32	RW	0000_0000
B8	IPI VFP (IPI_VFP_LINES)	32	RW	0000_0000
BC	IPI VACTIVE (IPI_VACTIVE_LINES)	32	RW	0000_0000
E0	Fatal Interruption Caused by PHY (INT_ST_PHY_FATAL)	32	R	0000_0000
E4	Mask for Fatal Interruption Caused by PHY (INT_MSK_PHY_FATAL)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
E8	Force for Fatal Interruption Caused by PHY (INT_FORCE_PHY_FATAL)	32	RW	0000_0000
F0	Fatal Interruption Caused During Packet Construction (INT_ST_PKT_FATAL)	32	R	0000_0000
F4	Mask for Fatal Interruption Caused During Packet Construction (INT_MSK_PKT_FATAL)	32	RW	0000_0000
F8	Force for Fatal Interruption Caused During Packet Construction (INT_FORCE_PKT_FATAL)	32	RW	0000_0000
110	Interruption Caused by PHY (INT_ST_PHY)	32	R	0000_0000
114	Mask for Interruption Caused by PHY (INT_MSK_PHY)	32	RW	0000_0000
118	Force for Interruption Caused by PHY (INT_FORCE_PHY)	32	RW	0000_0000
140	Fatal Interruption Caused by IPI Interface (INT_ST_IPI_FATAL)	32	R	0000_0000
144	Mask for Fatal Interruption Caused by IPI Interface (INT_MSK_IPI_FATAL)	32	RW	0000_0000
148	Force for Fatal Interruption Caused by IPI Interface (INT_FORCE_IPI_FATAL)	32	RW	0000_0000
280	Fatal Interruption Caused by Frame Boundaries (INT_ST_BNDRY_FRAME_FATAL)	32	R	0000_0000
284	Mask for Fatal Interruption Caused by Frame Boundaries (INT_MSK_BNDRY_FRAME_FATAL)	32	RW	0000_0000
288	Force for Fatal Interruption Caused by Frame Boundaries (INT_FORCE_BNDRY_FRAME_FATAL)	32	RW	0000_0000
290	Fatal Interruption Caused by Frame Sequence (INT_ST_SEQ_FRAME_FATAL)	32	R	0000_0000
294	Mask for Fatal Interruption Caused by Frame Sequence (INT_MSK_SEQ_FRAME_FATAL)	32	RW	0000_0000
298	Force for Fatal Interruption Caused by Frame Sequence (INT_FORCE_SEQ_FRAME_FATAL)	32	RW	0000_0000
2A0	Fatal Interruption Caused by Frame CRC (INT_ST_CRC_FRAME_FATAL)	32	R	0000_0000
2A4	Mask for Fatal Interruption Caused by Frame CRC (INT_MSK_CRC_FRAME_FATAL)	32	RW	0000_0000
2A8	Force for Fatal Interruption Caused by Frame CRC (INT_FORCE_CRC_FRAME_FATAL)	32	RW	0000_0000
2B0	Fatal Interruption Caused by Payload CRC (INT_ST_PLD_CRC_FATAL)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2B4	Mask for Fatal Interruption Caused by Payload CRC (INT_MSK_PLD_CRC_FATAL)	32	RW	0000_0000
2B8	Force for Fatal Interruption Caused by Payload CRC (INT_FORCE_PLD_CRC_FATAL)	32	RW	0000_0000
2C0	Interruption Caused by Data Type (INT_ST_DATA_ID)	32	R	0000_0000
2C4	Mask for Interruption Caused by Data Type (INT_MSK_DATA_ID)	32	RW	0000_0000
2C8	Force for Interruption Caused by Data Type (INT_FORCE_DATA_ID)	32	RW	0000_0000
2D0	Interruption Caused by Header Single Bit Errors (INT_ST_ECC_CORRECTED)	32	R	0000_0000
2D4	Mask for Interruption Caused by Header Single Bit Errors (INT_MSK_ECC_CORRECTED)	32	RW	0000_0000
2D8	Force for Interruption Caused by Header Single Bit Errors (INT_FORCE_ECC_CORRECTED)	32	RW	0000_0000
300	Data De-Scrambling (SCRAMBLING)	32	RW	0000_0000
304	De-scrambler Seed for Lane 1 (SCRAMBLING_SEED1)	32	RW	0000_1008
308	De-scrambler Seed for Lane 2 (SCRAMBLING_SEED2)	32	RW	0000_1188

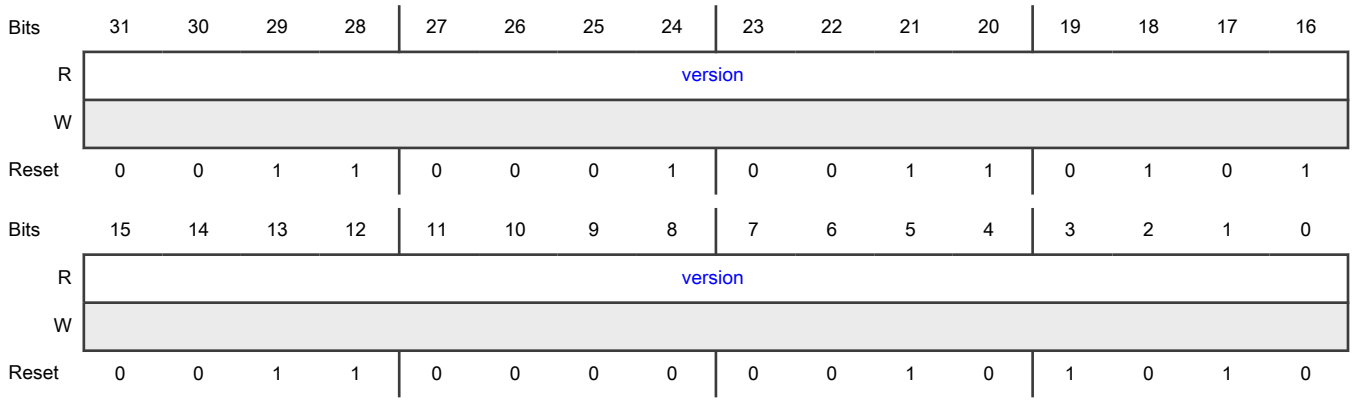
54.5.1.2 Core Version (VERSION)

Contains the version of MIPI_CSI2 host coded in 32-bit ASCII code.

Offset

Register	Offset
VERSION	0h

Diagram



Fields

Field	Description
31-0 version	Indicates the version of the MIPI_CSI2 host.

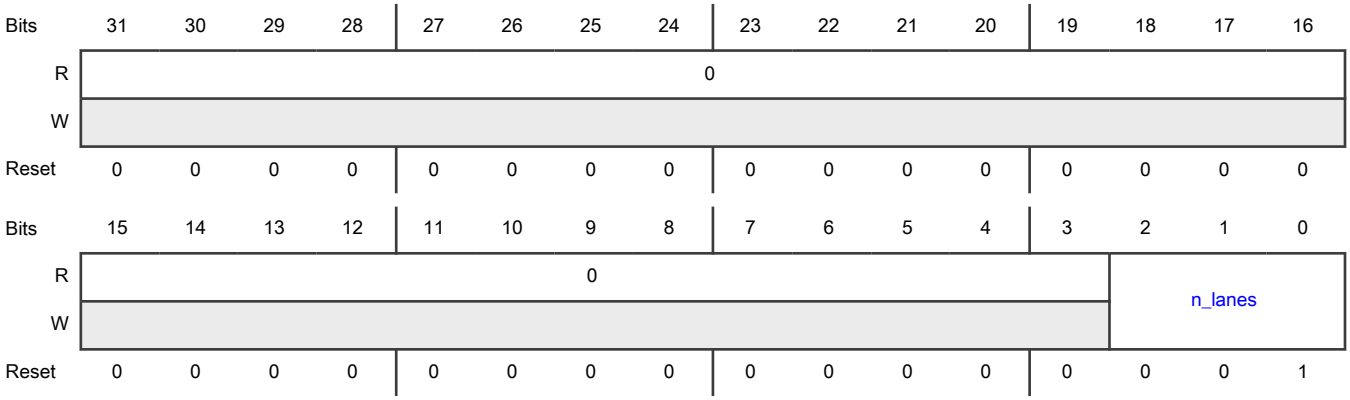
54.5.1.3 Number of Lanes (N_LANES)

Configures the number of active lanes that the MIPI_CSI2 host uses to receive the camera device data.

Offset

Register	Offset
N_LANES	4h

Diagram



Fields

Field	Description
31-3 —	Reserved and read as zero.
2-0 n_lanes	N_LANES can only be updated when the PHY lane is in stop state. Number of active data lanes: 000 - 1 Data Lane 001 - 2 Data Lanes 010 - Reserved 011 - Reserved 100 - Reserved 101 - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
	110 - Reserved 111 - Reserved

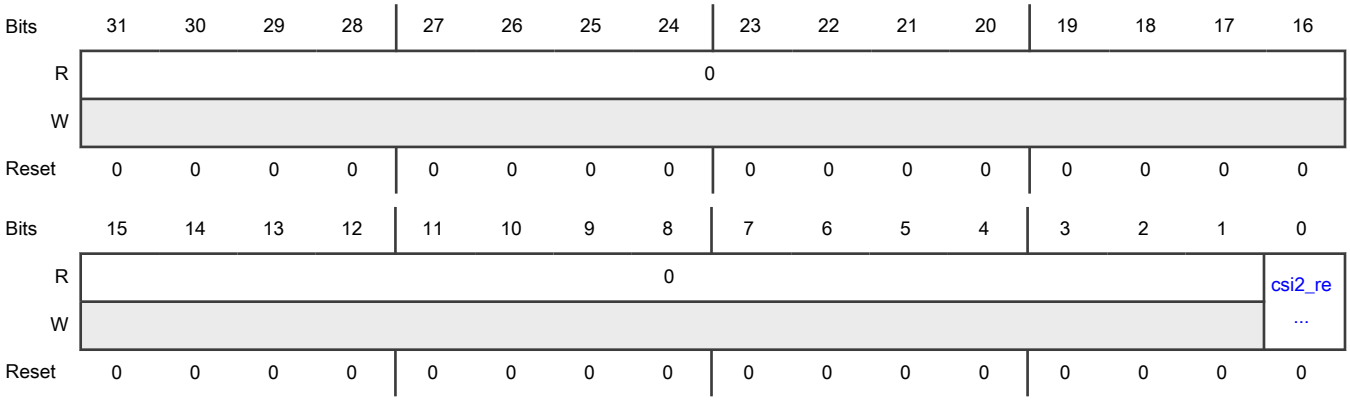
54.5.1.4 Logic Reset (CSI2_RESETN)

Controls the MIPI_CSI2 host logic reset state. When activated, the internal logic of the controller goes into the reset state. The configuration is not reset to default values with this register, instead, only the internal controller logic is affected.

Offset

Register	Offset
CSI2_RESETN	8h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 csi2_resetn	MIPI_CSI2 host reset output. Active Low.

54.5.1.5 Main Interrupt Status (INT_ST_MAIN)

Contains the status of individual interrupt sources, regardless of the contents of the associated interrupt mask registers, so it is possible to service the interrupt status registers by polling. Reading INT_ST_MAIN register clears the interrupt pin.

Offset

Register	Offset
INT_ST_MAIN	Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						0	0	0	0	0	0	0	status_...	0	status_...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								status_...	status_...	status_...	status_...	status_...	status_...	status_...	status_...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-26 —	Reserved and read as zero.
25 —	Reserved and read as zero.
24 —	Reserved and read as zero.
23 —	Reserved and read as zero.
22 —	Reserved and read as zero.
21 —	Reserved and read as zero.
20 —	Reserved and read as zero.
19	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
18 status_int_ipi_fatal	Status of Fatal Interrupt caused by IPI Interrupt Status of INT_ST_IPI_FATAL. After a read operation, the field's value clears to 0.
17 —	Reserved and read as zero.
16 status_int_phy	Status of Interrupt caused by PHY Status of INT_ST_PHY. After a read operation, the field's value clears to 0.
15-8 —	Reserved and read as zero.
7 status_int_ecc_corrected	Status of Interrupt caused by header error detected and corrected Status of INT_ST_ECC_CORRECTED. After a read operation, the field's value clears to 0.
6 status_int_data_id	Status of Interrupt caused by Data Type Status of INT_ST_DATA_ID. After a read operation, the field's value clears to 0.
5 status_int_pld_crc_fatal	Status of Fatal Interrupt caused by Payload CRC Status of INT_ST_PLD_CRC_FATAL. After a read operation, the field's value clears to 0.
4 status_int_crc_frame_fatal	Status of Fatal Interrupt caused by Frame CRC Status of INT_ST_CRC_FRAME_FATAL. After a read operation, the field's value clears to 0.
3 status_int_seq_frame_fatal	Status of Interrupt caused by Frame Sequence Status of INT_ST_SEQ_FRAME_FATAL. After a read operation, the field's value clears to 0.
2 status_int_bndry_frame_fatal	Status of Fatal Interrupt caused by Frame Boundaries Status of INT_ST_BNDRY_FRAME_FATAL. After a read operation, the field's value clears to 0.
1	Status of Fatal Interrupt caused during Packet Construction.

Table continues on the next page...

Table continued from the previous page...

Field	Description
status_int_pkt_fatal	Status of INT_ST_PKT_FATAL. After a read operation, the field's value clears to 0.
0 status_int_phy_fatal	Status of Fatal Interrupt caused by PHY Status of INT_ST_PHY_FATAL. After a read operation, the field's value clears to 0.

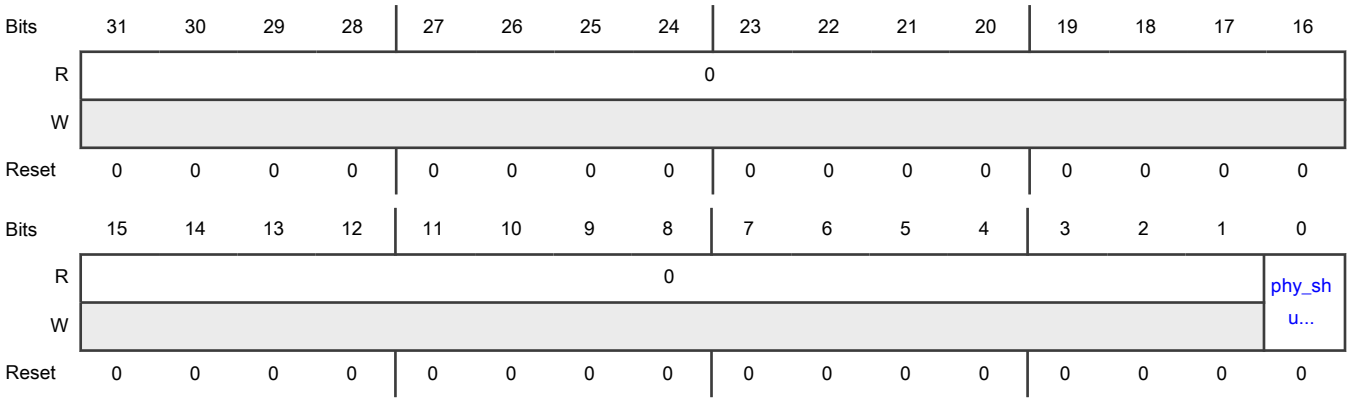
54.5.1.6 PHY Shutdown (PHY_SHUTDOWNZ)

Controls the PHY Shutdown mode. In this state, the PHY sets the analog and digital circuitry in the Reset state.

Offset

Register	Offset
PHY_SHUTDOWNZ	40h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 phy_shutdownz	Shutdown input. Used to place the complete macro in power down. All analog blocks are in Power-down mode and digital logic is cleared. Active Low.

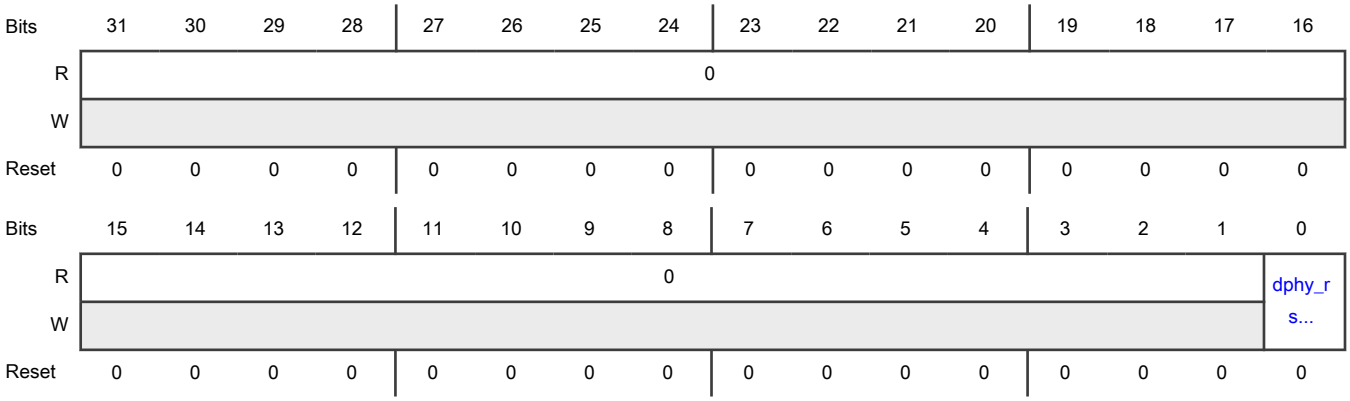
54.5.1.7 DPHY Reset (DPHY_RSTZ)

Controls the PHY Reset mode. In this state, the PHY sets the digital circuitry in the Reset State.

Offset

Register	Offset
DPHY_RSTZ	44h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 dphy_rstz	PHY reset output. Active Low.

54.5.1.8 RX PHY Status (PHY_RX)

Contains the status of RX-related signals from PHY:

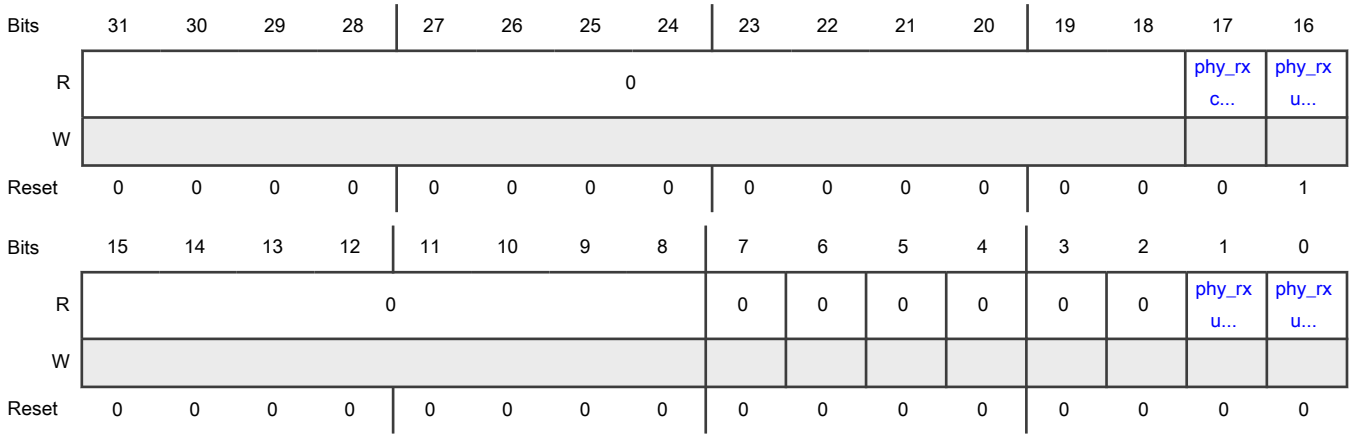
RXULPSESC* RXCLKACTIVEHS (D-PHY)

It contains information such as, if the clock lane is receiving high-speed clock signal, or if the data and clock lanes are in Ultra Low Power Mode (ULPM).

Offset

Register	Offset
PHY_RX	48h

Diagram



Fields

Field	Description
31-18 —	Reserved and read as zero.
17 phy_rxclkactivehs	Indicates that D-PHY clock lane is actively receiving a DDR clock
16 phy_rxulpsclknot	Active Low. Indicates that D-PHY Clock Lane module has entered the Ultra Low Power state
15-8 —	Reserved and read as zero.
7 —	Reserved and read as zero.
6 —	Reserved and read as zero.
5 —	Reserved and read as zero.
4 —	Reserved and read as zero.
3 —	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 —	Reserved and read as zero.
1 phy_rxulpsesc_1	Lane module 1 has entered the Ultra Low Power mode
0 phy_rxulpsesc_0	Lane module 0 has entered the Ultra Low Power mode.

54.5.1.9 STOPSTATE PHY Status (PHY_STOPSTATE)

Contains the STOPSTATE signal status from PHY.

Offset

Register	Offset
PHY_STOPSTATE	4Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															phy_stop...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0	0	0	0	0	phy_stop...	phy_stop...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-17 —	Reserved and read as zero.
16	D-PHY Clock lane in Stop state

Table continues on the next page...

Table continued from the previous page...

Field	Description
phy_stopstateclk	
15-8 —	Reserved and read as zero.
7 —	Reserved and read as zero.
6 —	Reserved and read as zero.
5 —	Reserved and read as zero.
4 —	Reserved and read as zero.
3 —	Reserved and read as zero.
2 —	Reserved and read as zero.
1 phy_stopstatedata_1	Data lane 1 in Stop state
0 phy_stopstatedata_0	Data lane 0 in Stop state.

54.5.1.10 D-PHY Test and Control Interface 1 (PHY_TEST_CTRL0)

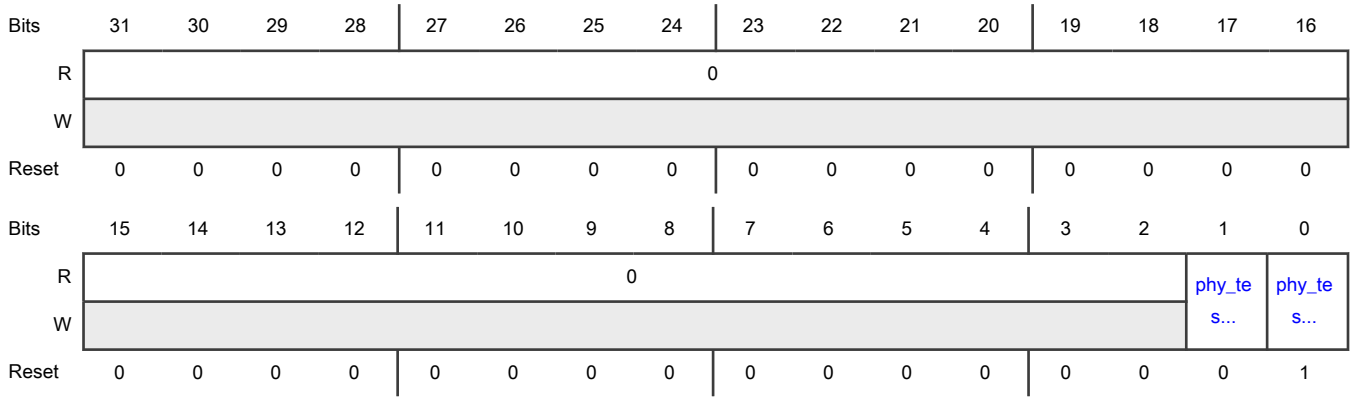
Contains control for vendor specific interface in the PHY.

Controls the D-PHY Test and Control interface. This register controls the TESTCLR and TESTCLK signals.

Offset

Register	Offset
PHY_TEST_CTRL0	50h

Diagram



Fields

Field	Description
31-2 —	Reserved and read as zero.
1 phy_testclk	Clock to capture testdin bus contents into the macro, with testen signal controlling the operation selection.
0 phy_testclr	When active, performs vendor specific interface initialization. Active High. **NOTE** This line needs an initial high pulse after power up for analog programmability default values to be preset.

54.5.1.11 D-PHY Test and Control Interface 2 (PHY_TEST_CTRL1)

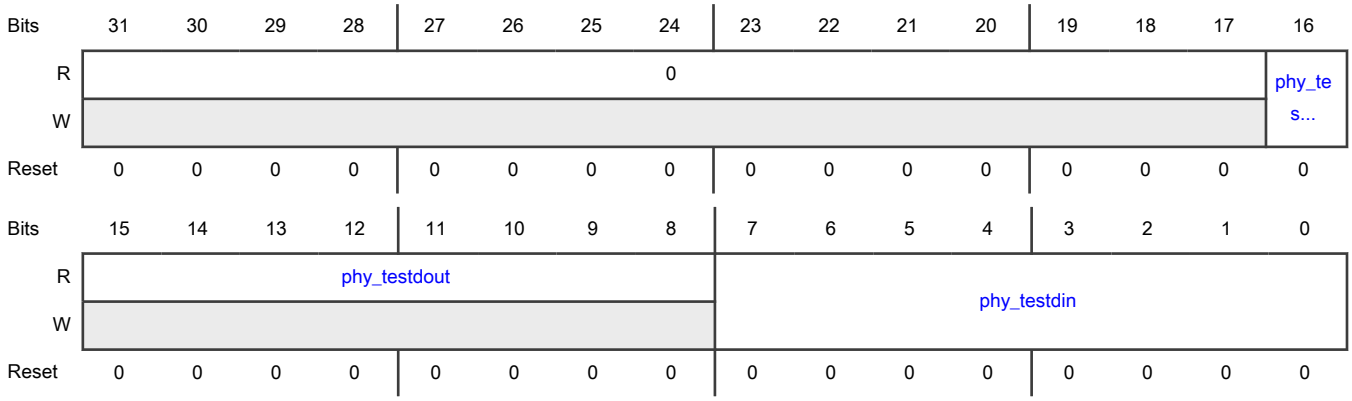
Contains control for vendor specific interface in the PHY.

Controls the D-PHY Test and Control interface. This register controls the TESTENABLE signal and the TESTDIN bus. Reading from this register retrieves the value from TESTOUT.

Offset

Register	Offset
PHY_TEST_CTRL1	54h

Diagram



Fields

Field	Description
31-17 —	Reserved and read as zero.
16 phy_testen	PHY test enable When asserted high, it configures an address write operation on the falling edge of testclk. When asserted low, it configures a data write operation on the rising edge of testclk.
15-8 phy_testdout	Vendor-specific 8-bit data output for reading data and other probing functionalities.
7-0 phy_testdin	Test interface 8-bit data input for programming internal registers and accessing test functionalities.

54.5.1.12 Pattern Generator Vertical Resolution (PPI_PG_PATTERN_VRES)

Configures the pattern vertical resolution of the PPI Pattern Generator.

Offset

Register	Offset
PPI_PG_PATTERN_VRES	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ppi_pg_pattern_vres															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved and read as zero.
15-0 ppi_pg_pattern_vres	Configures the pattern vertical resolution of the PPI Pattern Generator.

54.5.1.13 Pattern Generator Horizontal Resolution (PPI_PG_PATTERN_HRES)

Configures the pattern horizontal resolution of the PPI Pattern Generator.

Offset

Register	Offset
PPI_PG_PATTERN_HRES	64h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ppi_pg_pattern_hres															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved and read as zero.
15-0 ppi_pg_pattern_hres	Configures the pattern horizontal resolution of the PPI Pattern Generator.

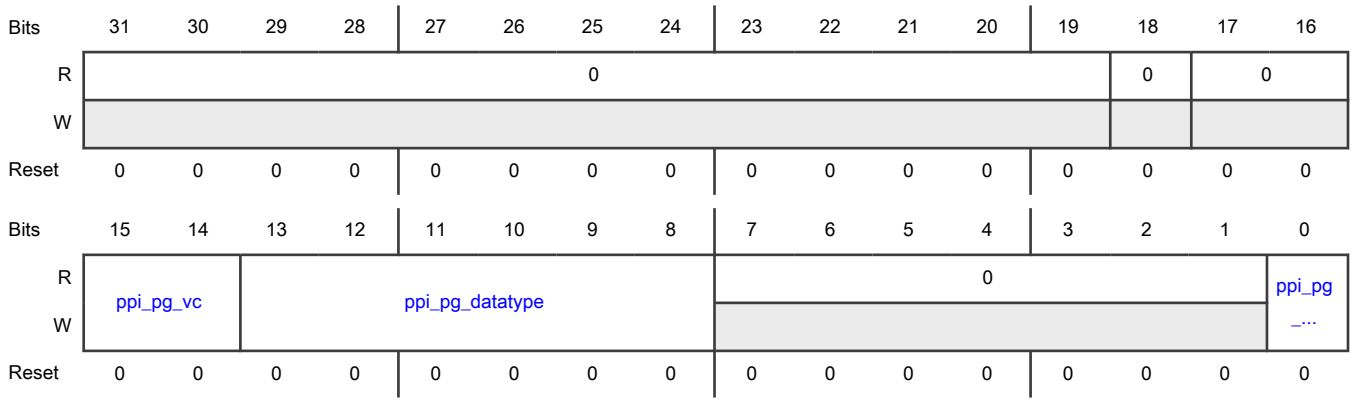
54.5.1.14 Pattern Generator (PPI_PG_CONFIG)

Configures the PPI Pattern Generator.

Offset

Register	Offset
PPI_PG_CONFIG	68h

Diagram



Fields

Field	Description
31-19 —	Reserved and read as zero.
18 —	Reserved and read as zero.
17-16 —	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-14 ppi_pg_vc	Configures the PPI Pattern Generator's virtual channel.
13-8 ppi_pg_datatype	Configures the PPI Pattern Generator's data type.
7-1 —	Reserved and read as zero.
0 ppi_pg_pattern	Configures the PPI Pattern Generator's pattern: 0 - Vertical Pattern 1 - Horizontal Pattern

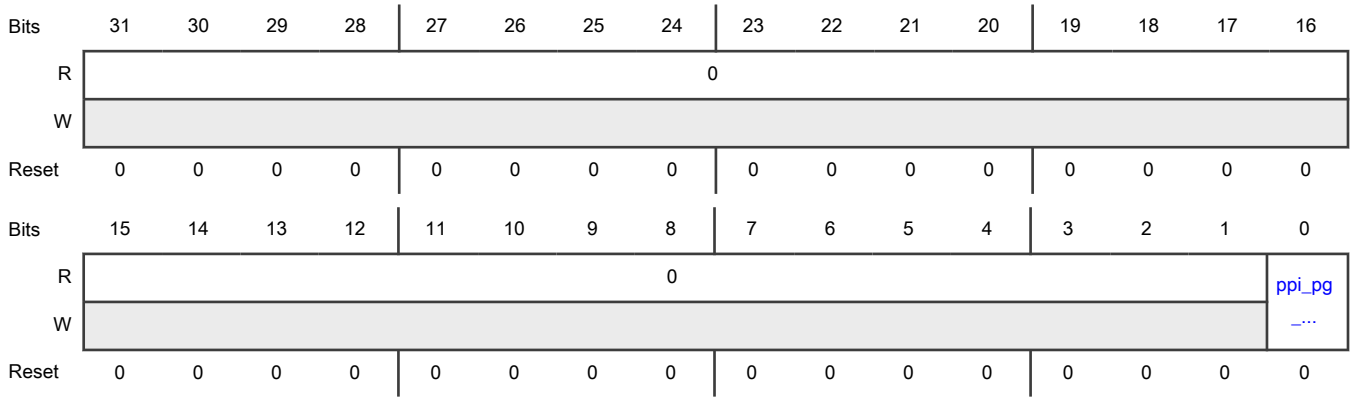
54.5.1.15 Pattern Generator Enable (PPI_PG_ENABLE)

Enables the PPI Pattern Generator.

Offset

Register	Offset
PPI_PG_ENABLE	6Ch

Diagram



Fields

Field	Description
31-1	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
0 ppi_pg_enable	Enables the PPI Pattern Generator.

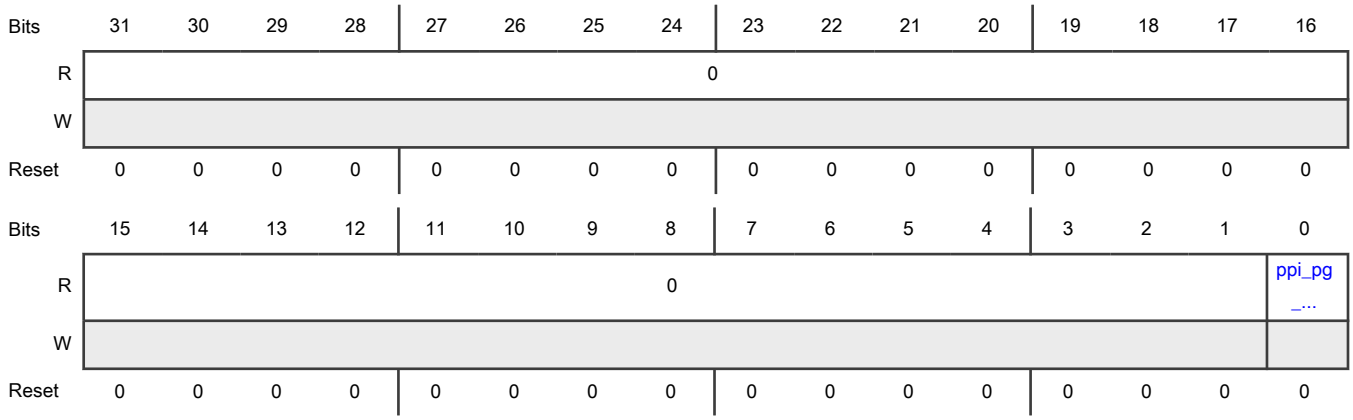
54.5.1.16 Pattern Generator Status (PPI_PG_STATUS)

Indicates PPI Pattern Generator Status.

Offset

Register	Offset
PPI_PG_STATUS	70h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 ppi_pg_status	PPI Pattern Generator status: 0 - PPIPG is inactive 1 - PPI PG is running

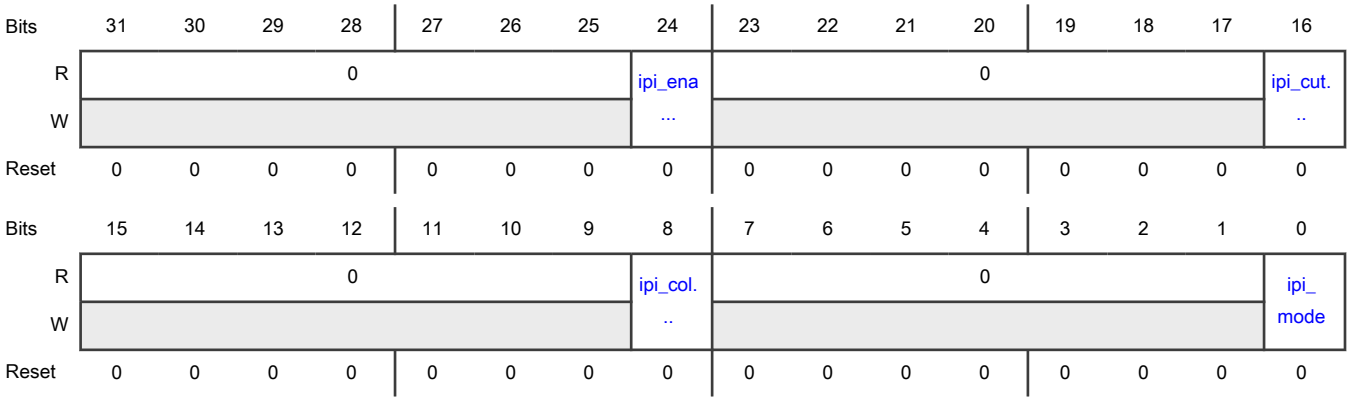
54.5.1.17 IPI Mode (IPI_MODE)

Selects how the IPI interface generates the video frame.

Offset

Register	Offset
IPI_MODE	80h

Diagram



Fields

Field	Description
31-25 —	Reserved and read as zero.
24 ipi_enable	Enables the interface.
23-17 —	Reserved and read as zero.
16 ipi_cut_through	Indicates cut-through mode state: 0 - Cut-through mode inactive 1 - Cut-through mode active
15-9 —	Reserved and read as zero.
8 ipi_color_com	Indicates how color mode components are delivered as follows: 0 - 48 bits interface 1 - 16 bits interface
7-1 —	Reserved and read as zero.
0	Indicates the video mode transmission type as follows:

Table continues on the next page...

Table continued from the previous page...

Field	Description
ipi_mode	0 - Camera timing 1 - Controller timing

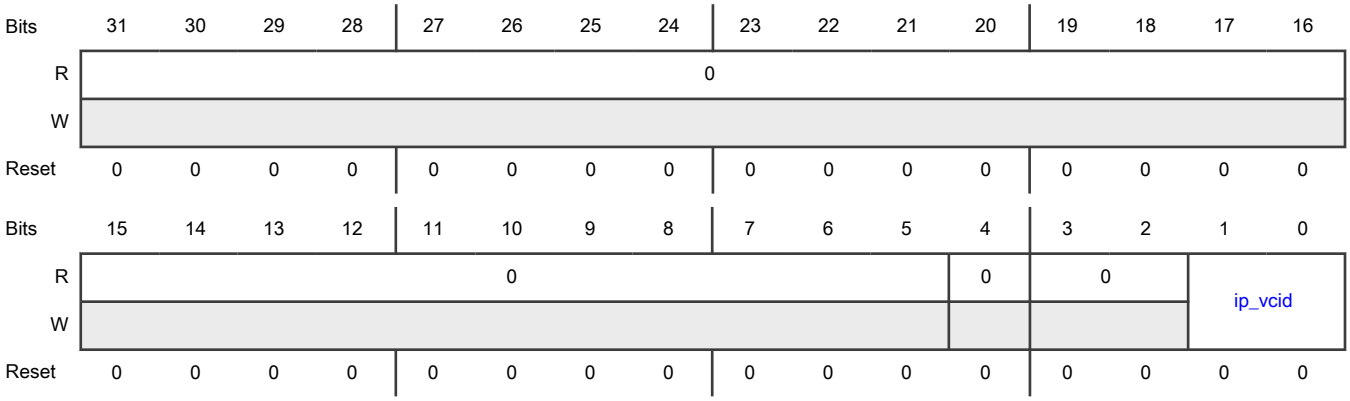
54.5.1.18 IPI Virtual Channel (IPI_VCID)

Selects the virtual channel processed by IPI.

Offset

Register	Offset
IPI_VCID	84h

Diagram



Fields

Field	Description
31-5 —	Reserved and read as zero.
4 —	Reserved and read as zero.
3-2 —	Reserved and read as zero.
1-0 ip_vcid	Virtual channel of data to be processed by pixel interface.

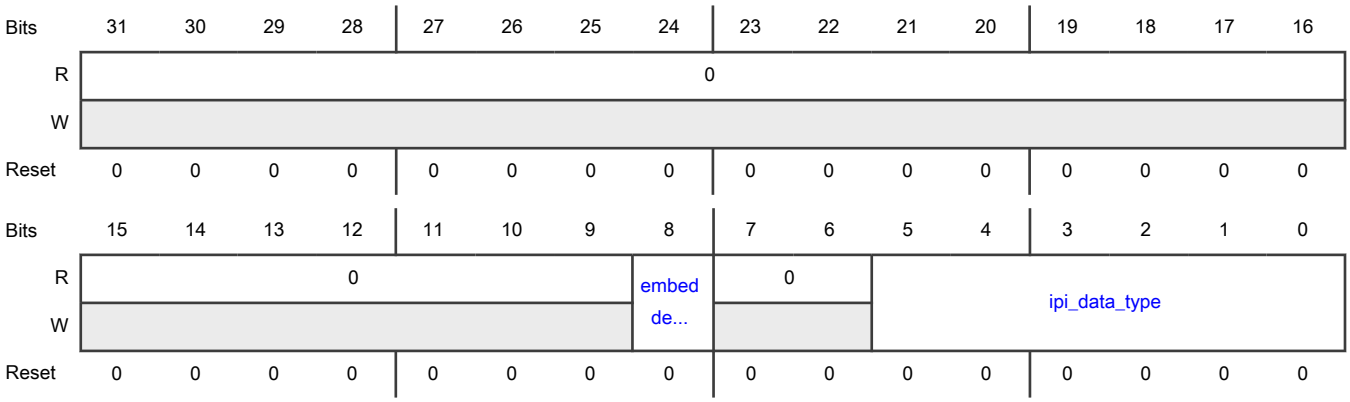
54.5.1.19 IPI Data Type (IPI_DATA_TYPE)

Selects the data type processed by IPI.

Offset

Register	Offset
IPI_DATA_TYPE	88h

Diagram



Fields

Field	Description
31-9 —	Reserved and read as zero.
8 embedded_data	Enables embedded data processing on IPI interface.
7-6 —	Reserved and read as zero.
5-0 ipi_data_type	Data type of data to be processed by pixel interface.

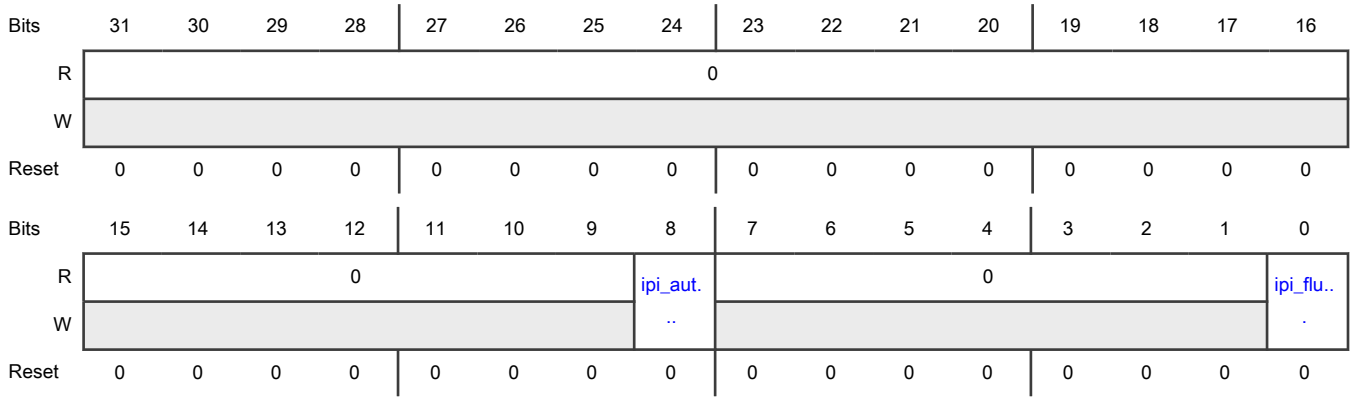
54.5.1.20 IPI Flush Memory (IPI_MEM_FLUSH)

Controls the flush of IPI memory.

Offset

Register	Offset
IPI_MEM_FLUSH	8Ch

Diagram



Fields

Field	Description
31-9 —	Reserved and read as zero.
8 ipi_auto_flush	Memory is automatically flushed at each Frame Start.
7-1 —	Reserved and read as zero.
0 ipi_flush	Flush IPI memory. This bit is auto clear.

54.5.1.21 IPI HSA (IPI_HSA_TIME)

Configures the video Horizontal Synchronism Active (HSA) time.

Offset

Register	Offset
IPI_HSA_TIME	90h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				ipi_hsa_time											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-12 —	Reserved and read as zero.
11-0 ipi_hsa_time	Configures the Horizontal Synchronism Active period in pixclk cycles.

54.5.1.22 IPI HBP (IPI_HBP_TIME)

Configures the video Horizontal Back Porch (HBP) time.

Offset

Register	Offset
IPI_HBP_TIME	94h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				ipi_hbp_time											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-12 —	Reserved and read as zero.
11-0 ipi_hbp_time	Configures the Horizontal Back Porch period in pixclk cycles.

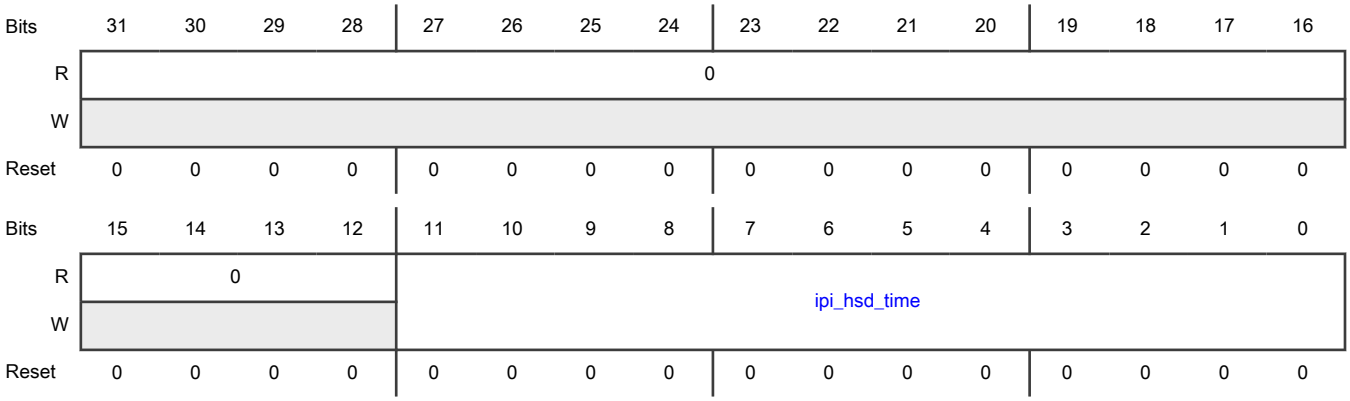
54.5.1.23 IPI HSD (IPI_HSD_TIME)

Configures the video Horizontal Sync Delay (HSD) time.

Offset

Register	Offset
IPI_HSD_TIME	98h

Diagram



Fields

Field	Description
31-12 —	Reserved and read as zero.
11-0 ipi_hsd_time	Configures the Horizontal Sync Porch delay period in pixclk cycles.

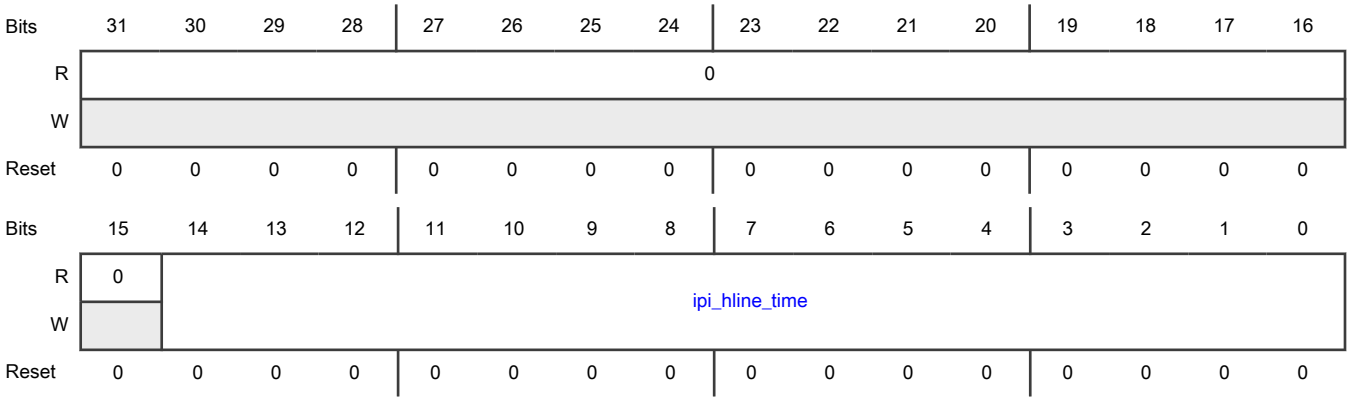
54.5.1.24 IPI HLINE (IPI_HLINE_TIME)

Configures the overall time for each video line.

Offset

Register	Offset
IPI_HLINE_TIME	9Ch

Diagram



Fields

Field	Description
31-15 —	Reserved and read as zero.
14-0 ipi_hline_time	Configures the size of the line time counted in pixclk cycles.

54.5.1.25 IPI Soft Reset (IPI_SOFTRSTN)

Controls the IPI logic reset state. When activated, the internal logic of the IPI goes into the reset state. Only the internal IPI logic is affected.

Offset

Register	Offset
IPI_SOFTRSTN	A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			0	0			0	0			0	0			0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			0	0			0	0			0	0			ipi_sof..
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31-29 —	Reserved and read as zero.
28 —	Reserved and read as zero.
27-25 —	Reserved and read as zero.
24 —	Reserved and read as zero.
23-21 —	Reserved and read as zero.
20 —	Reserved and read as zero.
19-17 —	Reserved and read as zero.
16 —	Reserved and read as zero.
15-13 —	Reserved and read as zero.
12 —	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
11-9 —	Reserved and read as zero.
8 —	Reserved and read as zero.
7-5 —	Reserved and read as zero.
4 —	Reserved and read as zero.
3-1 —	Reserved and read as zero.
0 ipi_softrstn	Resets IPI one. Active Low.

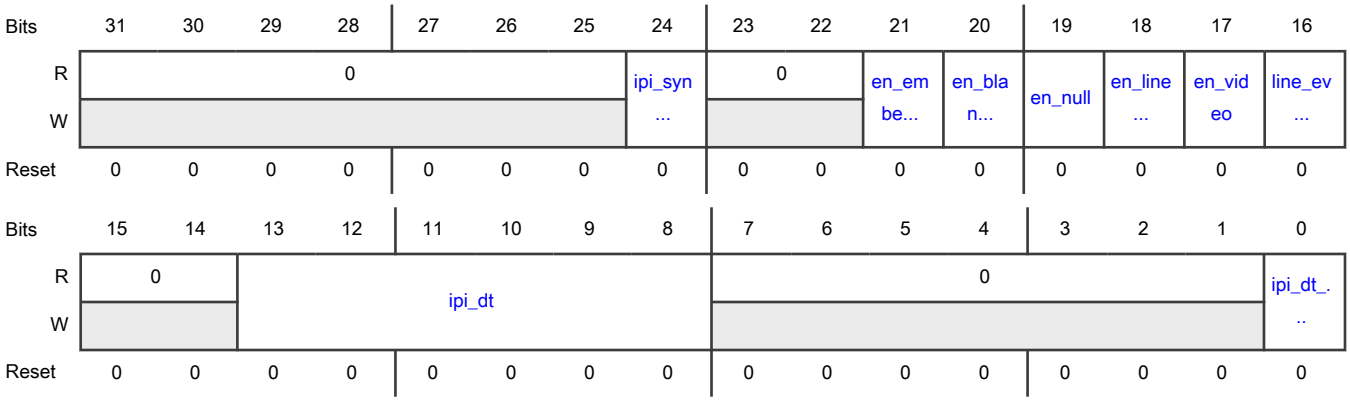
54.5.1.26 IPI Advanced Features (IPI_ADV_FEATURES)

Configures advanced features for IPI mode.

Offset

Register	Offset
IPI_ADV_FEATURES	ACh

Diagram



Fields

Field	Description
31-25 —	Reserved and read as zero.
24 ipi_sync_event_mode	For Camera Mode: 0 - Frame Start does not trigger any sync event. 1 - Legacy mode. Frame Start triggers a sync event.
23-22 —	Reserved and read as zero.
21 en_embedded	Allows the use of embedded packets for IPI synchronization events.
20 en_blanking	Allows the use of blanking packets for IPI synchronization events.
19 en_null	Allows the use of null packets for IPI synchronization events.
18 en_line_start	Allows the use of line start packets for IPI synchronization events.
17 en_video	Allows the use of video packets for IPI synchronization events.
16 line_event_selection	For Camera Mode, allows manual selection of the Packet for line delimiter as follows: 0 - Controller selects it automatically 1 - Select packets from list programmed in [17:21]
15-14 —	Reserved and read as zero.
13-8 ipi_dt	Datatype to overwrite.
7-1 —	Reserved and read as zero.
0 ipi_dt_overwrite	Ignore datatype of the header using the programmed datatype for decoding.

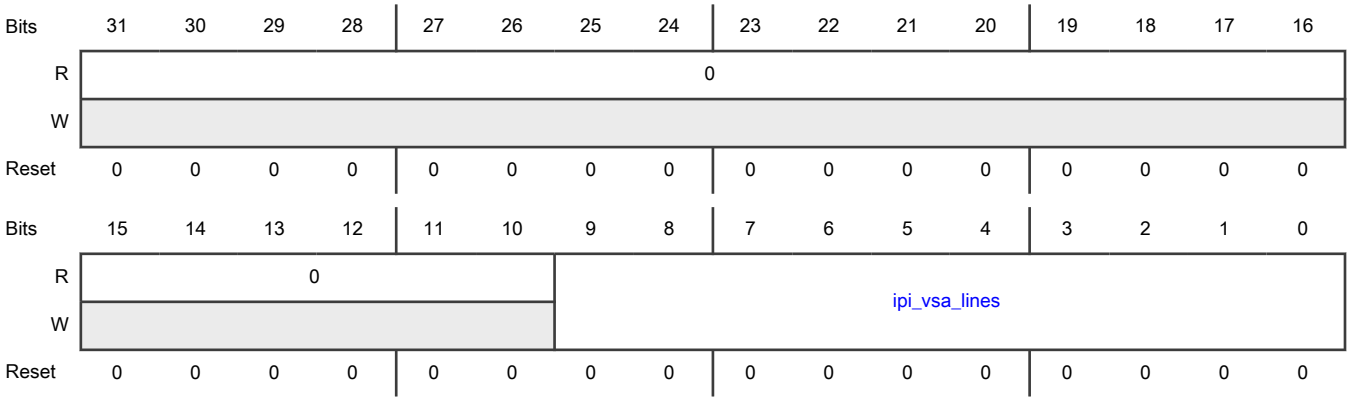
54.5.1.27 IPI VSA (IPI_VSA_LINES)

Configures the Vertical Synchronism Active (VSA) period.

Offset

Register	Offset
IPI_VSA_LINES	B0h

Diagram



Fields

Field	Description
31-10 —	Reserved and read as zero.
9-0 ipi_vsa_lines	Configures the Vertical Synchronism Active period measured in number of horizontal lines.

54.5.1.28 IPI VBP (IPI_VBP_LINES)

Configures the Vertical Back Porch (VBP) period.

Offset

Register	Offset
IPI_VBP_LINES	B4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						ipi_vbp_lines									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-10 —	Reserved and read as zero.
9-0 ipi_vbp_lines	Configures the Vertical Back Porch period measured in number of horizontal lines.

54.5.1.29 IPI VFP (IPI_VFP_LINES)

Configures the Vertical Front Porch (VFP) period.

Offset

Register	Offset
IPI_VFP_LINES	B8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						ipi_vfp_lines									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-10 —	Reserved and read as zero.
9-0 ipi_vfp_lines	Configures the Vertical Front Porch period measured in number of horizontal lines.

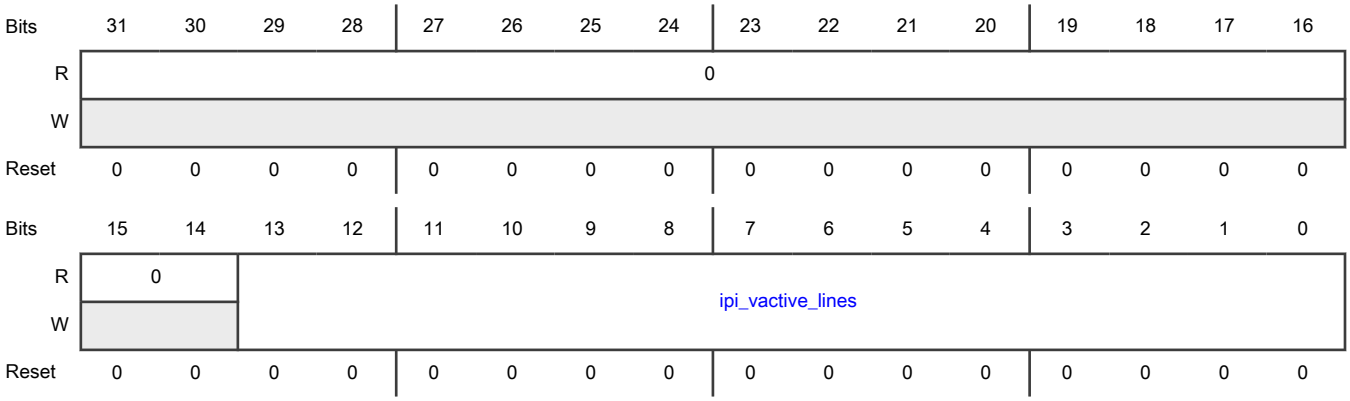
54.5.1.30 IPI VACTIVE (IPI_VACTIVE_LINES)

Configures the vertical resolution of the video.

Offset

Register	Offset
IPI_VACTIVE_LINES	BCh

Diagram



Fields

Field	Description
31-14 —	Reserved and read as zero.
13-0 ipi_vactive_lines	Configures the Vertical Active period measured in number of horizontal lines.

54.5.1.31 Fatal Interruption Caused by PHY (INT_ST_PHY_FATAL)

Groups the fatal interruptions caused by PHY Packet discarded.

Stores the source of the error.

Reading INT_ST_PHY_FATAL register does not clear the interrupt pin.

Offset

Register	Offset
INT_ST_PHY_FATAL	E0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							0	0	0	0	0	0	0	phy_err...	phy_err...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-19 —	Reserved and read as zero.
18 —	Reserved and read as zero.
17 —	Reserved and read as zero.
16 —	Reserved and read as zero.
15-9 —	Reserved and read as zero.
8 —	Reserved and read as zero.
7 —	Reserved and read as zero.
6 —	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
5 —	Reserved and read as zero.
4 —	Reserved and read as zero.
3 —	Reserved and read as zero.
2 —	Reserved and read as zero.
1 phy_errsotsynchs_1	Start of transmission error on data lane 1 (no synchronization achieved) After a read operation, the field's value clears to 0.
0 phy_errsotsynchs_0	Start of transmission error on data lane 0 (no synchronization achieved). After a read operation, the field's value clears to 0.

54.5.1.32 Mask for Fatal Interruption Caused by PHY (INT_MSK_PHY_FATAL)

Contains the interrupt mask for INT_ST_PHY_FATAL and controls which interrupt status bits trigger the interrupt pin. Bit at 1 - Enable the interrupt source. Bit at 0 - Interrupt source is masked.

Offset

Register	Offset
INT_MSK_PHY_FATAL	E4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							0	0	0	0	0	0	0	mask_ph...	mask_ph...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-19 —	Reserved and read as zero.
18 —	Reserved and read as zero.
17 —	Reserved and read as zero.
16 —	Reserved and read as zero.
15-9 —	Reserved and read as zero.
8 —	Reserved and read as zero.
7 —	Reserved and read as zero.
6 —	Reserved and read as zero.
5 —	Reserved and read as zero.
4 —	Reserved and read as zero.
3 —	Reserved and read as zero.
2 —	Reserved and read as zero.
1 mask_phy_errsotsynchs_1	Mask for phy_errsotsynchs_1
0 mask_phy_errsotsynchs_0	Mask for phy_errsotsynchs_0.

54.5.1.33 Force for Fatal Interruption Caused by PHY (INT_FORCE_PHY_FATAL)

Use for test purposes. Allows triggering INT_ST_PHY_FATAL interrupt events individually, without the need to activate the conditions that trigger the interrupt sources, since it may be extremely complex to generate stimuli for that purpose. This register is auto-clear.

Offset

Register	Offset
INT_FORCE_PHY_FATAL	E8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							0	0	0	0	0	0	0	force_p...	force_p...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-19 —	Reserved and read as zero.
18 —	Reserved and read as zero.
17 —	Reserved and read as zero.
16 —	Reserved and read as zero.
15-9 —	Reserved and read as zero.
8 —	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 —	Reserved and read as zero.
6 —	Reserved and read as zero.
5 —	Reserved and read as zero.
4 —	Reserved and read as zero.
3 —	Reserved and read as zero.
2 —	Reserved and read as zero.
1 force_phy_errsotsynchs_1	Force phy_errsotsynchs_1
0 force_phy_errsotsynchs_0	Force phy_errsotsynchs_0.

54.5.1.34 Fatal Interruption Caused During Packet Construction (INT_ST_PKT_FATAL)

Groups the fatal interruption related to Packet construction. Packet discarded.

Notifies which interruption bit has caused the interruption.

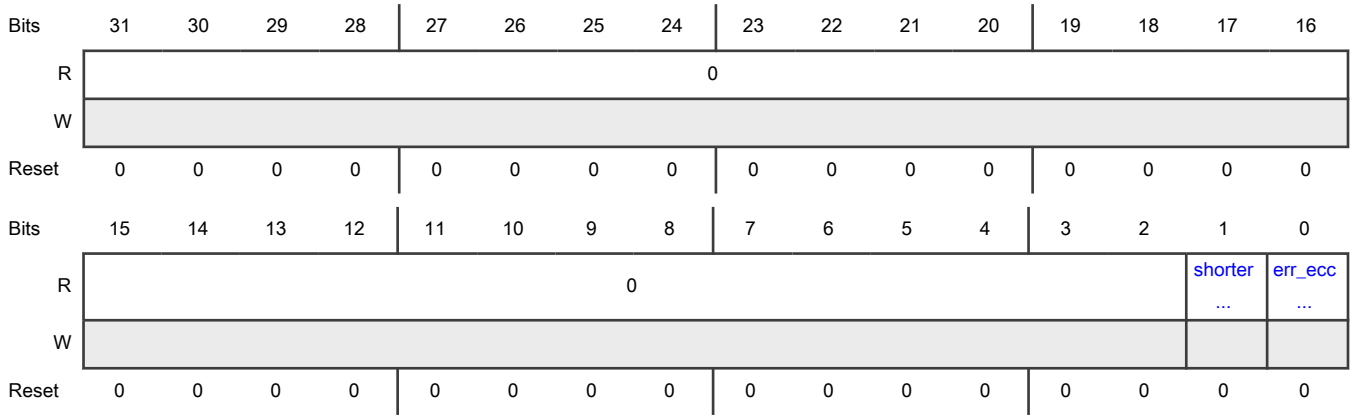
Stores the source of the error.

Reading INT_ST_PKT_FATAL register does not clear the interrupt pin.

Offset

Register	Offset
INT_ST_PKT_FATAL	F0h

Diagram



Fields

Field	Description
31-2 —	Reserved and read as zero.
1 shorter_payload	Reported greater WC than received, unrecoverable. After a read operation, the field's value clears to 0.
0 err_ecc_double	Header ECC contains at least 2 errors, unrecoverable. After a read operation, the field's value clears to 0.

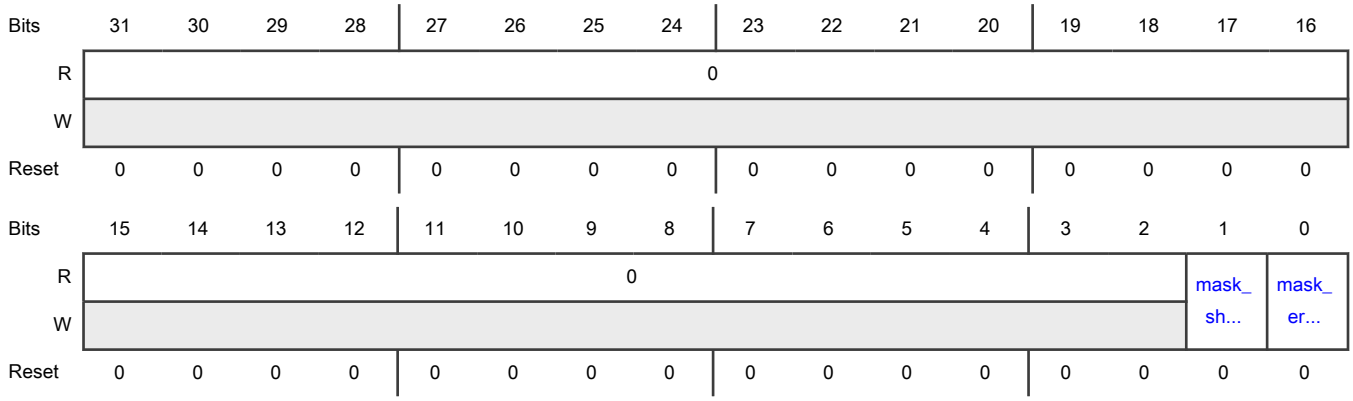
54.5.1.35 Mask for Fatal Interruption Caused During Packet Construction (INT_MSK_PKT_FATAL)

Controls which interrupt status bits trigger the interrupt pin. Interrupt mask for INT_ST_PKT_FATAL. Bit at 1 - Enable the interrupt source. Bit at 0 - Interrupt source is masked.

Offset

Register	Offset
INT_MSK_PKT_FATAL	F4h

Diagram



Fields

Field	Description
31-2 —	Reserved and read as zero.
1 mask_shorter_payload	Mask for shorter_payload.
0 mask_err_ecc_double	Mask for err_ecc_double.

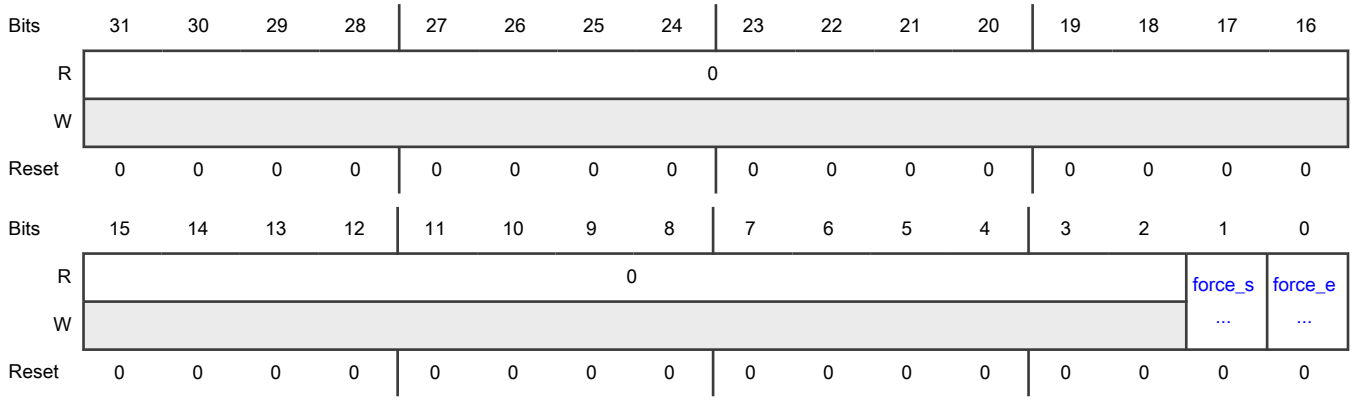
54.5.1.36 Force for Fatal Interruption Caused During Packet Construction (INT_FORCE_PKT_FATAL)

Use for test purposes. Allows triggering INT_ST_PKT_FATAL interrupt events individually, without the need to activate the conditions that trigger the interrupt sources, since it may be extremely complex to generate stimuli for that purpose. This register is auto-clear.

Offset

Register	Offset
INT_FORCE_PKT_FATAL	F8h

Diagram



Fields

Field	Description
31-2 —	Reserved and read as zero.
1 force_shorter_payload	Force shorter_payload.
0 force_err_ecc_double	Force err_ecc_double.

54.5.1.37 Interruption Caused by PHY (INT_ST_PHY)

Groups and notifies which interruption bits caused the interruption.

Interruption caused by PHY.

Stores the source of the error.

Reading INT_ST_PHY register does not clear the interrupt pin.

Offset

Register	Offset
INT_ST_PHY	110h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0	0	0	0	0	0	phy_err...	phy_err...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0	0	0	0	0	phy_err...	phy_err...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved and read as zero.
23 —	Reserved and read as zero.
22 —	Reserved and read as zero.
21 —	Reserved and read as zero.
20 —	Reserved and read as zero.
19 —	Reserved and read as zero.
18 —	Reserved and read as zero.
17 phy_erresc_1	Escape Entry Error on data lane 1 After a read operation, the field's value clears to 0.
16 phy_erresc_0	Escape Entry Error on data lane 0. After a read operation, the field's value clears to 0.
15-8	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7 —	Reserved and read as zero.
6 —	Reserved and read as zero.
5 —	Reserved and read as zero.
4 —	Reserved and read as zero.
3 —	Reserved and read as zero.
2 —	Reserved and read as zero.
1 phy_errsoths_1	Start of transmission error on data lane 1 (synchronization can still be achieved) After a read operation, the field's value clears to 0.
0 phy_errsoths_0	Start of transmission error on data lane 0 (synchronization can still be achieved). After a read operation, the field's value clears to 0.

54.5.1.38 Mask for Interruption Caused by PHY (INT_MSK_PHY)

Controls which interrupt status bits trigger the interrupt pin. Interrupt mask for INT_ST_PHY. Bit at 1 - Enable the interrupt source. Bit at 0 - Interrupt source is masked.

Offset

Register	Offset
INT_MSK_PHY	114h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0	0	0	0	0	0	mask_ph...	mask_ph...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0	0	0	0	0	mask_ph...	mask_ph...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved and read as zero.
23 —	Reserved and read as zero.
22 —	Reserved and read as zero.
21 —	Reserved and read as zero.
20 —	Reserved and read as zero.
19 —	Reserved and read as zero.
18 —	Reserved and read as zero.
17 mask_phy_erresc_1	Mask for phy_erresc_1
16 mask_phy_erresc_0	Mask for phy_erresc_0.
15-8	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7 —	Reserved and read as zero.
6 —	Reserved and read as zero.
5 —	Reserved and read as zero.
4 —	Reserved and read as zero.
3 —	Reserved and read as zero.
2 —	Reserved and read as zero.
1 mask_phy_errsoths_1	Mask for phy_errsoths_1
0 mask_phy_errsoths_0	Mask for phy_errsoths_0.

54.5.1.39 Force for Interruption Caused by PHY (INT_FORCE_PHY)

Use for test purposes. Allows triggering INT_ST_PHY interrupt events individually. There is no need to activate the conditions that trigger the interrupt sources, because it may be extremely complex to generate stimuli for that purpose. This register is auto-clear.

Offset

Register	Offset
INT_FORCE_PHY	118h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0	0	0	0	0	0	force_p...	force_p...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0	0	0	0	0	force_p...	force_p...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved and read as zero.
23 —	Reserved and read as zero.
22 —	Reserved and read as zero.
21 —	Reserved and read as zero.
20 —	Reserved and read as zero.
19 —	Reserved and read as zero.
18 —	Reserved and read as zero.
17 force_phy_erresc_1	Force phy_erresc_1
16 force_phy_erresc_0	Force phy_erresc_0
15-8	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7 —	Reserved and read as zero.
6 —	Reserved and read as zero.
5 —	Reserved and read as zero.
4 —	Reserved and read as zero.
3 —	Reserved and read as zero.
2 —	Reserved and read as zero.
1 force_phy_errsoths_1	Force phy_errsoths_1
0 force_phy_errsoths_0	Force phy_errsoths_0.

54.5.1.40 Fatal Interruption Caused by IPI Interface (INT_ST_IPI_FATAL)

Groups and notifies which interruption bits caused the interruption.

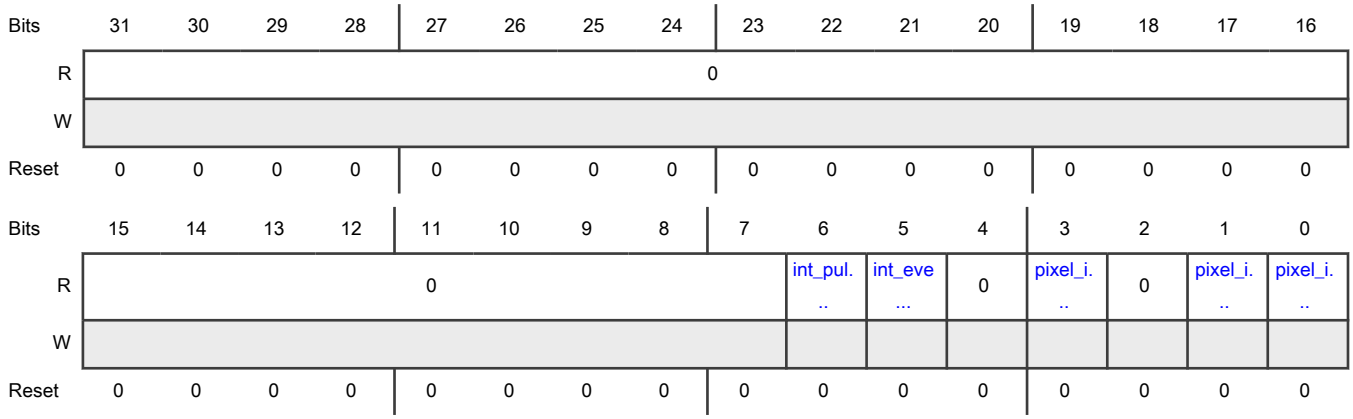
Stores the source of the error.

Reading INT_ST_IPI_FATAL register does not clear the interrupt pin.

Offset

Register	Offset
INT_ST_IPI_FATAL	140h

Diagram



Fields

Field	Description
31-7 —	Reserved and read as zero.
6 int_pulse_delay_overflow	Reporting internal fifo overflow on pulse delay block. After a read operation, the field's value clears to 0.
5 int_event_fifo_overflow	Reporting internal fifo overflow. After a read operation, the field's value clears to 0.
4 —	Reserved and read as zero.
3 pixel_if_fifo_nempty_fs	Pixel interface FIFO is not empty Controller timing mode: The FIFO of pixel interface is not empty at the start of a new frame. If this is expected this interrupt should be masked. Camera timing mode: The FIFO of pixel interface is not empty at the start of a new frame. There are some scenarios where synchronization events of new frame can be lost on IPI interface. After a read operation, the field's value clears to 0.
2 —	Reserved and read as zero.
1 pixel_if_fifo_overflow	The FIFO of pixel interface has lost information because some data arrived and FIFO is already full. After a read operation, the field's value clears to 0.
0	The FIFO has become empty before the expected number of pixels (calculated from the packet header) could be extracted to the pixel interface.

Table continues on the next page...

Table continued from the previous page...

Field	Description
pixel_if_fifo_underflow	After a read operation, the field's value clears to 0.

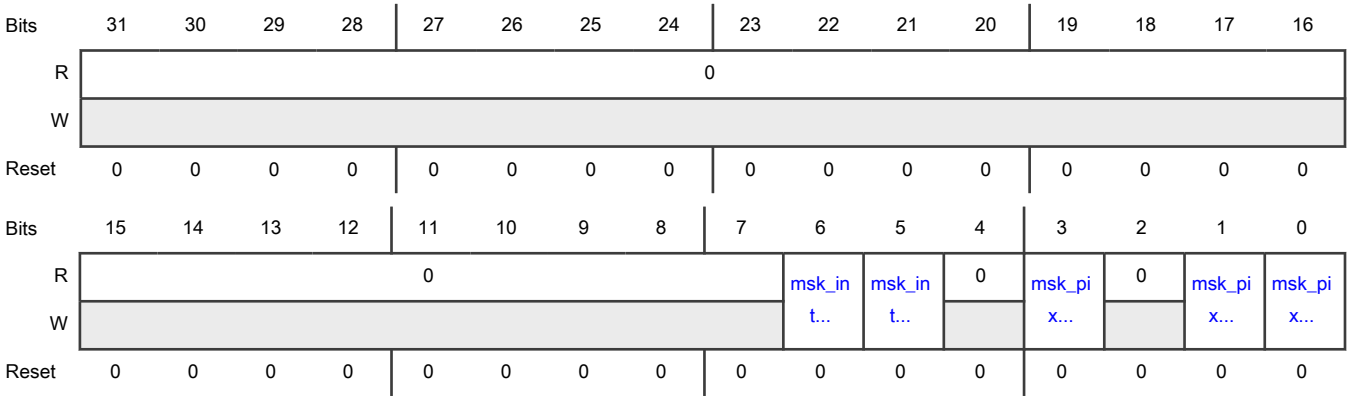
54.5.1.41 Mask for Fatal Interruption Caused by IPI Interface (INT_MSK_IPI_FATAL)

Controls which interrupt status bits trigger the interrupt pin. Interrupt mask for INT_ST_IPI_FATAL. Bit at 1 - Enable the interrupt source. Bit at 0 - Interrupt source is masked.

Offset

Register	Offset
INT_MSK_IPI_FATAL	144h

Diagram



Fields

Field	Description
31-7 —	Reserved and read as zero.
6 msk_int_pulse_delay_overflow	Mask int_pulse_delay_overflow.
5 msk_int_event_fifo_overflow	Mask int_event_fifo_overflow.
4	Reserved and read as zero.

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
3 msk_pixel_if_fifo_nempty_fs	Mask pixel_if_fifo_nempty_fs.
2 —	Reserved and read as zero.
1 msk_pixel_if_fifo_overflow	Mask for pixel_if_fifo_overflow.
0 msk_pixel_if_fifo_underflow	Mask for pixel_if_fifo_underflow.

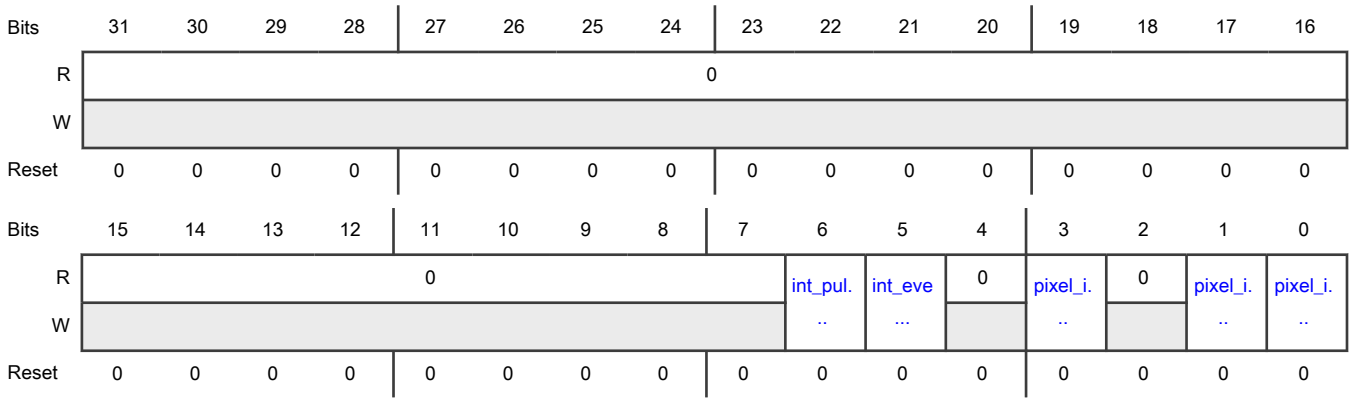
54.5.1.42 Force for Fatal Interruption Caused by IPI Interface (INT_FORCE_IPI_FATAL)

Use for test purposes. Allows triggering INT_ST_IPI_FATAL interrupt events individually. There is no need to activate the conditions that trigger the interrupt sources, because it may be extremely complex to generate stimuli for that purpose. This register is auto-clear.

Offset

Register	Offset
INT_FORCE_IPI_FATAL	148h

Diagram



Fields

Field	Description
31-7 —	Reserved and read as zero.
6 int_pulse_delay_overflow	Force int_pulse_delay_overflow.
5 int_event_fifo_overflow	Force int_event_fifo_overflow.
4 —	Reserved and read as zero.
3 pixel_if_fifo_nempty_fs	Force pixel_if_fifo_nempty_fs.
2 —	Reserved and read as zero.
1 pixel_if_fifo_overflow	Force for pixel_if_fifo_overflow.
0 pixel_if_fifo_underflow	Force for pixel_if_fifo_underflow.

54.5.1.43 Fatal Interruption Caused by Frame Boundaries (INT_ST_BNDRY_FRAME_FATAL)

Groups and notifies which interruption bits caused the interruption.

Fatal interruption related to matching Frame Start with Frame End for a specific virtual channel. Packet discarded.

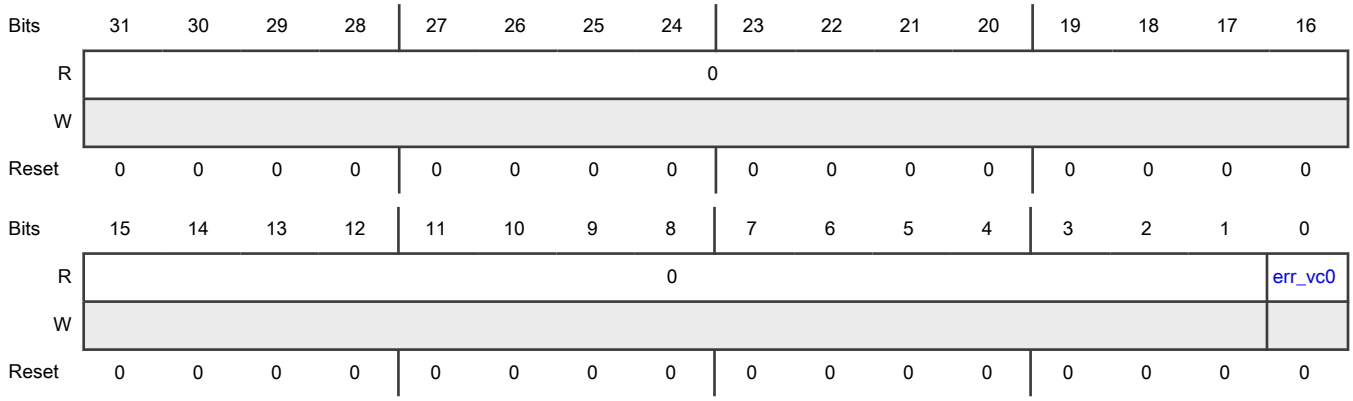
Stores the source of the error.

Reading INT_ST_BNDRY_FRAME_FATAL register does not clear the interrupt pin.

Offset

Register	Offset
INT_ST_BNDRY_FRAME_FATAL	280h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_vc0	Error matching Frame Start with Frame End for virtual channel 0. After a read operation, the field's value clears to 0.

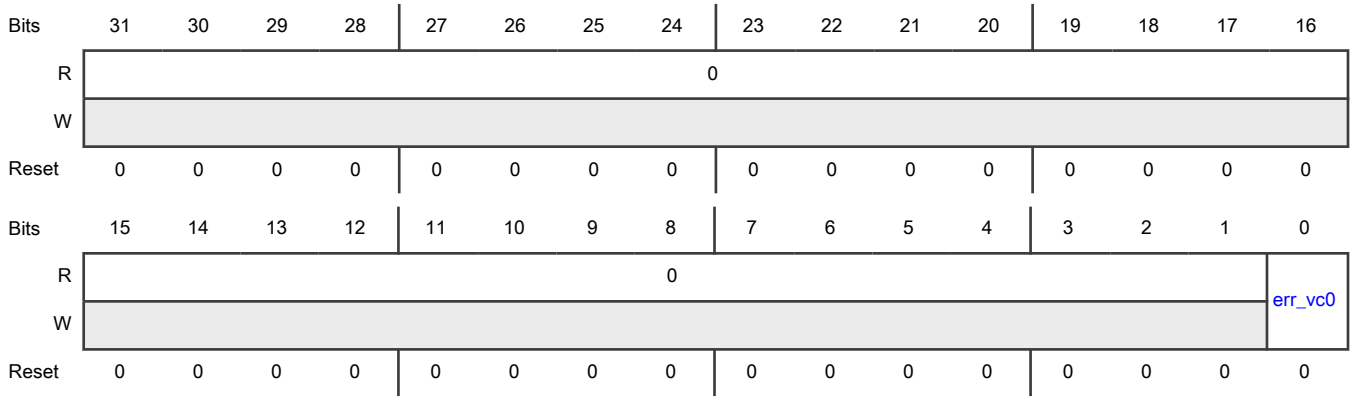
54.5.1.44 Mask for Fatal Interruption Caused by Frame Boundaries (INT_MSK_BNDRY_FRAME_FATAL)

Interrupt mask for INT_ST_BNDRY_FRAME_FATAL controls which interrupt status bits trigger the interrupt pin. Bit at 1 - Enable the interrupt source. Bit at 0 - Interrupt source is masked.

Offset

Register	Offset
INT_MSK_BNDRY_FRAME_FATAL	284h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_vc0	Mask for err_f_bndry_match_vc0.

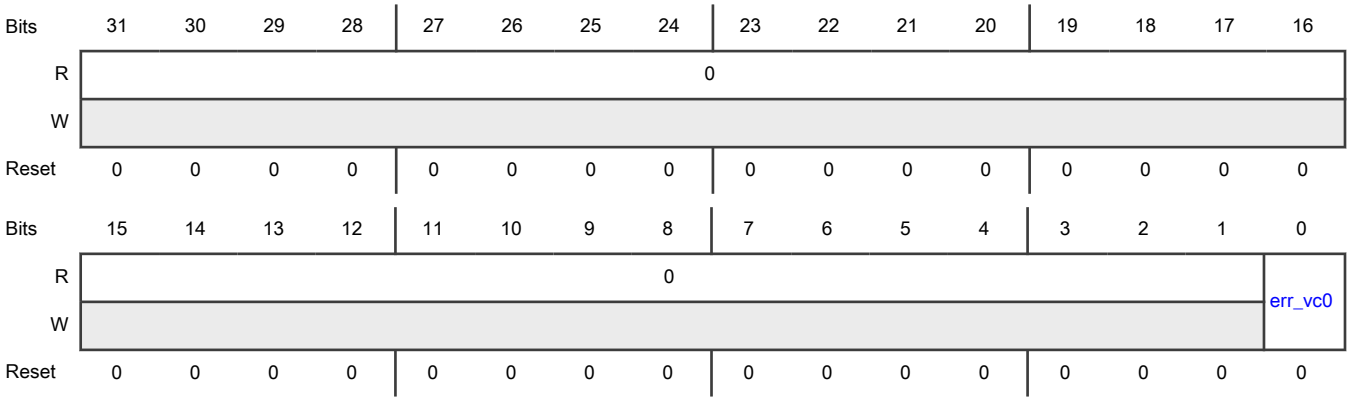
54.5.1.45 Force for Fatal Interruption Caused by Frame Boundaries (INT_FORCE_BNDRY_FRAME_FATAL)

Use for test purposes. Allows triggering INT_ST_BNDRY_FRAME_FATAL interrupt events individually, without the need to activate the conditions that trigger the interrupt sources, since it may be extremely complex to generate stimuli for that purpose. This register is auto-clear.

Offset

Register	Offset
INT_FORCE_BNDRY_FRAME_FATAL	288h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_vc0	Force err_f_bndry_match_vc0.

54.5.1.46 Fatal Interruption Caused by Frame Sequence (INT_ST_SEQ_FRAME_FATAL)

Groups and notifies which interruption bits caused the interruption.

Fatal interruption related to incorrect Frame sequence for a specific virtual channel. Packet discarded.

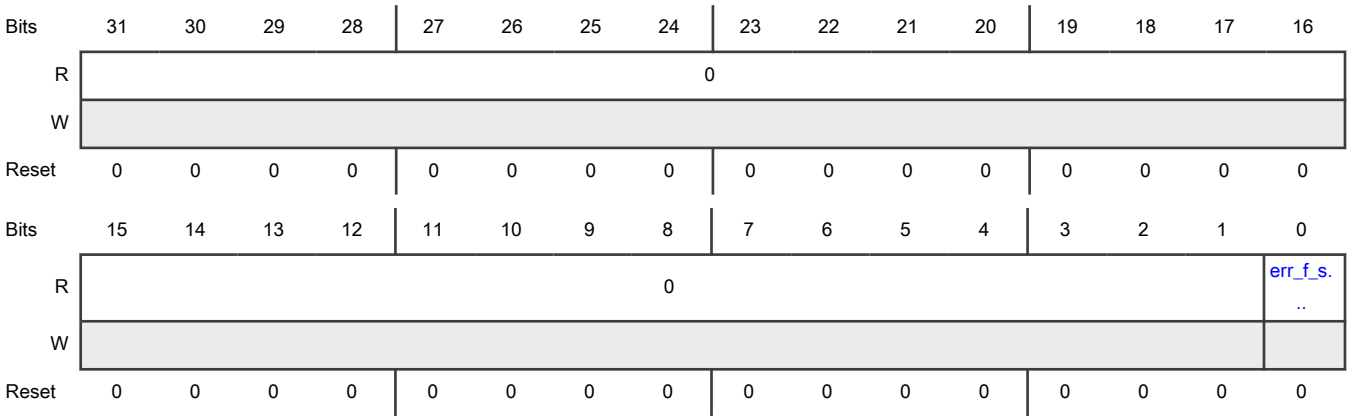
Stores the source of the error.

Reading INT_ST_SEQ_FRAME_FATAL register does not clear the interrupt pin.

Offset

Register	Offset
INT_ST_SEQ_FRAME_FATAL	290h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_f_seq_vc0	Incorrect Frame sequence detected in virtual channel 0. After a read operation, the field's value clears to 0.

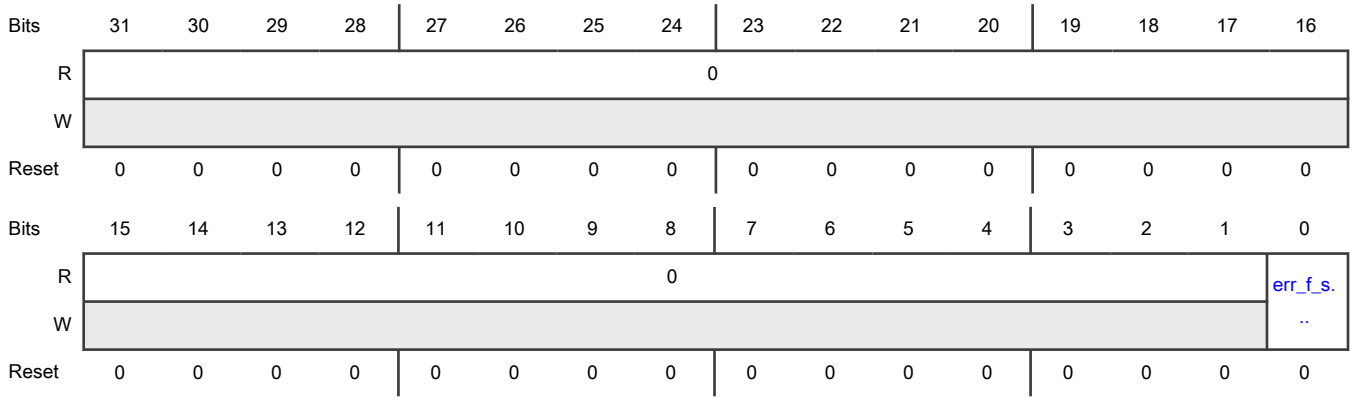
54.5.1.47 Mask for Fatal Interruption Caused by Frame Sequence (INT_MSK_SEQ_FRAME_FATAL)

Controls which interrupt status bits trigger the interrupt pin. Interrupt mask for INT_ST_SEQ_FRAME_FATAL. Bit at 1 - Enable the interrupt source. Bit at 0 - Interrupt source is masked.

Offset

Register	Offset
INT_MSK_SEQ_FRAME_FATAL	294h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_f_seq_vc0	Mask for err_f_seq_vc0.

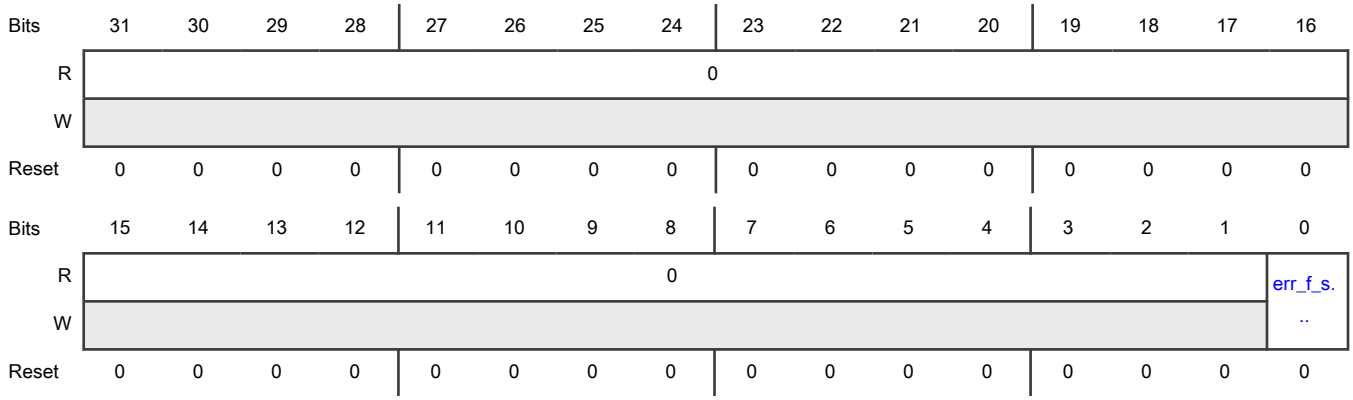
54.5.1.48 Force for Fatal Interruption Caused by Frame Sequence (INT_FORCE_SEQ_FRAME_FATAL)

Use for test purposes. Allows triggering INT_ST_SEQ_FRAME_FATAL interrupt events individually. There is no need to activate the conditions that trigger the interrupt sources, because it may be extremely complex to generate stimuli for that purpose. This register is auto-clear.

Offset

Register	Offset
INT_FORCE_SEQ_FRAME_FATAL	298h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_f_seq_vc0	Force err_f_seq_vc0.

54.5.1.49 Fatal Interruption Caused by Frame CRC (INT_ST_CRC_FRAME_FATAL)

Groups and notifies which interruption bits caused the interruption.

Fatal interruption related to Frames with at least one CRC error for a specific virtual channel. Packet discarded.

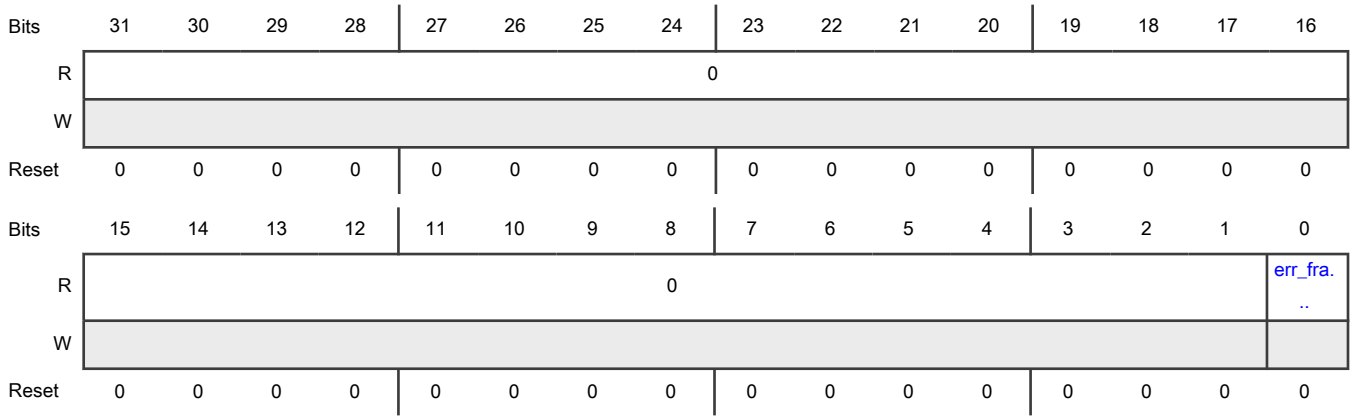
Stores the source of the error.

Reading INT_ST_CRC_FRAME_FATAL register does not clear the interrupt pin.

Offset

Register	Offset
INT_ST_CRC_FRAME_FATAL	2A0h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0	Last received Frame in virtual channel 0, had at least one CRC error.

Table continues on the next page...

Table continued from the previous page...

Field	Description
err_frame_data_vc0	After a read operation, the field's value clears to 0.

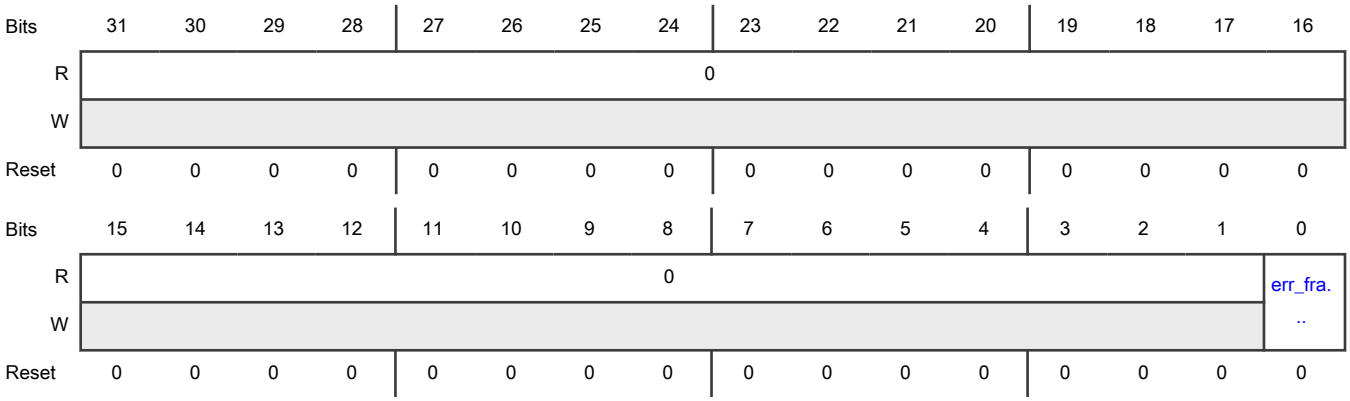
54.5.1.50 Mask for Fatal Interruption Caused by Frame CRC (INT_MSK_CRC_FRAME_FATAL)

Controls which interrupt status bits trigger the interrupt pin. Interrupt mask for INT_ST_CRC_FRAME_FATAL. Bit at 1 - Enable the interrupt source. Bit at 0 - Interrupt source is masked.

Offset

Register	Offset
INT_MSK_CRC_FRAME_FATAL	2A4h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_frame_data_vc0	Mask for err_frame_data_vc0.

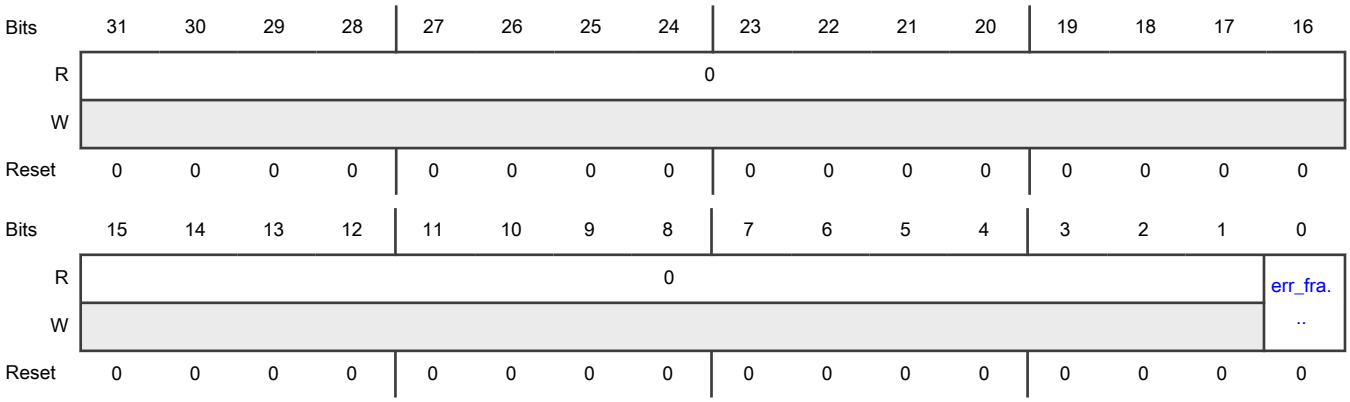
54.5.1.51 Force for Fatal Interruption Caused by Frame CRC (INT_FORCE_CRC_FRAME_FATAL)

Use for test purposes. Allows triggering INT_ST_CRC_FRAME_FATAL interrupt events individually. There is no need to activate the conditions that trigger the interrupt sources, because it may be extremely complex to generate stimuli for that purpose. This register is auto-clear.

Offset

Register	Offset
INT_FORCE_CRC_FRAME_FATAL	2A8h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_frame_data_vc0	Force err_frame_data_vc0.

54.5.1.52 Fatal Interruption Caused by Payload CRC (INT_ST_PLD_CRC_FATAL)

Groups and notifies which interruption bits caused the interruption.

Fatal interruption related to Payload Checksum error. Packet discarded.

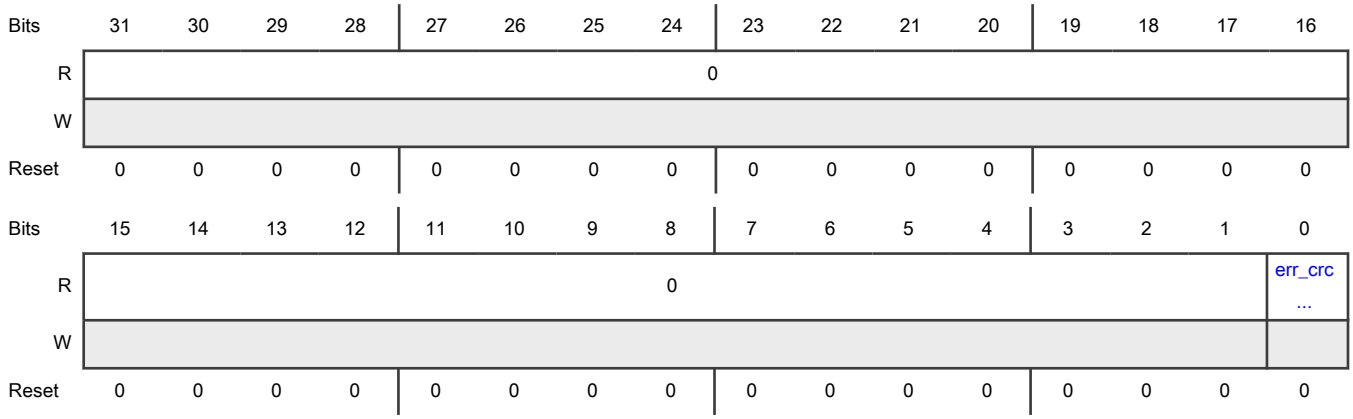
Stores the source of the error.

Reading INT_ST_PLD_CRC_FATAL register does not clear the interrupt pin.

Offset

Register	Offset
INT_ST_PLD_CRC_FATAL	2B0h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_crc_vc0	Payload Checksum error detected on virtual channel 0. After a read operation, the field's value clears to 0.

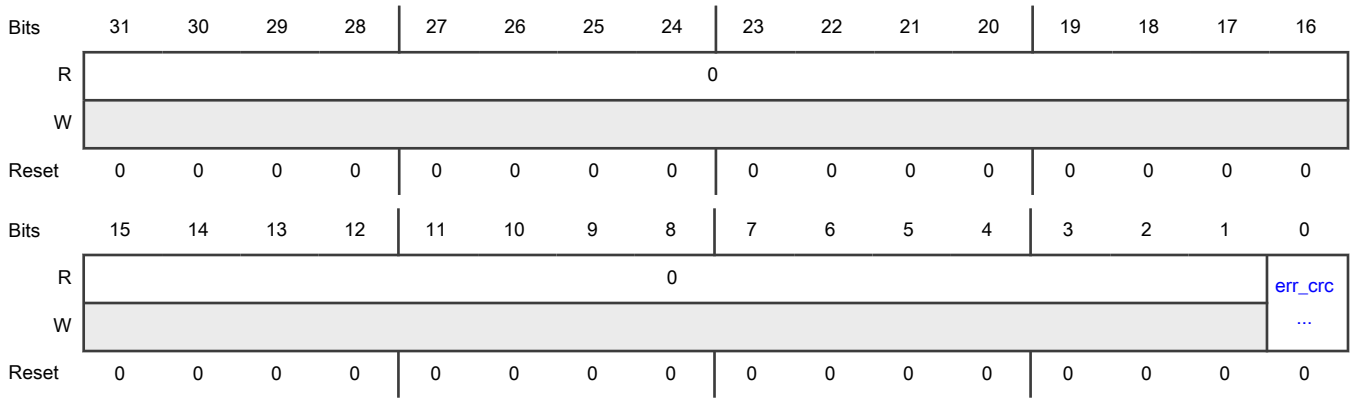
54.5.1.53 Mask for Fatal Interruption Caused by Payload CRC (INT_MSK_PLD_CRC_FATAL)

Controls which interrupt status bits trigger the interrupt pin. Interrupt mask for INT_ST_PLD_CRC_FATAL. Bit at 1 - Enable the interrupt source. Bit at 0 - Interrupt source is masked.

Offset

Register	Offset
INT_MSK_PLD_CRC_FATAL	2B4h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_crc_vc0	Mask for err_crc_vc0.

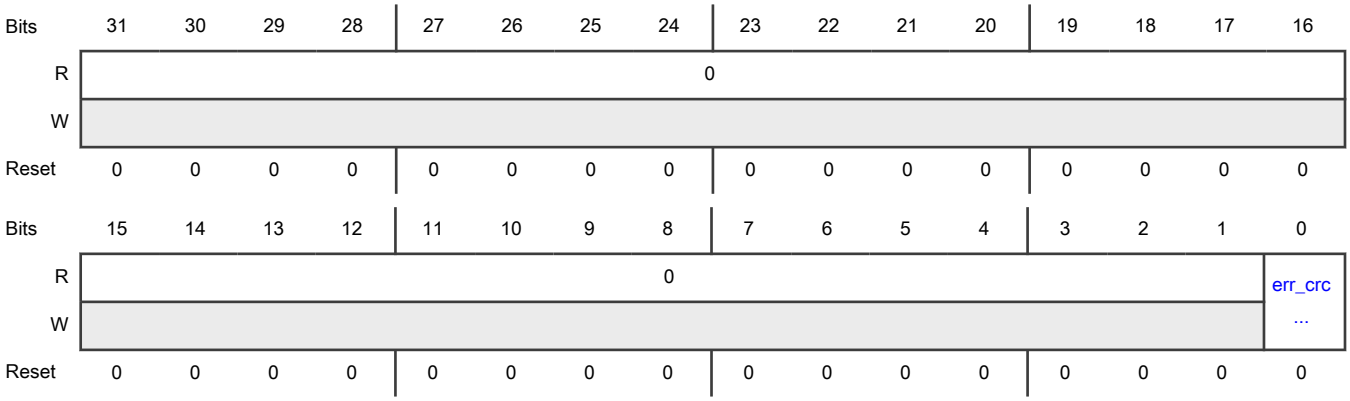
54.5.1.54 Force for Fatal Interruption Caused by Payload CRC (INT_FORCE_PLD_CRC_FATAL)

Use for test purposes. Allows triggering INT_ST_PLD_CRC_FATAL interrupt events individually. There is no need to activate the conditions that trigger the interrupt sources, because it may be extremely complex to generate stimuli for that purpose. This register is auto-clear.

Offset

Register	Offset
INT_FORCE_PLD_CRC_FATAL	2B8h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_crc_vc0	Force err_crc_vc0.

54.5.1.55 Interruption Caused by Data Type (INT_ST_DATA_ID)

Groups and notifies which interruption bits caused the interruption.

Interruption related to unrecognized or unimplemented data type detected in a specific virtual channel. Packet discarded.

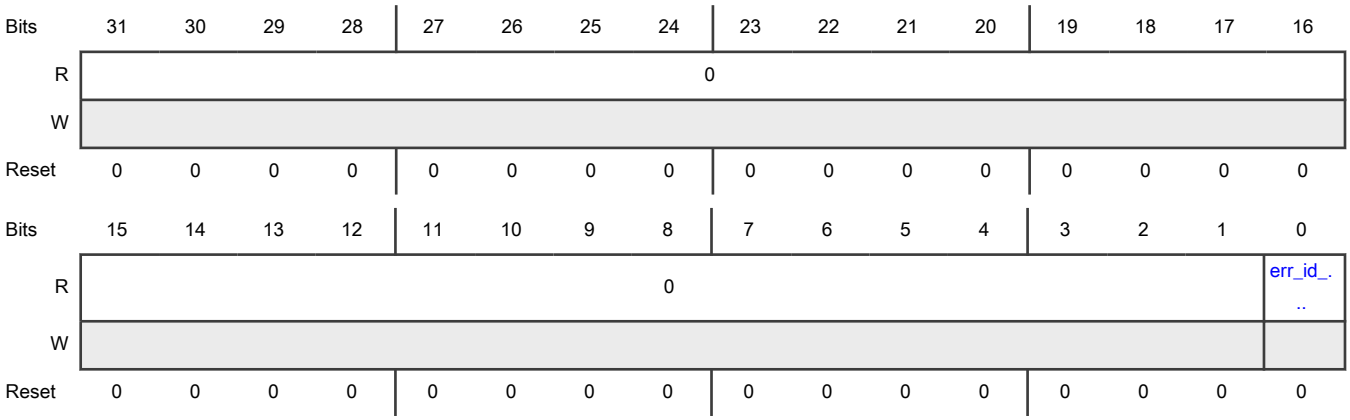
Stores the source of the error.

Reading INT_ST_DATA_ID register does not clear the interrupt pin.

Offset

Register	Offset
INT_ST_DATA_ID	2C0h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_id_vc0	Unrecognized or unimplemented data type detected in virtual channel 0. After a read operation, the field's value clears to 0.

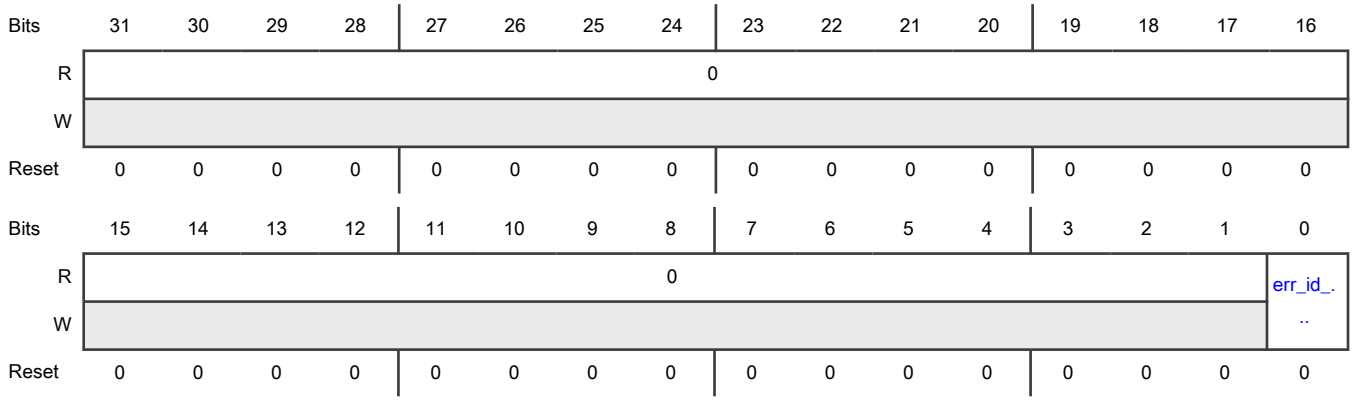
54.5.1.56 Mask for Interruption Caused by Data Type (INT_MSK_DATA_ID)

Controls which interrupt status bits trigger the interrupt pin. Interrupt mask for INT_ST_DATA_ID. Bit at 1 - Enable the interrupt source. Bit at 0 - Interrupt source is masked.

Offset

Register	Offset
INT_MSK_DATA_ID	2C4h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_id_vc0	Mask for err_id_vc0.

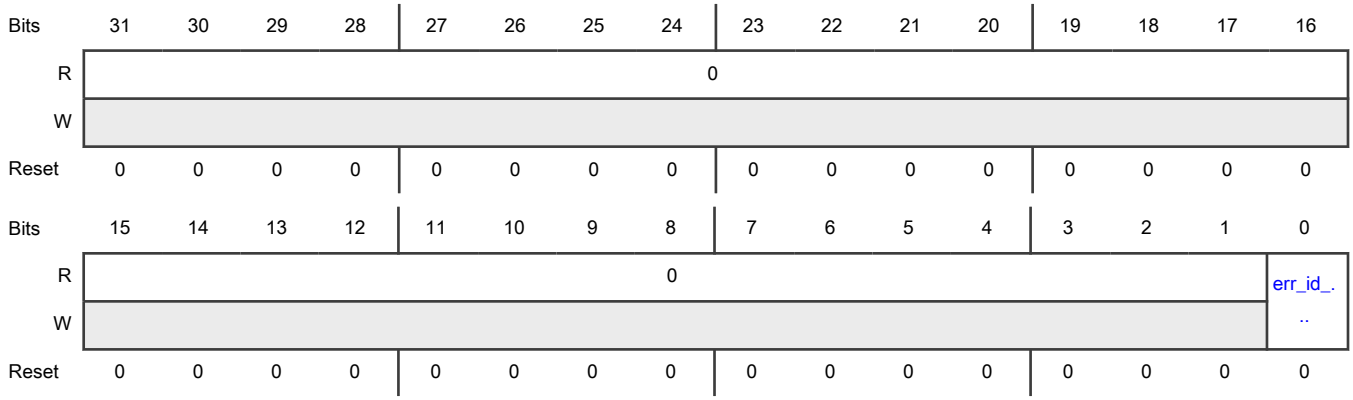
54.5.1.57 Force for Interruption Caused by Data Type (INT_FORCE_DATA_ID)

Use for test purposes. Allows triggering INT_ST_DATA_ID interrupt events individually. There is no need to activate the conditions that trigger the interrupt sources, because it may be extremely complex to generate stimuli for that purpose. This register is auto-clear.

Offset

Register	Offset
INT_FORCE_DATA_ID	2C8h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_id_vc0	Force err_id_vc0.

54.5.1.58 Interruption Caused by Header Single Bit Errors (INT_ST_ECC_CORRECTED)

Groups and notifies which interruption bits caused the interruption.

Interruption related to header error detected and corrected for a specific virtual channel.

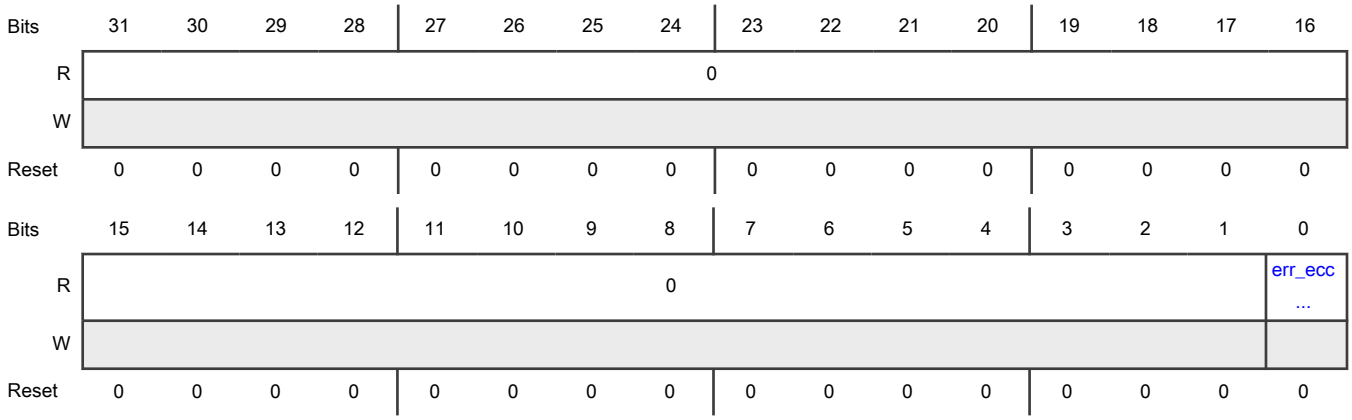
Stores the source of the error.

Reading INT_ST_ECC_CORRECTED register does not clear the interrupt pin.

Offset

Register	Offset
INT_ST_ECC_CORRECTED	2D0h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0	Header error detected and corrected on virtual channel 0.

Table continues on the next page...

Table continued from the previous page...

Field	Description
err_ecc_corrected_vc0	After a read operation, the field's value clears to 0.

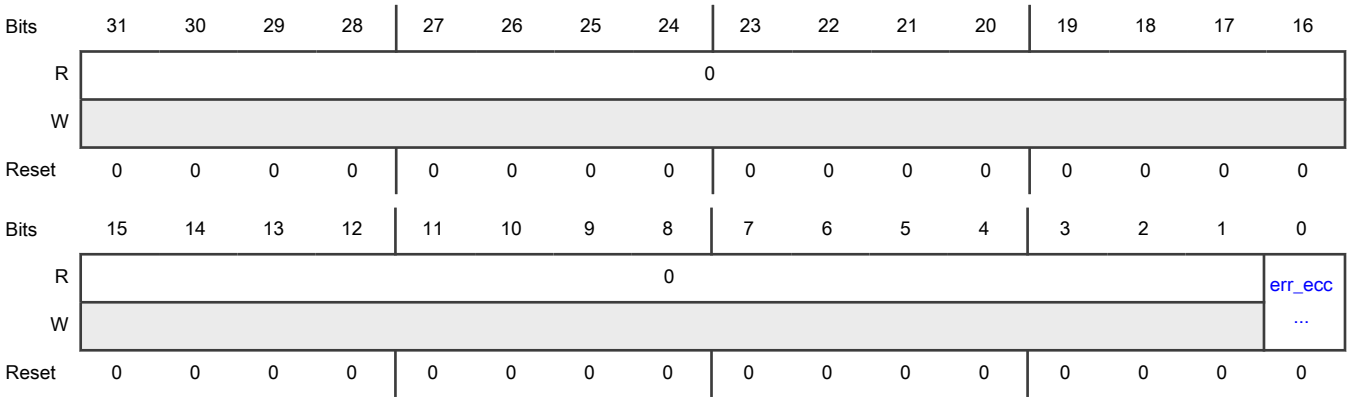
54.5.1.59 Mask for Interruption Caused by Header Single Bit Errors (INT_MSK_ECC_CORRECTED)

Controls which interrupt status bits trigger the interrupt pin. Interrupt mask for INT_ST_ECC_CORRECTED. Bit at 1 - Enable the interrupt source. Bit at 0 - Interrupt source is masked.

Offset

Register	Offset
INT_MSK_ECC_CORRECTED	2D4h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_ecc_corrected_vc0	Mask for err_ecc_corrected_vc0.

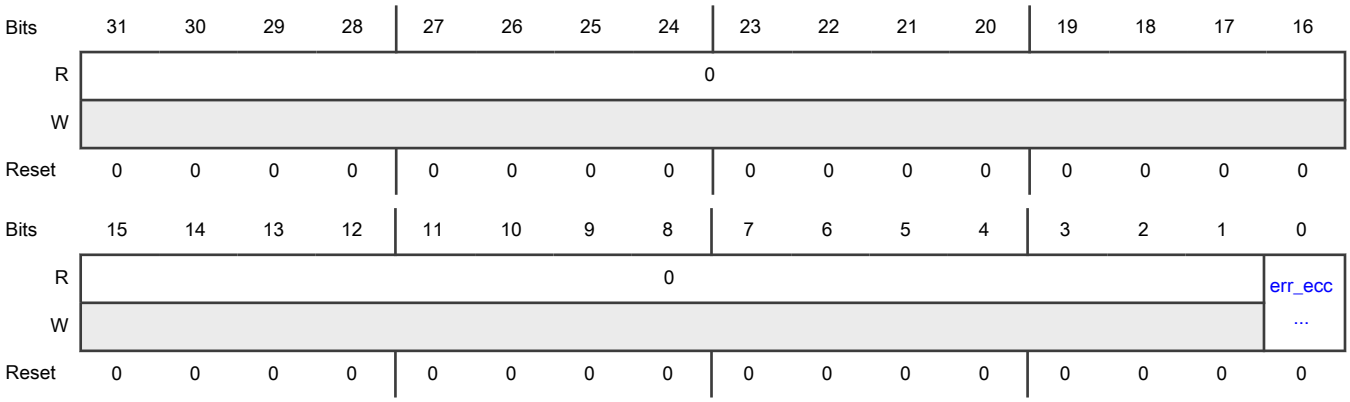
54.5.1.60 Force for Interruption Caused by Header Single Bit Errors (INT_FORCE_ECC_CORRECTED)

Use for test purposes. Allows triggering INT_ST_ECC_CORRECTED interrupt events individually. There is no need to activate the conditions that trigger the interrupt sources, because it may be extremely complex to generate stimuli for that purpose. This register is auto-clear.

Offset

Register	Offset
INT_FORCE_ECC_CORRECTED	2D8h

Diagram



Fields

Field	Description
31-1 —	Reserved and read as zero.
0 err_ecc_corrected_vc0	Force err_ecc_corrected_vc0.

54.5.1.61 Data De-Scrambling (SCRAMBLING)

Configures the De-scrambler block.

Offset

Register	Offset
SCRAMBLING	300h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															scramb l...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-1 —	Reserved and read as zero.
0 scramble_enable	Enables data de-scrambling on the controller side.

54.5.1.62 De-scrambler Seed for Lane 1 (SCRAMBLING_SEED1)

Configures the seed used by De-scrambler block for lane 1.

Offset

Register	Offset
SCRAMBLING_SEED1	304h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	scramble_seed_lane1															
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

Fields

Field	Description
31-16 —	Reserved and read as zero.
15-0 scramble_seed_lane1	Seed used by De-scrambler block for lane 1

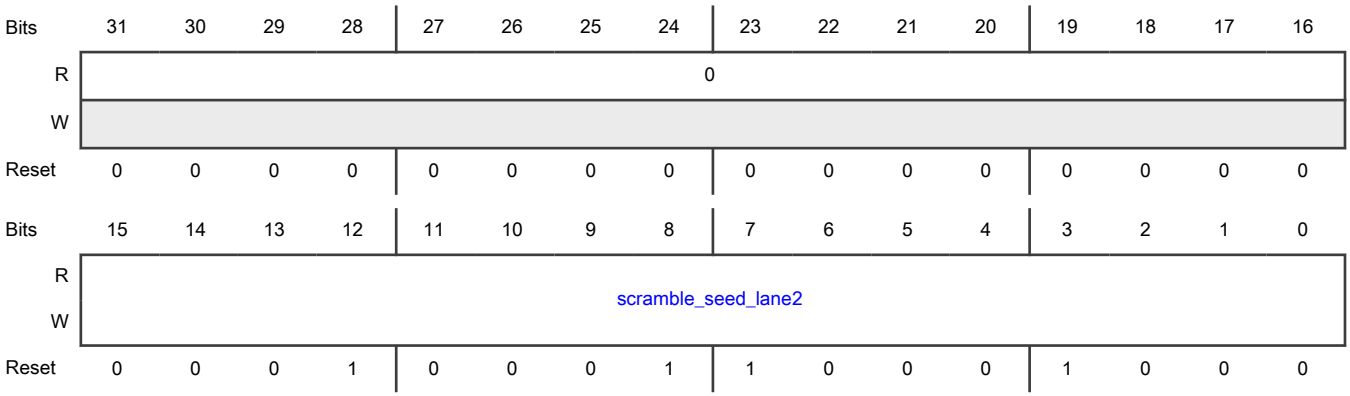
54.5.1.63 De-scrambler Seed for Lane 2 (SCRAMBLING_SEED2)

Configures the seed used by De-scrambler block for lane 2.

Offset

Register	Offset
SCRAMBLING_SEED2	308h

Diagram



Fields

Field	Description
31-16 —	Reserved and read as zero.
15-0 scramble_seed_lane2	Seed used by De-scrambler block for lane 2

Chapter 55
MIPI-CSI2 DPHY

55.1 Overview

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

The MIPI 2-lane Rx D-PHY module implements the physical layer for the MIPI D-PHY interface. It is targeted for high-speed digital data reception between a host processor and camera or display interfaces. The module includes two data lanes and one clock lane.

55.1.1 Block diagram

The block diagram of the MIPI Rx D-PHY is given below.

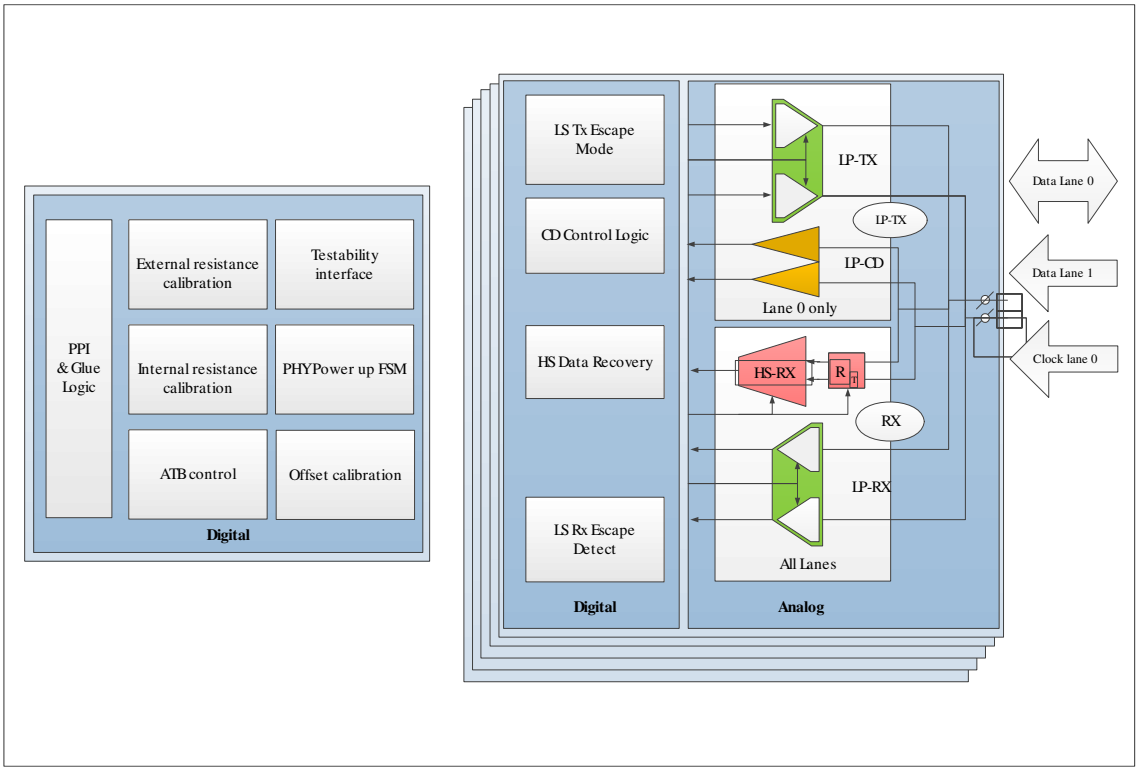


Figure 321. MIPI Rx D-PHY block diagram

- HS receiver for each lane to implement high-speed Rx functionalities
- LP receiver for each lane to implement low-power Rx functionalities
- LP driver only on data lane 0
- Low power contention detection
- Logic related to analog to digital interface, high-speed data path, data alignment, and PPI logic
- Logic responsible for PHY power up, resistor termination calibration, and tester interface

55.1.2 Features

The MIPI Rx D-PHY includes the following features:

- Compliant to MIPI D-PHY interface specification, revision 1.2
- Lane operation ranging from 80 Mbps to 1.5 Gbps in forward direction
- Aggregate throughput up to 3 Gbps with two data lanes
- Maximum LP data rate supported of 10 Mbps
- PHY-Protocol Interface (PPI) for clock and data lanes
- Low-power Escape modes and Ultra Low Power state
- HS RX Equalizer up to 3dB
- Optional usage of external reference resistor (200 ohm)
- HS RX automatic offset cancellation

55.2 Functional description

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

The following sections describe functional details of the MIPI RX D-PHY module.

55.2.1 HS RX equalizer

The HS RX equalizer allows users to select a fixed emphasis level for the RX to help with robustness when input signals are degraded.

Refer to Table 537 and Figure 322 for reference values for the equalizer curves. The HS RX Equalizer is disabled by default and set to no gain.

Control registers 0x509, 0x709 (lane0 and lane1 respectively) can be used to override the equalizer setting. By default the equalizer is set to 3'b000.

Table 537. Reference Values for Equalizer Curves

equalizer[2:0]	A_{DC} (dB)	A_{AC} (dB)	K_{AC_DC} (dB)	f_Z (MHz)	f_{p1} (MHz)	f_{p2} (MHz)
000	4.82	4.82	0	-	-	-
001	3.74	3.88	0.14	0.61	0.64	4.8
010	3.2	3.74	0.54	0.57	0.64	4.71
011	2.67	3.76	1.09	0.48	0.58	4.62
100	2.19	3.86	1.67	0.38	0.49	4.59
101	1.75	3.99	2.24	0.34	0.47	4.54
110	1.26	3.74	2.48	0.31	0.44	4.5
111	0.89	3.84	2.95	0.28	0.42	4.48

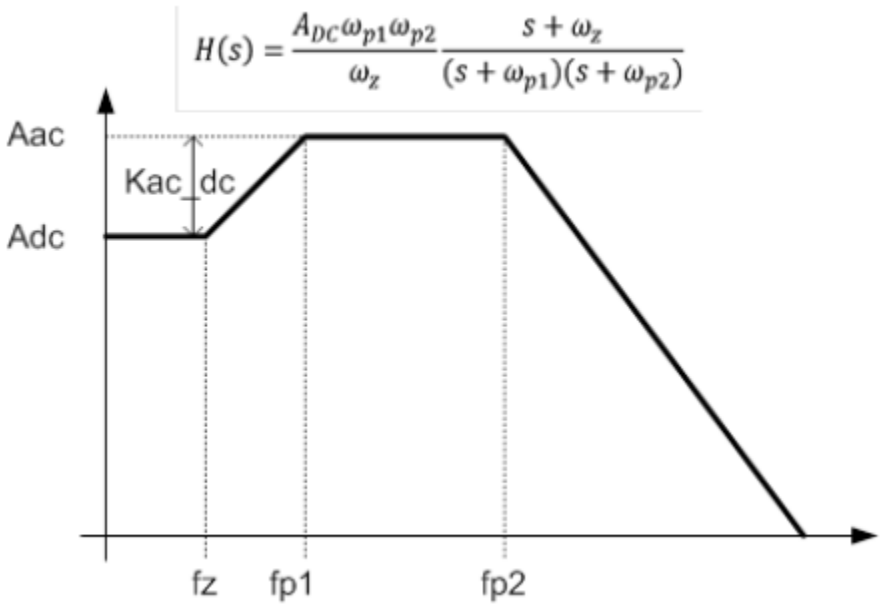


Figure 322. HS RX Equalizer Curves

55.2.2 External resistor calibration

The MIPI RX D-PHY supports the use of an optional external reference resistor. The calibration machine relies on external resistor Rext (200 Ω ±1%). The tx_cb_cal_repl[3:0] bits are progressively enabled and disabled resulting in an overall internal replica resistance value higher or lower than Rext. The results “higher than” and “lower than” are the output of internal comparator compared to Rext.

55.2.3 Power up and ULPS control FSM

The PHY supports a Finite State Machine (FSM) that defines the power up states and controls for ULPS.

Table 538. Power up and ULPS control FSM

State	Codification	Description
POWERDWN	4'b0000	Power down
BGPON	4'b0001	Common Clock Enable
CAL_TYPE	4'b0010	Termination calibration type
BURNIN_CAL	4'b0011	Termination calibration without Rext
TERMCAL	4'b0100	Termination Calibration with Rext
OFFSETCAL	4'b0101	Clock lane offset calibration
OFFSET_LANE	4’b0110	Data lane offset calibration
IDLE	4'b0111	System boot completed
ULP	4'b1000	System in global ULP
DDLTUNING	4'b1001	DDL tuning calibration

Table continues on the next page...

Table 538. Power up and ULPS control FSM (continued)

State	Codification	Description
SKEW_BACKWARD	4'b1010	Backwards compatibility skew compensation

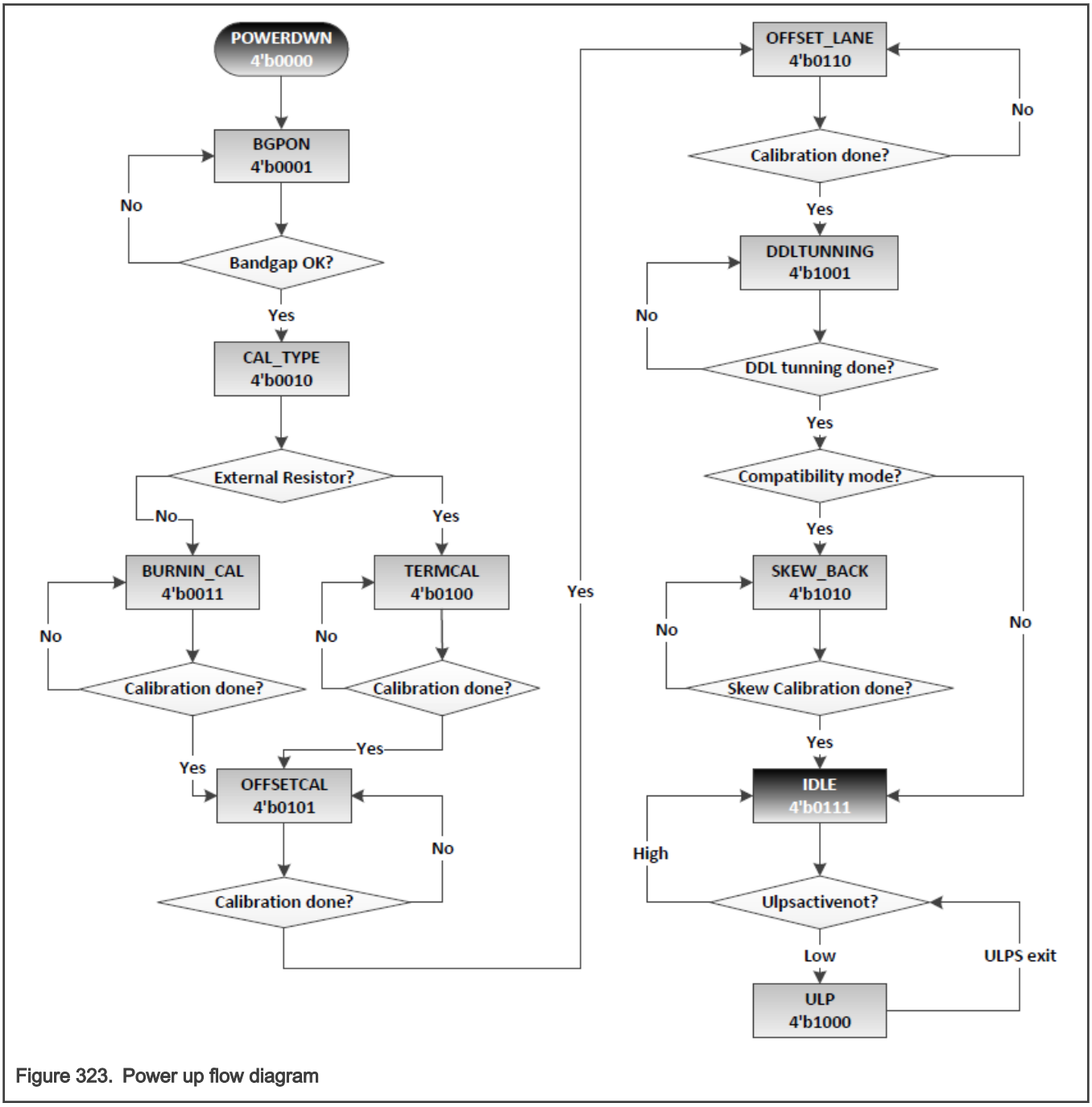


Figure 323. Power up flow diagram

55.2.4 Operating Modes

This section describes the various operating modes of the PHY.

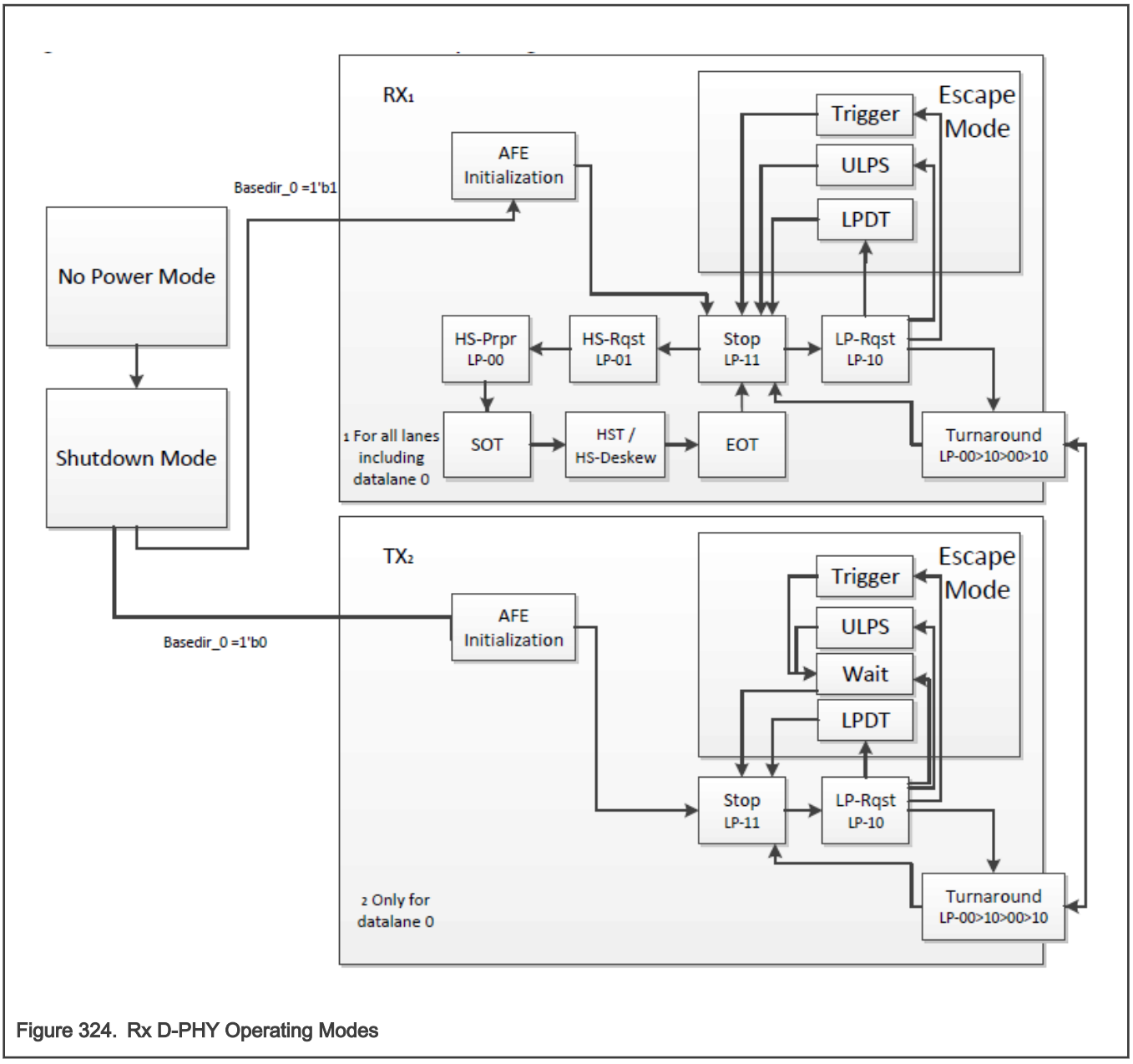


Figure 324. Rx D-PHY Operating Modes

55.2.4.1 Initialization

55.2.4.1.1 No Power Mode

The No Power mode is characterized by the non-existence of any supply voltage applied to the PHY. In order to get to the powered modes, proper voltages should be applied to the PHY.

55.2.4.1.2 Shutdown Mode

This mode is the lowest power consumption mode, where all analog blocks are disabled, and digital logic is reset. The shutdown mode is entered when rstz and shutdownz are in the low state. In this mode, the differential lines datap/datan and clkp/clkn are high impedance (Hi-Z).

All necessary MIPI D-PHY configurations to set the desired operation should be made with the MIPI D-PHY on this mode, before power-up.

55.2.4.1.3 DDL Tuning

A Digital Delay Line (DDL) is used on the clock and data paths to support the deskew mechanism introduced in the D-PHY 1.2 spec. The same mechanism is also exercised during D-PHY 1.1 compatibility mode operation, but with no pattern training sequence requirements. Each data lane has a DDL for the clock (clock received from the clock lane) and a DDL for the data path.

The DDL for the data path is only exercised when the D-PHY is working in backwards compatibility mode for operating frequency lower than 1.5 Gbps.

The DDL target oscillation frequency needs to be configured for the desired operating frequency according toTable 539 . It can be configured through the following registers:

- global control over all lanes through 0xe2, 0xe3, 0xe4
- independent lane 0 control through 0x60a, 0x60b, 0x60c
- independent lane 1 control through 0x80a, 0x80b, 0x80c

Table 539. Frequency Ranges and Defaults

Range (Mbps)	Default Bit Rate (Mbps)	hsfreqrange_ovr[6_0]	DDL target oscillation frequency (bits 11:0) in decimal
80 - 97.125	80	0000000	460
80 - 107.625	90	0010000	460
83.125 - 118.125	100	0100000	460
92.625 - 128.625	110	0110000	460
102.125 - 139.125	120	0000001	460
111.625 - 149.625	130	0010001	460
121.125 - 160.125	140	0100001	460
130.625 - 170.625	150	0110001	460
140.125 - 181.125	160	0000010	460
149.625 - 191.625	170	0010010	460
159.125 - 202.125	180	0100010	460
168.625 - 212.625	190	0110010	460
182.875 - 228.375	205	0000011	460
197.125 - 244.125	220	0010011	460
211.375 - 259.875	235	0100011	460
225.625 - 275.625	250	0110011	460
249.375 - 301.875	275	0000100	460
273.125 - 328.125	300	0010100	460
296.875 - 354.375	325	0100101	460
320.625 - 380.625	350	0110101	460
368.125 - 433.125	400	0000101	460
415.625 - 485.625	450	0010110	460

Table continues on the next page...

Table 539. Frequency Ranges and Defaults (continued)

Range (Mbps)	Default Bit Rate (Mbps)	hsfreqrange_ovr[6_0]	DDL target oscillation frequency (bits 11:0) in decimal
463.125 - 538.125	500	0100110	460
510.625 - 590.625	550	0110111	460
558.125 - 643.125	600	0000111	460
605.625 - 695.625	650	0011000	460
653.125 - 748.125	700	0101000	460
700.625 - 800.625	750	0111001	460
748.125 - 853.125	800	0001001	460
795.625 - 905.625	850	0011001	460
843.125 - 958.125	900	0101001	460
890.625 - 1010.625	950	0111010	460
938.125 - 1063.125	1000	0001010	460
985.625 - 1115.625	1050	0011010	460
1033.125 - 1168.125	1100	0101010	460
1080.625 - 1220.625	1150	0111011	460
1128.125 - 1273.125	1200	0001011	460
1175.625 - 1325.625	1250	0011011	460
1223.125 - 1378.125	1300	0101011	460
1270.625 - 1430.625	1350	0111100	460
1318.125 - 1483.125	1400	0001100	460
1365.625 - 1500	1450	0011100	460
1413.125 - 1500	1500	0101100	460

55.2.4.2 Active modes

55.2.4.2.1 Control Mode

Control mode is the default operating mode. After initialization is completed, the D-PHY remains in this default mode until a request is made either by the protocol layer, or directly through the sequence of low power signals in the lanes in case of RX. While in control mode, the transmitter side sets the LP-11 state in the lines. The LP-11 state is also called the Stop state.

The receiver side remains in control mode while receiving LP-11 in the lines. Any request must start from and end in Stop state. Following a request, a lane can leave control mode for either high-speed data transfer mode, Escape mode, or Ultra Low Power state.

55.2.4.2.2 High-Speed Data Reception

Once the initialization sequence is completed, the D-PHY remains in control mode, which is the default operating mode, until some request appears. High-speed is one of the possible requests at this point. High-speed data transfer occurs in bursts. Only during

these bursts that the lane is in high-speed mode. A high-speed burst must start from and return to a Stop state (control mode). A high-speed burst allows for the transmission of payload data by the data lanes.

A valid DDR clock in the clock lane is active during data reception.

High speed data bursts should start synchronously in all data lanes (one UI difference in HS-SYNC is tolerated), but may end independently of the state of the remaining data lanes due to an unequal amount of received bytes per data lane.

For a clock lane, processing the high-speed starts with the reception of a low- power sequence (LP11 > LP01 > LP00) to enable HS-RX line termination followed by a HS-0 during $T_{CLK-ZERO}$toenable the HS-RX clock receiver.

Fora data lane, the high-speed entry sequence is the similar as for clocklanes, that is low-power sequence to enable HS-RX line termination followed by a HS-0 during $T_{HS-ZERO}$ to enable the HS-RX data receiver.

RX will wait for $T_{CLK-SETTLE}$ and $T_{HS-SETTLE}$ to expire in order to neglect transition effects on clock and data, respectively. Before the payload data the leader sequence (HS-SYNC value 1'h1D) is transmitted to that allow for receiver synchronization. The transmission of the synchronization sequence requires the existence of a valid high-speed clock signal in the clock lane. The clock lane will enter the high-speed mode before a high-speed data transmission ($T_{CLK-PRE}$) begins and it must not leave this state before all the lanes finish their respective high-speed. The operation sequence when leaving the high-speed mode is slightly different for data and clock lanes. For a clock lane, the high-speed transfer always ends with a HS-0 state, this absence of clock transitions will be detected by RX ($T_{CLK-MISS}$) followed by the disabling of the HS-RX of the clock lane. As for a data lane, transmission ends with the differential state opposite to the last bit transmitted and then Stop state is observed in the lines (LP-11) and the PHY returns to Control Mode.

55.2.4.2.3 Escape Mode

Escape mode is a special mode of operation that uses the data lanes to communicate using the low-power states at low-speed. The PHY supports this mode in both directions for data lane 0 and only in forward direction for the other lanes. A data lane enters the Escape mode through an Escape mode entry procedure (LP-11, LP-10, LP-00, LP-01, LP-00). If an LP-11 is detected before reaching LP-00 state, the entry is aborted and the receiver returns to the Stop state. Once the sequence is correctly completed, the transmitter sends an 8-bit command to indicate a requested action.

Table 540 shows the Escape mode supported actions. If the entry command is not valid, it is ignored, ERRESC error flag goes high, and the receiver waits until the transmitter returns to the Stop state. The PHY applies Spaced-One-Hot encoding (a Mark state is interleaved with a Space state) on commands and data.

Each symbol consists of the following two parts:

- One-Hot phase
- Space state to transmit one bit, a Mark-1 should be sent followed by the Space state. In the case of a zero bit, a Mark-0 should be sent followed by Space state.

Table 540. Escape Mode Sequences for Data Lanes

Escape Mode Action	Entry Command Pattern (First Bit to Last Bit to be Transmitted)	Command Type
Low-Power Data Transmission	8'b11100001	mode
Ultra- Lower Power State	8'b00011110	mode
Reset Trigger	8'b01100010	trigger
Unknown - 3	8'b01011101	trigger
Unknown - 4	8'b00100001	trigger
Unknown - 5	8'b10100000	trigger

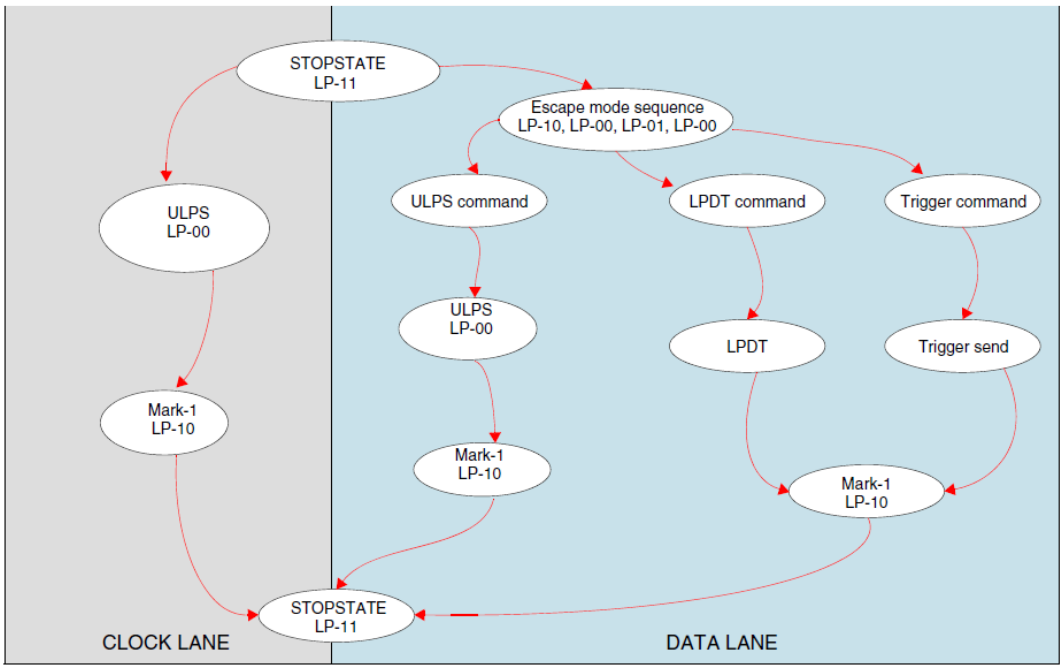


Figure 325. Escape Mode Sequences State Diagram

55.2.4.2.3.1 Low-Power Data Transmission (LPDT)

In LPDT mode, the data can be transmitted by the protocol at low speed in Low-Power mode—high-speed receivers are off and low-power receivers are on. During LPDT, the protocol can pause by maintaining a Space state on the lines. The last state before exiting Escape mode is Mark-1—not considered as 1 bit— because it is followed by the Stop state.

Data shall be encoded on the lines with the same Spaced-One-Hot code as used for the Entry Commands. This encoding allows data to be self-clocked and does not rely on the clock lane.

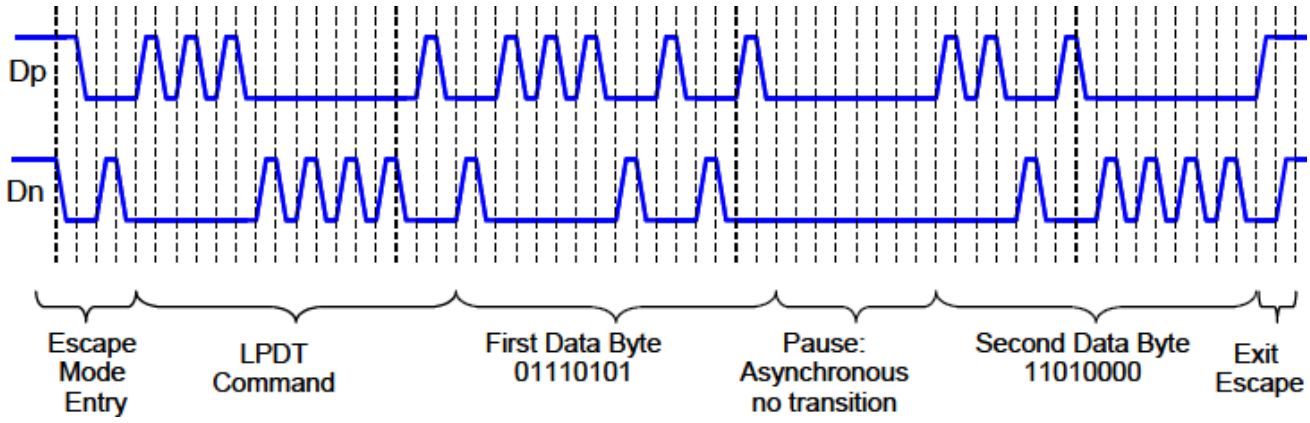


Figure 326. LPDT Example

55.2.4.2.3.2 Remote Trigger

The Remote Trigger mode allows the protocol to send a flag to the receiving side, on request of the transmitting side. If the Entry Command pattern matches the Reset-Trigger Command (8'b01100010) a Trigger is flagged to the protocol at the receive side via the logical PPI. Any bit received after a Trigger Command but before the Lines go to Stop state shall be ignored. Therefore, dummy bytes can be concatenated in order to provide clock information to the receive side.

55.2.4.2.3.3 Ultra Low Power State (ULPS)

The ULPS mode has the lowest power consumption, excluding the Shutdown mode. For data lanes, this mode is entered by sending an Ultra Lower Power State entry command (8'b00011110), after the Escape mode entry command. During this mode, the lines are in the Space state (LP-00). Although the clock lane does not support regular Escape mode, the clock lane supports ULPS.

To exit ULPS, the transmitter side drives a Mark-1 followed by a Stop state.

55.2.4.2.4 Turnaround

The PHY allows the transmission direction of the data lane 0 to be swapped by means of a turnaround request. The sequence of a turnaround operation occurs as follows:

- Transmitter side
  - Starts when turnrequest_0 input is set high on transmitter side, assuming the PHY is in Stop state
  - Transmitter sends specific low-power sequence—LP-11, LP-10, LP-00, L-10, L-00
- Receiver side
  - After receiving complete low-power sequence, receiver enables low-power driver
  - Transmits LP-00, indicating change in direction
  - Overlapping LP-00 transmitted by initial transmitter
  - Initial receiver side transmits LP-10, followed by LP-11, returning to Control mode

The turnaround procedure is the same for a Forward-to-Reverse direction or Reverse-to-Forward direction. However, in the Reverse-to-Forward direction, lane0 only supports Escape Mode. Figure 327 shows the state diagram for a turnaround sequence.

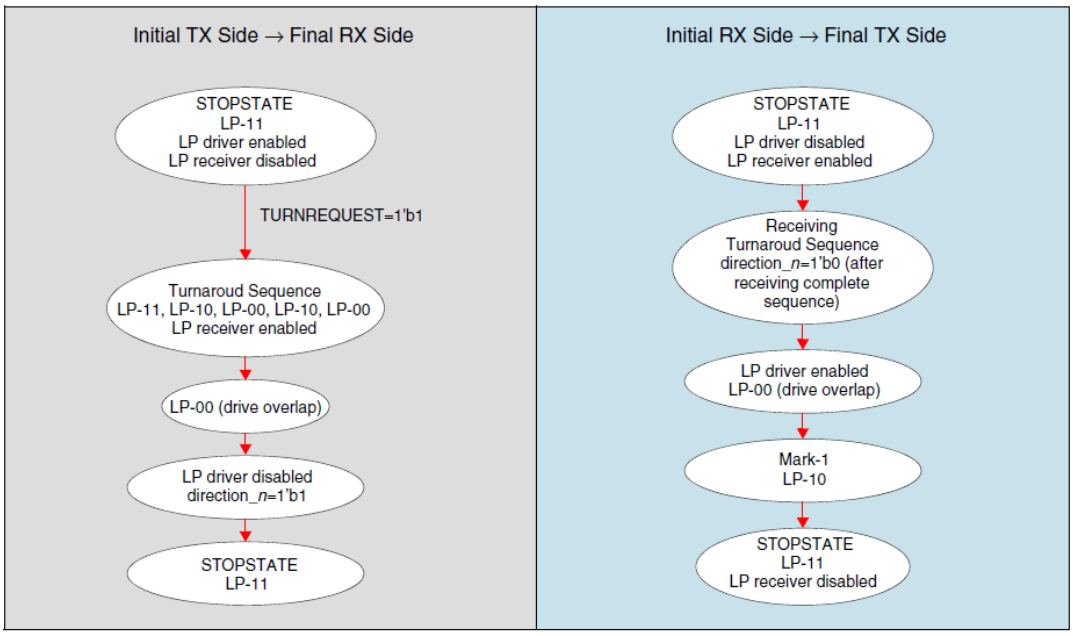


Figure 327. Turnaround Sequence State Diagram

55.3 Initialization

Synopsys Proprietary. Used with permission. This notice must be maintained on any copies or excerpts of this document.

55.3.1 Start-up sequence

The D-PHY operating frequency, input reference clock frequency and DDL target oscillation frequency need to be configured through hsfreqrange[6:0], cfgclkfreqrange[5:0] and osc_freq_target[11:0]. Refer to Table 539.

hsfreqrange can only be configured before the D-PHY power-up. It cannot be changed during active operation.

cfgclkfreqrange[5:0] needs to be configured according to the following formula: cfgclkfreqrange[5:0] = round[(Fcfg_clk(MHz) - 17) * 4].

Depending on the obtained cfgclkfreqrange, counter_for_des_en_config_if need also to configured according to Table 541.

Table 541. counter_for_des_en configuration

cfgclkfreqrange (decimal)	counter_for_des_en_config_if[3:0] (decimal)
0 ≤ cfgclkfreqrange[5:0] ≤ 12	0
13 ≤ cfgclkfreqrange[5:0] ≤ 38	1
39 ≤ cfgclkfreqrange[5:0] ≤ 40	2

After the reset signal RSTZ and SHUTDOWNZ are released, the PHY begins an initialization sequence that allows its correct operation. SHUTDOWNZ should precede RSTZ. After current references settle, the following the internal calibrations are performed:

1. Termination resistors calibration
2. Data and clock lane offset compensation
3. Digital Delay Line (DDL) tuning calibration
4. For operating frequencies < 1.5 Gbps (DPHY1.1 backwards compatibility mode), an internal deskew compensation algorithm is executed with internal stimuli provided to data and clock lanes.

The deskew algorithm selection is done automatically based on hsfreqrange[6:0] configuration but can be overridden through test control registers.

Chapter 56
Pixel Pipeline (PXP)

56.1 Chip-specific PXP information

Table 542. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The chip includes one instance of PXP in MEDIAMIX.

NOTE

Only one axi bus is available on this chip, the INPUT_FETCH_CTRL_CH0[ARBITEN] bit must be configured to 1'b1.

NOTE

INPUT_FETCH_CTRL_CH0[ARBITEN] bit description on this chip should be as below:

0 = Arbitration disabled. Not supported when PXP is enabled.

1 = Arbitration enabled. Must be set whenever PXP is enabled.

56.2 Overview

PXP processes graphics buffers or composite video and graphics data before sending to an LCD display or TV encoder. It minimizes the memory footprint required for the display pipeline and provide an area and performance optimized to both SDRAM-less and SRAM-based systems.

PXP integrates of several independent processing stages into a cohesive strategy to create flexible pixel pipeline.

PXP combines the following into a single processing engine:

- Scaling
- Color Space Conversion (CSC)
- Secondary Color Space Conversion (CSC2)
- Rotation

By integrating multiple blocks, intermediate buffer operations to external memory are removed, reducing external memory bandwidth, power, and software control complexity.

56.2.1 Block Diagram

PXP block diagram is shown below.

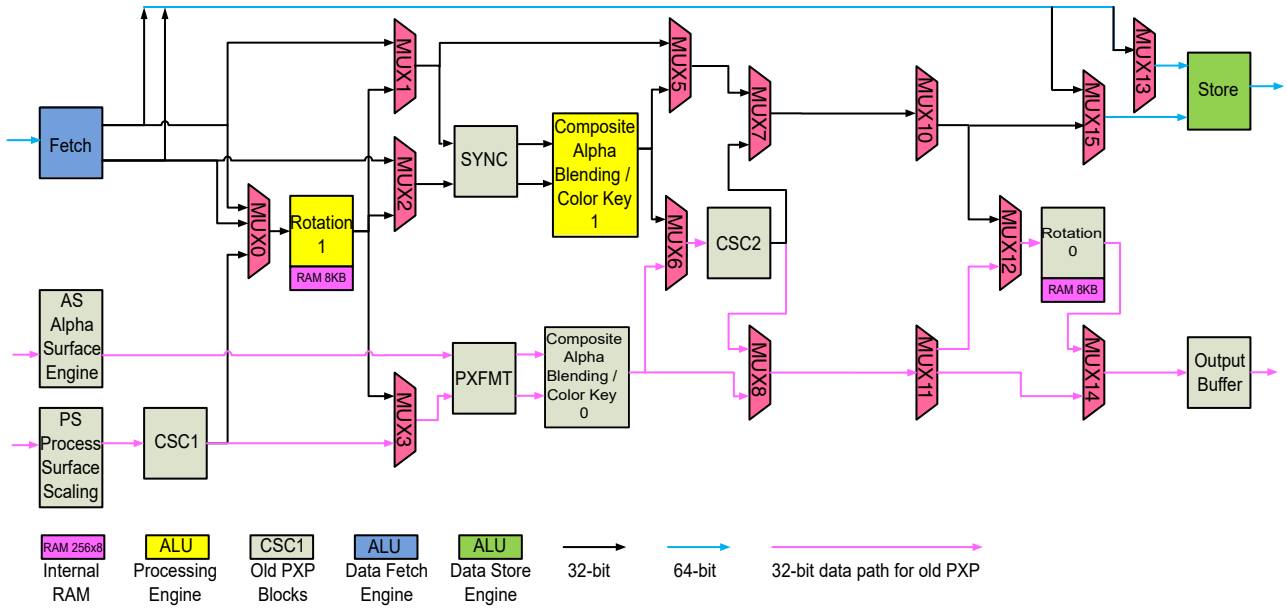


Figure 328. PXP Architecture

56.2.2 Features

PXP module includes following features:

- Scaling
  - Supports YUV 422, 420, 444 and any RGB formatted pixels
  - The maximum downscaling factor can be up to 16
- Color Space Conversion (CSC)
  - Supports converting from YUV to RGB
  - Supports bypass mode
  - Supports color space coefficients configuration
- Secondary Color Space Conversion (CSC2)
  - Supports converting to RGB or YUV color spaces depending on the desired output pixel format
  - Supports color space coefficients configuration
- Rotation
  - Supports 0, 90, 180, 270 degrees
  - Supports vertical and horizontal flips
- Composite Alpha Blending and Color Key
  - Supports normal alpha blending and Porter-Duff alpha blending.
  - Supports color key function to create transparent effects on the output pixel

56.3 Functional Description

PXP consist of several pipelined blocks that perform the video source frame scaling, color space conversion, alpha-blending and color key algorithm, secondary CSC, input and output rotation.

The legacy blocks operate within the requirements of the legacy PXP architecture, and perform operations on either 8x8 or 16x16 pixel blocks in the representative source buffers. The legacy pipeline operate within the context of two iteration counters that iterate through the appropriate grid of input blocks to produce the rotated output grid blocks in scan-line order.

Figure 328 shows the high-level architecture of the scaling, color space conversion, blending, rotation engines. The Alpha Surface Engine fetches one RGB graphics plane alpha surface (AS). The scaling engine fetches a single processed surface (PS), which can be blended with the AS surface. The scaling engine also supports an alpha channel for the PS image. Although the legacy PXP processes NxN pixel macro blocks, each of the AS or PS surfaces can have any pixel alignment within the output buffer. There are no restrictions and any pixel coordinates within the output buffer are valid. The upper left origin of the output buffer is defined as pixel 0,0. The upper left and lower right coordinates for each of the AS and PS are inclusive within the output buffer.

Figure 329 represents a sample output buffer configuration with both an alpha surface (AS) and a processed surface (PS) included. The alignment of each AS and PS within the output buffer can be at any arbitrary pixel locations. For example, the PS has an upper left coordinate (ULC) of 2,2 and a lower right coordinate (LRC) at pixel 13,13. The maximum value for the ULC and LRC for each of the AS and PS is bounded by the LRC of the output buffer, 15,15 for this example.

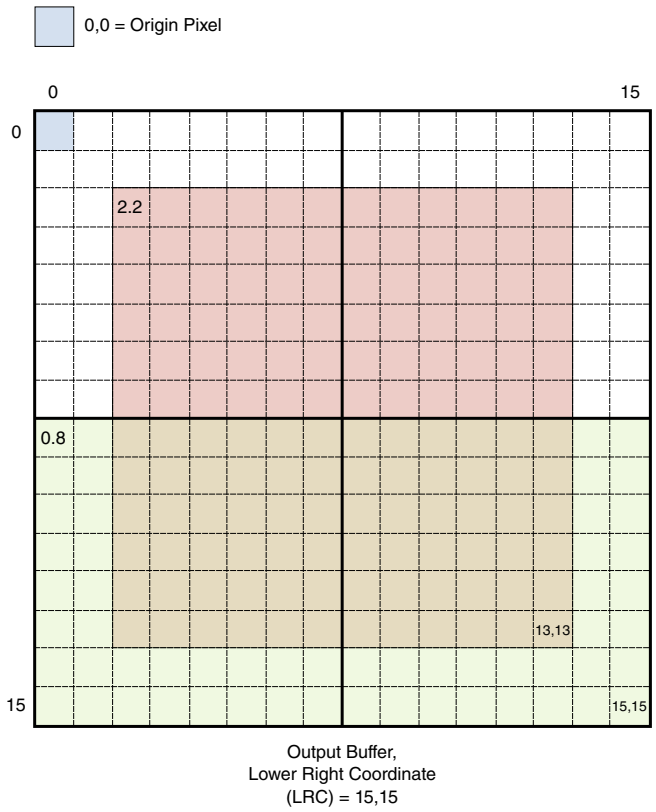


Figure 329. Sample output buffer configuration

The AS engine supports RGB pixel formats, and the PS engine supports ARGB, RGB, YUV, and YCbCr pixel formats. The CSC1 can be used to convert to RGB pixel formats so that the PS surface can be blended with the AS surfaces in the compositing engine in the RGB color space. There are two rotate engines in PXP pipeline. Rotation can occur after image composition, at the output of the PS engine, or both. In the first scenario, all the data produced by the AS and PS engines is rotated. When the rotation module is programmed to rotate only PS images, the AS is not rotated, and AS pixels are combined with rotated PS surfaces. The CSC2 unit can convert to any RGB, YUV, or YCbCr color spaces for final output.

PXP also supports two parallel image processing paths. The legacy flow can operate in parallel with fetch and store processing engines. See CTRL and CTRL2 for details on how to enable each of the individual data flows.

56.3.1 Operations

56.3.1.1 Processing Details

The legacy PXP architecture has been driven primarily by the requirement that the output buffer must be processed and rotated without intermediate frame buffer stored in external memory.

This reduces the use of external memory bandwidth requirements thus reducing overall system power consumed. The new architecture has fetch and store engines to enable parallel image processing.

Since the output of the rotation block must be NxN pixel blocks in scan order, the entire legacy pipeline will operate on NxN pixel blocks. In essence, the pipeline will be able to operate on blocks in a random access fashion, but the entire pipeline will operate within the context of two iteration counters that will iterate through the horizontal and vertical input blocks to generate the required output block.

Legacy Processing Pipeline

The control block will coordinate the processing of the pixel blocks within the source and destination image buffers. It begins by issuing a command to each stage of the pipeline requesting that operations be done for the block at offset x, y. When the block accepts the command, it asserts its acknowledge signal for a single cycle to indicate the acceptance and allow the control unit to move to the next block.

When the PS and AS fetch engines have received a command, they will fetch the required data and place it into their fetch buffers. If compositing the RGB AS surface with the PS surface, then the output of the PS engine needs to be converted to the RGB color space using CSC1, since all compositing occurs in the RGB color space. For YUV output pixel formats, the CSC1 unit can be enabled to convert pixels into the RGB space for subsequent compositing with AS pixels. Then, the CSC2 module can convert the resulting pixels back into the YUV output color space. If the final output color space is YUV and there is no compositing required (AS not present, for example), then both the CSC units can be bypassed and the pixel data path will pass the YUV pixels to the rotation engine. For YUV output formats, scaling operations, LUT, rotation, and dithering operations are still valid, but blending RGB AS surfaces with YUV PS surfaces is NOT supported. The two CSC units in the overall pixel data path must be used to achieve the desired source frame compositing and output pixel formatting.

The alpha blender/color key module will process a pixel any time that both inputs present valid data.

The New Processing Pipelines

The fetch and store engines work in block and scan modes. The fetch engine fetchs data from AXI bus. The store engine output the processed data flow that comes from fetch engine.

56.3.1.2 Scaling Operation

The scaling engine operates on YUV (or YCbCr) 422 or 420 and any RGB formatted pixels. Each color plane is sourced from color planes indicated by different base address registers. The scaling source data can be stored as 3 individual planes for each Y, U, and V data, stored as two planes as a single Y and interleaved UV plane, or stored as a single plane with YUV/RGB interleaved on a per byte basis.

The scaled output image is presented to the CSC module as YUV444 or RGB888 pixels with a single byte for each color channel. The scaler can reduce an input image by a maximum factor of 16. In this case, the output image will be 1/16 the dimension of the input image in each of the X and Y axis. There are no limits, essentially, on increasing the source image size. The theoretical maximum increase is 4096 since a 12 bit fractional step function is used when scaling an input image. Scaling in either axis, X or Y is independent, so a source image can appear stretched in either direction.

All source images pass through the scale engine. PXP alpha blend module and AS pixel streams are in the RGB888 format, so PS pixel buffers must be converted to the RGB888 format for alpha blending. The scaling engine works with the CSC1 module to translate YUV/YCbCr pixel formats to RGB888 for output frame buffer compositing using the alpha blender.

The CSC2 module can by bypassed or enabled to convert pixels to any output color space.

In the case of processing RGB pixels in the PS engine, the CSC1 unit can be bypassed so compositing can occur in the alpha engine.

The scaling operation is divided into two scaling steps. The first step is a decimation scaler, and the second step is a bilinear filter. The decimation filter provide a maximum down scaling factor of 8, and the subsequent bilinear filter provides a maximum

downscaling factor of 2. Combined, the maximum downscaling factor can be up to 16. The decimation and bilinear scaling engines are independently programmable. There is also an initial offset that is programmable to allow more source data to be considered in the bilinear scaling engine.

56.3.1.2.1 Decimation Image Scaling

The first of two scaling engines is the decimation filter. The intent of the decimation filter is to use as much source data as is possible to create the output image frame buffer. The decimation filter simply discards certain pixels from the source PS image depending on the reduction selected. For RGB pixel formats, each color channel is treated equally since there is the same amount of pixel data within each color plane. For YUV422/420 formats, the chroma samples are already subsampled by 2. In these decimation scenarios, the chroma decimation factor is adjusted to account for the pre-decimation of the chroma samples. For example, since YUV422 is already sub-sampled by 2 horizontally, an X decimation factor of 2 does not apply to the YUV422 pixels in the X direction. All the chroma samples are passed on to the bilinear filter in this case. As another example, an X decimation factor of 4 will decimate the chroma samples by 2, since this factor combined with the pre-decimation factor of 2 in the pixel source buffers totals an overall decimation factor of 4.

Figure 330 shows which pixels (in green) in a source RGB buffer that are passed to the bilinear filter for an X decimation factor of 2 and a Y decimation factor of 4. All pixels coincident with dashed lines are discarded.

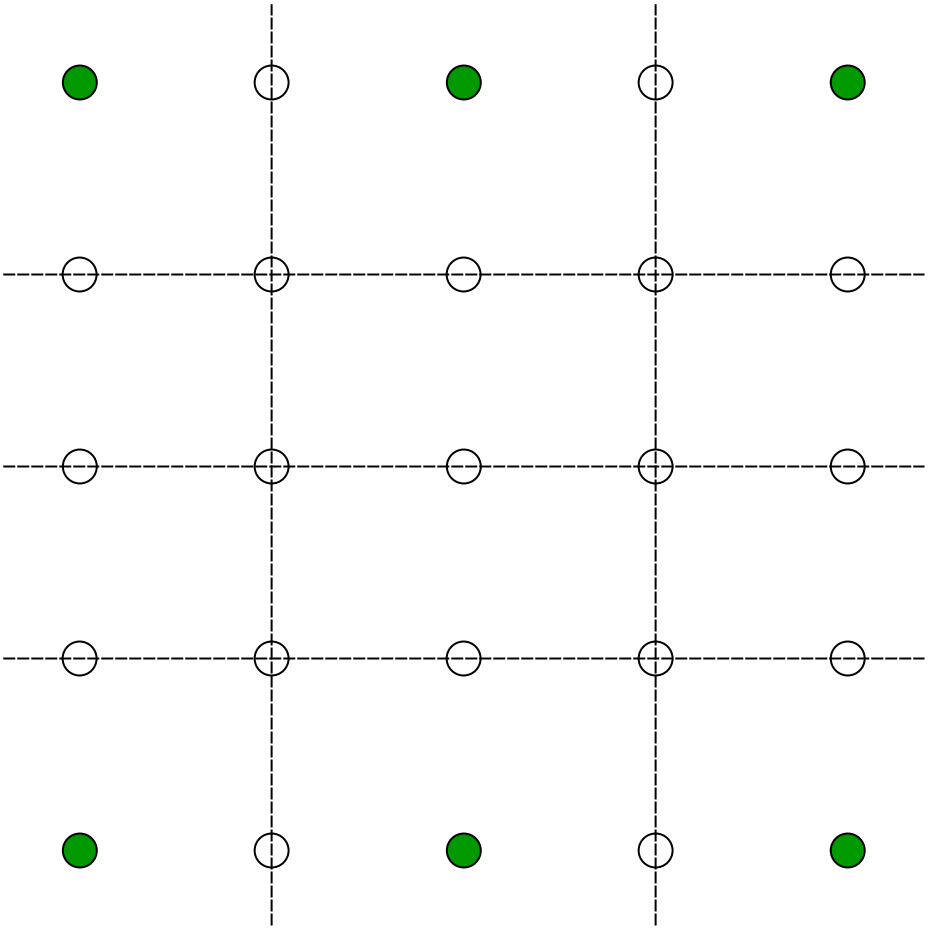


Figure 330. RGB decimation X /2, Y /4

Using the same decimation factor as the above scenario for RGB pixels, but using YUV420 source buffers, it can be shown that the decimation factor for the Y and UV components of data are decimated differently. This is due to the pre-decimation of the chroma samples in the source frame buffers. Figure 331 indicates that the U/V samples in the X direction are not decimated, but the Y samples in the X direction are decimated by the factor of 2.

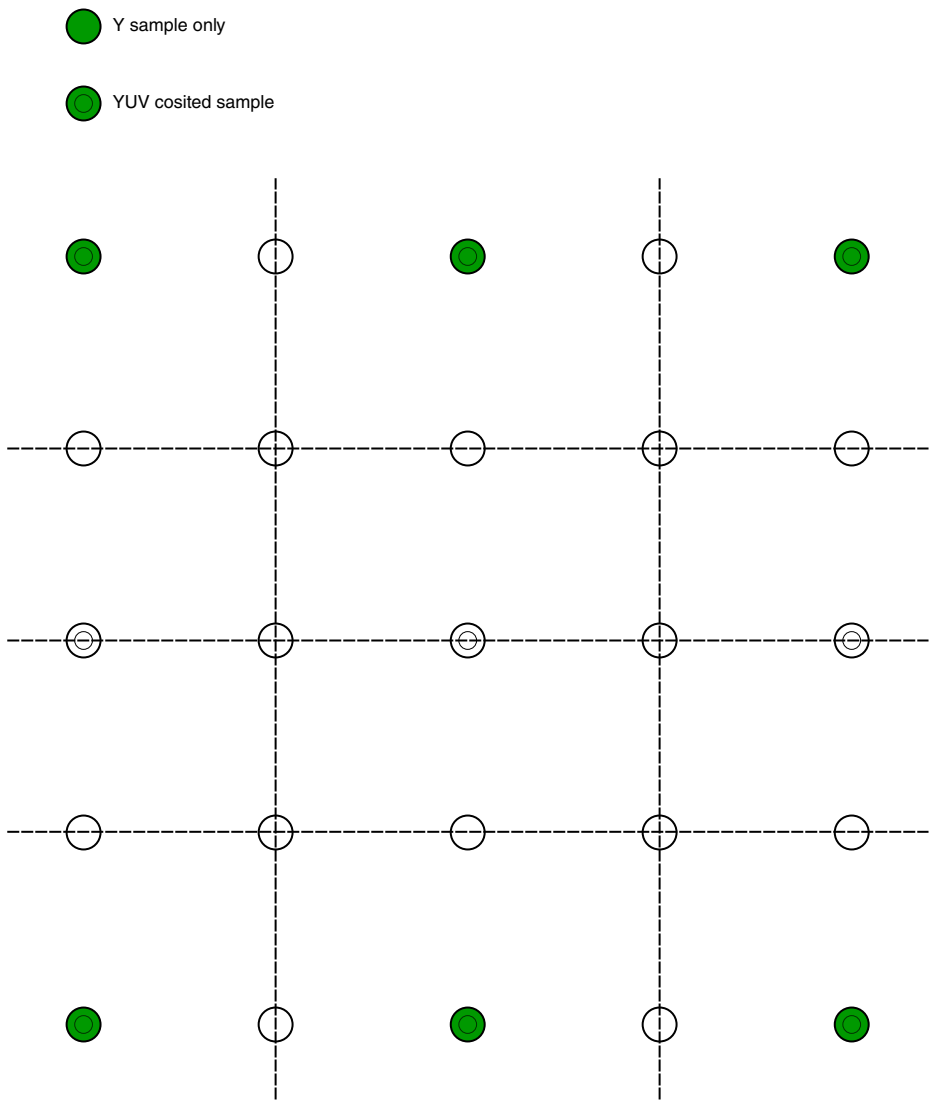


Figure 331. YUV420 decimation X /2, Y /4

56.3.1.2.2 Bilinear Image Scaling Filter

PXP implements a bilinear scaling filter to resize an input image to a different resolution for display output. The bilinear filter is a weighted average of the four nearest pixels that can be sourced to approximate the pixel in the output frame buffer.

When scaling YUV data, the UV values are offset by 0x80 (top bit inverted) to shift the signed UV bits into an unsigned equivalent with a range of 0 to 255. YCbCr data does not have to be shifted since it is defined as an unsigned byte. CSC1_COEF0[YCBCR_MODE] controls whether this operation is applied to the input UV bytes.

After scaling, the offset is removed so that the range for UV data is signed from -128 to 127.

The reason for this adjustment is based on the implementation of an unsigned scaling engine, and therefore, to ensure that the scaled values are handled properly. Consider the Table 543:

Table 543.

Format	pixel0	pixel1	average	Result
decimal	-2	+2	0	Correct
CbCr	0x7E	0x82	0x80	Correct (0x80 is 0 in CbCr)
UV	0xFE	0x02	0x80	Incorrect (0x80 is -128 in UV)
decimal	-32	+16	-8	Correct
CbCr	0x60	0x90	0x78	Correct (0x78 is -8 in CbCr)
UV	0xE0	0x10	0x78	Incorrect (0x78 is +120 in UV)

To compute the output pixel value at position as indicated by P, consider the Figure 332.

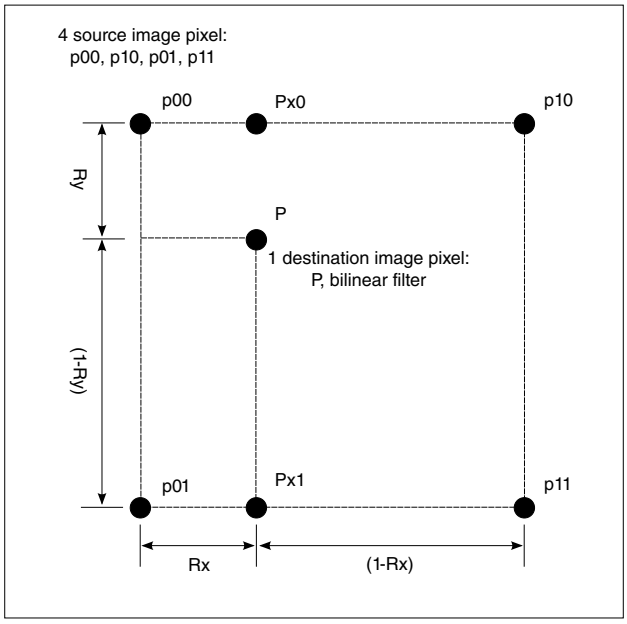


Figure 332. Output Pixel Value

A step function is used to indicate the position of the pixel "P" in the output frame. This position may not coincide with a single pixel position in the input frame buffer. In this case, the four closest pixels in the input frame are used to approximate the value of the pixel in the output frame.

PXP scaler first computes a linear filter in the X axis to create the two intermediate pixel values Px0 and Px1. The step function's X fractional component is used to provide the weighting factor for blending p00 with p10 to provide Px0. Likewise, Px1 is also derived from a linear filter using p01 and p11.

The equations for Px0 and Px1 are as follows:

Px0 = p00*(1-Rx) + p10*Rx

Px1 = p01*(1-Rx) + p11*Rx

PXP scaler uses the intermediate X pixels Px0 and Px1 and implements a bilinear filter on these two pixel values to produce the final pixel value at position P. The remainder of the step function for the Y axis is used to compute the weighted average pixel result. The equation for final filtered pixel is:

P = Px0*(1-Ry) + Px1*Ry

56.3.1.2.3 YUV 4:2:2 Image Scaling

Figure 333 illustrates the positioning of YUV samples for the 4:2:2 formats. There are twice as many Y luma samples then U and V chroma samples horizontally.

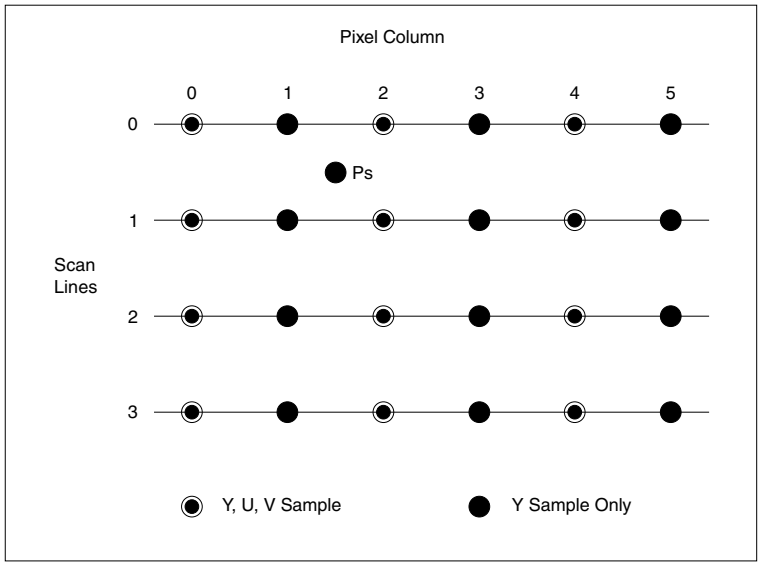


Figure 333. YUV Sample Positioning, 4:2:2

Consider the scaled output pixel Ps (pixel scaled) which has an accumulated step function of X=1.5 and Y=0.5. The remainder for the step function is Rx = 0.5 and Ry = 0.5. Or, the sub pixel position of output pixel Ps is half way between line 0 and 1 and half way between column 1 and 2.

The Y output component of Ps is simply the bilinear function of the four nearest Y samples from the input image. Specifically, the Y values at [1,0], [2,0], [1,1], and [2,1] are used to compute the Y for Ps.

For the U and V components of Ps, there are no samples present in the column position 1. The bilinear filter uses chroma components located at [0,0], [2,0], [0,1] and [2,1]. Since the chroma components are not sub sampled vertically, the remainder used to combine pixels vertically is Ry=0.5 (the same as for Y). However, horizontally, the scaling engine shifts the remainder by a factor of 2. So an X axis step function value of X=1.5 has a remainder Rx=0.75. Source chroma values are not replicated, they are completely interpolated using the four nearest chroma samples to approximate U and V at Ps.

56.3.1.2.4 YUV 4:2:0 Image Scaling

Figure 334 illustrates the positioning of YUV samples for the 4:2:0 formats. Chroma is sub sampled both horizontally and vertically. In this format, the chroma frame buffers contain ¼ the data that the luma frame buffers store.

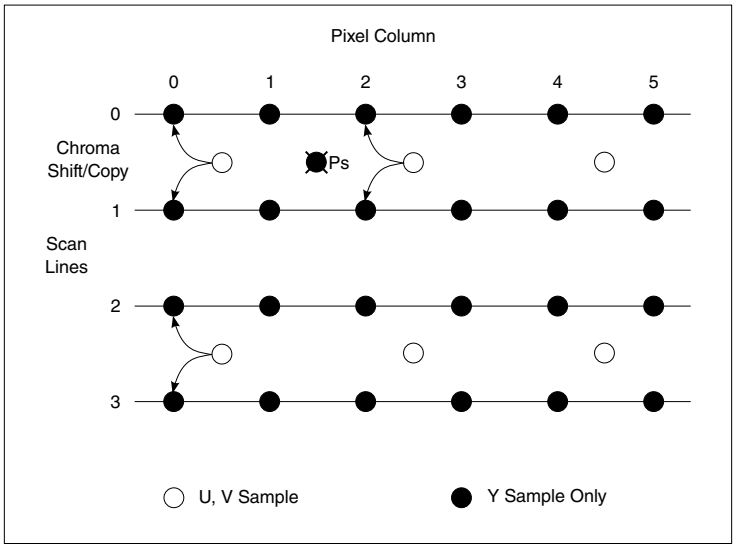


Figure 334. YUV Sample Positioning, 4:2:0

The Y output component for all scaled pixels in 4:2:0 formats are the same as for the 4:2:2 pixel formats.

The U and V output components have two considerations when computing the output pixel Ps.

1. All chroma samples from the input source image are shifted left and up by ½ a sample position of the input pixel matrix.
2. Odd scan lines are replicated using the previous even chroma scan line values. So, output image chroma values that map between even to odd scan lines are replicated in the vertical axis. In contrast, output image chroma values between odd to even scan lines are interpolated vertically.

The chroma values are interpolated horizontally as in the 4:2:2 pixel format.

As an example, consider the interpolated pixel Ps in Figure 334. For the Y component, the interpolated output luma is a function of the Y values in the source frame buffer at position [1,0], [2,0], [1,1], [2,1].

For the U and V interpolated samples, the chroma values on scan line position 0.5 are shifted so that they coincide with the even luma sample points. They are also replicated so that a single chroma scan line is used twice. The chroma scan line at 0.5 is replicated to represent the 4:2:2 sample points for scan line 0 and 1. The chroma scan line at 2.5 is replicated to represent the 4:2:2 sample points for scan line 2 and 3. This pattern of chroma replication occurs for the entire source frame buffer during the scaling operation.

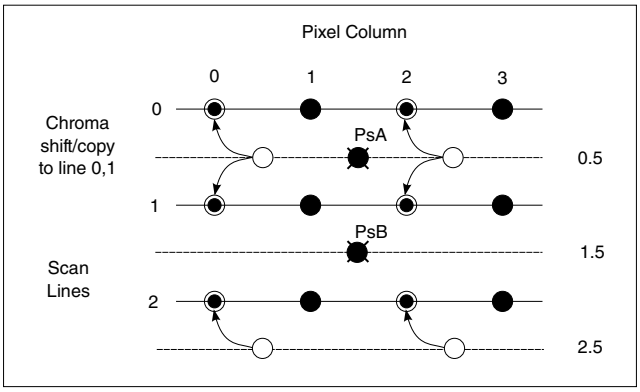


Figure 335. Scaled Chroma Computation Examples

Figure 335 has two examples for the computation of the scaled chroma output pixel. For chroma at output position PsA (vertical position 0.5), interpolation occurs in the X axis using chroma values at column 0 and column 2. However, since line 0 and line 1 have equal chroma values due to chroma line replication, scaling in the Y axis results in replication of chroma values.

For chroma at output position PsB (vertical position 1.5), interpolation occurs in both the X and Y axis. The Y axis is an interpolation since the chroma values copied to scan line 1 and 2 and not the same.

In summary, any output image pixels that map to an odd scan line above and an even scan line below are interpolated vertically. Output image pixels that map to an even scan line above and an odd scan line below are replicated vertically.

56.3.1.2.5 RGB/YUV444 Image Scaling

For all RGB formats, the RGB pixels are converted up to RGB888 with 8 bits per each color component. Then each color component is passed to the scaling engine and each component is treated in the same manor. The RGB scaling operation is the same as for the Y scaling operation described in the preceding sections. Also, YUV444 contains a byte for each color plane at each pixel location, so all three color components are scaled in the same manor.

56.3.1.3 Color Space Conversion (CSC)

The CSC unit converts pixels between color spaces. They are referred to as CSC1 and CSC2. The CSC1 unit is right after the scaling unit and is dedicated to converting from YUV to RGB. The CSC2 unit is a full-duplex color space converter that can convert into either RGB or YUV (or YCbCr) color spaces depending on the desired output pixel format. All coefficients are programmed as two's compliment numbers and CSC units can be bypassed if conversion is not desired at either position of these CSC units in the pixel data path.

56.3.1.3.1 CSC1 Operation

The CSC1 module receives scaled YUV/YCbCr444 pixels from the scale engine and converts the pixels to the RGB888 color space only if CSC1 is enabled. The CSC1 module will convert only to the RGB color space and it can be bypassed to allow YUV pixels through the data path. These pixels are loaded into the pixel FIFO for processing by subsequent modules in the pixel data path.

The following equations are used to perform YUV/YCbCr -> RGB conversion. The constants will be stored in PXP control registers as two's compliment values to allow flexibility in the implementation and to allow for differences in the video encode and decode operations. In addition, this provides a software mechanism to manipulate brightness or contrast.

R = C0(Y+Yoffset) + C1(V+UVoffset)

G = C0(Y+Yoffset) + C3(U+UVoffset) + C2(V+UVoffset)

B = C0(Y+Yoffset) + C4(U+UVoffset)

NOTE

In the equations above, U and V are synonymous with Cb and Cr in regards to the color space format of the source frame buffer.

Saturation of each color channel is checked and corrected for excursions outside the nominal YUV/YCbCr color spaces. Overflow for the three channels are saturated at 0x255 and underflow is saturated at 0x00.

Table 544 indicates the expected coefficients for YUV and YCbCr modes of operation:

Table 544. CSC Coefficients for YUV and YCbCr modes

Coefficient	YUV	YCbCr
Yoffset	0x000	0x1F0 (-16)
UVoffset	0x000	0x180 (-128)
C0	0x100 (1.00)	0x12A (1.164)
C1	0x123 (1.140)	0x198 (1.596)
C2	0x76B (-0.581)	0x730 (-0.813)
C3	0x79B (-0.394)	0x79C (-0.392)
C4	0x208 (2.032)	0x204 (2.017)

56.3.1.3.2 YUV versus YCbCr Support

By default, PXP color space coefficients are set to support the conversion of YUV data to RGB data. If YCbCr input is present, software must change the coefficient registers appropriately (see the register definitions for values). Software must also set CSC1_COEF0[YCBCR_MODE] to ensure proper conversion of YUV versus YCBCR data.

56.3.1.3.3 CSC2 operation

The CSC2 module receives pixels in any color space and can convert the pixels into any of RGB, YUV, or YCbCr color spaces. All coefficients are programmable and in the two's compliment notation. The output pixels are passed onto the rotation engine for further processing.

The following equations indicate the CSC2 modules ALU architecture.

Selecting RGB output in CSC2_CTRL[CSC_MODE] configures the ALU in the following manor:

R = A1(Y-D1) + A2(U-D2) + A3(V-D3)

G = B1(Y-D1) + B2(U-D2) + B3(V-D3)

B = C1(Y-D1) + C2(U-D2) + C3(V-D3)

Selecting YUV output configures the ALU in the alternate manor:

Y = A1*R + A2*G + A3*B + D1

U = B1*R + B2*G + B3*B + D2

V = C1*R + C2*G + C3*B + D3

Saturation of each color channel is checked and corrected for excursions outside the nominal color space. Overflow for the three channels are saturated at 0x255 and underflow is saturated at 0x00.

56.3.1.4 Alpha Blending/Color Key

There are two alpha blending/color key engines in the new PXP. Regardless of pixel input format, the PS and AS pixels are normalized to 32-bits, organized as one alpha and three data bytes. For the second alpha blending engine, the fetch engine should be programmed to fetch data in the RGB format. You can either use the 2 fetch channels or use either fetch channel with the PS.

Alpha blending occurs in the RGB space, if blending is required, PS pixels should be converted to RGB space. If no alpha blending is required, then YUV pixels can bypass the alpha blending ALU without color space conversion. All pixels are processed by the pixel ALU, but the ALU operations can be disabled to achieve pixel pass through for either PS or AS or fetch data channels source pixels.

56.3.1.4.1 Alpha Blend

There are two alpha blend modes in this module. First is the normal blending mode using single alpha parameter. Second is the porter-duff blending mode used in 2D. Both modes alpha blend modes are described in the sub-sections below.

56.3.1.4.1.1 Normal Alpha Blend

The alpha value for an individual pixel represents a mathematical weighting factor applied to the AS pixel. An alpha value of 0x00 corresponds to a transparent pixel and a value of 0xFF corresponds to an opaque pixel.

The effective alpha value for an AS pixel is determined by AS_CTRL[ALPHA] and AS_CTRL[ALPHA_CTRL] . If AS_CTRL[ALPHA_CTRL] = ALPHA_OVERRIDE, the alpha value for the pixel is taken from AS_CTRL[ALPHA]. This can be useful for applying a constant alpha to an entire image or for image formats that don't include an alpha value. If AS_CTRL[ALPHA_CTRL] = ALPHA_MULTIPLY, the pixel's alpha value will be multiplied by AS_CTRL[ALPHA] value in order to allow scaling of the pixel's alpha or to provide better control for pixel formats such as RGB1555, which only contains a single bit of alpha.

For each color channel, the equation used to blend two source pixels is defined below:

Gá = Programmed global alpha (AS_CTRL[ALPHA]).

Eá = Embedded alpha associated with AS pixel.

á = (Gá * Eá + 0x80)/256

The result for the red channel as an example:

R[7:0] = ((á * AS.r) + ((1 - á) * PS.r))/256

When á is 0x00, the PS pixel will not be blended with the AS pixel, but PS will be passed as the output pixel and will not be blended with AS. In this case, AS will be discarded. Likewise, if á is 0xff for a given pixel, AS will be loaded as the output pixel.

AS_CTRL[ALPHA0_INVERT] provides the option to invert the final alpha value. This essentially inverts the effect the alpha value has on the AS and PS blending operation.

56.3.1.4.1.2 Porter-Duff Alpha Blend

Porter-Duff blend includes 12 blending modes to describe digital image composite. These processes include Clear, Source Only, Destination Only, Source Over, Source In, Source Out, Source Atop, Destination Over, Destination In, Destination Out, Destination Atop and XOR. Through these process it can achieve any 2D image composite.

To control the blending modes, see Figure 336. All the fields shown below are included in Alpha Engine A Control Register (ALPHA_A_CTRL) and Alpha Engine B Control Register (ALPHA_B_CTRL).

The last value

alpha = (s10_alpha*s0_alpha'' + s01_alpha* s1_alpha'')/256

pixel = (s01_alpha * s1_pixel' + s10_alpha * s0_pixel')/256

NOTE

- S0 means PS/CH1.
- S1 means AS/CH0.

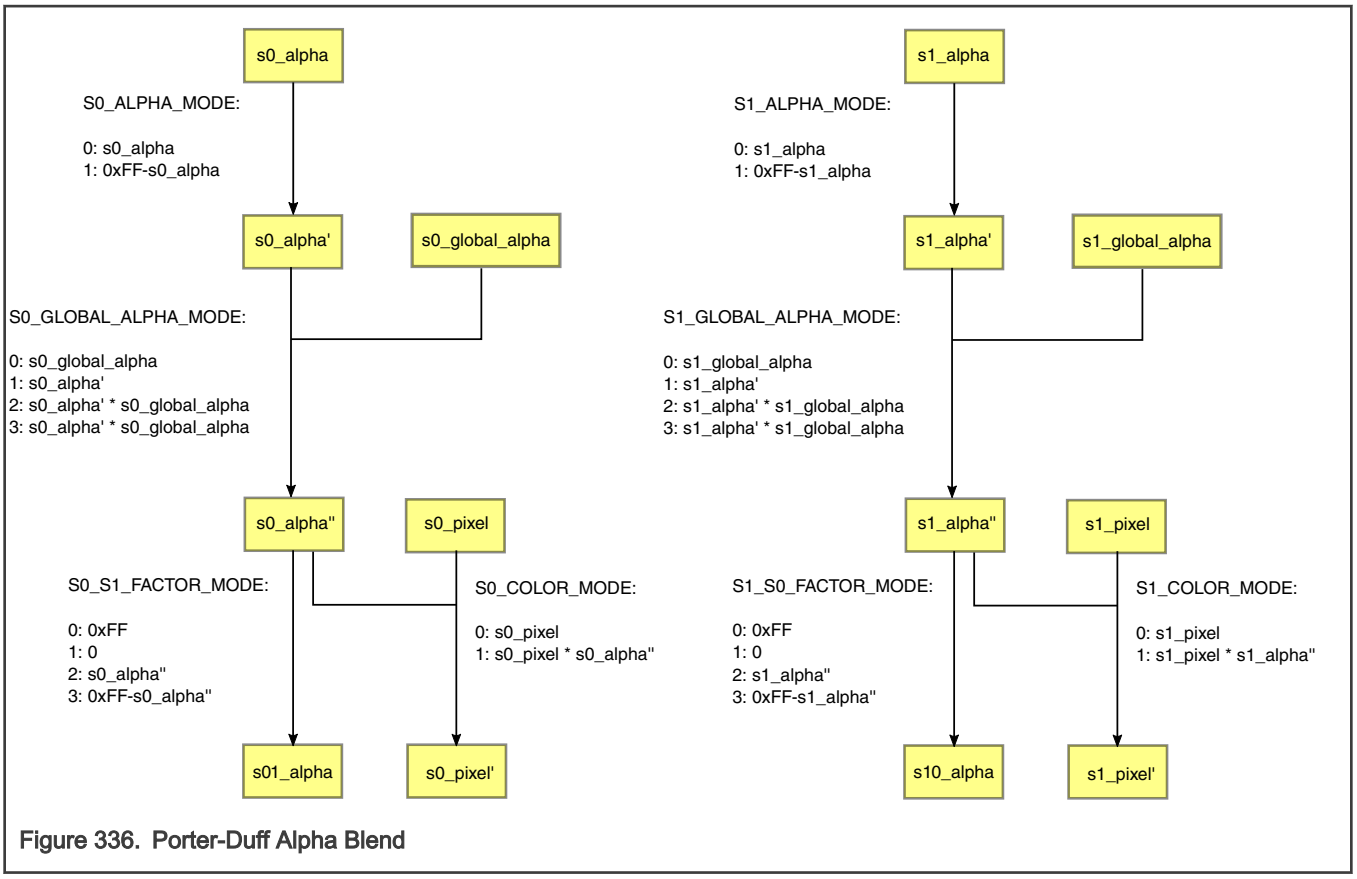


Figure 336. Porter-Duff Alpha Blend

56.3.1.4.2 Color Key

The color key function is provided to create transparent effects on the output pixel. Color keying is applied on the input pixels after they are converted to 8-bits for each red, green, and blue color channels (color keys are not applied directly to 16-bit pixel formats but to their corresponding 24-bit representation). A color key range is programmable for both PS and AS pixels. If the PS 24-bit pixel is within the PS color key range, then AS is passed through the pixel pipeline. In this case, alpha blending does NOT occur. Conversely, if AS is within the AS color key range, then PS is passed via PXP data pipeline. If both PS and AS color key tests pass, then the back ground color register is passed onto following PXP processing components in the pipeline.

The condition for color keying to be satisfied is:

CK0.r.low <= PS.r <= CK0.r.high

CK0.g.low <= PS.g <= CK0.g.high

CK0.b.low <= PS.b <= CK0.b.high

For example, if the "red" 8-bit value for the PS pixel (or PS.r) is between the color key low and high values (CK0.r.l and CK0.r.h), the condition is true for the red color plane. When ALL three color planes meet this condition, then only the AS pixel is loaded into the output register.

To disable color keying, program the low color key register value to 0xff and the high value to 0x00. This will guarantee that the color key range test will never be true.

56.3.1.5 Rotation

There are two rotation engines integrated into PXP. Rotation can occur after compositing the AS and PS buffers in the output stage.

The PS buffer can be rotated and later composited with the AS surface that is not rotated. The rotation1 engine can also be used with the Input Fetch engine.

To rotate graphics, the hardware must read pixels in one direction across a frame buffer and write them in a alternate orientation. For the 90 and 270 degree cases, this means that lines of pixels must either be read or written vertically in a frame buffer.

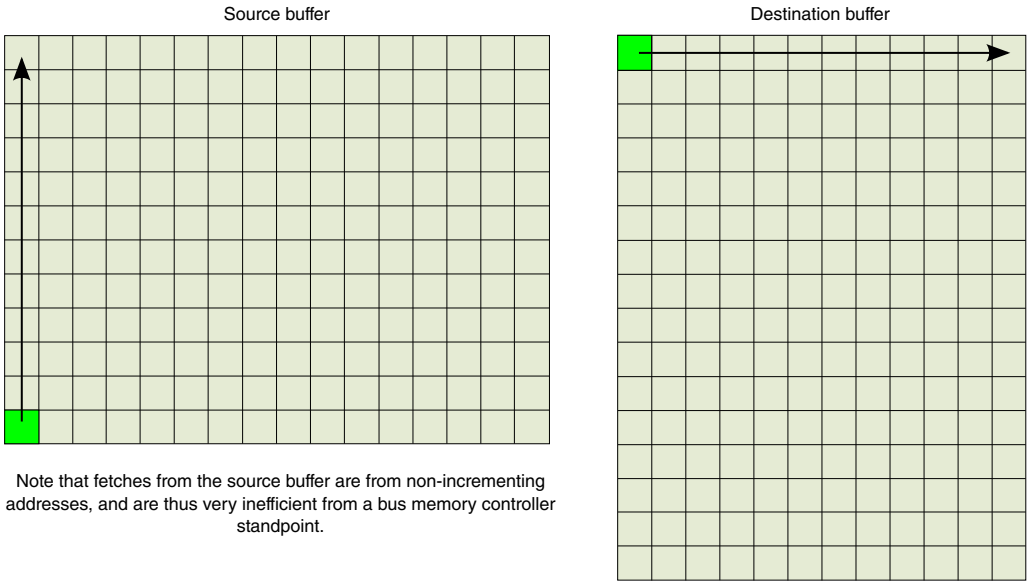


Figure 337. Rotation Read and Write

In order to rotate efficiently, multiple columns must be rotated to enable the engine to both fetch and store bursts of pixels, thus improving memory performance. The simplest method of doing this is to operate on square blocks of pixels. To rotate the image, each sub-block of pixels must be rotated by the required rotation angle.

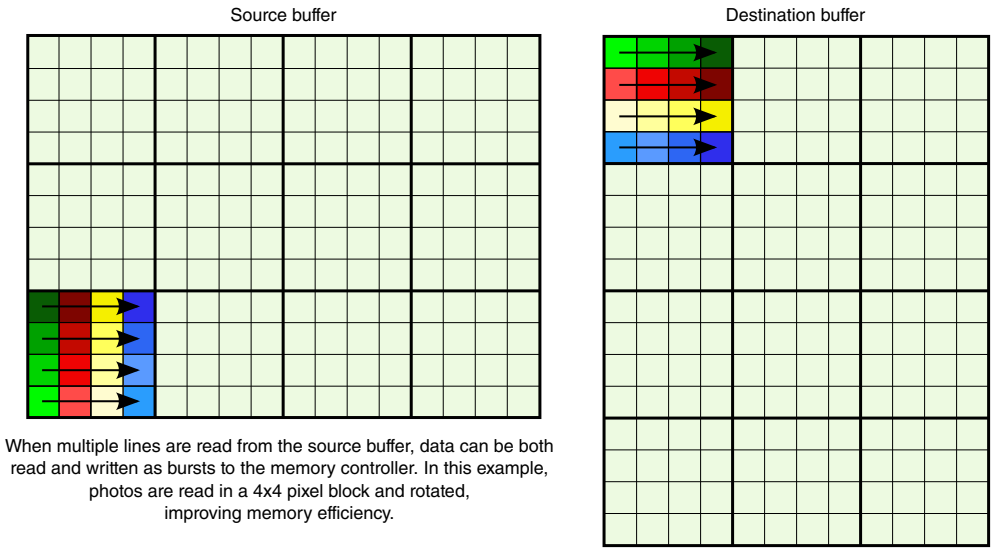


Figure 338. Rotated Sub-blocks

To manage the rotation process, the source image can be broken into a grid of sub-blocks that have coordinates as shown in Figure 339. In addition to rotating the sub-block, each block must be translated to a new coordinate location. For each of the rotation angles (0, 90, 180, 270), it is possible to define a simple algorithm for computing the new translated grid address. The hardware must then simply compute the memory address from the base grid address for both load and store operations.

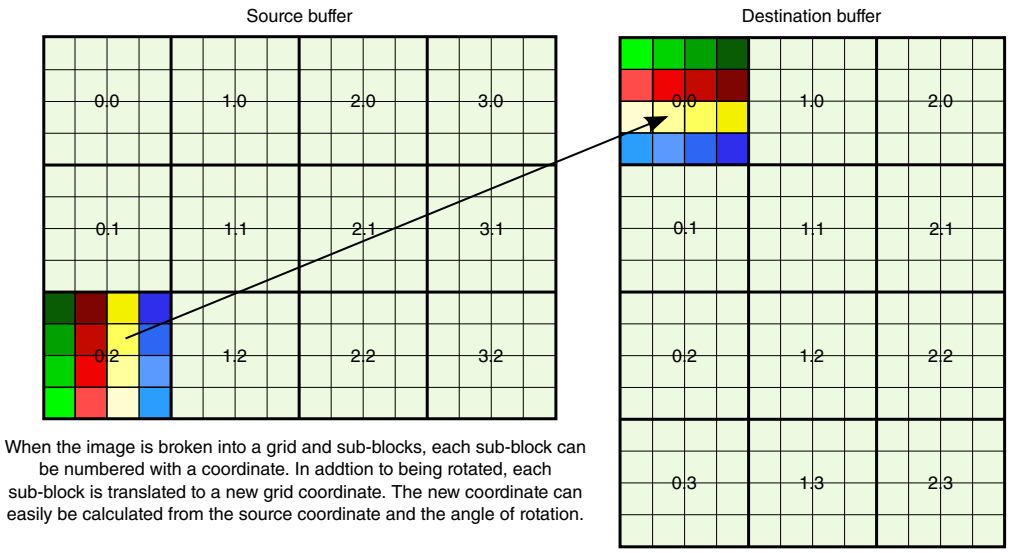


Figure 339. Grid of Sub-Blocks with Coordinates

In order to balance the requirements of reasonable burst sizes to the memory controller as well as keep the hardware storage requirements to a minimum, the blending/rotation engine will operate on either 8x8 or 16x16 pixel blocks. When using the Rotate engine with the input fetch engine, you need to program the input fetch engine to work in 8x8 block mode.

NOTE

An important artifact of PXP is when rotating a source image and the output is NOT divisible by the block size selected. The output engine essentially truncates any output pixels after the desired number of pixels has been written. Since the output buffer is written as a horizontal row of blocks, the incorrect pixels could be truncated and the final output image can look shifted. In the case where the block size is programmed to 8x8, and the output size that is programmed is 12x12, then there is a remainder of 4 pixels that will be truncated in either the X and/or Y axis when the PXP operation is complete. The output will be shifted by 4 pixels in this example. To compensate for this, the source base address needs to be adjusted so the correct pixels get truncated and the image does not look shifted. In this example, with 90 degrees of rotation, the PS base address should be adjusted by 4 times the actual PS base address -(4*pitch).

56.3.1.6 Output Buffer

The output buffer engine accepts data from PXP pixel pipeline and issues requests to transfer the output pixels to external DRAM or the internal SRAM double buffer row of blocks.

56.3.1.7 Address calculator

Each of the blocks will manage its own fetch address using a common address calculator block that computes real addresses from a base address and relative block offset from the base. Each block will then perform the multiple line fetches (or stores) required to perform the operation. This hides all the address buffer computations from the processing blocks and allows each block to simply track the coordinate of the block it is working on.

56.3.1.8 Block size selection

PXP can be configured to process blocks that are either 8x8 pixels or 16x16 pixels with CTRL[BLOCK_SIZE]. When selecting a 16x16 pixel block size, the accesses to fetch AS and PS images and write the final frame buffer are more efficient since twice as much data is requested and processed per memory request. When optimizing the system for memory bandwidth and image processing time, configure PXP to process 16x16 pixel blocks.

56.3.1.9 Interlaced Video Support

PXP has some minimal ability to generate interlaced video content from a progressive source. There two available options, based on the bandwidth requirements and how software is managing video frames. The PXP can either interlace on the input side (by reading every other line of input data) or on the output side (by writing the individual lines of video into two separate fields). Generally, output interleaving should be used since it is the most flexible mode (it allows scaling and full overlay support) and it only requires a single pass of PXP to generate two separate output fields.

Input interleaving can be beneficial in cases where PXP is running at 60fps, since it requires fewer fetches to produce the output data. There is no direct hardware support for input interleaving, in that, there is no configuration bit that can be set to alter how PXP processes a frame for input interleaving. Input interleaving is achieved by simply setting the source frame buffer pitch value to twice the value it would normally be set to for the equivalent progressive frame. The output parameters also need to be consistent with the desired processing effect. For example, the vertical resolution would be set to account for the reduced resolution to process the interlaced input buffers.

56.3.1.10 Theory of Operation

PXP can be used to accelerate graphics operations by offloading graphics processing from the processor. The block can perform alpha blending and color key substitution on two RGB graphics buffers.

PXP is organized as having a processed surface (PS) and an alpha surface (AS) that can be blended with the processed surface. There are no restrictions on the location of the AS or PS within the output surface (OS). As PXP processes NxN blocks, operations are performed on a pixel by pixel basis. The AS and PS pixels are alpha blended, color keyed, processed by CSC resources as individual pixel components. This allows efficient block processing with supporting arbitrary alignment for both the AS and PS surfaces. The resulting pixel block is then written to the corresponding block in the output buffer.

56.3.1.11 Pixel Handling

All pixels are internally represented as 32-bit values regardless of input or output pixel formats. The pixels get converted in the AS and PS buffer engines to 24-bit pixel values. There is also an 8-bit alpha value at stages up to the alpha blender within PXP for blending within the RGB color space. Compositing of AS and PS images can only occur in the RGB color space. If compositing is not required, then YUV pixels can be transferred and processed at all PXP pixel resource components. The color orientation of pixels within PXP can be controlled by the CSC1, CSC2 resources.

For RGB, input pixels are converted into -bit pixel values using the following rules for both AS and PS:

1. 32-bit ARGB8888 pixels are read directly with no conversion.
2. 32-bit RGB888 pixels are assumed to have an alpha value of 0xFF (full opaque).
3. 16-bit RGB565 and RGB555 values are expanded into the corresponding 24-bit color space and assigned an alpha value of 0xFF (opaque). The expansion process replicates the upper pixel bits into the lower pixel bits (for instance a 16-bit RGB555 triplet of 0x1F/0x10/0x07 would be expanded to 0xFF/0x84/0x39).
4. 16-bit ARGB1555 values are expanded into the corresponding 24-bit color space and assigned an alpha value of either 0x00 or 0xFF, based on the 1-bit alpha value in the pixel. The ALPHA_MULTIPLY function is useful in this scenario to allow scaling of the opaque pixels to a semi-transparent value.

Alpha values can be passed through the entire PXP data path and output in ARGB8888 and ARGB1555 pixel modes. Also, output pixels can be assigned an alpha value using OUT_CTRL[ALPHA] . 16-bit pixels values are formed from the most significant bits of the 24-bit pixel values.

When YUV/YCbCr output formats are selected, all pixels are internally represented as either RGB or YUV pixels values. The CSC2 can convert internal RGB/YUV pixels into the correct output format. In this way, any PS color space can be blended with AS RGB pixels and output in any color space.

56.3.1.12 Output Buffer Composition

The output buffer will be rendered by composing each pixel block from the associated PS and AS buffers. The AS pixel buffer can be blended or color-keyed with the associated data from the PS buffer (either the PS image pixels or PS Background Color Register (PS_BACKGROUND_0) based on PS programmed coordinates).

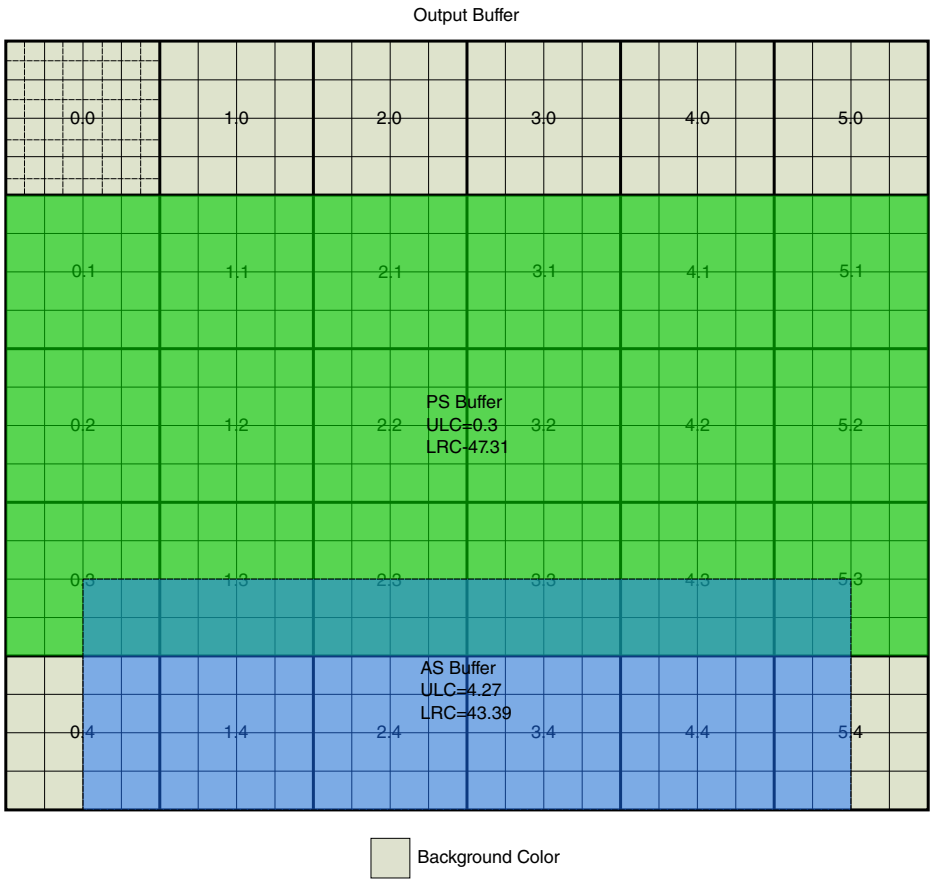


Figure 340. Output Buffer Composition

56.3.1.13 PS Image Processing

As PXP processes image buffers, it iterates over the output buffer by fetching the corresponding input buffer blocks and processing the pixels embedded in these.

56.3.1.13.1 Letterboxing

At each pixel coordinate, the control logic determines if the PS pixel (argument also applies to AS pixels) will be used in rendering the output pixel. This is determined by checking the output pixel's coordinates against Processed Surface Upper Left Coordinate Register (OUT_PS_ULC) and Processed Surface Lower Right Coordinate Register (OUT_PS_LRC) (ULC and LRC in short). For pixels outside this region, the PS pixel will be loaded with the pixel value from PS Background Color Register (PS_BACKGROUND_0) , which can be used to effectively control the letterboxing color. There are no block size or

block boundary restrictions when setting OUT_PS_ULC or OUT_PS_LRC for either the AS or PS. The only restriction is that OUT_PS_ULC and OUT_PS_LRC are within Output Surface Lower Right Coordinate Register (OUT_LRC) extents.

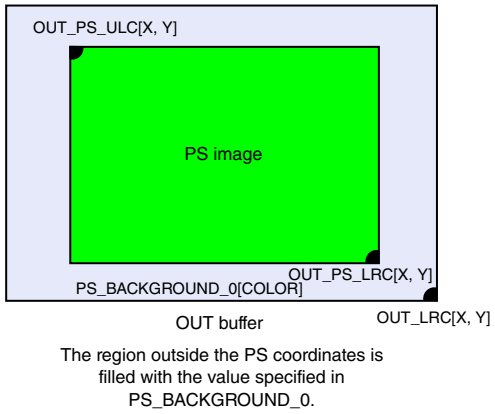


Figure 341. OUT Buffer

56.3.1.13.2 Clipping source images

A subset of the PS buffer can be used in rendering the output buffer. PS Input Buffer Address Register (PS_BUF) can indicate an offset into the PS buffer that will be used for display within the OUTPUT buffer. The pixel at the address defined in PS_BUF will be the pixel that is displayed at the pixel coordinate indicated by Processed Surface Upper Left Coordinate Register (OUT_PS_ULC) within the output buffer. Essentially, PS_BUF can be used to establish an offset into the PS buffer thus clipping all PS buffer pixels that are at a lower address. Processed Surface Pitch Register (PS_PITCH) will always indicate the number of bytes that are vertically adjacent in the PS buffer. The settings in PS Input Buffer Address Register (PS_BUF) , Processed Surface Upper Left Coordinate Register (OUT_PS_ULC) , and Processed Surface Lower Right Coordinate Register (OUT_PS_LRC) will determine the subset of the PS buffer, or clipped PS source buffer, that will be used in the output buffer.

NOTE

When scaling the PS buffer, the coordinates of the PS buffer within the output buffer need to be consistent with the scaling factors and original PS buffer size.

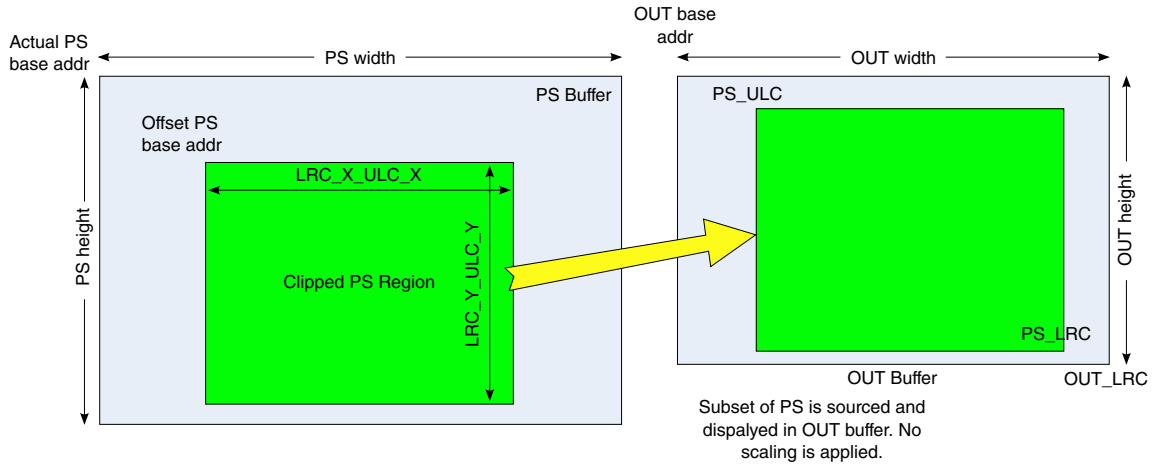


Figure 342. PS Buffer Scaling

When sourcing a subset of the PS image, it should fall completely within the PS buffer to avoid displaying incorrect data. The following conditions should be met:

x_base_addr_offset + x_scale * (LRC_X-ULC_X) <= PS_pitch

y_base_addr_offset + y_scale * (LRC_Y-ULC_Y) <= PS_size

PXP hardware does not check for these conditions and will render the image as programmed. The following case could indicate invalid programming parameters for PXP:

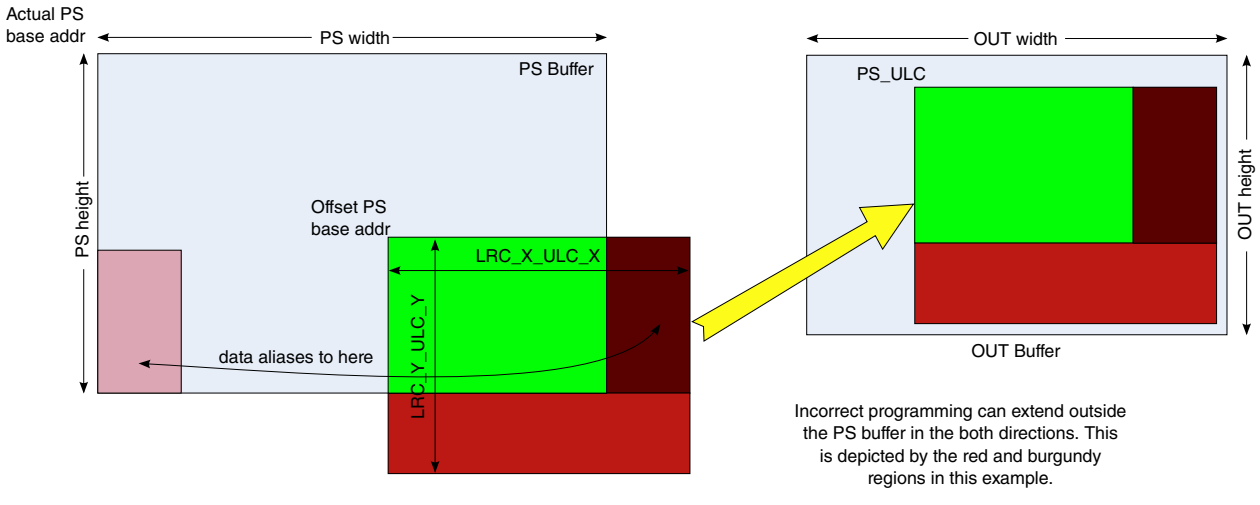


Figure 343. Example with Invalid Parameters

56.3.1.13.3 Color Key Processing

Pixels may be made transparent to the corresponding AS by using the PS color key registers. If a PS pixel matches the range specified by PS Color Key Low Register (PS_CLRKEYLOW_0) and PS Color Key High Register (PS_CLRKEYHIGH_0) , the

pixel from the associated AS will be displayed. If no AS is present for the pixel, a black pixel will be generated since the default AS pixel is 0x00000000 (transparent black pixel).

The most common use for this is when a bitmap does not support an alpha-field or for applications such as "green screen" where an image is substituted for a solid background color .

Figure 344. The PS image (player) and AS image (stadium)

The green portion of the background image can be color keyed to display the contents of the AS buffer for locations that match the color range. For this example, the color range is:

PS Colorkey: 00<R<80 70<G<ff 00<B<80

The resulting image becomes:

Figure 345. Resulting Image

56.3.1.13.4 In Place Processing (PS buffer is destination buffer)

PXP also has the ability to process an image and write the resulting buffer back to the original PS buffer. This is referred to as "in place" rendering. This could be useful for basic blit operations into the PS buffer. IN_PLACE operations are achieved by programming the OUT base address to the pixel location in the PS buffer that marks the upper left pixel of the update region. The actual region that is updated should be indicated by programming the ULC = (0,0) and the LRC = (X,Y). The region bounded by the coordinates will be updated, and the rest of the PS buffer will not be modified.

56.3.1.14 Alpha Surface (AS) Processing

The AS surface has a complete set of registers that determines how the AS effects the final OUT surface. Most of the registers that exist for the PS surface also are defined for the AS surface where applicable. This is provided to replicate the SW interface for each PS and AS processes.

56.3.1.14.1 Alpha Handling

Alpha values in the AS are embedded in the source image pixels. For AS pixel formats that do not support an alpha value, the pixel is assigned an alpha value of 0xFF (opaque).

This can be modified by the AS control by setting AS_CTRL[ALPHA_CTRL] to enable either the ALPHA_MULTIPLY or ALPHA_OVERRIDE functions. If ALPHA_MULTIPLY is enabled, the 8-bit ALPHA value from AS_CTRL is multiplied by the source alpha before blending with the PS image. If the ALPHA_OVERRIDE is set, the 8-bit ALPHA value is simply substituted for the pixel.

56.3.1.14.2 Color Key Processing (AS_CTRL)

AS_CTRL[ENABLE_COLORKEY] that can be used to enable or disable color key substitution for the AS. When enabled, the pixel values are compared to Overlay Color Key Low Register (AS_CLRKEYLOW_0) and Overlay Color Key High Register (AS_CLRKEYHIGH_0) to determine if a match has occurred. When an AS pixel matches the color key range, the pixel from the AS image is considered transparent and the corresponding PS pixel is rendered. If both the PS and AS pixels match their corresponding color key ranges, the AS pixel is displayed unmodified.

AS color keys are handled in a manner similar to PS color keys. The same images used in the PS color key example could be used with the images swapped. In this case, matches on the AS image to Overlay Color Key Low Register (AS_CLRKEYLOW_0) and Overlay Color Key High Register (AS_CLRKEYHIGH_0) would display the PS pixels.

56.3.2 Output Image Processing

Several PXP options affect the resulting output image.

56.3.2.1 Output Image Size

PXP generates an output image in the resolution programmed by Output Surface Lower Right Coordinate Register (OUT_LRC) . As PXP processes pixels, it iterates over the NxN blocks (in output scan-block order) based on the final image resolution.

56.3.2.2 Output Format

The result of PXP operations are written to the buffer pointed to by Output Frame Buffer Pointer Register (OUT_BUF) or Output Frame Buffer Pointer #2 Register (OUT_BUF2) . The pixel format is controlled by OUT_CTRL[FORMAT] . 32-bit pixels are formed directly from the internal 24-bit representations and 16-bit pixel formats are generated by truncating the internal 24-bit values to the appropriate number of bits. For formats supporting an alpha value, PXP assigns the alpha using the 8-bit value in OUT_CTRL[ALPHA] . For ARGB1555, the most significant alpha bit is appended to the output pixel. Also, for ARGB4444, the most significant nibble is appended to the output pixel. Single and dual buffer YUV output formats are also available. Since each pixel in the data path is represented by a full YUV444 24bpp value, decimation reduces the output in cases of YUV422/420 output formats.

56.3.2.3 Rotation/Flip operations

PXP supports four rotation angles in conjunction with vertical and horizontal flip options. The flip operations effectively take place before the rotation. Rotations of 0, 90, 180, and 270 degrees are supported and any combination of rotation and flip are supported. There is no performance difference between any of these modes of operation.

56.3.3 Queuing PXP transactions

PXP supports a primitive ability to queue up one operation while the current operation is running. This is enabled through the use of Next Frame Pointer Register (NEXT) . When this register is written, it enables PXP to reload its current register contents with the data found at the location pointed to by this address when it completes processing of the current frame. This feature may be useful in helping to reduce the interrupt latency in servicing PXP, especially in cases where PXP is using the on-chip SRAM buffer handshake (since PXP must begin generating next frame data immediately).

If PXP is idle when NEXT is written, PXP treats this as an indication that it should immediately load the values at the pointer and begin processing the frame. This ability should allow software to use the same routines when programming PXP (so that the first frame doesn't differ from subsequent frames).

When loading values from NEXT, all registers in PXP are reloaded. Some register loads have no effect.

After writing NEXT, PXP will set NEXT[ENABLED] to indicate that the next command has been queued. Software should first check the status of this bit to ensure that a previous command has not been enabled. Likewise, after programming the first frame in a sequence of frames, software should poll this bit until it is sampled logic 1'b0 before queuing the next operation.

PXP will issue interrupts from frames as they complete, regardless of whether they were started by writing the control registers directly or using NEXT. When software receives an interrupt, it should check/clear PXP's status register as normal, poll NEXT[ENABLED], and then issue the next operation. A queued operation may be cancelled by issuing a CLEAR operation to NEXT[ENABLED]. The SET and TOGGLE operations should never be used with this register.

56.3.4 Error Handling

PXP does minimal checking on the control registers, so it is important that these are correctly specified. PXP monitors the bus transactions for errors and reports the errors in Status Register (STAT) . Upon receipt of a bus error, PXP sets the ERROR interrupt and stops any further operations. Bus errors can be generated from any system access that results in an error response returned from the internal SIM Bus errors in PXP are signaled as either a read or a write error, but do not indicate the failing address. Software may deduce the failing address from the current block status indicators.

56.3.4.1 Known PXP Limitations/Issues

PXP has the following known limitations:

1. When using Next Frame Pointer Register (NEXT) , the interrupt enable setting should remain the same for all frames. If not, PXP will change the interrupt enable register value and possible cause the loss of an interrupt.

56.3.5 PXP Store Engine Block Description

56.3.5.1 Overview

The essential function of the Store Engine is to store input data from the input sub-block to memory or directly output to next sub-block. The input source is 32 bits per channel per valid clock. There is one instance of the Store Engine in PXP. The main supported features are:

- Two DMA controllers thus it can work on two channel data storage at the output at the same time. If there is only one stream data input, store engine should use channel 0 but not channel1.
- Accepts as input up to 8 bytes data (two channels, 32 bits each) and 8 bits flag input. After shift operation it can store up to 64bpp per valid clock to memory.
- Data fill function for initialization. It can fill a fixed 32 bit data value to a specified memory buffer.
- Handshake mode with a downstream connected data Fetch Engine. It supports 1 line, 4 lines, 8 lines and 16 lines scan handshake with and 1 line, 3 lines and 5 lines array handshake modes. The Store Engine signals to the Fetch Engine when data is available for processing thus eliminating the need for software intervention through interrupt service routines.
- Bypass mode which directly outputs the data to the downstream connected Fetch Engine after shift operation. This is done through a valid/ready data interface.
- Support for 8x8 and 16x16 block mode and scanline mode.
- Support for 8, 16, 32 and 64pp input data format.

56.3.5.2 Top-Level Architecture

Figure 346 shows the high level architecture of the block. The basic flow is as follows:

1. Interface with driving sub-block, read in and synchronize data channels.
2. Shift data function swizzles data fields as specified.
3. Data is steered to the output channels and packed for output.

4. Data is output to memory through the AXI masters or sent to receiving sub-block through the bypass interface.

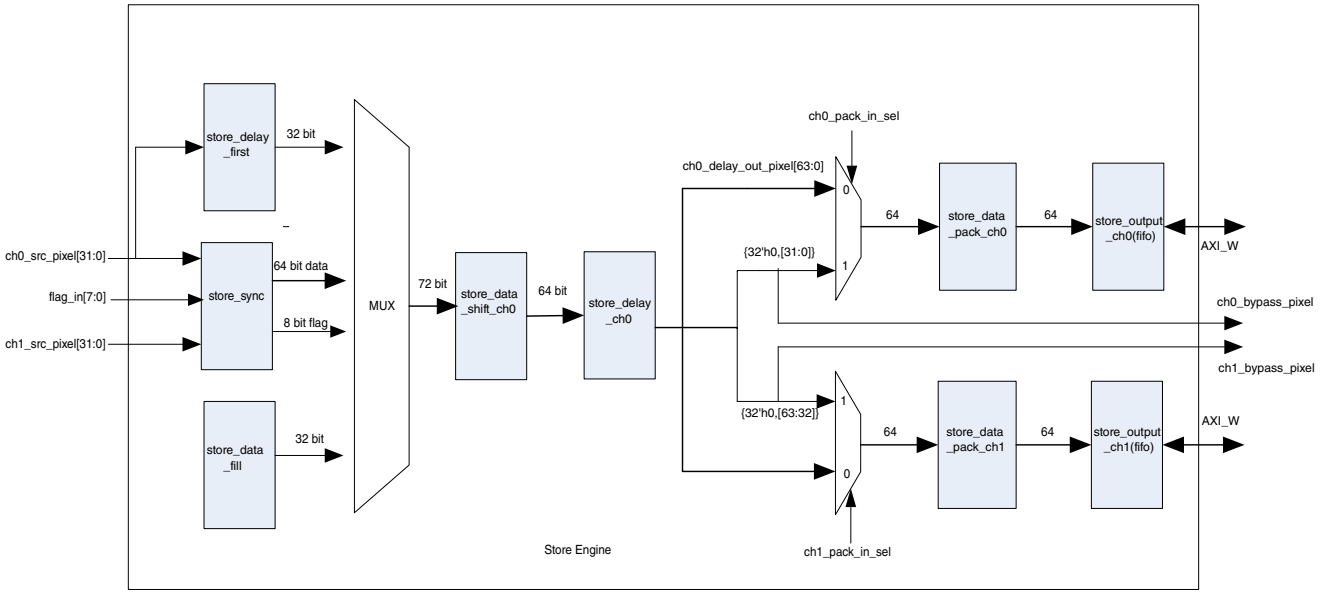
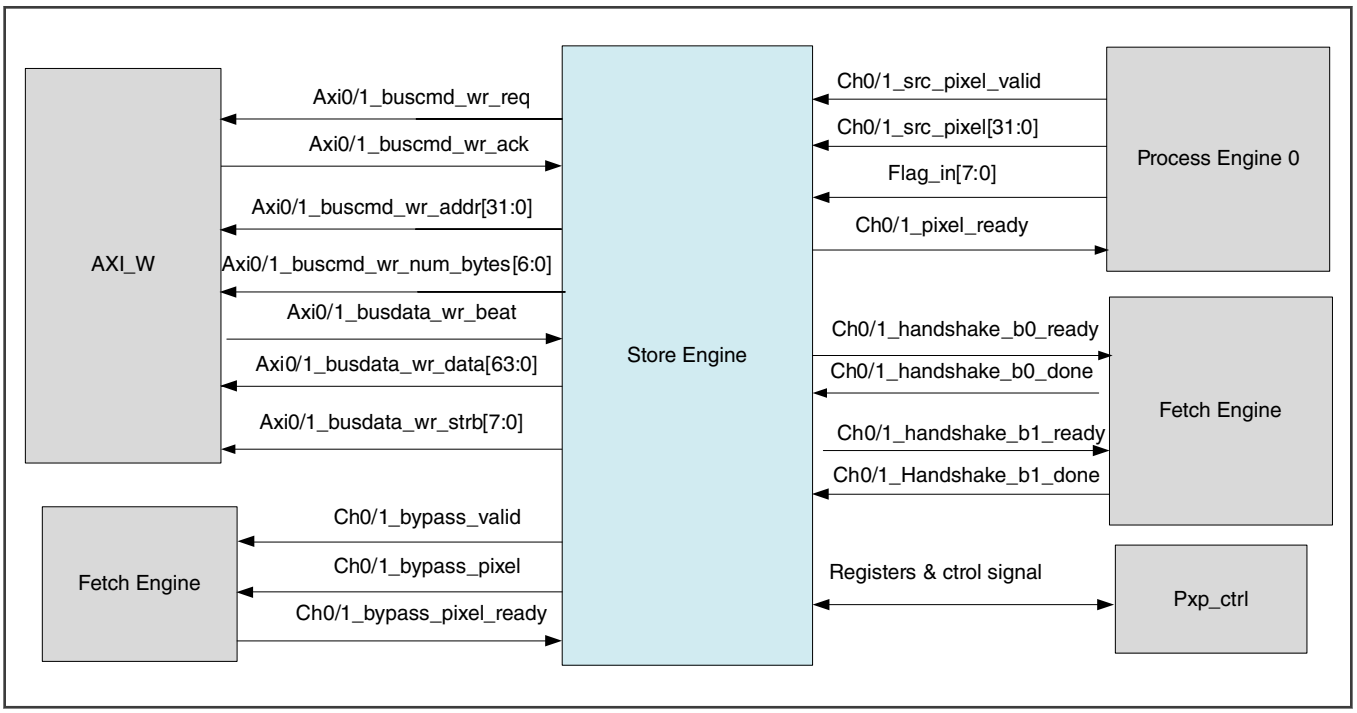


Figure 346. High Level Block Diagram

The following figure shows the Store Engine supported connections. It can output data to memory through the AXI arbiter. The block is configured and controlled through software programming.



56.3.5.3 Store Engine Design

56.3.5.3.1 Input Data Source

1. **Fixed Data Input**

   The Store Engine supports a fill function which writes a fixed value to a specific set of memory locations. The data received from the input is replaced by INPUT_STORE_FILL_DATA_CH0 . This function is usually used to initialize a memory buffer. The value in INPUT_STORE_FILL_DATA_CH0 is the initialized value stored to the memory buffer. Also, this mode can only work through channel 0 and not channel 1.

   In this mode, INPUT_STORE_CTRL_CH0[FILL_DATA_EN] = 1. The fixed value is set in INPUT_STORE_FILL_DATA_CH0 . How many bits are valid from INPUT_STORE_FILL_DATA_CH0 is decided by INPUT_STORE_SHIFT_CTRL_CH0[OUTPUT_ACTIVE_BPP] .

2. **Normal Data Input**

   Store Engine supports two 32 bits wide data channels and 8 bits flag input. If there is only one channel of data input, the channel 0 must be used not channel 1. In this case, INPUT_FETCH_CTRL_CH0[COMBINE_2CHANNEL], and INPUT_STORE_CTRL_CH0[COMBINE_2CHANNEL] = 0.

   If both channels and 8 bits flag input contain valid data, then INPUT_FETCH_CTRL_CH0[COMBINE_2CHANNEL], and INPUT_STORE_CTRL_CH0[COMBINE_2CHANNEL] should be set to 1. The Store Engine will then combine the two channels of data into 64 bits of data send them with the 8 bits of flags to the shift module. After shift, the Store Engine will choose the active bits from the 64 bit output according to INPUT_STORE_SHIFT_CTRL_CHn[OUTPUT_ACTIVE_BPP] and then pack them in preparation for output.

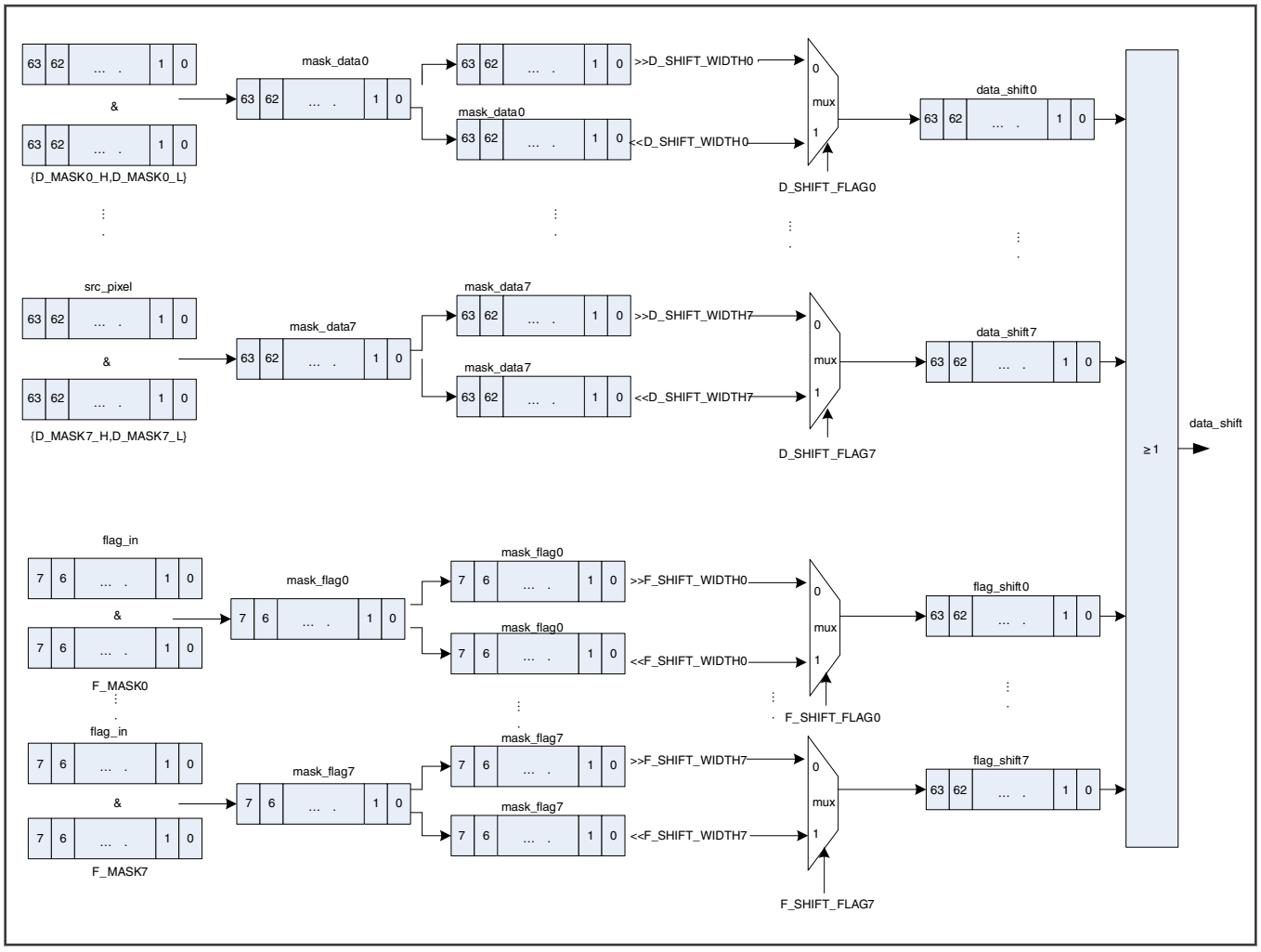
56.3.5.3.2 Store data shift operation

The data shift operation supports up to 8 data fields and 8 flag components to be shifted to any position within the 64 bit word. The data fields can be any width. This function is a way to reorder or reformat the component parts of the 64 bit word output. As data flows through the Store Engine each data word is processed the same way according to register configuration.

The following registers are used in shift operation. The pairs of (INPUT_STORE_D_MASKn_H_CH0, and INPUT_STORE_D_MASKn_L_CH0) make up the 64 bits of mask for each MASKn component. There are a set of masks for each channel in each instance. INPUT_STORE_D_SHIFT_L_CH0[D_SHIFT_FLAGn], INPUT_STORE_D_SHIFT_H_CH0[D_SHIFT_FLAGn], INPUT_STORE_F_SHIFT_L_CH0[F_SHIFT_FLAGn], INPUT_STORE_F_SHIFT_H_CH0[F_SHIFT_FLAGn] specify which direction to shift (0: right shift 1: left shift) each data component0 ~ component7 after the mask is applied. INPUT_STORE_D_SHIFT_L_CH0[D_SHIFT_WIDTHn], INPUT_STORE_D_SHIFT_H_CH0[D_SHIFT_WIDTHn] specify how many bits to right or left shift each data component 0 through7 after the mask is applied.

The shift function works the same for the 8 bits of flag data. There are analogous “F_SHIFT” registers that contain INPUT_STORE_F_SHIFT_L_CH0[F_MASKn], INPUT_STORE_F_SHIFT_H_CH0[F_MASKn], INPUT_STORE_F_SHIFT_L_CH0[F_SHIFT_FLAGn], INPUT_STORE_F_SHIFT_H_CH0[F_SHIFT_FLAGn] and INPUT_STORE_D_SHIFT_L_CH0[F_SHIFT_WIDTHn], INPUT_STORE_D_SHIFT_H_CH0[F_SHIFT_WIDTHn] for swizzling the input flags.

The following figure shows how the shift function works.



56.3.5.3.3 Data Packing

After data shift, the 64 bits of data can be divided to two 32bit words depending on the value of INPUT_FETCH_CTRL_CHn[PACK_IN_SEL], and INPUT_STORE_CTRL_CHn[PACK_IN_SEL]. If INPUT_FETCH_CTRL_CH0[PACK_IN_SEL], and INPUT_STORE_CTRL_CH0[PACK_IN_SEL] = 0, it will select active bits from all 64 bits of data. If equal to 1, it will select active bits from the lower 32 bits data.

For channel 1, if INPUT_FETCH_CTRL_CHn[PACK_IN_SEL], and INPUT_STORE_CTRL_CHn[PACK_IN_SEL] = 0, it will also select active bits from all 64 bits data and if equal to 1, it will select active bits from the higher 32 bits data.

Store Engine supports 8bpp, 16bpp, 32bpp and 64bpp active bits per pixel according to INPUT_STORE_SHIFT_CTRL_CHn[OUTPUT_ACTIVE_BPP]. When either channel's INPUT_FETCH_CTRL_CHn[PACK_IN_SEL], and INPUT_STORE_CTRL_CHn[PACK_IN_SEL] = 1 only 8bpp, 16bpp, 32bpp can be selected.

After shift and data output selection, the packing module will pack the data into a 64 bit string and store to a FIFO before writing out to memory.

Packing style according to active bpp is showed below.

OUTPUT_ACTIVE_BPP	Active bpp from 64 shift output data	Packed data
0	8 bits	{pix7, pix6, pix5, pix4, pix3, pix2, pix1, pix0}
1	16 bits	{pix3, pix2, pix1, pix0}
2	32 bits	{pix1, pix0}
3	64 bits	{pix0}

56.3.5.3.4 Data Store Format

- **Scanline mode**

  When INPUT_FETCH_CTRL_CHn[BLOCK_EN], and INPUT_STORE_CTRL_CHn[BLOCK_EN] = 0, the Store Engine will store the data line by line in scanline mode. The AXI burst length is set by INPUT_FETCH_CTRL_CHn[WR_NUM_BYTES], and INPUT_STORE_CTRL_CHn[WR_NUM_BYTES]. In Scanline mode, it can support 8, 16, 32 and 64 bytes in a burst.

- **Block mode**

  When INPUT_FETCH_CTRL_CHn[BLOCK_EN], and INPUT_STORE_CTRL_CHn[BLOCK_EN] = 1, the Store Engine supports 8x8 and 16x16 block mode according to INPUT_FETCH_CTRL_CHn[BLOCK_16], and INPUT_STORE_CTRL_CHn[BLOCK_16]. When INPUT_FETCH_CTRL_CHn[BLOCK_16], and INPUT_STORE_CTRL_CHn[BLOCK_16] = 1, the block size is 16*16 while if INPUT_FETCH_CTRL_CHn[BLOCK_16], and INPUT_STORE_CTRL_CHn[BLOCK_16] = 0 the size is 8*8 blocks. When INPUT_FETCH_CTRL_CHn[BLOCK_EN], and INPUT_STORE_CTRL_CHn[BLOCK_EN] = 1, then INPUT_STORE_SHIFT_CTRL_CHn[OUTPUT_ACTIVE_BPP] cannot be 3, that is, block mode can not work on a 64bpp data stream.

  When the Store Engine is configured block mode, the upstream Fetch Engine should be configured in block mode and with the same block size as well.

  When the total width or total height of image is no divisible by 8 or 16, the upstream Fetch Engine will fetch extra pixels or lines to fill out the final partial blocks. For example, if the size is 23*23, and block mode is 8*8, Fetch Engine will fetch 24*24 data pixels and output to store engine. The Store Engine has a crop function for this. That is, the extra pixels or lines will not be stored to memory by the Store Engine. This is accomplished by the hardware using the AXI bus byte enables.

  In block mode, the number of AXI bus write bytes is internally calculated from INPUT_STORE_SHIFT_CTRL_CHn[OUTPUT_ACTIVE_BPP] and INPUT_FETCH_CTRL_CHn[BLOCK_16], and INPUT_STORE_CTRL_CHn[BLOCK_16] (block size) rather than INPUT_FETCH_CTRL_CHn[WR_NUM_BYTES], and INPUT_STORE_CTRL_CHn[WR_NUM_BYTES].

  One limitation exists with block mode configuration. No YUV422 output format is not supported. So INPUT_STORE_SHIFT_CTRL_CHn[OUT_YUV422_1P_EN] and INPUT_STORE_SHIFT_CTRL_CHn[OUT_YUV422_2P_EN] must be disabled.

56.3.5.3.5 Output Format Modes

- **Normal mode**

  If INPUT_STORE_CTRL_CHn[STORE_BYPASS_EN] = 0,

  INPUT_STORE_CTRL_CHn[STORE_MEMORY_EN] = 1 and

  INPUT_STORE_CTRL_CHn[HANDSHAKE_EN] = 0, the Store Engine is working in normal mode.

  In normal mode, the Store Engine will store the input data to memory frame by frame. The store frame start address is set in INPUT_STORE_ADDR_n_CH0, INPUT_STORE_ADDR_n_CH1 . Two address registers exist for each channel. The second one is used for YUV422 2 plane mode.

The Store Engine will stop storing data when the frame of data is finished. Another pxp_start signal will trigger the next frame operation. The frame size is specified by INPUT_STORE_SIZE_CHn. The values should be set up for each channel in use. If both channels are being used, the height and width values in INPUT_STORE_SIZE_CH0 and INPUT_STORE_SIZE_CH1 may be different but the total number of pixels in the frame must be the same. The only exception to this is when both channels are being used and the Store Engine is outputting data in handshake mode. In handshake mode the width and height must be set the same in both channel size registers.

If the data shift mode is set to YUV422 single plane or 2 plane, two pixels, 64 bits, are read in and 32 bits are written. If the YUV data is coming from the upstream Fetch Engine it will be in YUV444 format as {8'h0, Y0, U0, V0, 8'h0, Y1, U1, V1}. Even though the YUV422 mode is set, the shift function must be set to shift the components into their proper position. For instance, for 1p mode, the shift MASK, FLAG and WIDTH parameters need to be set up on the 64 bits of pixel data to format each 32 bits (2 pixels) to be written out as {Y0, U0, Y1, V0}. Likewise in 2p mode, the ch0 data would be shifted to yield {Y0, Y1} with ch1 as {U0, V0} for the first two pixels.

- **Bypass mode**

  If INPUT_STORE_CTRL_CHn[STORE_BYPASS_EN] = 1, INPUT_STORE_CTRL_CHn[STORE_MEMORY_EN] = 0, the Store Engine will output the input data, after the shift function, directly to the downstream Fetch Engine instead of storing to memory. The AXI bus is inactive in this case. The data is output using the valid/ready external interface to the downstream connected Fetch Engine.

  In bypass mode, if INPUT_STORE_CTRL_CH0[CH_EN] = 1, it will output the lower 32 bits of data. If INPUT_STORE_CTRL_CH1[CH_EN] = 1, it will output the high 32 bits of data.

- **Dual mode**

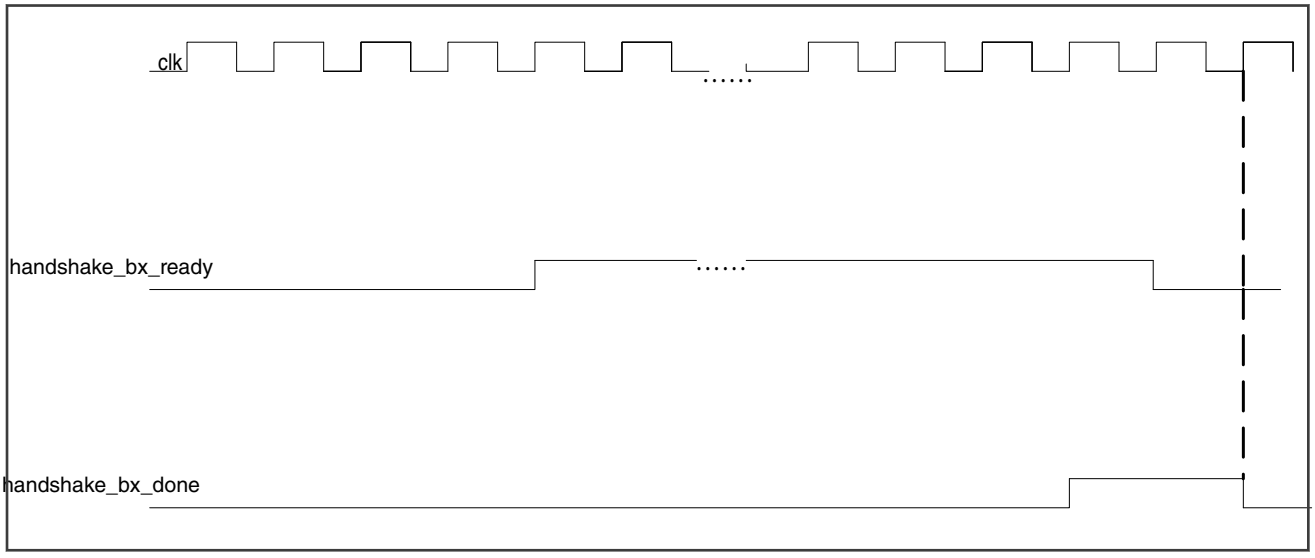
  If STORE_BYPASS_EN = 1 and INPUT_STORE_CTRL_CHn[STORE_MEMORY_EN] = 1, the Store Engine is configured in Dual mode. That is, the Store Engine supports bypassing the data and also storing it to memory at the same time.

- **Handshake mode**

  The array handshake is used in cases where the Fetch Engine is in array mode. One line handshake config indicates that the Fetch Engine is working on 1x1 array mode. The 3 line handshake indicates the Fetch Engine work is in 3x3 array while 5 line handshake indicates 5x5 array mode.

  To accomplish the buffer sharing, Store Engine and the attached Fetch Engine will maintain buffer status using a pair of handshake signals. When a buffer is filled by the Store Engine, it will assert the handshake_bx_ready (where x is 0 or 1) signal to indicate to the fetch engine that the buffer has valid data. The prefetch engine will then release the buffer by asserting the handshake_bx_done signal.

  The basic protocol and timing is shown in the following figure:



Unlike scanline handshake, there is only 1 buffer in memory. The buffer size is (array_line + 1).

— 1 line array handshake

When INPUT_STORE_CTRL_CHn[ARRAY_LINE_NUM] = 0, the configuration is 1 line handshake mode. The buffer size is 2 * OUT_PITCH (2 lines). The first ready occurs when the Store Engine finishes writing 1 line.

— 3 line array handshake

When INPUT_STORE_CTRL_CHn[ARRAY_LINE_NUM] = 1, the configuration is 3 line handshake mode. The buffer size is 4 * OUT_PITCH. The first ready occurs when the Store Engine finishes writing 2 lines.

— 5 line array handshake

When INPUT_STORE_CTRL_CHn[ARRAY_LINE_NUM] = 2, the configuration is 5 line handshake mode. The buffer size is 6*OUT_PITCH. The first ready occurs when store engine finishes writing 3 lines.

56.3.5.3.6 Limitations

When both channels are active, they should be configured to work in the same mode. For example, if channel 0 is configured for bypass mode then channel 1 must also be configured the same way.

56.3.6 PXP Fetch Engine Block Description

56.3.6.1 Overview

The main function of Fetch Engine is to fetch data from memory or directly from the input bypass channel and output to another downstream module as 32 bits per pixel per valid clock.

The Fetch Engine has two channel DMA controllers and thus can fetch two memory data buffers at the same time. The Fetch Engine supports an expansion and shift function on the fetched data and then output as 32 bits per pixel according to different data format. It supports 8x8 or 16x16 block mode and scanline mode. The block also supports rotation function with rotation module when in block mode. It can support vertical flip, horizontal flip, 90/180/270 degree rotation angle.Using the expansion function, the user can read in rgb555, rgb565, rgb444, argb1555, argb4444, yuv422_2p or yuv422_1p input data format and convert to argb8888, rgb888 or yuv444 format at the output. The Fetch Engine can interface with a connected Store Engine block directly in one of two ways, handshake mode and bypass mode as described further below. In handshake mode 1 line, 8 line and 16 line configurations are possible. In bypass mode memory accesses are bypassed altogether and data passes directly from a Store Engine to its connected Fetch. Input formats of 8bpp, 16bpp, 32bpp, 64bpp are supported.

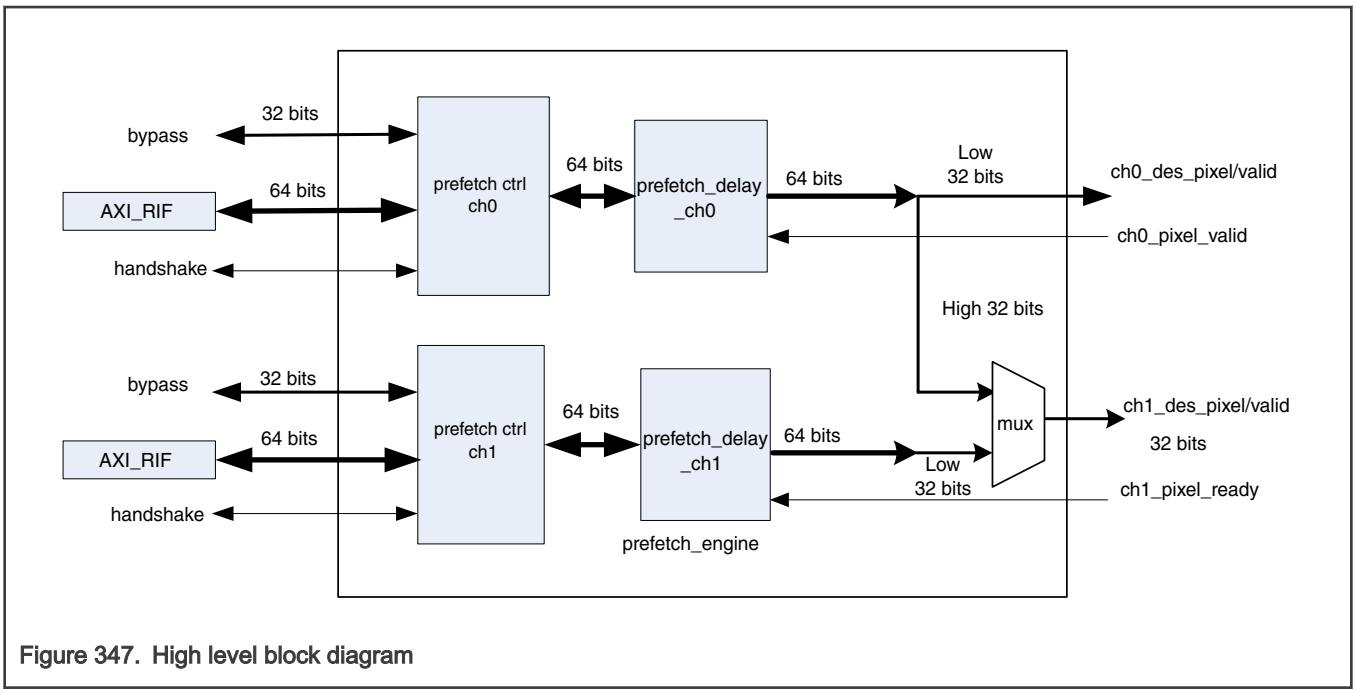


Figure 347. High level block diagram

Figure 348 shows the internal structure of the Fetch_ctrl engine.

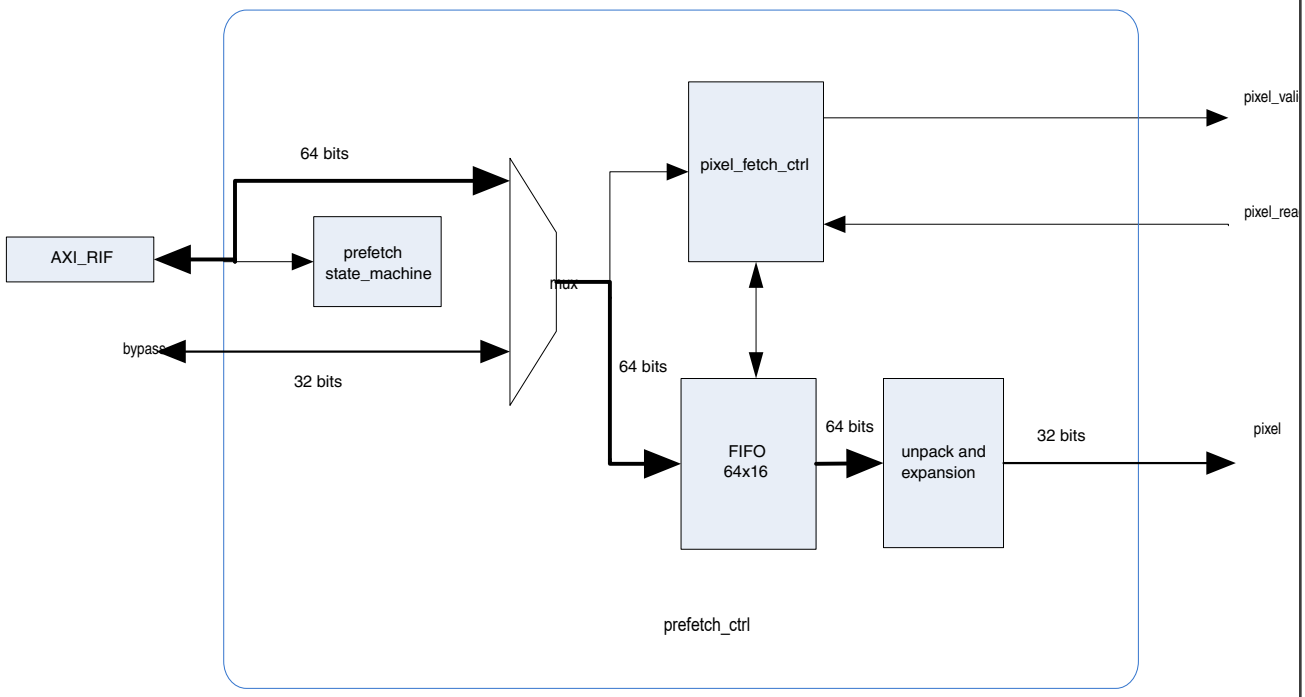


Figure 348. Fetch Control block diagram

56.3.6.2 Fetch Data formats

The Fetch Engine supports input 8bpp, 16bpp, 32bpp, 64bpp data format according to INPUT_FETCH_SHIFT_CTRL_CHn[INPUT_ACTIVE_BPP] .

When input data is 64bpp, since output data bus is 32 bits per channel, the Fetch Engine will need to be configured to use both output channels (32 bits each) to transmit the 64 bits of data at the output. Thus, when input data is 64bpp, the Fetch Engine supports only channel 0 data input but not channel 1 at the same time.

The Fetch Engine supports an expansion function. When you set INPUT_FETCH_SHIFT_CTRL_CH0[EXPAND_EN], INPUT_FETCH_SHIFT_CTRL_CH1[EXPAND_EN] = 1, the engine will change the input format to a unified YUV444 or ARGB888/RGB888 format. Otherwise it will remain in the same format and output to an external module. Note the expand function is not available if the input data is YUV422 format, 1 plane or 2 plane data fetching is supported. When EXPAND_FORMAT=7, prefetch_engine will fetch YUV422 format data from two plane. One is for Y, the other is for UV. The start address of these two buffer planes are set by INPUT_BASE_ADDR_0 (Y buffer) and INPUT_BASE_ADDR_1 (UV buffer) in INPUT_FETCH_ADDR_n_CH0, INPUT_FETCH_ADDR_n_CH1 for that Fetch instance. Also, if input format is YUV422_2P, INPUT_FETCH_SHIFT_CTRL_CH0[EXPAND_EN], INPUT_FETCH_SHIFT_CTRL_CH1[EXPAND_EN] must always be asserted. This combination is only used to transform YUV422_2P to YUV444 format.

The data format in memory and output is shown in Table 545. For input formats that read multiple pixels within the 64 bit input word, there will be multiple output cycles for each input read to expand and output all the pixels in the input word.

X means this parameter is a “don’t care”.

Table 545. Fetch data format in memory and output

INPUT_ACTIVE_BPP	Input data valid bits(bpp)	Data store in memory	EXPAND_EN	EXPAND_FORMAT	After expanding(32 bit word)	Data output format in 32 bit bus
0	8 bits	{pix7, pix6, pix5, pix4, pix3, pix2, pix1, pix0}	0	Any value except 7	NA	{24'h0, pix0}
1	16bits	{pix3, pix2, pix1, pix0}	0	Any value except 7	NA	{16'h0, pix0}
2	32bits	{pix1, pix0}	0	Any value except 7	NA	{pix0}
3	64bits	{pix0}	X	X	NA	Channel 0 bus output {pix0[31:0]} , Channel 1 bus output {pix0[63:32]}
1	16bits	{pix3, pix2, pix1, pix0}	1	0 (rgb565)	{8'hff,rgb888}	{pix0}
1	16bits	{pix3, pix2, pix1, pix0}	1	1 (rgb555)	{8'hff,rgb888}	{pix0}
1	16bits	{pix3, pix2, pix1, pix0}	1	2 (argb1555)	{8{a}},rgb888 }	{pix0}
1	16bits	{pix3, pix2, pix1, pix0}	1	3 (rgb444)	{8'hff,rgb888}	{pix0}
1	16bits	{pix3, pix2, pix1, pix0}	1	4 (argb4444)	{ {2{a}}, rgb888 }	{pix0}
1	16bits	{pix3, pix2, pix1, pix0}	1	5 (yuyv/yvyu)	{8'hff,yuv444} /{8'hff,yvu444}	{pix0}
1	16bits	{pix3, pix2, pix1, pix0}	1	6 (uyvy/vyuy)	{8'hff,yuv444} /{8'hff,yvu444}	{pix0}
x	Any value except 64bits	Address0 {Y7,Y6,Y5,Y4,Y3,Y2,Y1,Y0} Address1 {U6V6U4V4U2V2U0V0}	1	7 (yuv422_2p)	{8'hff,yuv444}	{pix0}

NOTE

In 64 bpp mode, the expand and shift functions are not supported.

56.3.6.3 Data Fetching Format

- **Scanline mode**

  When INPUT_FETCH_CTRL_CHn[BLOCK_EN] = 0, the Fetch Engine will fetch data line by line using scanline mode. The burst length can be set by INPUT_FETCH_CTRL_CHn[RD_NUM_BYTES]. In Scanline mode, it can support 8, 16, 32 and 64 bytes for a burst. The values INPUT_FETCH_CTRL_CHn[RD_NUM_BYTES] = 0,1,2,3 indicates 8,16,32,64 bytes per burst.

- **Block mode**

  When INPUT_FETCH_CTRL_CHn[BLOCK_EN] = 1, Fetch Engine support 8*8 and 16*16 block mode according to INPUT_FETCH_CTRL_CHn[BLOCK_16], and INPUT_STORE_CTRL_CHn[BLOCK_16]. When INPUT_FETCH_CTRL_CHn[BLOCK_16], and INPUT_STORE_CTRL_CHn[BLOCK_16] = 1 the block size is 16*16. When INPUT_FETCH_CTRL_CHn[BLOCK_16], and INPUT_STORE_CTRL_CHn[BLOCK_16] = 0, it is 8*8 block mode. When INPUT_FETCH_CTRL_CHn[BLOCK_EN] = 1, INPUT_FETCH_SHIFT_CTRL_CHn[INPUT_ACTIVE_BPP] cannot be 3, that is, block mode cannot work on a 64bpp data stream.

  When the total width or total height of image is not divisible by 8, 16 (according to block size) it will fetch extra pixels or lines to compensate to a complete block. For example, if the size is 23*23, and block mode is 8*8, Fetch Engine will fetch 24*24 data pixels and output to store engine. Store engine has a crop function for this.

  In block mode, AXI bus read number bytes is decided by input format and block size and overrides INPUT_FETCH_CTRL_CHn[RD_NUM_BYTES] .

56.3.6.4 Fetch Data Shift Function

Fetch Engine supports a shift operation that will swizzle register defined data fields in the incoming data words. The post-shifted data is then output to the proper channel. Up to 4 component shift operations are supported.

The shift function can be disabled by asserting INPUT_FETCH_SHIFT_CTRL_CH0[SHIFT_BYPASS], INPUT_FETCH_SHIFT_CTRL_CH1[SHIFT_BYPASS] . This function is bypassed by default.

The shift is configured with INPUT_FETCH_SHIFT_OFFSET_CH0[OFFSETn], INPUT_FETCH_SHIFT_OFFSET_CH1[OFFSETn] , INPUT_FETCH_SHIFT_WIDTH_CH0[WIDTHn], INPUT_FETCH_SHIFT_WIDTH_CH1[WIDTHn] . Each channel is controlled separately. INPUT_FETCH_SHIFT_OFFSET_CH0[OFFSETn], INPUT_FETCH_SHIFT_OFFSET_CH1[OFFSETn] specify the starting position of the component fields in the incoming data word. INPUT_FETCH_SHIFT_WIDTH_CH0[WIDTHn], INPUT_FETCH_SHIFT_WIDTH_CH1[WIDTHn] specify the width of that component. There is register support for 4 component fields for each channel. The specified fields are shifted in order and packed to the least significant part of the data word. The unspecified bits in the input word are discarded. If there are any unspecified bits in the most significant part of the output word they are zeroed out.

Not all fields must be specified. For example if field number three is not used then WIDTH3 should be left with a value of 0.

A few examples are shown in Figure 349. In the example in the left part of Figure 349, data format of YUV444 is swizzled to YVU444. The fields should be set as follows:

INPUT_FETCH_SHIFT_OFFSET_CH0[OFFSET0], INPUT_FETCH_SHIFT_OFFSET_CH1[OFFSET0] = 8,
INPUT_FETCH_SHIFT_WIDTH_CH0[WIDTH0], INPUT_FETCH_SHIFT_WIDTH_CH1[WIDTH0] = 8,
INPUT_FETCH_SHIFT_OFFSET_CH0[OFFSET1], INPUT_FETCH_SHIFT_OFFSET_CH1[OFFSET1] = 0,
INPUT_FETCH_SHIFT_WIDTH_CH0[WIDTH1], INPUT_FETCH_SHIFT_WIDTH_CH1[WIDTH1] = 8,
INPUT_FETCH_SHIFT_OFFSET_CH0[OFFSET2], INPUT_FETCH_SHIFT_OFFSET_CH1[OFFSET2] = 16,
INPUT_FETCH_SHIFT_WIDTH_CH0[WIDTH2], INPUT_FETCH_SHIFT_WIDTH_CH1[WIDTH2] = 8,
INPUT_FETCH_SHIFT_OFFSET_CH0[OFFSET3], INPUT_FETCH_SHIFT_OFFSET_CH1[OFFSET3] = 0,
INPUT_FETCH_SHIFT_WIDTH_CH0[WIDTH3], INPUT_FETCH_SHIFT_WIDTH_CH1[WIDTH3] = 0

The result switches the position of the 8-bit U and V components. The Y component remains in the same place and the high byte remains unchanged.

In the example in Figure 349, right, we are swizzling the data format of ARGB888 to BGRA888. The register settings are:

INPUT_FETCH_SHIFT_OFFSET_CH0[OFFSET0], INPUT_FETCH_SHIFT_OFFSET_CH1[OFFSET0] = 24,
INPUT_FETCH_SHIFT_WIDTH_CH0[WIDTH0], INPUT_FETCH_SHIFT_WIDTH_CH1[WIDTH0] = 8,
INPUT_FETCH_SHIFT_OFFSET_CH0[OFFSET1], INPUT_FETCH_SHIFT_OFFSET_CH1[OFFSET1] = 16,
INPUT_FETCH_SHIFT_WIDTH_CH0[WIDTH1], INPUT_FETCH_SHIFT_WIDTH_CH1[WIDTH1] = 8,
INPUT_FETCH_SHIFT_OFFSET_CH0[OFFSET2], INPUT_FETCH_SHIFT_OFFSET_CH1[OFFSET2] = 8,
INPUT_FETCH_SHIFT_WIDTH_CH0[WIDTH2], INPUT_FETCH_SHIFT_WIDTH_CH1[WIDTH2] = 8,
INPUT_FETCH_SHIFT_OFFSET_CH0[OFFSET3], INPUT_FETCH_SHIFT_OFFSET_CH1[OFFSET3] = 0,
INPUT_FETCH_SHIFT_WIDTH_CH0[WIDTH3], INPUT_FETCH_SHIFT_WIDTH_CH1[WIDTH3] = 8

All the components are swizzled and packed to the right in order from component 0 to 3.

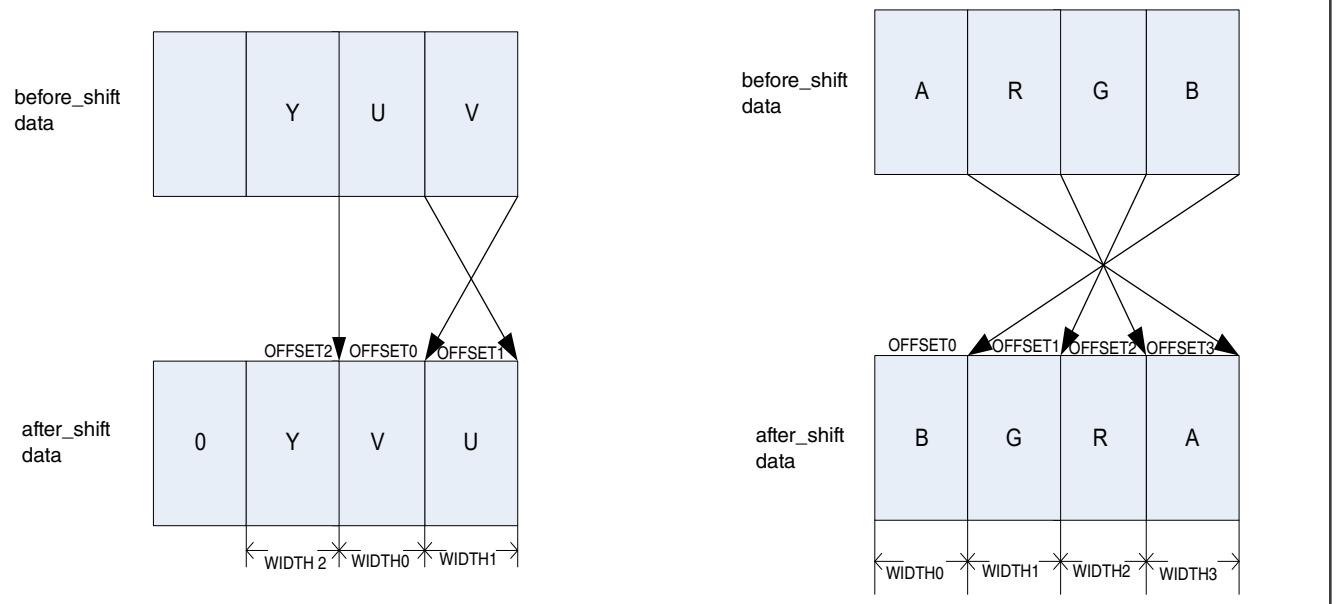


Figure 349. Fetch data shift

56.3.6.5 Fetch Interface Modes

- **Normal mode**

  If INPUT_FETCH_CTRL_CH0[BYPASS_PIXEL_EN], INPUT_FETCH_CTRL_CH1[BYPASS_PIXEL_EN] = 0 and INPUT_FETCH_CTRL_CH0[HANDSHAKE_EN], INPUT_FETCH_CTRL_CH1[HANDSHAKE_EN] = 0, the Fetch Engine is working in the “normal” or default mode.

  In normal mode, the Fetch engine will fetch data from output memory usually DDR frame by frame. The fetch frame start address is set in INPUT_FETCH_ADDR_0_CH0[INPUT_BASE_ADDR0], INPUT_FETCH_ADDR_1_CH0[INPUT_BASE_ADDR0] depending on which channel and plane format is used.

  The Fetch Engine will stop to fetch data when one frame data is finished. Another pxp_start signal will trigger another frame fetching work.

- **Bypass mode**

  In the Fetch Engine, if INPUT_FETCH_CTRL_CH0[BYPASS_PIXEL_EN], INPUT_FETCH_CTRL_CH1[BYPASS_PIXEL_EN] = 1 it will fetch data directly from the input pixel of the upstream Store Engine and not from memory. In this case the AXI bus is inactive.

  In bypass mode, the pixel input will be 32 bits per pixel per valid cycle. The bypassed data will be stored to an internal FIFO first and then output to the next downstream block after the shift operation takes place.

When INPUT_STORE_CTRL_CHn[STORE_BYPASS_EN] = 1, INPUT_FETCH_CTRL_CHn[HANDSHAKE_EN] should be equal to 0. Bypass mode and Handshake mode can not be set at the same time.

- **Handshake mode**

  If INPUT_FETCH_CTRL_CHn[HANDSHAKE_EN] = 1, the Fetch Engine is set to function in handshake mode.

  The Fetch Engine supports 1 line/ 8line/16lines handshake with the Store Engine module according to INPUT_FETCH_CTRL_CHn[HANDSHAKE_SCAN_LINE_NUM].

  INPUT_FETCH_CTRL_CHn[HANDSHAKE_SCAN_LINE_NUM] = 0, 1, 2 indicates 1 line, 8 lines, 16 lines handshake mode separately.

  One line handshake can only work in scanline mode, that is, under the condition of INPUT_FETCH_CTRL_CHn[BLOCK_EN], INPUT_STORE_CTRL_CHn[BLOCK_EN] = 0. Eight lines handshake can only work on block 8x8 mode -- INPUT_FETCH_CTRL_CHn[BLOCK_EN], INPUT_STORE_CTRL_CHn[BLOCK_EN] = 1, INPUT_FETCH_CTRL_CHn[BLOCK_16], INPUT_STORE_CTRL_CHn[BLOCK_16] = 0.

  Sixteen lines handshake configuration only works in block 16x16 mode -- INPUT_FETCH_CTRL_CHn[BLOCK_EN], INPUT_STORE_CTRL_CHn[BLOCK_EN] = 1, INPUT_FETCH_CTRL_CHn[BLOCK_16], and INPUT_STORE_CTRL_CHn[BLOCK_16] = 1.

  This is accomplished by creating two buffers in memory, where each correspond to 1/8/16-lines of the frame buffer. The buffers must be consecutive and allocated as a single block of data.

The storage required can be calculated for 1 line handshake as:

Storage = 1(line) * 2(buffer) * width * bytes/pixel;

For 8 lines handshake config the requirements are:

Storage = 8(line) * 2(buffer) * width * bytes/pixel;

For 16 lines handshake config:

Storage =16(line) * 2(buffer) * width * bytes/pixel;

To accomplish the buffer sharing, the Fetch Engine and store engine will maintain buffer status using a pair of handshake signals. When a buffer is filled by store engine, it will assert the handshake_bx_ready (where x is 0 or 1) signal to indicate to the Fetch Engine that the buffer has valid data. After reading it the Fetch Engine will then release the buffer by asserting the handshake_bx_done signal. The Store Engine is then free to refill it and assert the handshake_bx_ready again.

The 1 scanline mode is used when a Fetch Engine is connected to a Store Engine. An example would be the Input_Store communicating with the Dither_Fetch. The 8 and 16 scanline modes are used when the Output Buffer is communicating with the Dither_Fetch.

The basic protocol is shown in Figure 350:

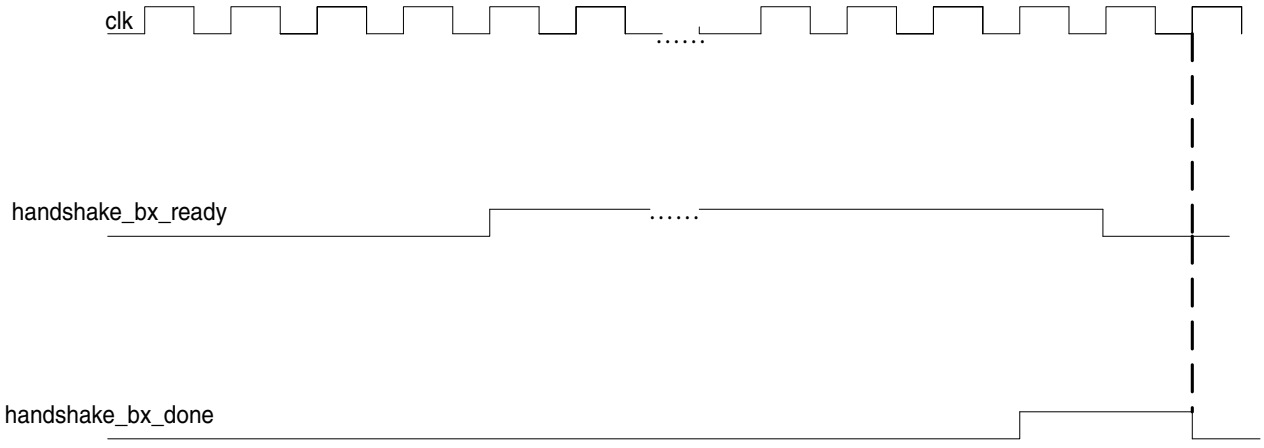


Figure 350. Handshake basic protocol

56.3.7 Clocking

PXP has 2 clock inputs: APB clock and function clock. APB clock is used for APB bus access. Function clock is used for internal logic and AXI bus. PXP has 2 clock outputs for rotation memory, they are gated version of function clock. See the appropriate Clocking module for clock setting, configuration and gating information for your device.

56.3.8 Reset

PXP has software reset bit in Control Register 0 (CTRL) . Set this bit to zero to enable normal PXP operation. Set this bit to one to disable clocking with PXP and hold it in its reset state. This bit can be turned on and then off to reset PXP to its default state.

56.3.9 Interrupts

PXP has 2 interrupts. The first interrupt source includes:

- STAT[IRQ0] indicates current PXP interrupt status.
- STAT[NEXT_IRQ] indicates that a command issued with the "Next Command" functionality has been issued and that a new command may be initiated with a write to Next Frame Pointer Register (NEXT) .

The second interrupt source includes:

- IRQ[FIRST_CH0_PREFETCH_IRQ] indicates initial ch0 prefetch engine interrupt
- IRQ[FIRST_CH1_PREFETCH_IRQ] indicates initial ch1 prefetch engine interrupt
- IRQ[FIRST_CH0_STORE_IRQ] indicates initial ch0 store engine interrupt
- IRQ[FIRST_CH1_STORE_IRQ] indicates initial ch1 store engine interrupt
- IRQ[FIRST_STORE_IRQ] indicates initial ch1 or ch0 store engine interrupt

56.4 External Signals

This module has no external signals.

56.5 Initialization

To initialize PXP:

1. Write 0 to CTRL[CLKGATE] to enable PXP clock
2. Write 0 to CTRL[SFTRST] to release software reset

3. Configure CTRL to enable function modules
4. Configure AS, PS, OUT_BUF, FE and SE related registers to input detail information
5. Configure Color Space Conversion 1 Coefficient 0 Register (CSC1_COEF0), Color Space Conversion 1 Coefficient 1 Register (CSC1_COEF1), Color Space Conversion 1 Coefficient 2 Register (CSC1_COEF2), Color Space Conversion 2 Coefficient 0 Register (CSC2_COEF0), Color Space Conversion 2 Coefficient 1 Register (CSC2_COEF1), Color Space Conversion 2 Coefficient 2 Register (CSC2_COEF2), Color Space Conversion 2 Coefficient 3 Register (CSC2_COEF3), Color Space Conversion 2 Coefficient 4 Register (CSC2_COEF4), and Color Space Conversion 2 Coefficient 5 Register (CSC2_COEF5) if need color space conversion
6. Write 1 to CTRL[ENABLE] to enable PXP processing flow

56.6 PXP Memory Map/Register Definition

56.6.1 PXP register descriptions

PXP Hardware Register Format Summary

56.6.1.1 PXP memory map

pxp base address: 4AE2_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Control Register 0 (CTRL)	32	RW	C701_0000
4	Control Register 0 (CTRL_SET)	32	RW	C701_0000
8	Control Register 0 (CTRL_CLR)	32	RW	C701_0000
C	Control Register 0 (CTRL_TOG)	32	RW	C701_0000
10	Status Register (STAT)	32	RW	0000_0000
14	Status Register (STAT_SET)	32	RW	0000_0000
18	Status Register (STAT_CLR)	32	RW	0000_0000
1C	Status Register (STAT_TOG)	32	RW	0000_0000
20	Output Buffer Control Register (OUT_CTRL)	32	RW	0000_0000
24	Output Buffer Control Register (OUT_CTRL_SET)	32	RW	0000_0000
28	Output Buffer Control Register (OUT_CTRL_CLR)	32	RW	0000_0000
2C	Output Buffer Control Register (OUT_CTRL_TOG)	32	RW	0000_0000
30	Output Frame Buffer Pointer Register (OUT_BUF)	32	RW	0000_0000
40	Output Frame Buffer Pointer #2 Register (OUT_BUF2)	32	RW	0000_0000
50	Output Buffer Pitch Register (OUT_PITCH)	32	RW	0000_0000
60	Output Surface Lower Right Coordinate Register (OUT_LRC)	32	RW	0000_0000
70	Processed Surface Upper Left Coordinate Register (OUT_PS_ULC)	32	RW	0000_0000
80	Processed Surface Lower Right Coordinate Register (OUT_PS_LRC)	32	RW	0000_0000
90	Alpha Surface Upper Left Coordinate Register (OUT_AS_ULC)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
A0	Alpha Surface Lower Right Coordinate Register (OUT_AS_LRC)	32	RW	0000_0000
B0	Processed Surface (PS) Control Register (PS_CTRL)	32	RW	0000_0000
B4	Processed Surface (PS) Control Register (PS_CTRL_SET)	32	RW	0000_0000
B8	Processed Surface (PS) Control Register (PS_CTRL_CLR)	32	RW	0000_0000
BC	Processed Surface (PS) Control Register (PS_CTRL_TOG)	32	RW	0000_0000
C0	PS Input Buffer Address Register (PS_BUF)	32	RW	0000_0000
D0	PS U/Cb or 2 Plane UV Input Buffer Address Register (PS_UBUF)	32	RW	0000_0000
E0	PS V/Cr Input Buffer Address Register (PS_VBUF)	32	RW	0000_0000
F0	Processed Surface Pitch Register (PS_PITCH)	32	RW	0000_0000
100	PS Background Color Register (PS_BACKGROUND_0)	32	RW	0000_0000
110	PS Scale Factor Register (PS_SCALE)	32	RW	1000_1000
120	PS Scale Offset Register (PS_OFFSET)	32	RW	0000_0000
130	PS Color Key Low Register (PS_CLRKEYLOW_0)	32	RW	00FF_FFFF
140	PS Color Key High Register (PS_CLRKEYHIGH_0)	32	RW	0000_0000
150	Alpha Surface Control Register (AS_CTRL)	32	RW	0000_0000
160	Alpha Surface Buffer Pointer Register (AS_BUF)	32	RW	0000_0000
170	Alpha Surface Pitch Register (AS_PITCH)	32	RW	0000_0000
180	Overlay Color Key Low Register (AS_CLRKEYLOW_0)	32	RW	00FF_FFFF
190	Overlay Color Key High Register (AS_CLRKEYHIGH_0)	32	RW	0000_0000
1A0	Color Space Conversion 1 Coefficient 0 Register (CSC1_COEF0)	32	RW	0400_0000
1B0	Color Space Conversion 1 Coefficient 1 Register (CSC1_COEF1)	32	RW	0123_0208
1C0	Color Space Conversion 1 Coefficient 2 Register (CSC1_COEF2)	32	RW	079B_076C
1D0	Color Space Conversion 2 Control Register (CSC2_CTRL)	32	RW	0000_0001
1E0	Color Space Conversion 2 Coefficient 0 Register (CSC2_COEF0)	32	RW	0000_0000
1F0	Color Space Conversion 2 Coefficient 1 Register (CSC2_COEF1)	32	RW	0000_0000
200	Color Space Conversion 2 Coefficient 2 Register (CSC2_COEF2)	32	RW	0000_0000
210	Color Space Conversion 2 Coefficient 3 Register (CSC2_COEF3)	32	RW	0000_0000
220	Color Space Conversion 2 Coefficient 4 Register (CSC2_COEF4)	32	RW	0000_0000
230	Color Space Conversion 2 Coefficient 5 Register (CSC2_COEF5)	32	RW	0000_0000
290	Alpha Engine A Control Register (ALPHA_A_CTRL)	32	RW	0000_0000
2A0	Alpha Engine B Control Register (ALPHA_B_CTRL)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2B0	Alpha Engine B Control 1 Register (ALPHA_B_CTRL_1)	32	RW	0000_0000
2C0	PS Background Color 1 Register (PS_BACKGROUND_1)	32	RW	0000_0000
2D0	PS Color Key Low Register (PS_CLRKEYLOW_1)	32	RW	00FF_FFFF
2E0	PS Color Key High Register (PS_CLRKEYHIGH_1)	32	RW	0000_0000
2F0	Overlay Color Key Low Register (AS_CLRKEYLOW_1)	32	RW	00FF_FFFF
300	Overlay Color Key High Register (AS_CLRKEYHIGH_1)	32	RW	0000_0000
310	Control Register 2 (CTRL2)	32	RW	0000_0000
314	Control Register 2 (CTRL2_SET)	32	RW	0000_0000
318	Control Register 2 (CTRL2_CLR)	32	RW	0000_0000
31C	Control Register 2 (CTRL2_TOG)	32	RW	0000_0000
320	Power Control Register 0 (POWER_REG0)	32	RW	0000_0000
330	Power Control Register 1 (POWER_REG1)	32	RW	0000_0000
340	Data Path Control 0 Register (DATA_PATH_CTRL0)	32	RW	4104_5A2B
344	Data Path Control 0 Register (DATA_PATH_CTRL0_SET)	32	RW	4104_5A2B
348	Data Path Control 0 Register (DATA_PATH_CTRL0_CLR)	32	RW	4104_5A2B
34C	Data Path Control 0 Register (DATA_PATH_CTRL0_TOG)	32	RW	4104_5A2B
390	IRQ Mask Register (IRQ_MASK)	32	RW	0000_0000
394	IRQ Mask Register (IRQ_MASK_SET)	32	RW	0000_0000
398	IRQ Mask Register (IRQ_MASK_CLR)	32	RW	0000_0000
39C	IRQ Mask Register (IRQ_MASK_TOG)	32	RW	0000_0000
3A0	Interrupt Register (IRQ)	32	RW	0000_0000
3A4	Interrupt Register (IRQ_SET)	32	RW	0000_0000
3A8	Interrupt Register (IRQ_CLR)	32	RW	0000_0000
3AC	Interrupt Register (IRQ_TOG)	32	RW	0000_0000
400	Next Frame Pointer Register (NEXT)	32	RW	0000_0000
450	Input Fetch Control Channel 0 Register (INPUT_FETCH_CTRL_CH0)	32	RW	0002_0000
454	Input Fetch Control Channel 0 Register (INPUT_FETCH_CTRL_CH0_SET)	32	RW	0002_0000
458	Input Fetch Control Channel 0 Register (INPUT_FETCH_CTRL_CH0_CLR)	32	RW	0002_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
45C	Input Fetch Control Channel 0 Register (INPUT_FETCH_CTRL_CH0_TOG)	32	RW	0002_0000
460	Input Fetch Control Channel 1 Register (INPUT_FETCH_CTRL_CH1)	32	RW	0002_0000
464	Input Fetch Control Channel 1 Register (INPUT_FETCH_CTRL_CH1_SET)	32	RW	0002_0000
468	Input Fetch Control Channel 1 Register (INPUT_FETCH_CTRL_CH1_CLR)	32	RW	0002_0000
46C	Input Fetch Control Channel 1 Register (INPUT_FETCH_CTRL_CH1_TOG)	32	RW	0002_0000
470	Input Fetch Status Channel 0 Register (INPUT_FETCH_STATUS_CH0)	32	R	0000_0000
480	Input Fetch Status Channel 1 Register (INPUT_FETCH_STATUS_CH1)	32	R	0000_0000
490	Input Fetch Active Size ULC Channel 0 Register (INPUT_FETCH_ACTIVE_SIZE_ULC_CH0)	32	RW	0000_0000
4A0	Input Fetch Active Size LRC Channel 0 Register (INPUT_FETCH_ACTIVE_SIZE_LRC_CH0)	32	RW	0000_0000
4B0	Input Fetch Active Size ULC Channel 1 Register (INPUT_FETCH_ACTIVE_SIZE_ULC_CH1)	32	RW	0000_0000
4C0	Input Fetch Active Size LRC Channel 1 Register (INPUT_FETCH_ACTIVE_SIZE_LRC_CH1)	32	RW	0000_0000
4D0	Input Fetch Size Channel 0 Register (INPUT_FETCH_SIZE_CH0)	32	RW	0000_0000
4E0	Input Fetch Size Channel 1 Register (INPUT_FETCH_SIZE_CH1)	32	RW	0000_0000
4F0	Input Fetch Background Color Channel 0 Register (INPUT_FETCH_BACKGROUND_COLOR_CH0)	32	RW	0000_0000
500	Input Fetch Background Color Channel 1 Register (INPUT_FETCH_BACKGROUND_COLOR_CH1)	32	RW	0000_0000
510	Input Fetch Pitch Register (INPUT_FETCH_PITCH)	32	RW	0000_0000
520	Input Fetch Shift Control Channel 0 Register (INPUT_FETCH_SHIFT_CTRL_CH0)	32	RW	0000_1000
524	Input Fetch Shift Control Channel 0 Register (INPUT_FETCH_SHIFT_CTRL_CH0_SET)	32	RW	0000_1000
528	Input Fetch Shift Control Channel 0 Register (INPUT_FETCH_SHIFT_CTRL_CH0_CLR)	32	RW	0000_1000
52C	Input Fetch Shift Control Channel 0 Register (INPUT_FETCH_SHIFT_CTRL_CH0_TOG)	32	RW	0000_1000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
530	Input Fetch Shift Control Channel 1 Register (INPUT_FETCH_SHIFT_CTRL_CH1)	32	RW	0000_1000
534	Input Fetch Shift Control Channel 1 Register (INPUT_FETCH_SHIFT_CTRL_CH1_SET)	32	RW	0000_1000
538	Input Fetch Shift Control Channel 1 Register (INPUT_FETCH_SHIFT_CTRL_CH1_CLR)	32	RW	0000_1000
53C	Input Fetch Shift Control Channel 1 Register (INPUT_FETCH_SHIFT_CTRL_CH1_TOG)	32	RW	0000_1000
540	Input Fetch Shift Offset Channel 0 Register (INPUT_FETCH_SHIFT_OFFSET_CH0)	32	RW	0000_0000
544	Input Fetch Shift Offset Channel 0 Register (INPUT_FETCH_SHIFT_OFFSET_CH0_SET)	32	RW	0000_0000
548	Input Fetch Shift Offset Channel 0 Register (INPUT_FETCH_SHIFT_OFFSET_CH0_CLR)	32	RW	0000_0000
54C	Input Fetch Shift Offset Channel 0 Register (INPUT_FETCH_SHIFT_OFFSET_CH0_TOG)	32	RW	0000_0000
550	Input Fetch Shift Offset Channel 1 Register (INPUT_FETCH_SHIFT_OFFSET_CH1)	32	RW	0000_0000
554	Input Fetch Shift Offset Channel 1 Register (INPUT_FETCH_SHIFT_OFFSET_CH1_SET)	32	RW	0000_0000
558	Input Fetch Shift Offset Channel 1 Register (INPUT_FETCH_SHIFT_OFFSET_CH1_CLR)	32	RW	0000_0000
55C	Input Fetch Shift Offset Channel 1 Register (INPUT_FETCH_SHIFT_OFFSET_CH1_TOG)	32	RW	0000_0000
560	Input Fetch Shift Width Channel 0 Register (INPUT_FETCH_SHIFT_WIDTH_CH0)	32	RW	0000_8888
564	Input Fetch Shift Width Channel 0 Register (INPUT_FETCH_SHIFT_WIDTH_CH0_SET)	32	RW	0000_8888
568	Input Fetch Shift Width Channel 0 Register (INPUT_FETCH_SHIFT_WIDTH_CH0_CLR)	32	RW	0000_8888
56C	Input Fetch Shift Width Channel 0 Register (INPUT_FETCH_SHIFT_WIDTH_CH0_TOG)	32	RW	0000_8888
570	Input Fetch Shift Width Channel 1 Register (INPUT_FETCH_SHIFT_WIDTH_CH1)	32	RW	0000_8888
574	Input Fetch Shift Width Channel 1 Register (INPUT_FETCH_SHIFT_WIDTH_CH1_SET)	32	RW	0000_8888

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
578	Input Fetch Shift Width Channel 1 Register (INPUT_FETCH_SHIFT_WIDTH_CH1_CLR)	32	RW	0000_8888
57C	Input Fetch Shift Width Channel 1 Register (INPUT_FETCH_SHIFT_WIDTH_CH1_TOG)	32	RW	0000_8888
580	Input Fetch Address 0 Channel 0 Register (INPUT_FETCH_ADDR_0_CH0)	32	RW	0000_0000
590	Input Fetch Address 1 Channel 0 Register (INPUT_FETCH_ADDR_1_CH0)	32	RW	0000_0000
5A0	Input Fetch Address 0 Channel 1 Register (INPUT_FETCH_ADDR_0_CH1)	32	RW	0000_0000
5B0	Input Fetch Address 1 Channel 1 Register (INPUT_FETCH_ADDR_1_CH1)	32	RW	0000_0000
5C0	Input Store Control Channel 0 Register (INPUT_STORE_CTRL_CH0)	32	RW	0002_0200
5C4	Input Store Control Channel 0 Register (INPUT_STORE_CTRL_CH0_SET)	32	RW	0002_0200
5C8	Input Store Control Channel 0 Register (INPUT_STORE_CTRL_CH0_CLR)	32	RW	0002_0200
5CC	Input Store Control Channel 0 Register (INPUT_STORE_CTRL_CH0_TOG)	32	RW	0002_0200
5D0	Input Store Control Channel 1 Register (INPUT_STORE_CTRL_CH1)	32	RW	0002_0200
5D4	Input Store Control Channel 1 Register (INPUT_STORE_CTRL_CH1_SET)	32	RW	0002_0200
5D8	Input Store Control Channel 1 Register (INPUT_STORE_CTRL_CH1_CLR)	32	RW	0002_0200
5DC	Input Store Control Channel 1 Register (INPUT_STORE_CTRL_CH1_TOG)	32	RW	0002_0200
5E0	Input Store Status Channel 0 Register (INPUT_STORE_STATUS_CH0)	32	R	0000_0000
5F0	Input Store Status Channel 1 Register (INPUT_STORE_STATUS_CH1)	32	R	0000_0000
600	Input Store Size Channel 0 Register (INPUT_STORE_SIZE_CH0)	32	RW	0000_0000
610	Input Store Size Channel 1 Register (INPUT_STORE_SIZE_CH1)	32	RW	0000_0000
620	Input Store Pitch Register (INPUT_STORE_PITCH)	32	RW	0000_0000
630	Input Store Shift Control Channel 0 Register (INPUT_STORE_SHIFT_CTRL_CH0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
634	Input Store Shift Control Channel 0 Register (INPUT_STORE_SHIFT_CTRL_CH0_SET)	32	RW	0000_0000
638	Input Store Shift Control Channel 0 Register (INPUT_STORE_SHIFT_CTRL_CH0_CLR)	32	RW	0000_0000
63C	Input Store Shift Control Channel 0 Register (INPUT_STORE_SHIFT_CTRL_CH0_TOG)	32	RW	0000_0000
640	Input Store Shift Control Channel 1 Register (INPUT_STORE_SHIFT_CTRL_CH1)	32	RW	0000_0000
644	Input Store Shift Control Channel 1 Register (INPUT_STORE_SHIFT_CTRL_CH1_SET)	32	RW	0000_0000
648	Input Store Shift Control Channel 1 Register (INPUT_STORE_SHIFT_CTRL_CH1_CLR)	32	RW	0000_0000
64C	Input Store Shift Control Channel 1 Register (INPUT_STORE_SHIFT_CTRL_CH1_TOG)	32	RW	0000_0000
690	Input Store Address 0 Channel 0 Register (INPUT_STORE_ADDR_0_CH0)	32	RW	0000_0000
6A0	Input Store Address 1 Channel 0 Register (INPUT_STORE_ADDR_1_CH0)	32	RW	0000_0000
6B0	Input Store Fill Data Channel 0 Register (INPUT_STORE_FILL_DATA_CH0)	32	RW	0000_0000
6C0	Input Store Address 0 Channel 1 Register (INPUT_STORE_ADDR_0_CH1)	32	RW	0000_0000
6D0	Input Store Address 1 Channel 1 Register (INPUT_STORE_ADDR_1_CH1)	32	RW	0000_0000
6E0	Input Store Data Mask 0 High Byte Channel 0 Register (INPUT_STORE_D_MASK0_H_CH0)	32	RW	0000_0000
6F0	Input Store Data Mask 0 Low Byte Channel 0 Register (INPUT_STORE_D_MASK0_L_CH0)	32	RW	0000_0000
700	Input Store Data Mask 1 High Byte Channel 0 Register (INPUT_STORE_D_MASK1_H_CH0)	32	RW	0000_0000
710	Input Store Data Mask 1 Low Byte Channel 0 Register (INPUT_STORE_D_MASK1_L_CH0)	32	RW	0000_0000
720	Input Store Data Mask 2 High Byte Channel 0 Register (INPUT_STORE_D_MASK2_H_CH0)	32	RW	0000_0000
730	Input Store Data Mask 2 Low Byte Channel 0 Register (INPUT_STORE_D_MASK2_L_CH0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
740	Input Store Data Mask 3 High Byte Channel 0 Register (INPUT_STORE_D_MASK3_H_CH0)	32	RW	0000_0000
750	Input Store Data Mask 3 Low Byte Channel 0 Register (INPUT_STORE_D_MASK3_L_CH0)	32	RW	0000_0000
760	Input Store Data Mask 4 High Byte Channel 0 Register (INPUT_STORE_D_MASK4_H_CH0)	32	RW	0000_0000
770	Input Store Data Mask 4 Low Byte Channel 0 Register (INPUT_STORE_D_MASK4_L_CH0)	32	RW	0000_0000
780	Input Store Data Mask 5 High Byte Channel 0 Register (INPUT_STORE_D_MASK5_H_CH0)	32	RW	0000_0000
790	Input Store Data Mask 5 Low Byte Channel 0 Register (INPUT_STORE_D_MASK5_L_CH0)	32	RW	0000_0000
7A0	Input Store Data Mask 6 High Byte Channel 0 Register (INPUT_STORE_D_MASK6_H_CH0)	32	RW	0000_0000
7B0	Input Store Data Mask 6 Low Byte Channel 0 Register (INPUT_STORE_D_MASK6_L_CH0)	32	RW	0000_0000
7C0	Input Store Data Mask 7 High Byte Channel 0 Register (INPUT_STORE_D_MASK7_H_CH0)	32	RW	0000_0000
7E0	Input Store Data Mask 7 Low Byte Channel 0 Register (INPUT_STORE_D_MASK7_L_CH0)	32	RW	0000_0000
7F0	Input Store Data Shift Low Channel 0 Register (INPUT_STORE_D_SHIFT_L_CH0)	32	RW	0000_0000
800	Input Store Data Shift High Channel 0 Register (INPUT_STORE_D_SHIFT_H_CH0)	32	RW	0000_0000
810	Input Store Flag Shift Low Channel 0 Register (INPUT_STORE_F_SHIFT_L_CH0)	32	RW	0000_0000
820	Input Store Flag Shift High Channel 0 Register (INPUT_STORE_F_SHIFT_H_CH0)	32	RW	0000_0000
830	Input Store Flag Mask Low Channel 0 Register (INPUT_STORE_F_MASK_L_CH0)	32	RW	0000_0000
840	Input Store Flag Mask High Channel 0 Register (INPUT_STORE_F_MASK_H_CH0)	32	RW	0000_0000
2CE0	Bus MUX Register (BUS_MUX)	32	RW	0000_0000
2CF0	Handshake Ready MUX 0 Register (HANDSHAKE_READY_MUX0)	32	RW	7654_3210
2D00	Handshake Ready MUX 1 Register (HANDSHAKE_READY_MUX1)	32	RW	FEDC_BA98
2D10	Handshake Done MUX 0 Register (HANDSHAKE_DONE_MUX0)	32	RW	7654_3210

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2D20	Handshake Done MUX 1 Register (HANDSHAKE_DONE_MUX1)	32	RW	FEDC_BA98
2D30	Handshake CPU Fetch Register (HANDSHAKE_CPU_FETCH)	32	RW	0010_0010
2D34	Handshake CPU Fetch Register (HANDSHAKE_CPU_FETCH_SET)	32	RW	0010_0010
2D38	Handshake CPU Fetch Register (HANDSHAKE_CPU_FETCH_CLR)	32	RW	0010_0010
2D3C	Handshake CPU Fetch Register (HANDSHAKE_CPU_FETCH_TOG)	32	RW	0010_0010
2D40	Handshake CPU Store Register (HANDSHAKE_CPU_STORE)	32	RW	0010_0010
2D44	Handshake CPU Store Register (HANDSHAKE_CPU_STORE_SET)	32	RW	0010_0010
2D48	Handshake CPU Store Register (HANDSHAKE_CPU_STORE_CLR)	32	RW	0010_0010
2D4C	Handshake CPU Store Register (HANDSHAKE_CPU_STORE_TOG)	32	RW	0010_0010

56.6.1.2 Control Register 0 (CTRL)

The CTRL register contains controls for PXP module.

CTRL: 0x000

CTRL_SET: 0x004

CTRL_CLR: 0x008

CTRL_TOG: 0x00C

The Control register contains the primary controls for PXP block.

EXAMPLE

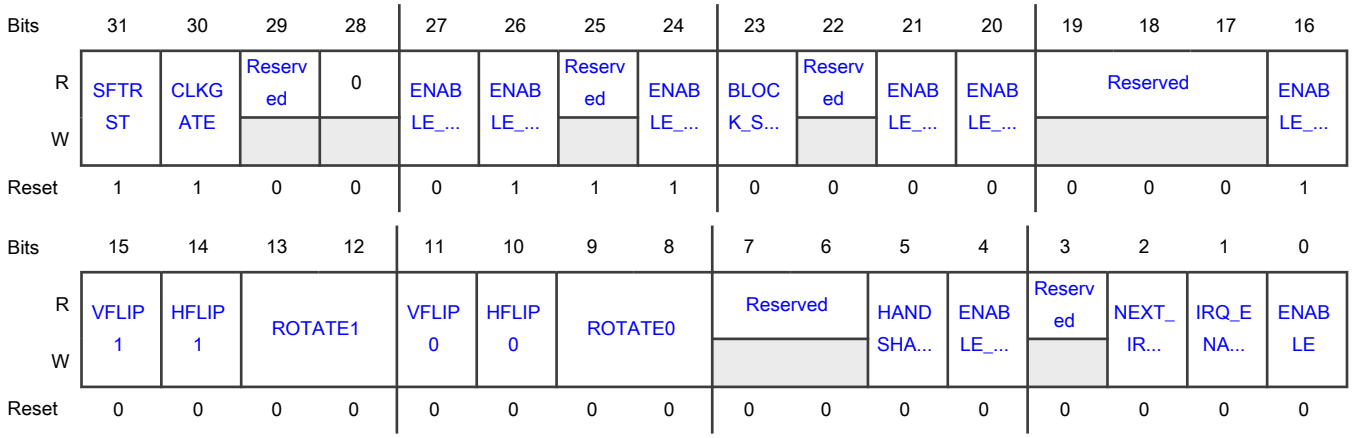
```
REG_CTRL_SET(BM_PXP_CTRL_SFTRST);
REG_CTRL_CLR(BM_PXP_CTRL_SFTRST | BM_PXP_CTRL_CLKGATE);
```

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
CTRL	0h	Control Register 0
CTRL_SET	4h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 1
CTRL_CLR	8h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 0
CTRL_TOG	Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CTRL

Diagram



Fields

Field	Description
31 SFTRST	SFTRST Set this bit to zero to enable normal PXP operation. Set this bit to one (default) to disable clocking with PXP and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset PXP block to its default state.
30 CLKGATE	CLKGATE This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block.
29 —	Reserved
28 —	Reserved
27 ENABLE_ROTATE1	ENABLE_ROTATE1 Enable the ROTATE1 engine in PXP primary processing flow.
26 ENABLE_ROTATE0	ENABLE_ROTATE0 Enable the ROTATE0 engine in PXP primary processing flow.
25 —	Reserved
24 ENABLE_CSC2	ENABLE_CSC2 Enable the CSC2 engine in PXP primary processing flow.
23	BLOCK_SIZE

Table continues on the next page...

Table continued from the previous page...

Field	Description
BLOCK_SIZE	Select the block size to process through the Rotate block. 0 - BLK_SIZE_8X8 : Process 8x8 pixel blocks. 1 - BLK_SIZE_16X16 : Process 16x16 pixel blocks.
22 —	Reserved
21 ENABLE_ALPHA_B	ENABLE_ALPHA_B Enable the Alpha-B engine in PXP primary processing flow.
20 ENABLE_INPUT_FETCH_STORE	ENABLE_INPUT_FETCH_STORE Enable the Input Fetch and Store engine in PXP primary processing flow.
19-17 —	Reserved
16 ENABLE_PS_AS_OUT	ENABLE_PS_AS_OUT Enable the PS engine, AS engine, OUTBUF in PXP primary processing flow.
15 VFLIP1	VFLIP1 Indicates that the input should be flipped vertically (effect applied before rotation).
14 HFLIP1	HFLIP1 Indicates that the input should be flipped horizontally (effect applied before rotation).
13-12 ROTATE1	ROTATE1 Indicates the clockwise rotation to be applied at the input buffer. The rotation effect is defined as occurring after the FLIP_X and FLIP_Y permutation. 00 - ROT_0 : rotate 0 degrees 01 - ROT_90 : rotate 90 degrees 10 - ROT_180 : rotate 180 degrees 11 - ROT_270 : rotate 270 degrees
11 VFLIP0	VFLIP0 Indicates that the output buffer should be flipped vertically (effect applied before rotation).
10 HFLIP0	HFLIP0 Indicates that the output buffer should be flipped horizontally (effect applied before rotation).

Table continues on the next page...

Table continued from the previous page...

Field	Description
9-8 ROTATE0	ROTATE0 Indicates the clockwise rotation to be applied at the output buffer. The rotation effect is defined as occurring after the FLIP_X and FLIP_Y permutation. 00 - ROT_0 : rotate 0 degrees 01 - ROT_90 : rotate 90 degrees 10 - ROT_180 : rotate 180 degrees 11 - ROT_270 : rotate 270 degrees
7-6 —	Reserved
5 HANDSHAKE_ABORT_SKIP	HANDSHAKE_ABORT_SKIP When skip is enable, even the abort asserted, pxp will not assert the ready directly but wait for whole block line complete.
4 ENABLE_LCD0_HANDSHAKE	ENABLE_LCD0_HANDSHAKE Enable handshake with LCD0 controller. When this is set, PXP will not process an entire frame buffer, but will instead process rows of NxN blocks in a double-buffer handshake with the LCDIF. This enables the use of the onboard SRAM for a partial frame buffer.
3 —	Reserved
2 NEXT_IRQ_ENABLE	NEXT_IRQ_ENABLE Next command interrupt enable. When set, PXP will issue an interrupt when a queued command initiated by a write to NEXT has been loaded into PXP"s registers. This interrupt also indicates that a new command may now be queued.
1 IRQ_ENABLE	IRQ_ENABLE Interrupt enable. **NOTE** When using the NEXT functionality to reprogram PXP, the new value of this bit will be used and may therefore enable or disable an interrupt unintentionally.
0 ENABLE	ENABLE Enables PXP operation with specified parameters. The ENABLE bit will remain set while PXP is active and will be cleared once the current operation completes. Software should use the IRQ bit in the STAT when polling for PXP completion.

56.6.1.3 Status Register (STAT)

PXP Interrupt Status register provides interrupt status information.

STAT: 0x010

STAT_SET: 0x014

STAT_CLR: 0x018

STAT_TOG: 0x01C

This register provides PXP interrupt status and the current X/Y block coordinate that is being processed.

EXAMPLE

```
REG_STAT_CLR(BM_PXP_STAT_IRQ); // clear CSC interrupt
```

NOTE

Writing on SET or TOG variants of these registers will have no impact on value.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
STAT	10h	Status Register
STAT_SET	14h	Writing 1 to a bit in this register ensures that the corresponding bit in STAT is 1
STAT_CLR	18h	Writing 1 to a bit in this register ensures that the corresponding bit in STAT is 0
STAT_TOG	1Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in STAT

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BLOCKX								BLOCKY							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AXI_ERROR_ID_1				Reserved	AXI_READ...	AXI_WRI...	Reserved	AXI_ERROR_ID_0				NEXT_IRQ	AXI_READ...	AXI_WRI...	IRQ0
W						W1C	W1C						W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 BLOCKX	BLOCKX Indicates the X coordinate of the block currently being rendered.
23-16	BLOCKY

Table continues on the next page...

Table continued from the previous page...

Field	Description
BLOCKY	Indicates the Y coordinate of the block currently being rendered.
15-12 AXI_ERROR_ID_1	AXI_ERROR_ID_1 Indicates the AXI1 ID of the failing bus operation.
11 —	Reserved
10 AXI_READ_ERROR_1	AXI_READ_ERROR_1 Indicates PXP encountered an AXI read error and processing has been terminated.
9 AXI_WRITE_ERROR_1	AXI_WRITE_ERROR_1 Indicates PXP encountered an AXI write error and processing has been terminated.
8 —	Reserved
7-4 AXI_ERROR_ID_0	AXI_ERROR_ID_0 Indicates the AXI0 ID of the failing bus operation.
3 NEXT_IRQ	NEXT_IRQ Indicates that a command issued with the quot;Next Commandquot; functionality has been issued and that a new command may be initiated with a write to NEXT.
2 AXI_READ_ERROR_0	AXI_READ_ERROR_0 Indicates PXP encountered an AXI read error and processing has been terminated.
1 AXI_WRITE_ERROR_0	AXI_WRITE_ERROR_0 Indicates PXP encountered an AXI write error and processing has been terminated.
0 IRQ0	IRQ0 Indicates current PXP interrupt status. The IRQ is routed through the pxp_irq when the IRQ_ENABLE bit in the control register is set.

56.6.1.4 Output Buffer Control Register (OUT_CTRL)

The OUT_CTRL register contains controls for the Output Buffer.

OUT_CTRL: 0x020

OUT_CTRL_SET: 0x024

OUT_CTRL_CLR: 0x028

OUT_CTRL_TOG: 0x02C

Control register to determine OUT buffer processing.

EXAMPLE

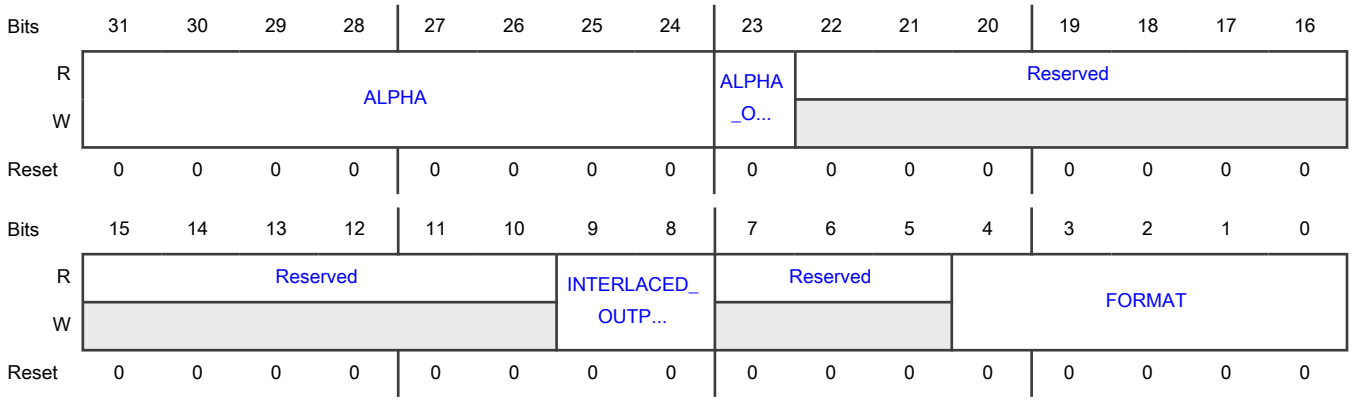
```
REG_CTRL_SET(BM_PXP_CTRL_SFTRST);
REG_CTRL_CLR(BM_PXP_CTRL_SFTRST | BM_PXP_CTRL_CLKGATE);
```

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
OUT_CTRL	20h	Output Buffer Control Register
OUT_CTRL_SET	24h	Writing 1 to a bit in this register ensures that the corresponding bit in OUT_CTRL is 1
OUT_CTRL_CLR	28h	Writing 1 to a bit in this register ensures that the corresponding bit in OUT_CTRL is 0
OUT_CTRL_TOG	2Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in OUT_CTRL

Diagram



Fields

Field	Description
31-24 ALPHA	ALPHA When generating an output buffer with an alpha component, the value in this field will be used when enabled to override the alpha passed through the pixel data pipeline.
23 ALPHA_OUTPUT	ALPHA_OUTPUT Indicates that alpha component in output buffer pixels should be overwritten by REG_OUT_CTRL[ALPHA] register. If 0, retain their alpha value from the computed alpha for that pixel.

Table continues on the next page...

Table continued from the previous page...

Field	Description
22-10 —	Reserved
9-8 INTERLACED_OUTPUT	INTERLACED_OUTPUT Determines how PXP writes its output data. Output interlacing should not be used in conjunction with input interlacing. Splitting frames into fields is most efficient using output interlacing. 2-plane output formats AND interlaced output is NOT supported. 00 - PROGRESSIVE : All data written in progressive format to the OUTBUF Pointer. 01 - FIELD0 : Interlaced output: only data for field 0 is written to the OUTBUF Pointer. 10 - FIELD1 : Interlaced output: only data for field 1 is written to the OUTBUF2 Pointer. 11 - INTERLACED : Interlaced output: data for field 0 is written to OUTBUF and data for field 1 is written to OUTBUF2.
7-5 —	Reserved
4-0 FORMAT	FORMAT Output frame buffer format. The UV byte lanes are synonymous with CbCr byte lanes for YUV output pixel formats. For example, the YUV2P420 format should be selected when the output is YCbCr 2-plane 420 output format. 0_0000 - ARGB8888 : 32-bit RGB format with blue in LSB 0_0100 - RGB888 : 32-bit RGB format with blue in LSB (unpacked 24-bit pixel in 32 bit DWORD) 0_0101 - RGB888P : 24-bit RGB format with blue in LSB (packed 24-bit format) 0_1000 - ARGB1555 : 16-bit RGB format with blue in LSB 0_1001 - ARGB4444 : 16-bit RGB format with blue in LSB 0_1100 - RGB555 : 16-bit RGB format with blue in LSB 0_1101 - RGB444 : 16-bit RGB format with blue in LSB 0_1110 - RGB565 : 16-bit RGB format with blue in LSB 1_0000 - YUV1P444 : 32-bit pixels (1-plane XYUV unpacked) 1_0010 - UYVY1P422 : 16-bit pixels (1-plane U0,Y0,V0,Y1 interleaved bytes) 1_0011 - VYUY1P422 : 16-bit pixels (1-plane V0,Y0,U0,Y1 interleaved bytes) 1_0100 - Y8 : 8-bit monochrome pixels (1-plane Y luma output) 1_0101 - Y4 : 4-bit monochrome pixels (1-plane Y luma, 4 bit truncation) 1_1000 - YUV2P422 : 16-bit pixels (2-plane UV interleaved bytes) 1_1001 - YUV2P420 : 16-bit pixels (2-plane UV) 1_1010 - YVU2P422 : 16-bit pixels (2-plane VU interleaved bytes) 1_1011 - YVU2P420 : 16-bit pixels (2-plane VU)

56.6.1.5 Output Frame Buffer Pointer Register (OUT_BUF)

Output Frame Buffer Pointer. This register points to the beginning of the output frame buffer. This pointer is used for progressive format and field 0 when generating interlaced output.

This register is used by the logic to point to the current output location for the output frame buffer.

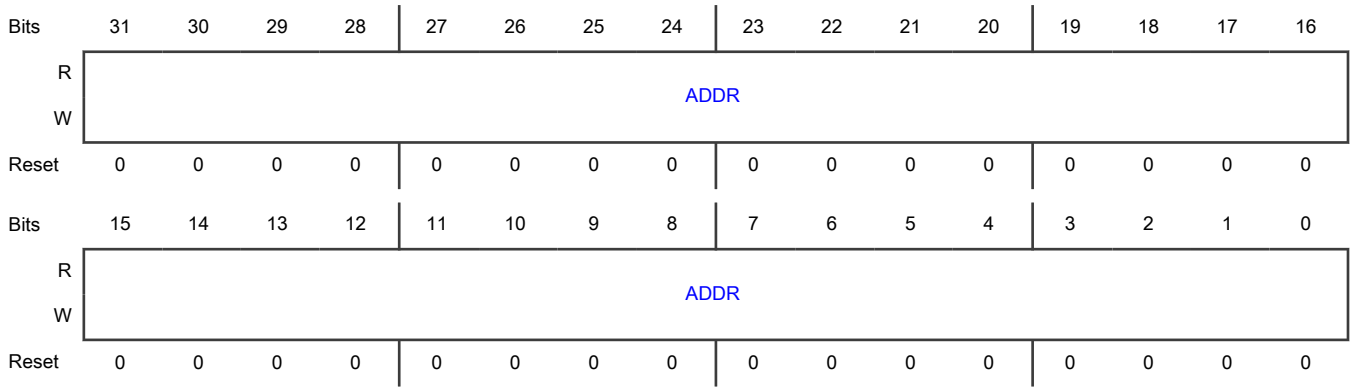
EXAMPLE

```
REG_OUT_BUF_WR( buffer );
```

Offset

Register	Offset
OUT_BUF	30h

Diagram



Fields

Field	Description
31-0 ADDR	ADDR Current address pointer for the output frame buffer. The address can have any byte alignment. 64B alignment is recommended for optimal performance.

56.6.1.6 Output Frame Buffer Pointer #2 Register (OUT_BUF2)

Output Frame Buffer Pointer #2. This register points to the beginning of the output frame buffer for either field 1 when generating interlaced output or for the UV buffer when in YUV 2-plane output modes. Both interlaced output AND 2-plane output modes are not supported in a single PXP operation.

This register is used by the logic to point to the current output location for the field 1 or UV output frame buffer.

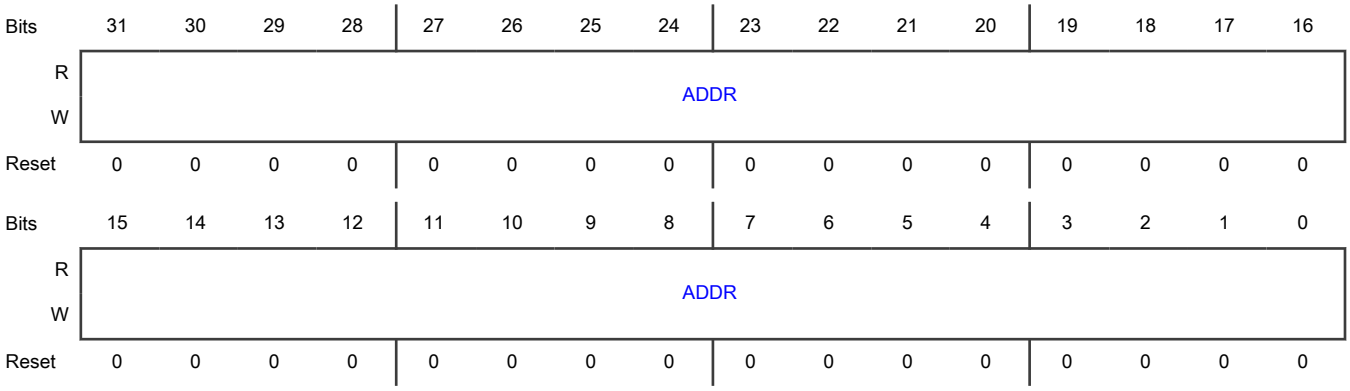
EXAMPLE

```
REG_OUT_BUF_WR( field0 ); // buffer for interlaced field 0
REG_OUT_BUF2_WR( field1 ); // buffer for interlaced field 1
```

Offset

Register	Offset
OUT_BUF2	40h

Diagram



Fields

Field	Description
31-0 ADDR	ADDR Current address pointer for the output frame buffer. The address can have any byte alignment. 64B alignment is recommended for optimal performance.

56.6.1.7 Output Buffer Pitch Register (OUT_PITCH)

This register contains the output buffer pitch in bytes.

Any byte value will indicate the vertical pitch. This value will be used in output pixel address calculations.

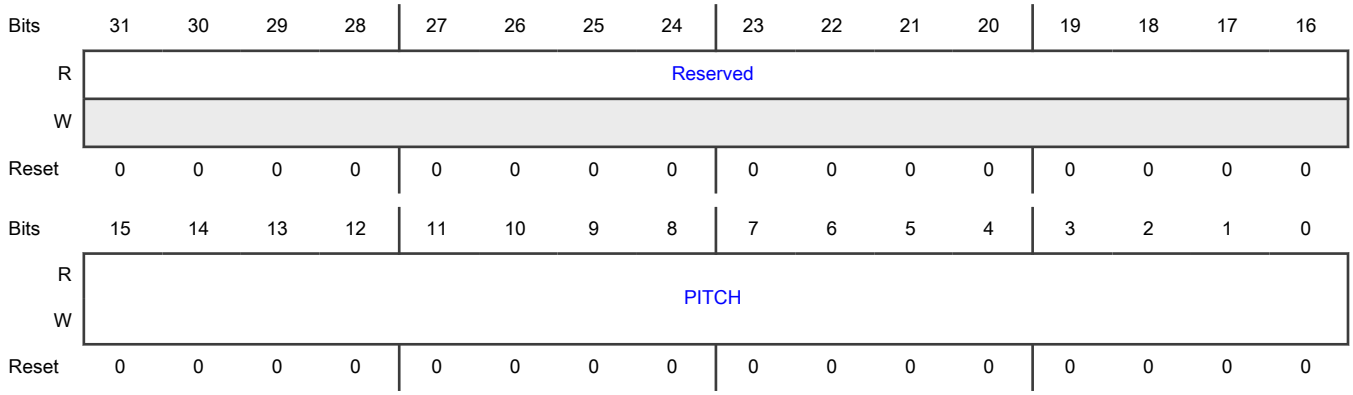
EXAMPLE

```
REG_OUT_PITCH_WR( 68 * 4 ); // The output buffer pitch is 68 pixels times 32 bits per pixel
```

Offset

Register	Offset
OUT_PITCH	50h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 PITCH	PITCH Indicates the number of bytes in memory between two vertically adjacent pixels.

56.6.1.8 Output Surface Lower Right Coordinate Register (OUT_LRC)

This register contains the size, or lower right coordinate, of the output buffer NOT rotated. It is implied that the upper left coordinate of the output surface is always [0,0]. When rotating the frame buffer, PXP will automatically swap the X/Y, or WIDTH/HEIGHT, to accommodate the rotated size.

This register sets the size of the output frame buffer in pixels, not blocks. The frame buffer need not be a multiple of NxN pixels. Partial blocks will be written for output frame buffer sizes that are not divisible by N pixels in either dimension.

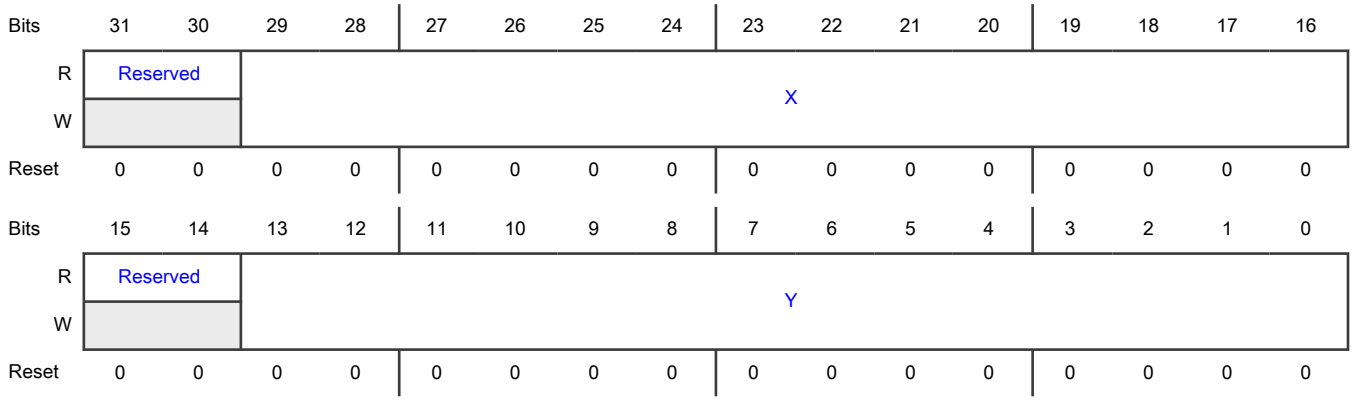
EXAMPLE

```
REG_OUT_LRC[X]=319; // set width of output frame buffer to 320 pixels
REG_OUT_LRC[Y]=243; // set height of output frame buffer to 244 pixels which is not
divisible by block size N
REG_OUT_LRC_WR( BF_PXP_OUT_LRC_X(319) | BF_PXP_OUT_LRC_Y(243) );
```

Offset

Register	Offset
OUT_LRC	60h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-16 X	X Indicates number of horizontal PIXELS in the output surface (non-rotated). The output buffer pixel width minus 1 should be programmed. The image size is not required to be a multiple of 8 pixels. PXP will clip the pixel output at this boundary.
15-14 —	Reserved
13-0 Y	Y Indicates the number of vertical PIXELS in the output surface (non-rotated). The output buffer pixel height minus 1 should be programmed. The image size is not required to be a multiple of 8 pixels. PXP will clip the pixel output at this boundary.

56.6.1.9 Processed Surface Upper Left Coordinate Register (OUT_PS_ULC)

This register contains the upper left pixel coordinate for the Processed Surface in the OUTPUT buffer.

This register contains the upper left coordinate of the Processed Surface in the output frame buffer (in pixels). Values that are within the REG_OUT_LRC[X,Y] extents are valid. The lowest valid value for these fields is 0,0. If the value of the REG_OUT_PS_ULC is greater than the REG_OUT_LRC, then no PS pixels will be fetched from memory, but only REG_PS_BACKGROUND pixels will be processed by the PS engine. Pixel locations that are greater than or equal to the PS upper left coordinates, less than or equal to the PS lower right coordinates, and within the REG_OUT_LRC extents will use the PS to render pixels into the output buffer.

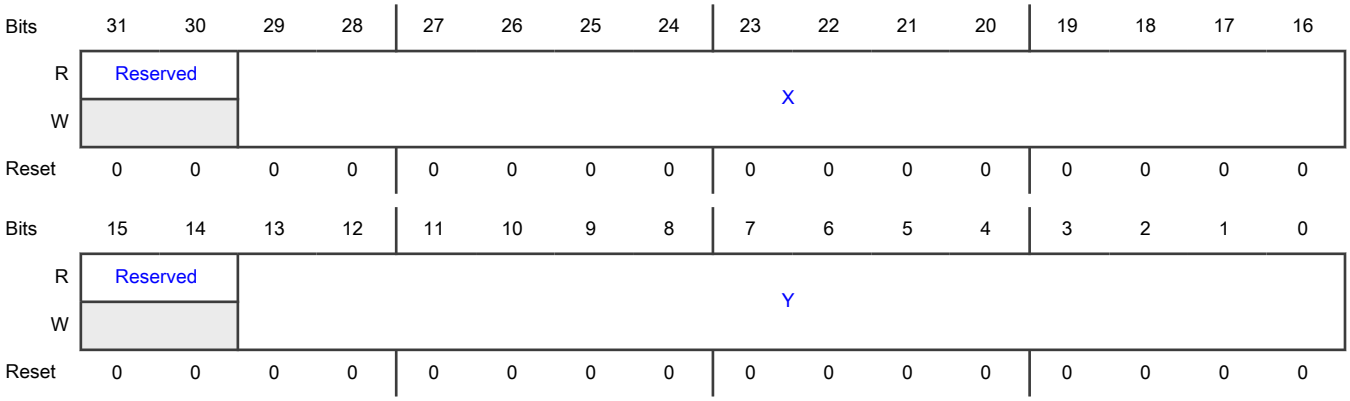
EXAMPLE

```
REG_OUT_PS_ULC_WR(0,0x0002_0002); // Processed Surface upper left coordinate at (X,Y) = 2,2.
The PS surface will not effect pixels in the first and second row and column of the output
buffer.
```

Offset

Register	Offset
OUT_PS_ULC	70h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-16 X	X This field indicates the upper left X-coordinate (in pixels) of the processed surface (PS) in the output buffer.
15-14 —	Reserved
13-0 Y	Y This field indicates the upper left Y-coordinate (in pixels) of the processed surface in the output buffer.

56.6.1.10 Processed Surface Lower Right Coordinate Register (OUT_PS_LRC)

This register contains the lower right extent for the Processed Surface in the OUTPUT buffer.

This register contains the lower right coordinate of the Processed Surface in the output frame buffer (in pixels). Values that are within the REG_OUT_LRC[X,Y] extents are valid. The lowest valid value for these fields is 0,0. Pixel locations that are greater than or equal to the PS upper left coordinates, less than or equal to the PS lower right coordinates, and within the REG_OUT_LRC extents will use the PS to render pixels into the output buffer.

EXAMPLE

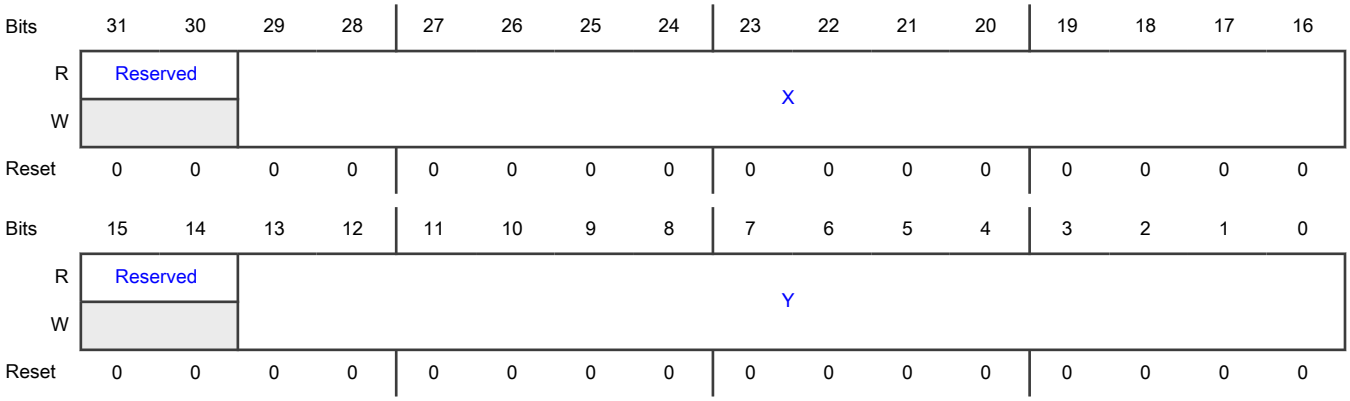
```
REG_OUT_PS_ULC_WR(0,0x03FF_03FF); // With this UL/LR pair of pixel coordinates, only one
pixel at OUT[X,Y]=1023,1023 will use the PS to contribute to its value.

REG_OUT_PS_LRC_WR(0,0x03FF_03FF);
```

Offset

Register	Offset
OUT_PS_LRC	80h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-16 X	X This field indicates the lower right X-coordinate (in pixels) of the processed surface (PS) in the output frame buffer.
15-14 —	Reserved
13-0 Y	Y This field indicates the lower right Y-coordinate (in pixels) of the processed surface in the output frame buffer.

56.6.1.11 Alpha Surface Upper Left Coordinate Register (OUT_AS_ULC)

This register contains the upper left location for the Alpha Surface in the output buffer.

This register contains the upper left coordinate of AS in the output frame buffer (in pixels). Values that are within the REG_OUT_LRC[X,Y] extents are valid. The lowest valid value for these fields is 0,0. Pixel locations that are greater than or equal to the upper left coordinates will use the AS to render pixels in the output buffer.

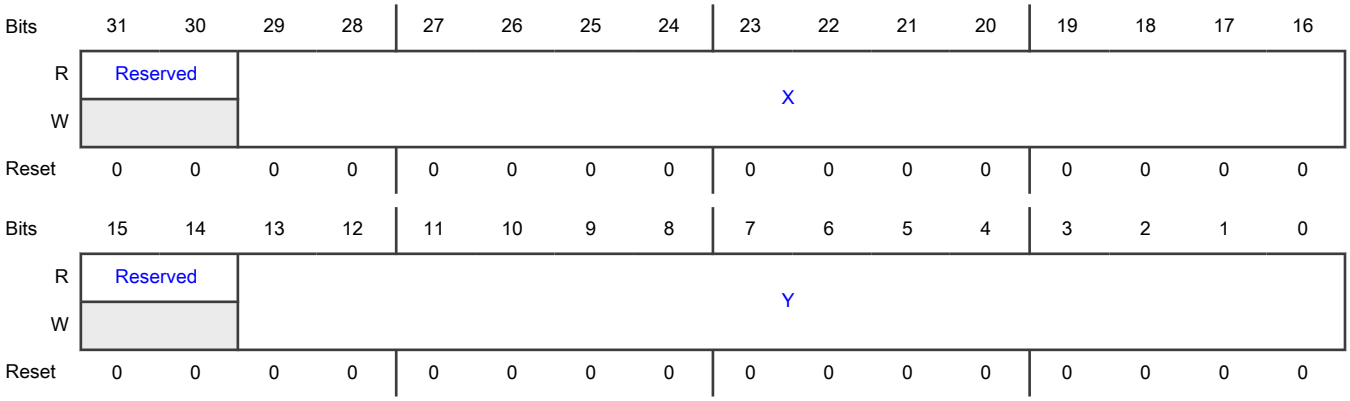
EXAMPLE

```
REG_OUT_AS_ULC_WR(0,0x0001_0001); // Alpha Surface upper left coordinate at (X,Y) = 1,1. The
AS surface will not effect pixels in the first row or first column of the output buffer.
```

Offset

Register	Offset
OUT_AS_ULC	90h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-16 X	X This field indicates the upper left X-coordinate (in pixels) of the alpha surface (AS) in the output frame buffer.
15-14 —	Reserved
13-0 Y	Y This field indicates the upper left Y-coordinate (in pixels) of the alpha surface in the output frame buffer.

56.6.1.12 Alpha Surface Lower Right Coordinate Register (OUT_AS_LRC)

This register contains the lower right extent for Alpha Surface in the output buffer.

This register contains the lower right coordinate of AS in the output frame buffer (in pixels). Values that are within the REG_OUT_LRC[X,Y] extents are valid. The lowest valid value for these fields is 0,0. Pixel locations that are less than or equal to the lower right coordinates will use the AS to render pixels in the output buffer.

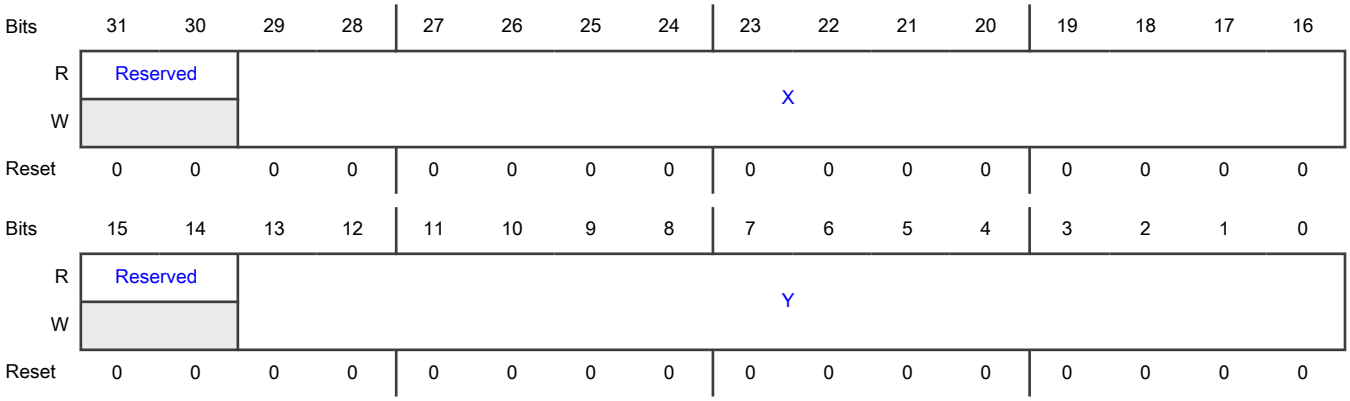
EXAMPLE

```
REG_AS_LRC_WR(0,0x03FF_03FF); // Alpha Surface lower right coordinate at (X,Y) = 1023,1023.
```

Offset

Register	Offset
OUT_AS_LRC	A0h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-16 X	X This field indicates the lower right X-coordinate (in pixels) of the alpha surface (AS) in the output frame buffer.
15-14 —	Reserved
13-0 Y	Y This field indicates the lower right Y-coordinate (in pixels) of the alpha surface in the output frame buffer.

56.6.1.13 Processed Surface (PS) Control Register (PS_CTRL)

The PS_CTRL register contains controls for the Processed Surface Buffer.

PS_CTRL: 0x0B0

PS_CTRL_SET: 0x0b4

PS_CTRL_CLR: 0x0B8

PS_CTRL_TOG: 0x0BC

Control register to determine PS buffer processing.

EXAMPLE

```
REG_CTRL_SET(BM_PXP_CTRL_SFTRST);
```

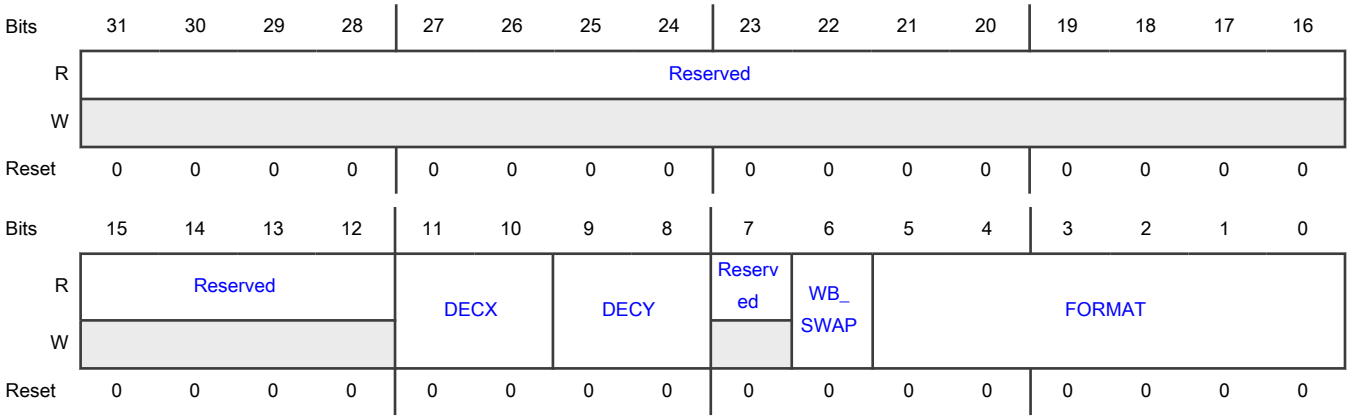
```
REG_CTRL_CLR(BM_PXP_CTRL_SFTRST | BM_PXP_CTRL_CLKGATE);
```

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
PS_CTRL	B0h	Processed Surface (PS) Control Register
PS_CTRL_SET	B4h	Writing 1 to a bit in this register ensures that the corresponding bit in PS_CTRL is 1
PS_CTRL_CLR	B8h	Writing 1 to a bit in this register ensures that the corresponding bit in PS_CTRL is 0
PS_CTRL_TOG	BCh	Writing 1 to a bit in this register inverts the value of the corresponding bit in PS_CTRL

Diagram



Fields

Field	Description
31-12 —	Reserved
11-10 DECX	DECX Horizontal pre decimation filter control. 00 - DISABLE : Disable pre-decimation filter. 01 - DECX2 : Decimate PS by 2. 10 - DECX4 : Decimate PS by 4. 11 - DECX8 : Decimate PS by 8.
9-8 DECY	DECY Vertical pre decimation filter control.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	00 - DISABLE : Disable pre-decimation filter. 01 - DECY2 : Decimate PS by 2. 10 - DECY4 : Decimate PS by 4. 11 - DECY8 : Decimate PS by 8.
7 —	Reserved
6 WB_SWAP	WB_SWAP Swap bytes in words. For each 16 bit word, the two bytes will be swapped.
5-0 FORMAT	FORMAT PS buffer format. To select between YUV and YCbCr formats, see CSC1_COEF0[YCBCR_MODE]. 00_0100 - RGB888 : 32-bit RGB format with blue in LSB (unpacked 24-bit pixel in 32 bit DWORD) 00_1100 - RGB555 : 16-bit RGB format with blue in LSB 00_1101 - RGB444 : 16-bit RGB format with blue in LSB 00_1110 - RGB565 : 16-bit RGB format with blue in LSB 01_0000 - YUV1P444 : 32-bit pixels (1-plane XYUV unpacked) 01_0010 - UYVY1P422 : 16-bit pixels (1-plane U0,Y0,V0,Y1 interleaved bytes) 01_0011 - VYUY1P422 : 16-bit pixels (1-plane V0,Y0,U0,Y1 interleaved bytes) 01_0100 - Y8 : 8-bit monochrome pixels (1-plane Y luma output) 01_0101 - Y4 : 4-bit monochrome pixels (1-plane Y luma, 4 bit truncation) 01_1000 - YUV2P422 : 16-bit pixels (2-plane UV interleaved bytes) 01_1001 - YUV2P420 : 16-bit pixels (2-plane UV) 01_1010 - YVU2P422 : 16-bit pixels (2-plane VU interleaved bytes) 01_1011 - YVU2P420 : 16-bit pixels (2-plane VU) 01_1110 - YUV422 : 16-bit pixels (3-plane format) 01_1111 - YUV420 : 16-bit pixels (3-plane format)

56.6.1.14 PS Input Buffer Address Register (PS_BUF)

PS Input Buffer Address. This should be programmed to the starting address of the RGB data or Y (luma) data for the PS plane.

This register contains the pointer to the Luma/RGB buffer. If the application requires an offset into the PS buffer, then this address can be set so that the desired offset is achieved. Any byte address is valid. For best performance, 64B alignment is recommended.

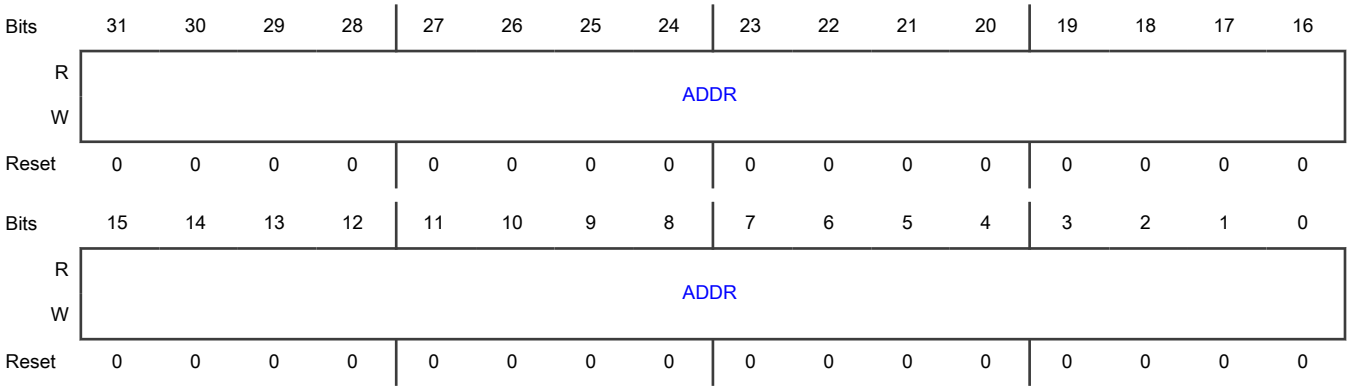
EXAMPLE

```
REG_PS_BUF_WR(image_rgb); // RGB image
REG_PS_BUF_WR(image_y); // Y (luma) image data
REG_PS_UBUF_WR(image_u); // U (Cb) image data
REG_PS_VBUF_WR(image_v); // V (Cr) image data
```

Offset

Register	Offset
PS_BUF	C0h

Diagram



Fields

Field	Description
31-0 ADDR	ADDR Address pointer for the PS RGB or Y (luma) input buffer.

56.6.1.15 PS U/Cb or 2 Plane UV Input Buffer Address Register (PS_UBUF)

PS Chroma (U/Cb/UV) Input Buffer Address. This register points to the beginning of the PS U/Cb input buffer. In two plane operation, this register points to the beginning of the PS UV chroma input buffer.

This register contains the pointer to the Chroma U/Cb or 2 plane UV buffer. This register is unused when processing 1-plane buffer formats. If the application requires an offset into the PS buffer, then this address can be set so that the desired offset is achieved. Any byte address is valid. For best performance, 64B alignment is recommended.

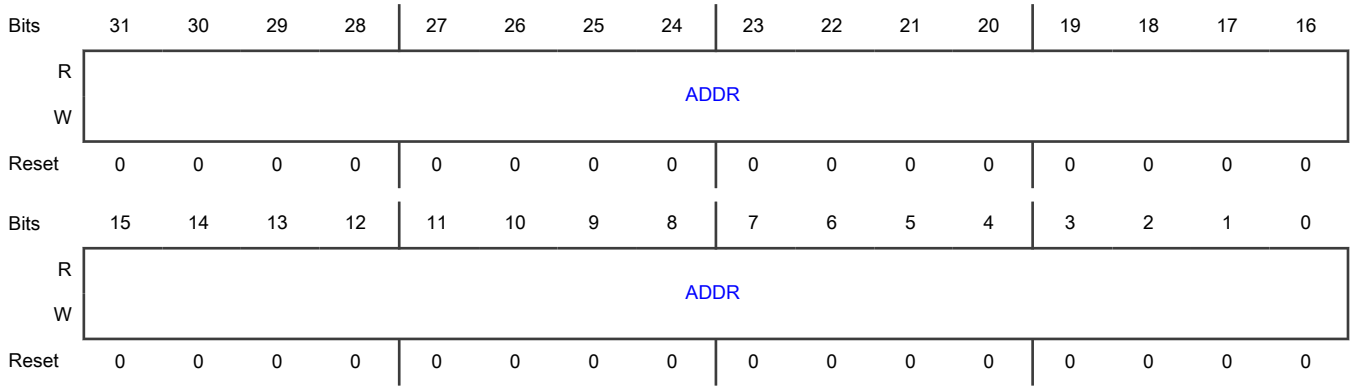
EXAMPLE

```
REG_PS_BUF_WR(image_y); // Y (luma) image data
REG_PS_UBUF_WR(image_u); // U (Cb) image data
REG_PS_VBUF_WR(image_v); // V (Cr) image data
```

Offset

Register	Offset
PS_UBUF	D0h

Diagram



Fields

Field	Description
31-0 ADDR	ADDR Address pointer for the PS U/Cb or 2 plane UV Chroma input buffer.

56.6.1.16 PS V/Cr Input Buffer Address Register (PS_VBUF)

PS Chroma (V/Cr) Input Buffer Address. This register points to the beginning of the PS V/Cr input buffer. In one or two plane operation, this register is not used. In monochrome modes Y8 and Y4, the low 16 bits are used as the U/V data in the datapath instead of sourcing U/V data from external buffers. In this case, it represents a fixed value 0x8080 for U/V data.

This register contains the pointer to the Chroma V/Cr buffer. For Y8/Y4 modes, the low 16 bits are used as the monochrome U and V values in the data path. Bits [15:8] represent the U data byte, and bits [7:0] represent the V data byte. Other than with Y8/Y4 input buffer formats, this register is unused when processing 1 or 2-plane buffer formats. If the application requires an offset into the PS buffer, then this address can be set so that the desired offset is achieved. Any byte address is valid. For best performance, 64B alignment is recommended.

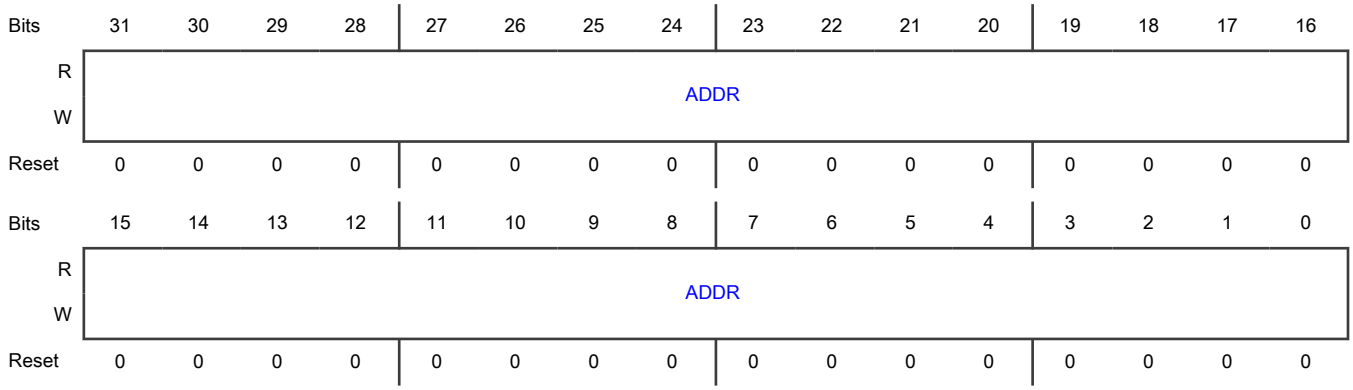
EXAMPLE

```
REG_PS_BUF_WR(image_y); // Y (luma) image data
REG_PS_UBUF_WR(image_u); // U (Cb) image data
REG_PS_VBUF_WR(image_v); // V (Cr) image data
```

Offset

Register	Offset
PS_VBUF	E0h

Diagram



Fields

Field	Description
31-0 ADDR	ADDR Address pointer for the PS V/Cr Chroma input buffer.

56.6.1.17 Processed Surface Pitch Register (PS_PITCH)

This register contains the processed surface pitch in bytes.

Any byte value will indicate the vertical pitch of the PS source frame buffer. This value will be used in PS pixel address calculations. This value has no relation to the UL and LR registers. It specifies how many bytes are between two vertically adjacent pixels in the input PS surface. For multi-plane formats, the Y buffer pitch should be programmed. For 2-plane YUV422, the UV pitch is the same as the Y pitch. For 3-plane YUV422, the U and V pitch is 1/2 the Y pitch. For 2-plane YUV420, the UV pitch is 1/2 the Y pitch. For 3-plane YUV420, the U and V pitch is 1/4 the Y pitch. All source buffers should comply with these U and V resolution reductions with respect to their Y source buffers.

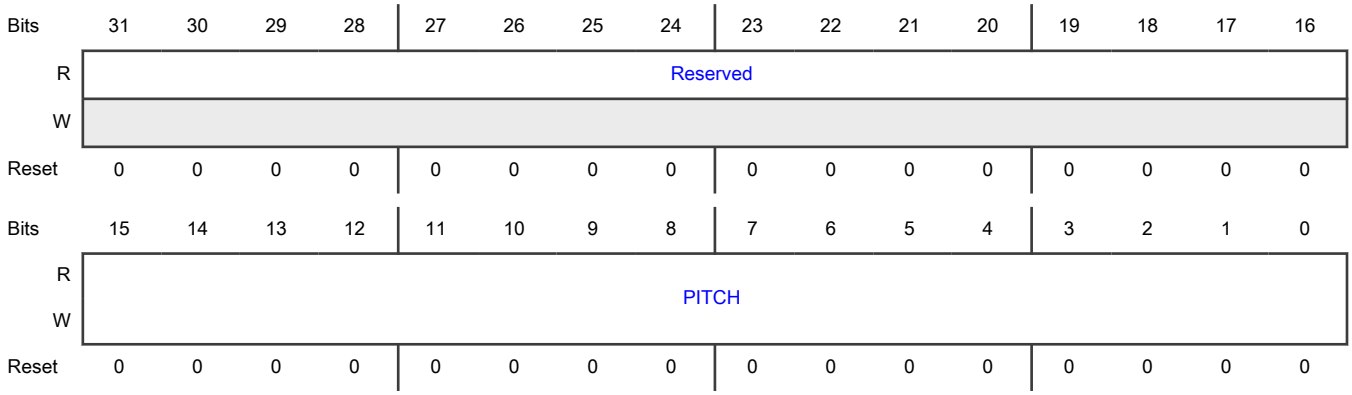
EXAMPLE

```
REG_PS_PITCH_WR( 64 * 4 ); // The output buffer pitch is 64 pixels times 32 bits per pixel
```

Offset

Register	Offset
PS_PITCH	F0h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 PITCH	PITCH Indicates the number of bytes in memory between two vertically adjacent pixels.

56.6.1.18 PS Background Color Register (PS_BACKGROUND_0)

PS Background Pixel Color. This register provides a pixel value used when processing pixels outside of the region specified by the PS Coordinate registers. This value can effectively be used to set the color of the letterboxing region around the PS image. This value is used by the Alpha A block.

This register contains a pixel value to be used for any PS pixels that fall outside the PS extents. This is effectively a background or letterbox color. The CSC1 control and datapath pixel format should be considered when selecting the background color.

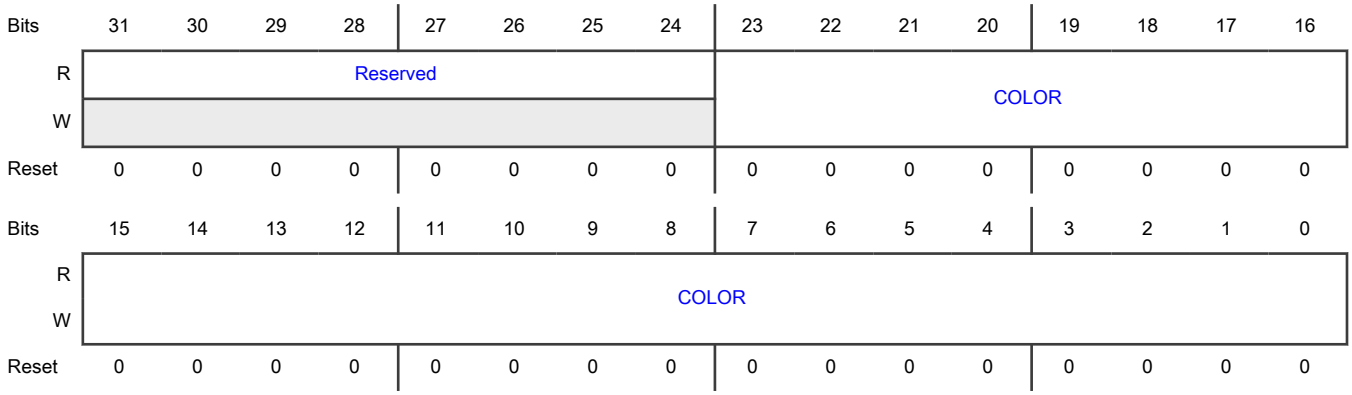
EXAMPLE

```
REG_PS_BACKGROUND_WR(0x00000000); // letterbox is black
REG_PS_BACKGROUND_WR(0x00800000); // letterbox is dark red
REG_PS_BACKGROUND_WR(0x00008000); // letterbox is dark green
REG_PS_BACKGROUND_WR(0x00000080); // letterbox is dark blue
```

Offset

Register	Offset
PS_BACKGROUND_0	100h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 COLOR	COLOR Background color (in 24bpp format) for any pixels not within the buffer range specified by the PS ULC/LRC.

56.6.1.19 PS Scale Factor Register (PS_SCALE)

PS Scale Factor. This register provides the scale factor for the PS buffer.

The maximum down scaling factor is 1/2 such that the output image in either axis is 1/2 the size of the source. The maximum up scaling factor is 2^12 for either axis. The reciprocal of the scale factor should be loaded into this register. To reduce the PS buffer by a factor of two in the output frame buffer, a value of 10.0000_0000_0000 should be loaded into this register. To scale up by a factor of 4, the value of 1/4, or 00.0100_0000_0000, should be loaded into this register. To scale up by 8/5, the value of 00.1010_0000_0000 should be loaded.

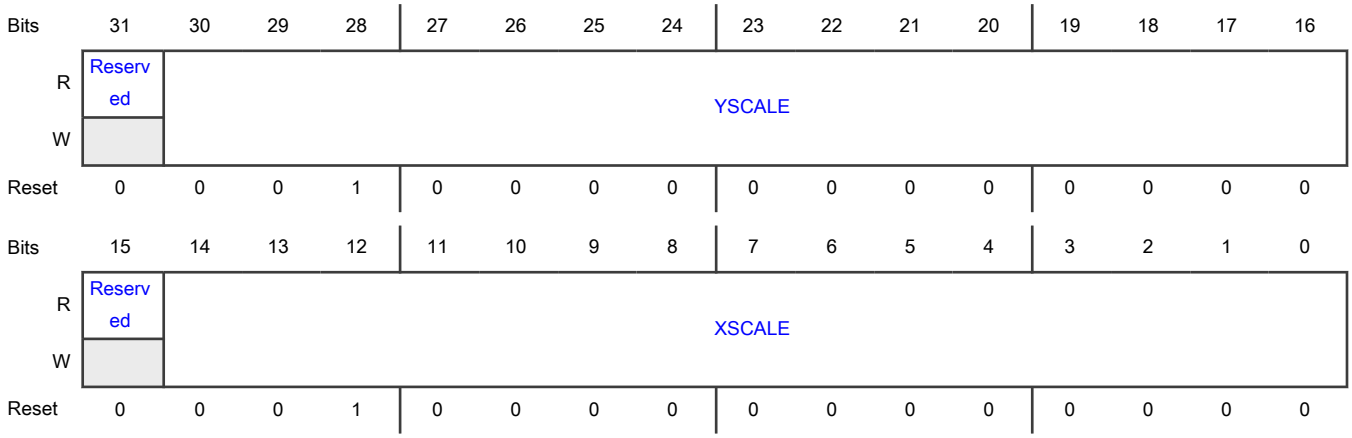
EXAMPLE

```
REG_PS_SCALE_WR(0x10001000); // 1:1 scaling (0x1.000)
REG_PS_SCALE_WR(0x08000800); // 2x scaling (0x0.800)
REG_PS_SCALE_WR(0x20002000); // 1/2x scaling (0x2.000)
```

Offset

Register	Offset
PS_SCALE	110h

Diagram



Fields

Field	Description
31 —	Reserved
30-16 YSCALE	YSCALE This is a two bit integer and 12 bit fractional representation (##.####_####_####) of the Y scaling factor for the PS source buffer. The maximum value programmed should be 2 since scaling down by a factor greater than 2 is not supported with the bilinear filter. Decimation and the bilinear filter should be used together to achieve scaling by more than a factor of 2.
15 —	Reserved
14-0 XSCALE	XSCALE This is a two bit integer and 12 bit fractional representation (##.####_####_####) of the X scaling factor for the PS source buffer. The maximum value programmed should be 2 since scaling down by a factor greater than 2 is not supported with the bilinear filter. Decimation and the bilinear filter should be used together to achieve scaling by more than a factor of 2.

56.6.1.20 PS Scale Offset Register (PS_OFFSET)

PS Scale Offset. This register provides the initial scale offset for the PS buffer.

The X and Y offset provides the ability to access the source image with a per sub-pixel granularity. This provides the capability to use all source pixels to effect the output PS image. The fixed offset values can be used for sub-pixel adjustments in the bilinear scaling filter. For example, when scaling an image down by a factor of 2, an initial offset of 0x0 would result in sub-sampling every other pixel. If a fixed offset of 0x800 (1/2), all pixels are used in scaling the final output pixel value. In this case, the first output pixel would be the sum of (1/2*P0) + (1/2*P1). This fixed offset is applied after the decimation filter stage, and before the bilinear filter stage.

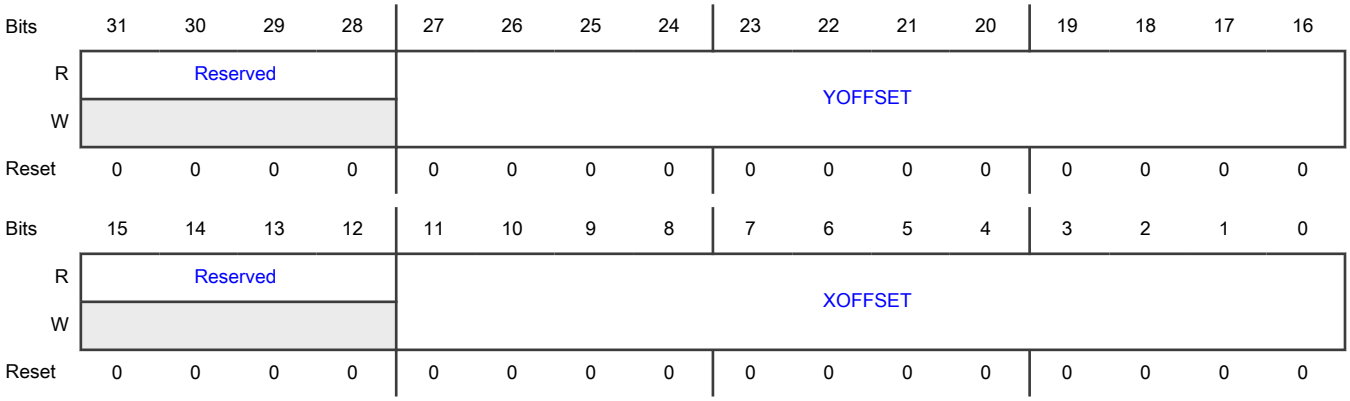
EXAMPLE

```
REG_PS_SCALE_WR(0x2000_2000); // 1/2x scaling (0x2.000)
REG_PS_OFFSET_WR(0x0800_0800); // half-pixel offset in both X and Y to ensure averaging versus pixel decimation
```

Offset

Register	Offset
PS_OFFSET	120h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 YOFFSET	YOFFSET This is a 12 bit fractional representation (0.####_####_####) of the Y scaling offset. This represents a fixed pixel offset which gets added to the scaled address to determine source data for the scaling engine.
15-12 —	Reserved
11-0 XOFFSET	XOFFSET This is a 12 bit fractional representation (0.####_####_####) of the X scaling offset. This represents a fixed pixel offset which gets added to the scaled address to determine source data for the scaling engine.

56.6.1.21 PS Color Key Low Register (PS_CLRKEYLOW_0)

This register contains the color key low value for the PS buffer. This value is used by the Alpha A block.

When processing an image, if PXP finds a pixel in the PS buffer with a color that falls in the range between REG_PS_CLRKEYLOW and REG_PS_CLRKEYHIGH, it will insert the pixel from the AS channel. If the current AS pixel is letterboxed or if the AS also matches its colorkey range, the REG_PS_BACKGROUND color is passed down the pixel pipeline.

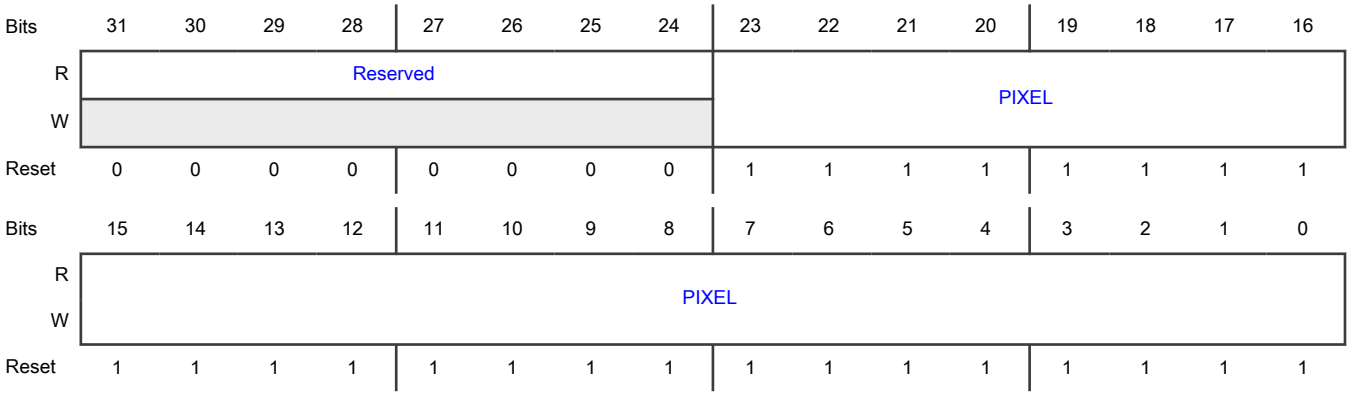
EXAMPLE

```
REG_PS_CLRKEYLOW_WR (0x008000); // colorkey values between medium green and
REG_PS_CLRKEYHIGH_WR(0x00FF00); // light green
```

Offset

Register	Offset
PS_CLRKEYLOW_0	130h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 PIXEL	PIXEL Low range of color key applied to PS buffer. To disable PS colorkeying, set the low colorkey to 0xFFFFFF and the high colorkey to 0x000000.

56.6.1.22 PS Color Key High Register (PS_CLRKEYHIGH_0)

This register contains the color key high value for the PS buffer. This value is used by the Alpha A block.

When processing an image, if PXP finds a pixel in the PS buffer with a color that falls in the range between REG_PS_CLRKEYLOW and REG_PS_CLRKEYHIGH, it will insert the pixel from the AS channel. If the current AS pixel is letterboxed or if the AS also matches its colorkey range, the REG_PS_BACKGROUND color is passed down the pixel pipeline.

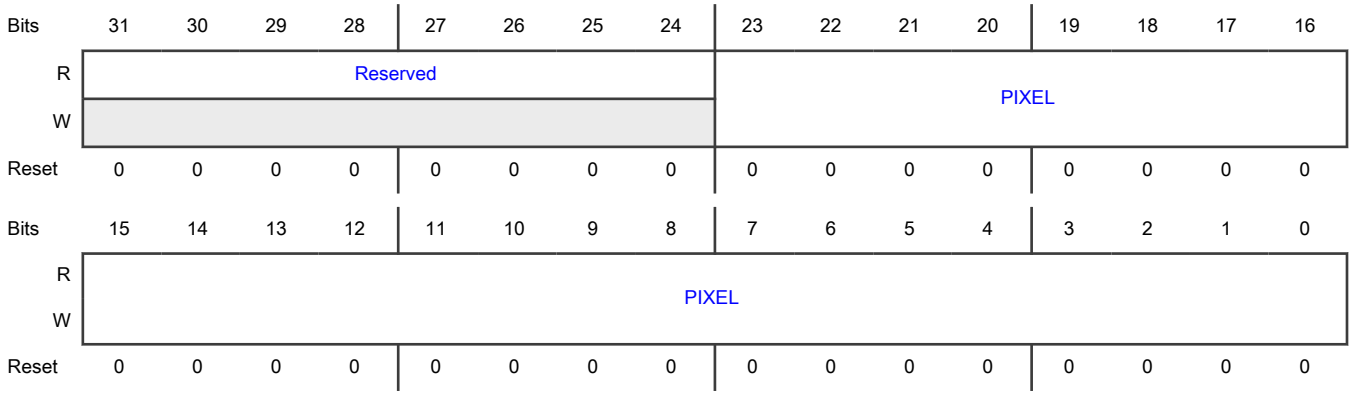
EXAMPLE

```
REG_PS_CLRKEYLOW_WR (0x008000); // colorkey values between medium green and
REG_PS_CLRKEYHIGH_WR(0x00FF00); // light green
```

Offset

Register	Offset
PS_CLRKEYHIGH_0	140h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 PIXEL	PIXEL High range of color key applied to PS buffer. To disable PS colorkeying, set the low colorkey to 0xFFFFFF and the high colorkey to 0x000000.

56.6.1.23 Alpha Surface Control Register (AS_CTRL)

This register contains buffer control for the Alpha Surface 0 input buffer.

Control register to determine AS buffer processing.

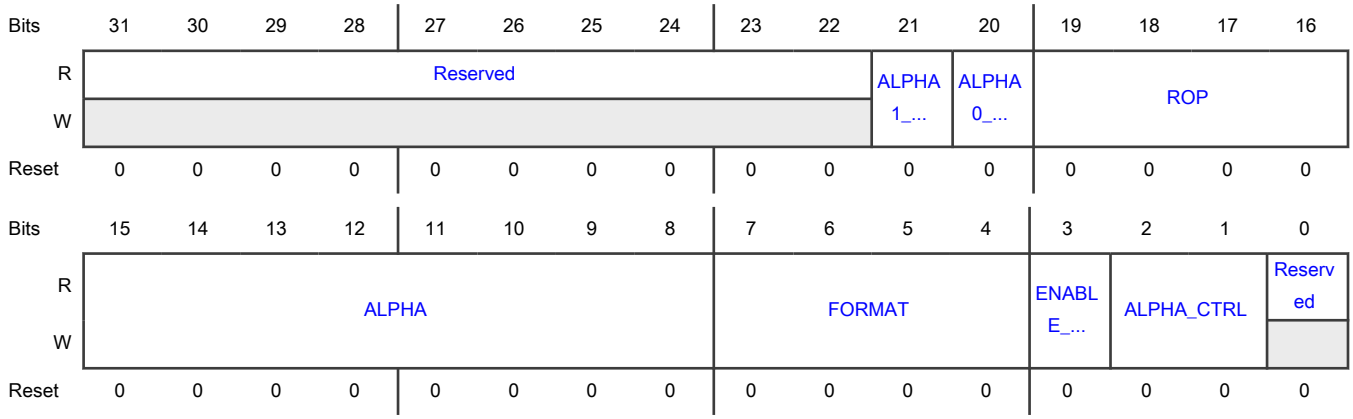
EXAMPLE

```
u32 asparam; asparam = BF_PXP_ASPARAM_ENABLE (1);
asparam |= BF_PXP_ASPARAM_ALPHA_CTRL(BV_PXP_ASPARAM_ALPHA_CTRL__ROPs);
asparam |= BF_PXP_ASPARAM_FORMAT (BV_PXP_ASPARAM_FORMAT__ARGB8888);
asparam |= BF_PXP_ASPARAM_ROP (BV_PXP_ASPARAM_ROP__XORAS);
REG_ASPARAM_WR(0,asparam); // enable alpha surface to perform XOR ROP using RGB8888 AS pixel format
```

Offset

Register	Offset
AS_CTRL	150h

Diagram



Fields

Field	Description
31-22 —	Reserved
21 ALPHA1_INVERT	ALPHA1_INVERT Setting this bit to logic 0 will not alter the alpha1 value. A logic 1 will invert the alpha1 value and apply (1-alpha) for image composition.
20 ALPHA0_INVERT	ALPHA0_INVERT Setting this bit to logic 0 will not alter the alpha0 value. A logic 1 will invert the alpha0 value and apply (1-alpha) for image composition.
19-16 ROP	ROP Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CTRL field. 0000 - MASKAS : AS AND PS 0001 - MASKNOTAS : nAS AND PS 0010 - MASKASNOT : AS AND nPS 0011 - MERGEAS : AS OR PS 0100 - MERGENOTAS : nAS OR PS 0101 - MERGEASNOT : AS OR nPS 0110 - NOTCOPYAS : nAS 0111 - NOT : nPS 1000 - NOTMASKAS : AS NAND PS 1001 - NOTMERGEAS : AS NOR PS 1010 - XORAS : AS XOR PS 1011 - NOTXORAS : AS XNOR PS

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-8 ALPHA	ALPHA Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE values are programmed in REG_AS_CTRL[ALPHA_CTRL]. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when selected.
7-4 FORMAT	FORMAT Indicates the input buffer format for AS. 0000 - ARGB8888 : 32-bit RGB format with blue in LSB 0001 - RGBA8888 : 32-bit RGB format with alpha in LSB 0100 - RGB888 : 32-bit RGB format with blue in LSB (unpacked 24-bit pixel in 32 bit DWORD) 1000 - ARGB1555 : 16-bit RGB format with blue in LSB 1001 - ARGB4444 : 16-bit RGB format with blue in LSB 1100 - RGB555 : 16-bit RGB format with blue in LSB 1101 - RGB444 : 16-bit RGB format with blue in LSB 1110 - RGB565 : 16-bit RGB format with blue in LSB
3 ENABLE_COLORKEY	ENABLE_COLORKEY Indicates that colorkey functionality is enabled for this alpha surface. Pixels found in the alpha surface colorkey range will be displayed as transparent (the PS pixel will be used).
2-1 ALPHA_CTRL	ALPHA_CTRL Determines how the alpha value is constructed for this alpha surface. Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. 00 - Embedded : Indicates that the AS pixel alpha value will be used to blend the AS with PS. The ALPHA field is ignored. 01 - Override : Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. 10 - Multiply : Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. 11 - ROPs : Enable ROPs. The ROP field indicates an operation to be performed on the alpha surface and PS pixels.
0 —	Reserved

56.6.1.24 Alpha Surface Buffer Pointer Register (AS_BUF)

Alpha Surface 0 Buffer Address Pointer. This register points to the beginning of the Alpha Surface 0 input buffer.

This register is used to indicate the base address of the AS buffer.

EXAMPLE

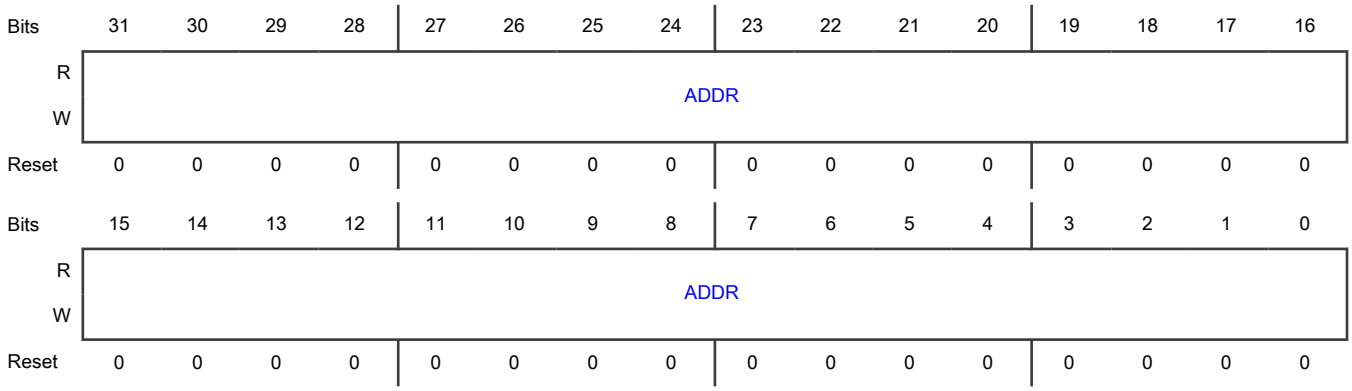
```
u32* alpha_ptr;
```

```
REG_ASn_WR(0,alpha_ptr);
```

Offset

Register	Offset
AS_BUF	160h

Diagram



Fields

Field	Description
31-0 ADDR	ADDR Address pointer for the alpha surface 0 buffer.

56.6.1.25 Alpha Surface Pitch Register (AS_PITCH)

This register contains the alpha surface pitch in bytes.

Any byte value will indicate the vertical pitch. This value will be used in AS pixel address calculations. This value has no relation to the UL and LR registers. It specifies how many bytes are between two vertically adjacent pixels in the input AS surface.

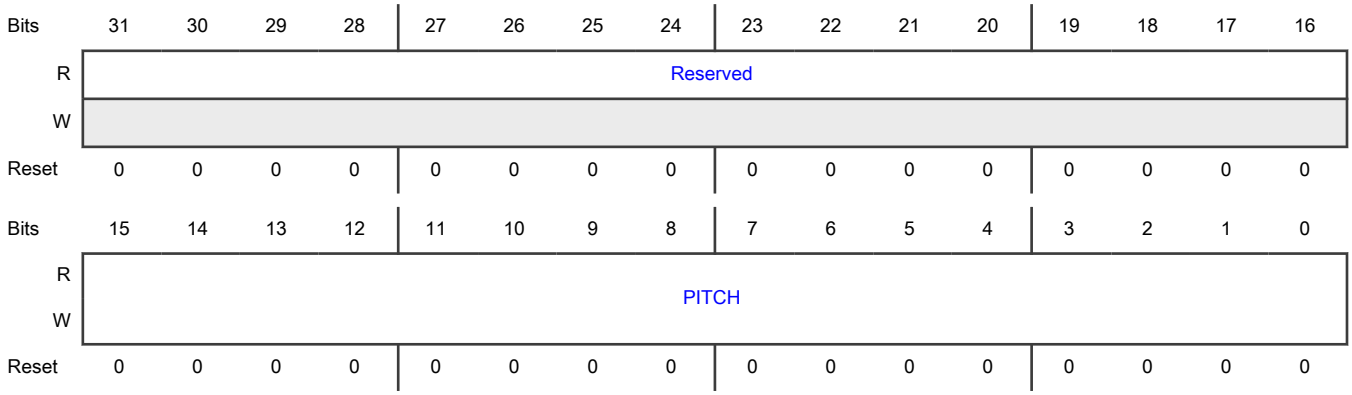
EXAMPLE

```
REG_AS_PITCH_WR( 1920 * 4 ); // The output buffer pitch is HD resolution at 32 bits per pixel
```

Offset

Register	Offset
AS_PITCH	170h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 PITCH	PITCH Indicates the number of bytes in memory between two vertically adjacent pixels.

56.6.1.26 Overlay Color Key Low Register (AS_CLRKEYLOW_0)

This register contains the color key low value for the AS buffer. This value is used by the Alpha A block.

When processing an image, if PXP finds a pixel in the current overlay image with a color that falls in the range from the ASCOLORKEYLOW to ASCOLORKEYHIGH range, it will use the PS pixel value for that location. If no PS image is present or if the PS image also matches its colorkey range, the PS background color is used. Colorkey operations are higher priority than alpha or ROP operations.

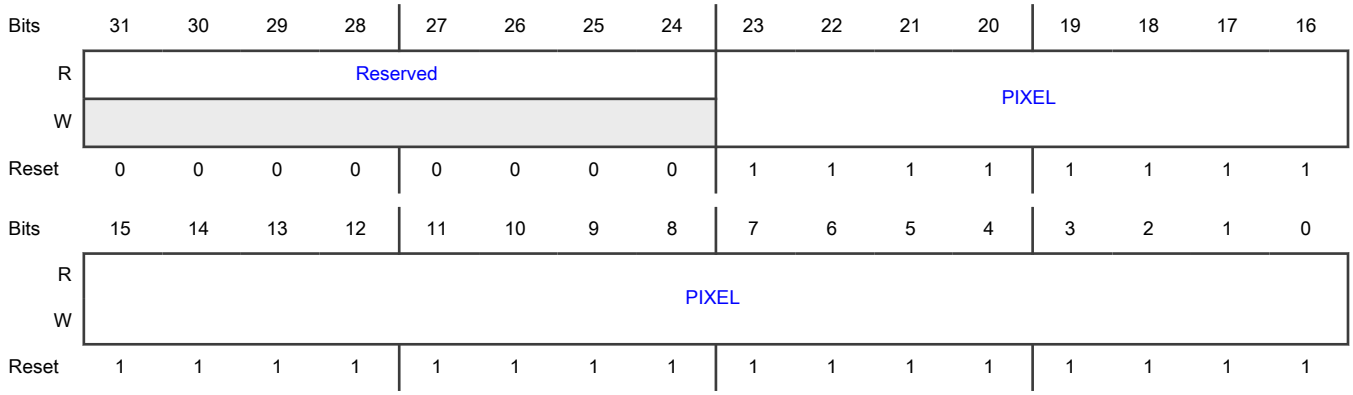
EXAMPLE

```
REG_AS_CLRKEYLOW_WR (0x000000); // colorkey values between black and
REG_AS_CLRKEYHIGH_WR(0x800000); // medium red
```

Offset

Register	Offset
AS_CLRKEYLOW_0	180h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 PIXEL	PIXEL Low range of RGB color key applied to AS buffer. Each overlay has an independent colorkey enable.

56.6.1.27 Overlay Color Key High Register (AS_CLRKEYHIGH_0)

This register contains the color key high value for the AS buffer. This value is used by the Alpha A block.

When processing an image, if PXP finds a pixel in the current overlay image with a color that falls in the range from the ASCOLORKEYLOW to ASCOLORKEYHIGH range, it will use the PS pixel value for that location. If no PS image is present or if the PS image also matches its colorkey range, the PS background color is used. Colorkey operations are higher priority than alpha or ROP operations.

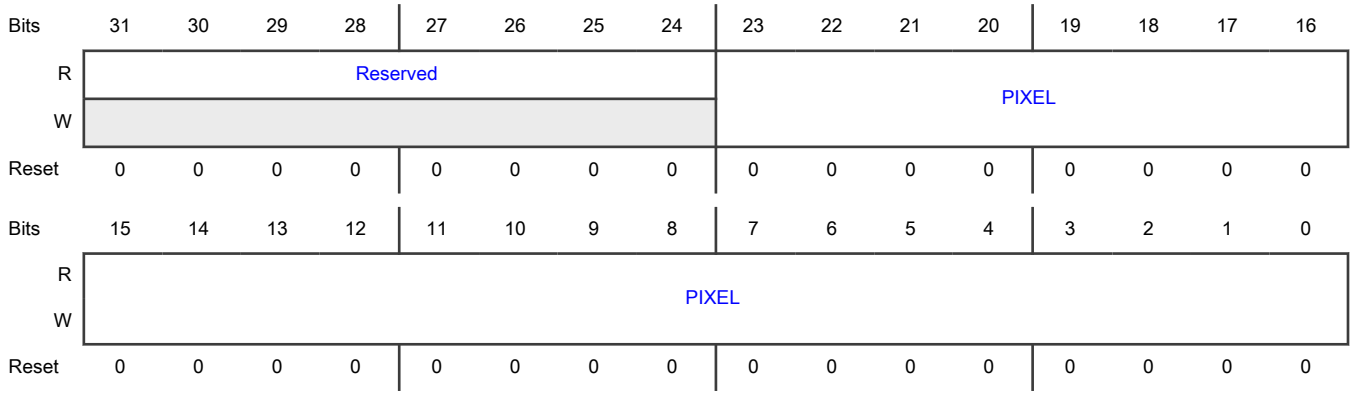
EXAMPLE

```
REG_AS_CLRKEYLOW_WR (0x000000); // colorkey values between black and
REG_AS_CLRKEYHIGH_WR(0x800000); // medium red
```

Offset

Register	Offset
AS_CLRKEYHIGH_0	190h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 PIXEL	PIXEL High range of RGB color key applied to AS buffer. Each overlay has an independent colorkey enable.

56.6.1.28 Color Space Conversion 1 Coefficient 0 Register (CSC1_COEF0)

This register contains color space conversion coefficients in two"s compliment notation.

The Coefficient 0 register contains coefficients used in the color space conversion algorithm. The Y and UV offsets are added to the source buffer to normalize them before the conversion. C0 is the coefficient that is used to multiply the luma component of the data for all three RGB components.

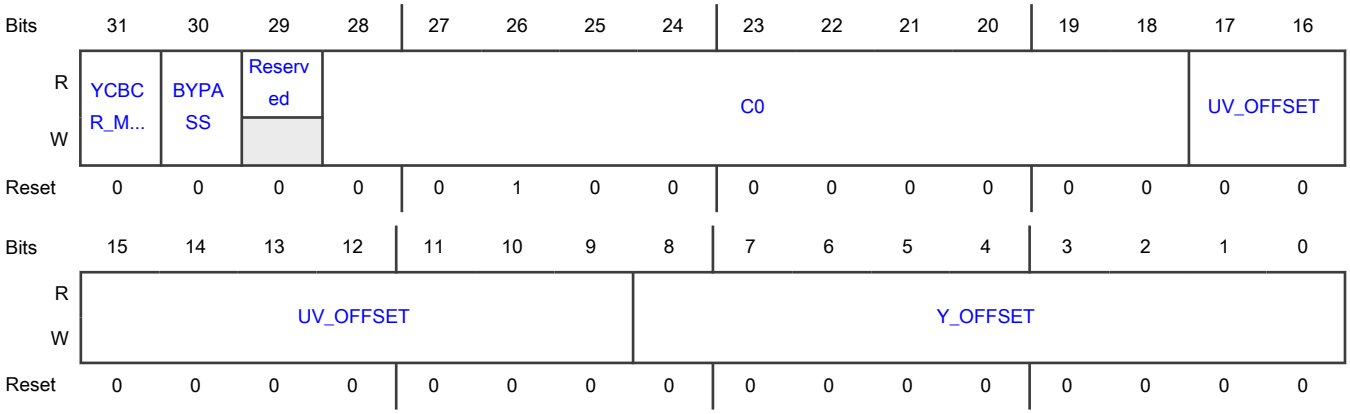
EXAMPLE

```
The equations used for Colorspace conversion are:
R = C0*(Y+YOFFSET) + C1(V+UV_OFFSET)
G = C0*(Y+YOFFSET) + C3(U+UV_OFFSET) + C2(V+UV_OFFSET)
R = C0*(Y+YOFFSET) + C4(U+UV_OFFSET)
REG_CSCCOEF0_WR(0x04030000); // YUV coefficients: C0, Yoffset, UVoffset
REG_CSCCOEF1_WR(0x01230208); // YUV coefficients: C1, C4
REG_CSCCOEF2_WR(0x076B079b); // YUV coefficients: C2, C3
```

Offset

Register	Offset
CSC1_COEF0	1A0h

Diagram



Fields

Field	Description
31 YCBCR_MODE	YCBCR_MODE Set to 1 when performing YCbCr conversion to RGB. Set to 0 when converting YUV to RGB data. This bit changes the behavior of the scaler when performing U/V scaling.
30 BYPASS	BYPASS Bypass the CSC unit in the scaling engine. When set to logic 1, bypass is enabled and the output pixels will be in the YUV/YCbCr color space. When set to logic 0, the CSC unit is enabled and the pixels will be converted based on the programmed coefficients.
29 —	Reserved
28-18 C0	C0 Two's compliment Y multiplier coefficient. YUV=0x100 (1.000) YCbCr=0x12A (1.164)
17-9 UV_OFFSET	UV_OFFSET Two's compliment phase offset implicit for CbCr data. Generally used for YCbCr to RGB conversion. YCbCr=0x180, YUV=0x000 (typically -128 or 0x180 to indicate normalized -0.5 to 0.5 range)
8-0 Y_OFFSET	Y_OFFSET Two"s compliment amplitude offset implicit in the Y data. For YUV, this is typically 0 and for YCbCr, this is typically -16 (0x1F0)

56.6.1.29 Color Space Conversion 1 Coefficient 1 Register (CSC1_COEF1)

This register contains color space conversion coefficients in two's compliment notation.

The Coefficient 1 register contains coefficients used in the color space conversion algorithm. C1 is the coefficient that is used to multiply the chroma (Cr/V) component of the data for the red component. C4 is the coefficient that is used to multiply the chroma (Cb/U) component of the data for the blue component. Both values should be coded as a two's compliment fixed point number with 8 bits right of the decimal.

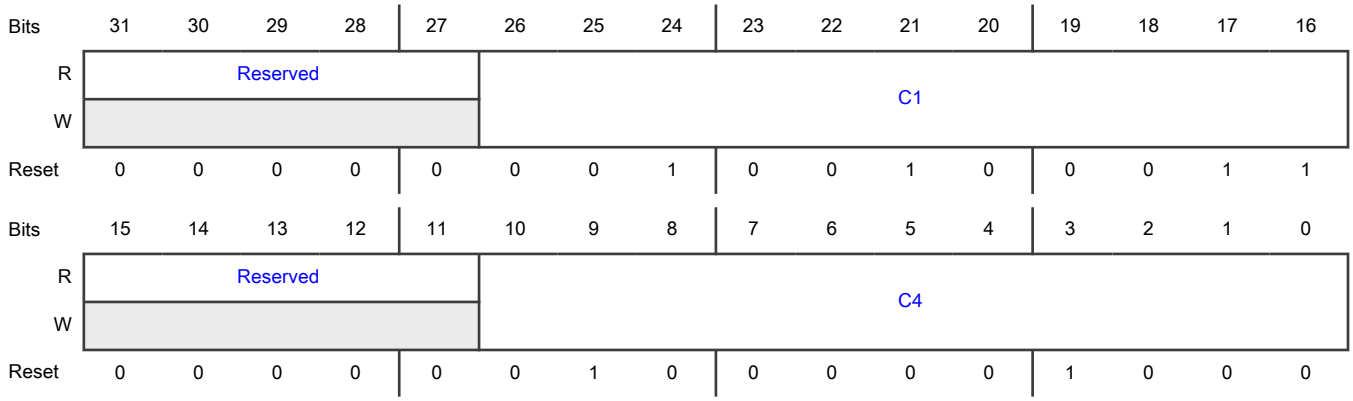
EXAMPLE

```
REG_CSCCOEF0_WR(0x04030000); // YUV coefficients: C0, Yoffset, UVoffset
REG_CSCCOEF1_WR(0x01230208); // YUV coefficients: C1, C4
REG_CSCCOEF2_WR(0x076B079b); // YUV coefficients: C2, C3
```

Offset

Register	Offset
CSC1_COEF1	1B0h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 C1	C1 Two's compliment Red V/Cr multiplier coefficient. YUV=0x123 (1.140) YCbCr=0x198 (1.596)
15-11 —	Reserved
10-0 C4	C4 Two's compliment Blue U/Cb multiplier coefficient. YUV=0x208 (2.032) YCbCr=0x204 (2.017)

56.6.1.30 Color Space Conversion 1 Coefficient 2 Register (CSC1_COEF2)

This register contains color space conversion coefficients in two's compliment notation.

The Coefficient 2 register contains coefficients used in the color space conversion algorithm. C2 is the coefficient that is used to multiply the chroma (Cr/V) component of the data for the green component. C3 is the coefficient that is used to multiply the chroma (Cb/U) component of the data for the green component. Both values should be coded as a two's compliment fixed point number with 8 bits right of the decimal.

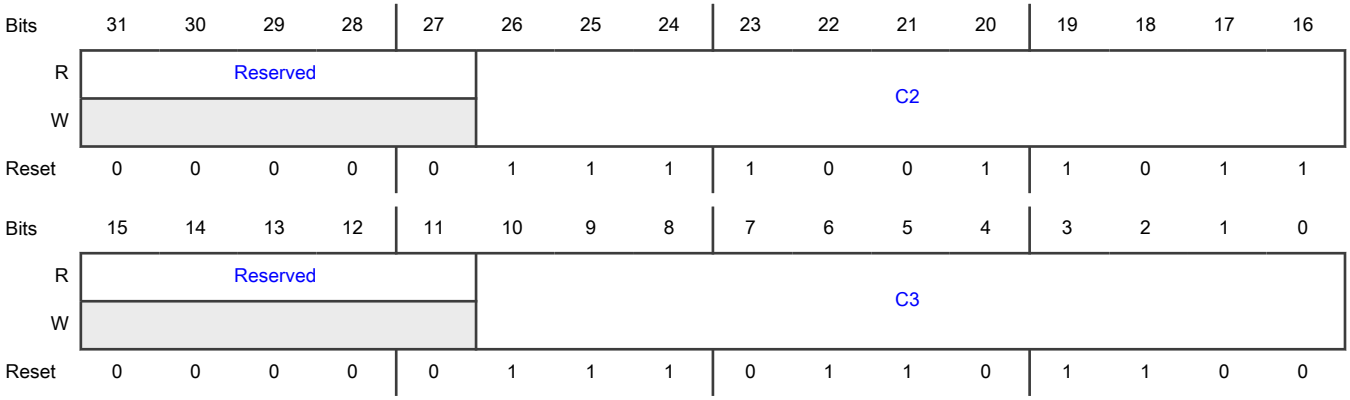
EXAMPLE

```
NOTE: The default values for the CSCCOEF2 register are incorrect. C2 should be 0x76B and C3
should be 0x79C for proper operation.
REG_CSCCOEF0_WR(0x04030000); // YUV coefficients: C0, Yoffset, UVoffset
REG_CSCCOEF1_WR(0x01230208); // YUV coefficients: C1, C4
REG_CSCCOEF2_WR(0x076B079b); // YUV coefficients: C2, C3
```

Offset

Register	Offset
CSC1_COEF2	1C0h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 C2	C2 Two's compliment Green V/Cr multiplier coefficient. YUV=0x76B (-0.581) YCbCr=0x730 (-0.813)
15-11 —	Reserved
10-0 C3	C3 Two's compliment Green U/Cb multiplier coefficient. YUV=0x79C (-0.394) YCbCr=0x79C (-0.392)

56.6.1.31 Color Space Conversion 2 Control Register (CSC2_CTRL)

This register contains the control registers to configure the CSC module.

The CSC control register will configure the CSC module to perform color space conversion between the RGB/YUV/YCbCr color spaces.

EXAMPLE

```
//Converting from YUV/YCbCr color spaces to the RGB color space uses the //following
equation structure: // // R = A1(Y-D1) + A2(U-D2) + A3(V-D3) // G = B1(Y-D1) + B2(U-D2) +
B3(V-D3) // B = C1(Y-D1) + C2(U-D2) + C3(V-D3) // //Converting from the RGB color space to
YUV/YCbCr color spaces uses the //following equation structure: // // Y = A1*R + A2*G + A3*B
+ D1 // U = B1*R + B2*G + B3*B + D2 // V = C1*R + C2*G + C3*B + D3 // //All math is signed,
so all coefficients come in as two's comp numbers //
```

Offset

Register	Offset
CSC2_CTRL	1D0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												CSC_MODE			BYPASS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31-3 —	Reserved
2-1 CSC_MODE	CSC_MODE This field controls how the CSC unit operates on pixels when the CSC is not bypassed. 00 - YUV2RGB : Convert from YUV to RGB. 01 - YCbCr2RGB : Convert from YCbCr to RGB. 10 - RGB2YUV : Convert from RGB to YUV. 11 - RGB2YCbCr : Convert from RGB to YCbCr.
0 BYPASS	BYPASS This bit controls whether the pixels entering the CSC2 unit get converted or not. When BYPASS is set, no operations occur on the pixels. When BYPASS is cleared, the selected CSC operation takes place.

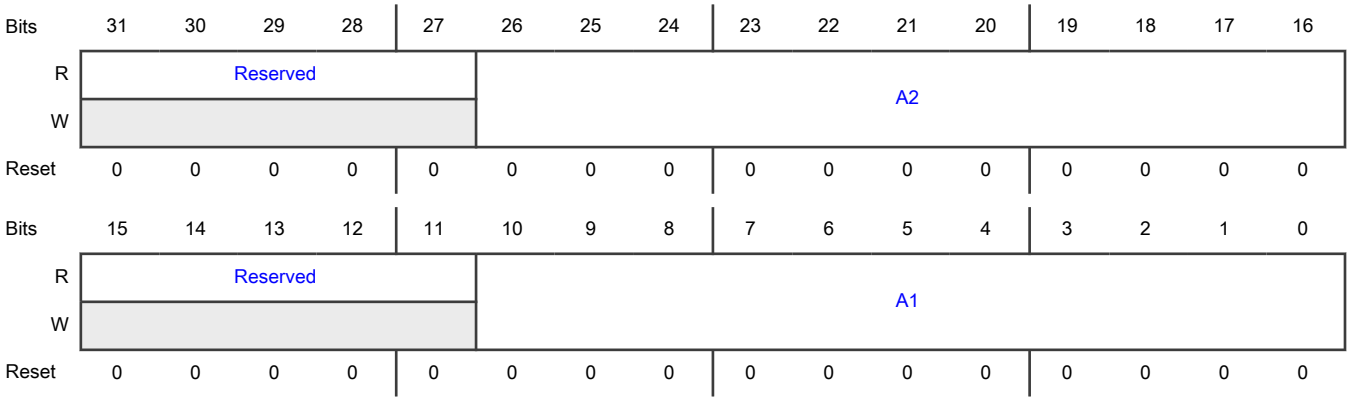
56.6.1.32 Color Space Conversion 2 Coefficient 0 Register (CSC2_COEF0)

This register contains color space conversion coefficients in two's compliment notation.

Offset

Register	Offset
CSC2_COEF0	1E0h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 A2	A2 Two's compliment coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 A1	A1 Two's compliment coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

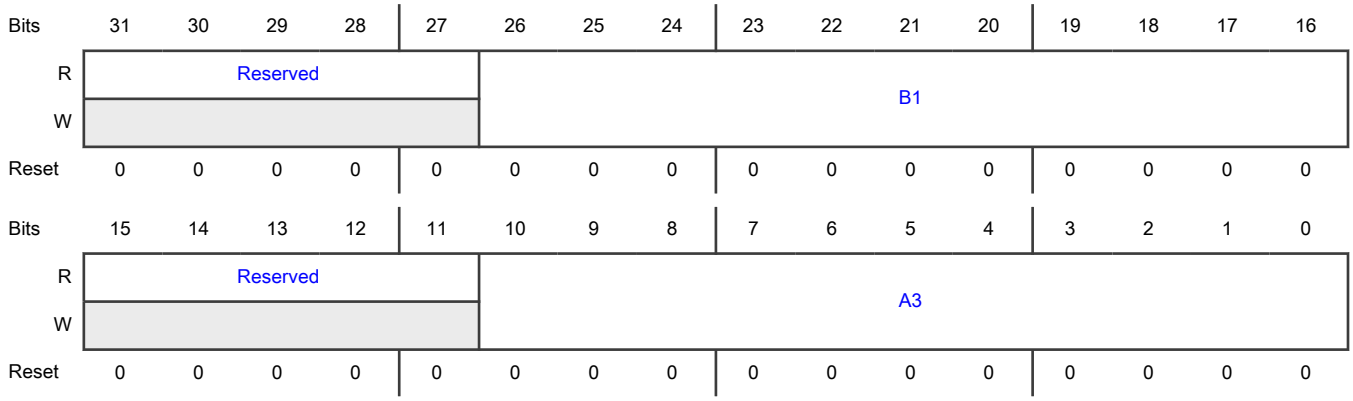
56.6.1.33 Color Space Conversion 2 Coefficient 1 Register (CSC2_COEF1)

This register contains color space conversion coefficients in two's compliment notation.

Offset

Register	Offset
CSC2_COEF1	1F0h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 B1	B1 Two's compliment coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 A3	A3 Two's compliment coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

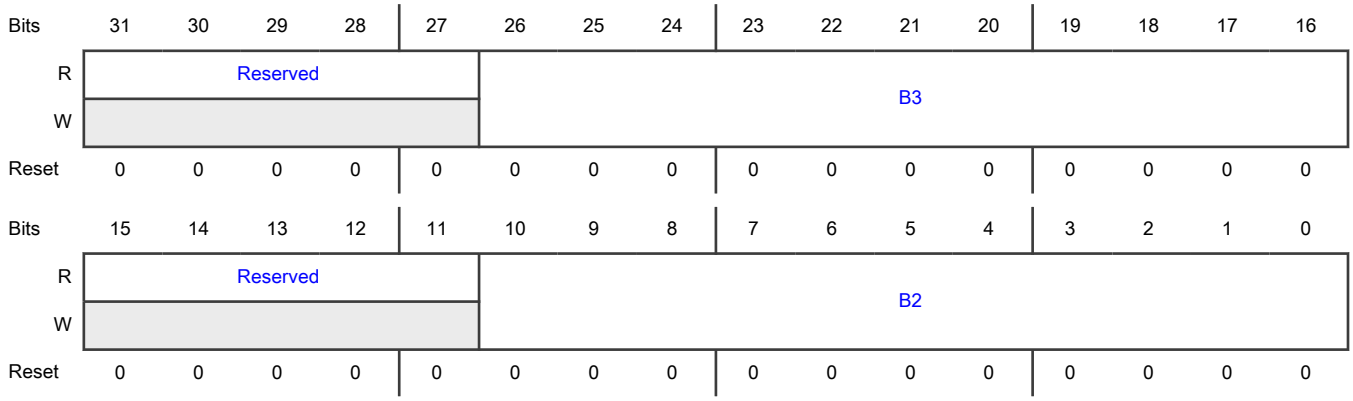
56.6.1.34 Color Space Conversion 2 Coefficient 2 Register (CSC2_COEF2)

This register contains color space conversion coefficients in two's compliment notation.

Offset

Register	Offset
CSC2_COEF2	200h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 B3	B3 Two's compliment coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 B2	B2 Two's compliment coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

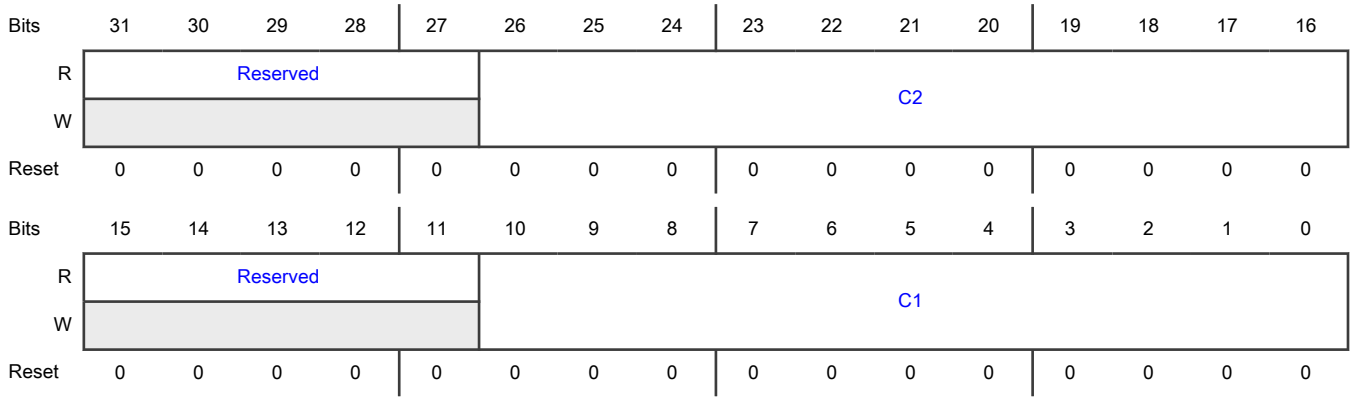
56.6.1.35 Color Space Conversion 2 Coefficient 3 Register (CSC2_COEF3)

This register contains color space conversion coefficients in two's compliment notation.

Offset

Register	Offset
CSC2_COEF3	210h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 C2	C2 Two's compliment coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 C1	C1 Two's compliment coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

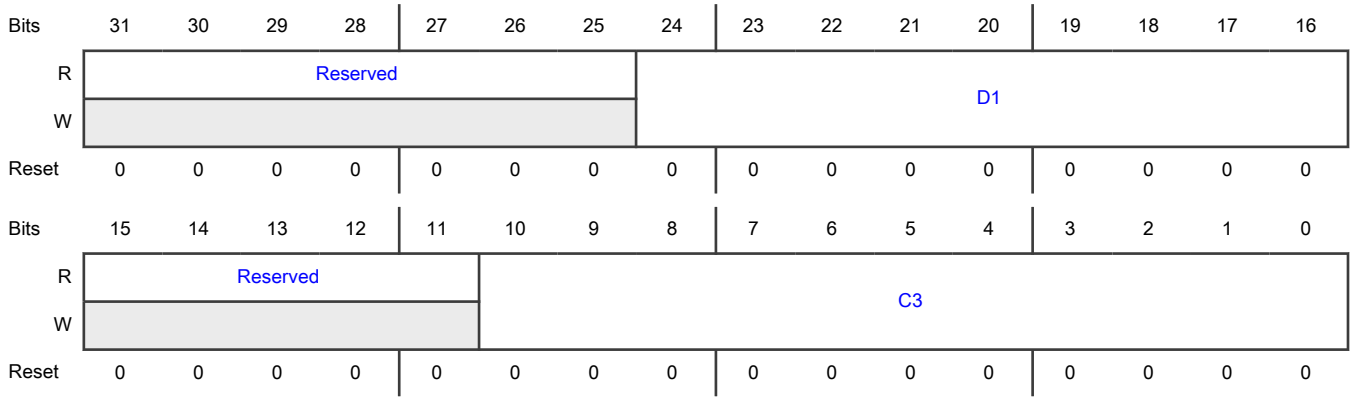
56.6.1.36 Color Space Conversion 2 Coefficient 4 Register (CSC2_COEF4)

This register contains color space conversion coefficients in two's compliment notation.

Offset

Register	Offset
CSC2_COEF4	220h

Diagram



Fields

Field	Description
31-25 —	Reserved
24-16 D1	D1 Two's compliment coefficient integer offset to be added.
15-11 —	Reserved
10-0 C3	C3 Two's compliment coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

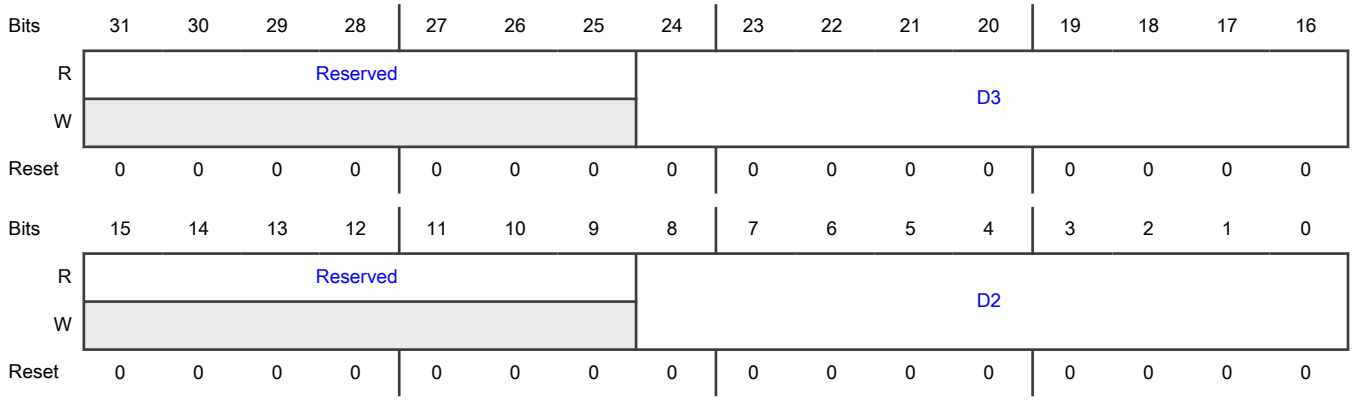
56.6.1.37 Color Space Conversion 2 Coefficient 5 Register (CSC2_COEF5)

This register contains color space conversion coefficients in two's compliment notation.

Offset

Register	Offset
CSC2_COEF5	230h

Diagram



Fields

Field	Description
31-25 —	Reserved
24-16 D3	D3 Two's compliment coefficient integer offset to be added.
15-9 —	Reserved
8-0 D2	D2 Two's compliment D1 coefficient integer offset to be added.

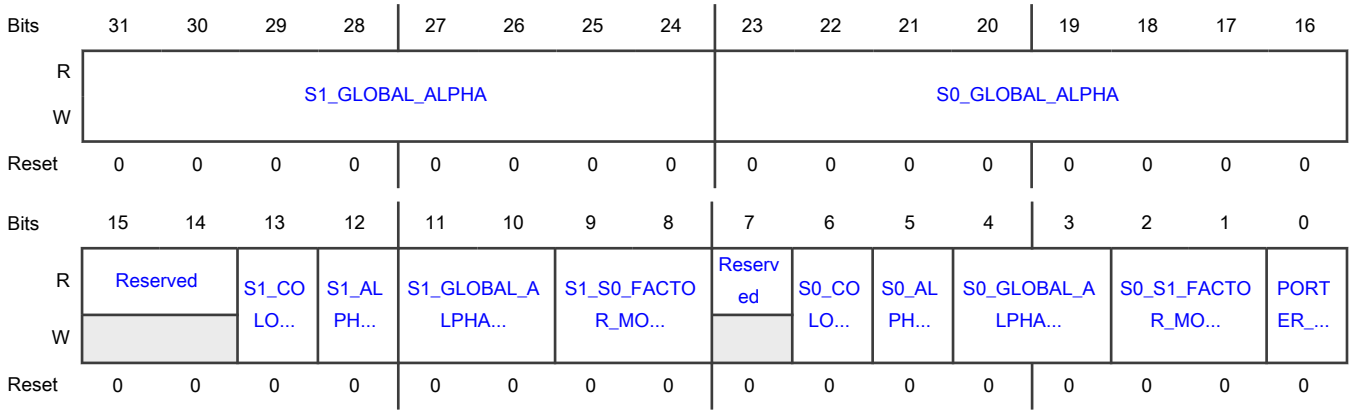
56.6.1.38 Alpha Engine A Control Register (ALPHA_A_CTRL)

This register controls alpha blending function in PXP.

Offset

Register	Offset
ALPHA_A_CTRL	290h

Diagram



Fields

Field	Description
31-24 S1_GLOBAL_ALPHA	S1_GLOBAL_ALPHA s1 global alpha
23-16 S0_GLOBAL_ALPHA	S0_GLOBAL_ALPHA s0 global alpha
15-14 —	Reserved
13 S1_COLOR_MODE	S1_COLOR_MODE s1 color mode 0 - straight mode for s1 color 1 - multiply mode for s1 color
12 S1_ALPHA_MODE	S1_ALPHA_MODE s1 alpha mode 0 - straight mode for s1 alpha 1 - inversed mode for s1 alpha
11-10 S1_GLOBAL_ALPHA_MODE	S1_GLOBAL_ALPHA_MODE s1 global alpha mode 00 - using global alpha. 01 - using local alpha. 10 - using scaled alpha. 11 - using scaled alpha.

Table continues on the next page...

Table continued from the previous page...

Field	Description
9-8 S1_S0_FACTOR_MODE	S1_S0_FACTOR_MODE s1 to s0 factor mode 00 - using 1. 01 - using 0. 10 - using straight alpha. 11 - using inverse alpha.
7 —	Reserved
6 S0_COLOR_MODE	S0_COLOR_MODE s0 color mode 0 - straight mode for s0 color 1 - multiply mode for s0 color
5 S0_ALPHA_MODE	S0_ALPHA_MODE s0 alpha mode 0 - straight mode for s0 alpha 1 - inversed mode for s0 alpha
4-3 S0_GLOBAL_ALPHA_MODE	S0_GLOBAL_ALPHA_MODE s0 global alpha mode 00 - using global alpha. 01 - using local alpha. 10 - using scaled alpha. 11 - using scaled alpha.
2-1 S0_S1_FACTOR_MODE	S0_S1_FACTOR_MODE s0 to s1 factor mode 00 - using 1. 01 - using 0. 10 - using straight alpha. 11 - using inverse alpha.
0 PORTER_DUFF_ENABLE	PORTER_DUFF_ENABLE porter_duff enable 0 - porter duff disable.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - porter duff enable.

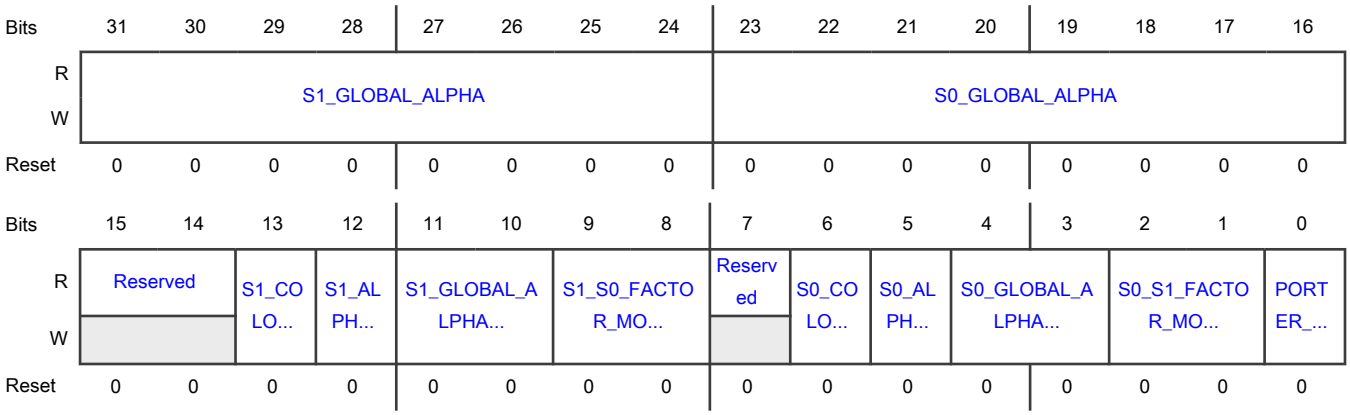
56.6.1.39 Alpha Engine B Control Register (ALPHA_B_CTRL)

This register controls alpha blending function in PXP.

Offset

Register	Offset
ALPHA_B_CTRL	2A0h

Diagram



Fields

Field	Description
31-24 S1_GLOBAL_ALPHA	S1_GLOBAL_ALPHA s1 global alpha
23-16 S0_GLOBAL_ALPHA	S0_GLOBAL_ALPHA s0 global alpha
15-14 —	Reserved
13	S1_COLOR_MODE s1 color mode

Table continues on the next page...

Table continued from the previous page...

Field	Description
S1_COLOR_MODE	0 - straight mode for s1 color 1 - multiply mode for s1 color
12 S1_ALPHA_MODE	S1_ALPHA_MODE s1 alpha mode 0 - straight mode for s1 alpha 1 - inversed mode for s1 alpha
11-10 S1_GLOBAL_ALPHA_MODE	S1_GLOBAL_ALPHA_MODE s1 global alpha mode 00 - using global alpha. 01 - using local alpha. 10 - using scaled alpha. 11 - using scaled alpha.
9-8 S1_S0_FACTOR_MODE	S1_S0_FACTOR_MODE s1 to s0 factor mode 00 - using 1. 01 - using 0. 10 - using straight alpha. 11 - using inverse alpha.
7 —	Reserved
6 S0_COLOR_MODE	S0_COLOR_MODE s0 color mode 0 - straight mode for s0 color 1 - multiply mode for s0 color
5 S0_ALPHA_MODE	S0_ALPHA_MODE s0 alpha mode 0 - straight mode for s0 alpha 1 - inversed mode for s0 alpha
4-3 S0_GLOBAL_ALPHA_MODE	S0_GLOBAL_ALPHA_MODE s0 global alpha mode 00 - using global alpha.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	01 - using local alpha. 10 - using scaled alpha. 11 - using scaled alpha.
2-1 S0_S1_FACTOR_MODE	S0_S1_FACTOR_MODE s0 to s1 factor mode 00 - using 1. 01 - using 0. 10 - using straight alpha. 11 - using inverse alpha.
0 PORTER_DUFF_ENABLE	PORTER_DUFF_ENABLE porter_duff enable 0 - porter duff disable. 1 - porter duff enable.

56.6.1.40 Alpha Engine B Control 1 Register (ALPHA_B_CTRL_1)

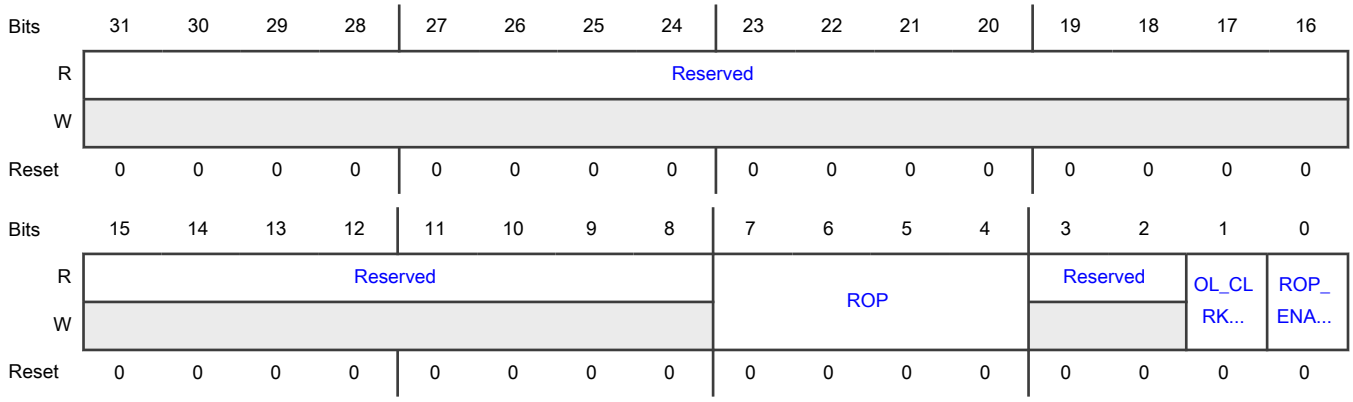
This register defines software define pixels for alpha blending.

This register defines alpha1 legacy blending parameters.

Offset

Register	Offset
ALPHA_B_CTRL_1	2B0h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-4 ROP	ROP Indicates a raster operation to perform when enabled. 0000 - MASKAS : AS AND PS 0001 - MASKNOTAS : nAS AND PS 0010 - MASKASNOT : AS AND nPS 0011 - MERGEAS : AS OR PS 0100 - MERGENOTAS : nAS OR PS 0101 - MERGEASNOT : AS OR nPS 0110 - NOTCOPYAS : nAS 0111 - NOT : nPS 1000 - NOTMASKAS : AS NAND PS 1001 - NOTMERGEAS : AS NOR PS 1010 - XORAS : AS XOR PS 1011 - NOTXORAS : AS XNOR PS
3-2 —	Reserved
1 OL_CLRKEY_ENABLE	OL_CLRKEY_ENABLE Indicates that colorkey functionality is enabled for this alpha surface. Pixels found in the alpha surface colorkey range will be displayed as transparent (the PS pixel will be used).
0 ROP_ENABLE	ROP_ENABLE ROP ENABLE

56.6.1.41 PS Background Color 1 Register (PS_BACKGROUND_1)

PS Background Pixel Color. This register provides a pixel value used when processing pixels outside of the region specified by the PS Coordinate registers. This value can effectively be used to set the color of the letterboxing region around the PS image. This value is used by the Alpha B block.

This register contains a pixel value to be used for any PS pixels that fall outside the PS extents. This is effectively a background or letterbox color. The CSC1 control and datapath pixel format should be considered when selecting the background color.

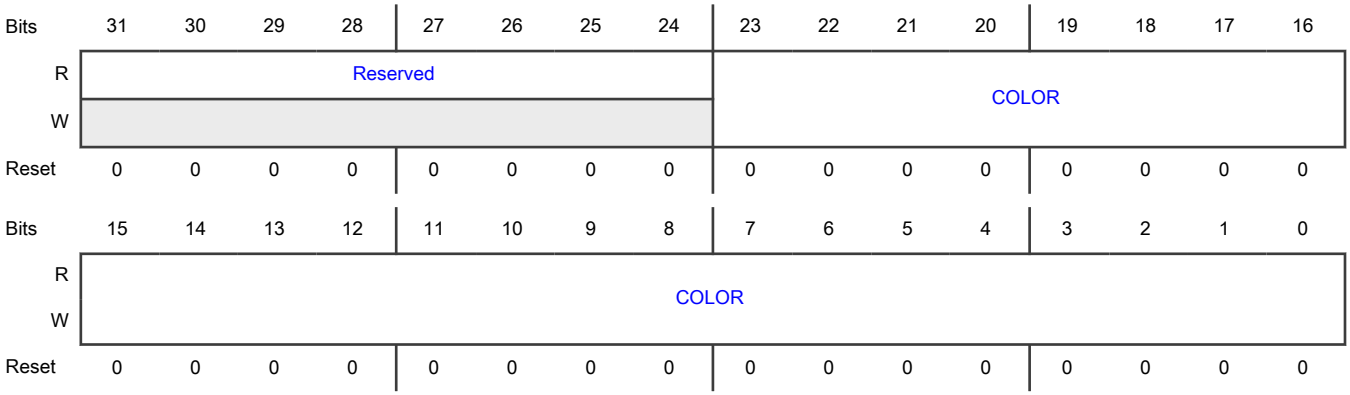
EXAMPLE

```
REG_PS_BACKGROUND_WR(0x00000000); // letterbox is black
REG_PS_BACKGROUND_WR(0x00800000); // letterbox is dark red
REG_PS_BACKGROUND_WR(0x00008000); // letterbox is dark green
REG_PS_BACKGROUND_WR(0x00000080); // letterbox is dark blue
```

Offset

Register	Offset
PS_BACKGROUND_1	2C0h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 COLOR	COLOR Background color (in 24bpp format) for any pixels not within the buffer range specified by the PS ULC/LRC.

56.6.1.42 PS Color Key Low Register (PS_CLRKEYLOW_1)

This register contains the color key low value for the PS buffer 1. This value is used by the Alpha B block.

When processing an image, if PXP finds a pixel in the PS buffer with a color that falls in the range between REG_PS_CLRKEYLOW and REG_PS_CLRKEYHIGH, it will insert the pixel from the AS channel. If the current AS pixel is letterboxed or if the AS also matches its colorkey range, the REG_PS_BACKGROUND color is passed down the pixel pipeline.

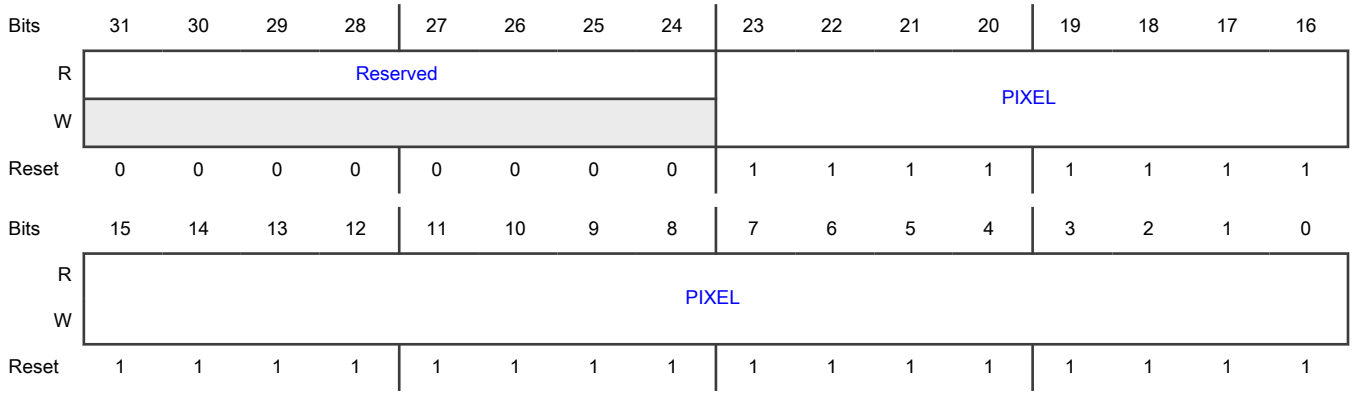
EXAMPLE

```
REG_PS_CLRKEYLOW_WR (0x008000); // colorkey values between medium green and
REG_PS_CLRKEYHIGH_WR(0x00FF00); // light green
```

Offset

Register	Offset
PS_CLRKEYLOW_1	2D0h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 PIXEL	PIXEL Low range of color key applied to PS buffer. To disable PS colorkeying, set the low colorkey to 0xFFFFFF and the high colorkey to 0x000000.

56.6.1.43 PS Color Key High Register (PS_CLRKEYHIGH_1)

This register contains the color key high value for the PS buffer. This value is used by the Alpha B block.

When processing an image, if PXP finds a pixel in the PS buffer with a color that falls in the range between REG_PS_CLRKEYLOW and REG_PS_CLRKEYHIGH, it will insert the pixel from the AS channel. If the current AS pixel is letterboxed or if the AS also matches its colorkey range, the REG_PS_BACKGROUND color is passed down the pixel pipeline.

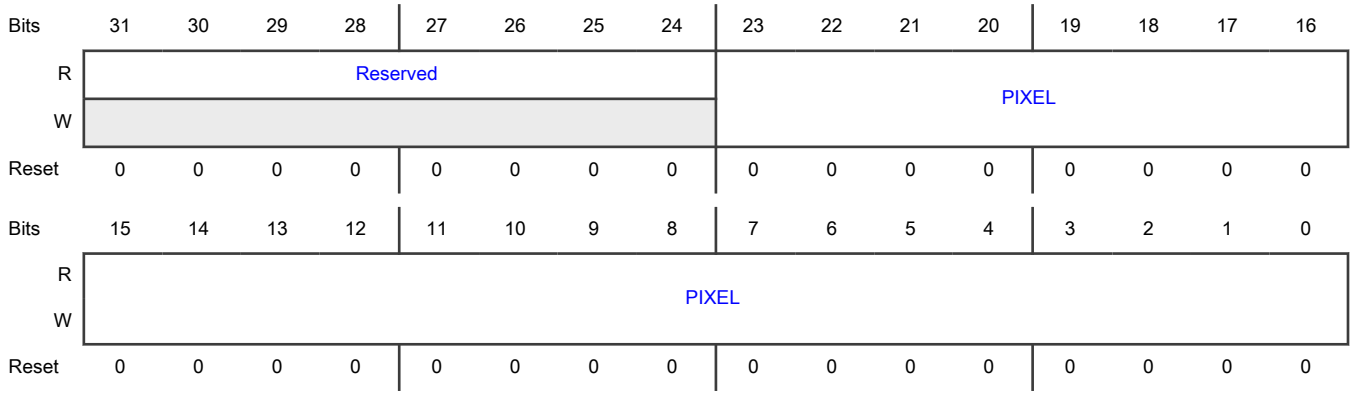
EXAMPLE

```
REG_PS_CLRKEYLOW_WR (0x008000); // colorkey values between medium green and
REG_PS_CLRKEYHIGH_WR(0x00FF00); // light green
```

Offset

Register	Offset
PS_CLRKEYHIGH_1	2E0h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 PIXEL	PIXEL High range of color key applied to PS buffer. To disable PS colorkeying, set the low colorkey to 0xFFFFFF and the high colorkey to 0x000000.

56.6.1.44 Overlay Color Key Low Register (AS_CLRKEYLOW_1)

This register contains the color key low value for the AS buffer. This value is used by the Alpha B block.

When processing an image, if PXP finds a pixel in the current overlay image with a color that falls in the range from the ASCOLORKEYLOW to ASCOLORKEYHIGH range, it will use the PS pixel value for that location. If no PS image is present or if the PS image also matches its colorkey range, the PS background color is used. Colorkey operations are higher priority than alpha or ROP operations.

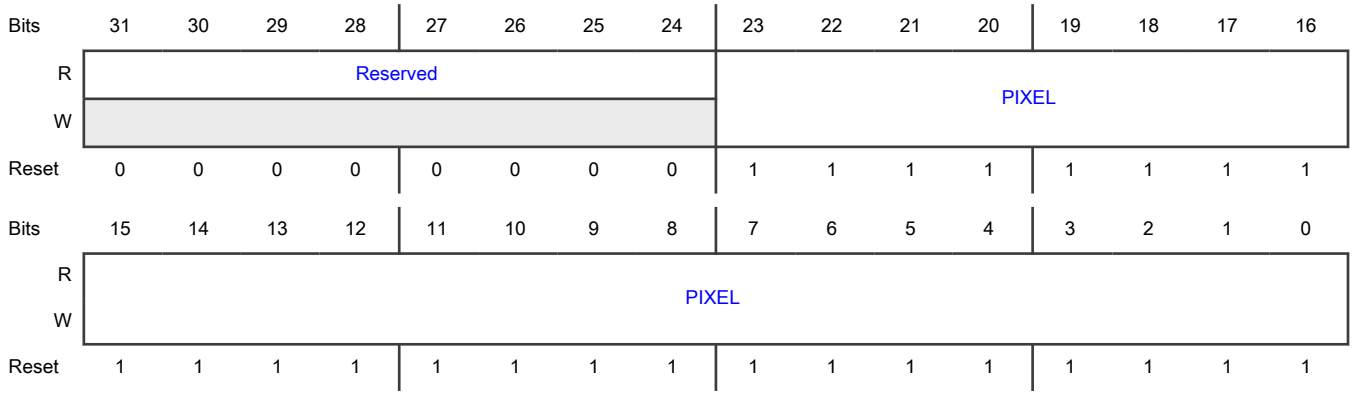
EXAMPLE

```
REG_AS_CLRKEYLOW_WR (0x000000); // colorkey values between black and
REG_AS_CLRKEYHIGH_WR(0x800000); // medium red
```

Offset

Register	Offset
AS_CLRKEYLOW_1	2F0h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 PIXEL	PIXEL Low range of RGB color key applied to AS buffer. Each overlay has an independent colorkey enable.

56.6.1.45 Overlay Color Key High Register (AS_CLRKEYHIGH_1)

This register contains the color key high value for the AS buffer. This value is used by the Alpha B block.

When processing an image, if PXP finds a pixel in the current overlay image with a color that falls in the range from the ASCOLORKEYLOW to ASCOLORKEYHIGH range, it will use the PS pixel value for that location. If no PS image is present or if the PS image also matches its colorkey range, the PS background color is used. Colorkey operations are higher priority than alpha or ROP operations.

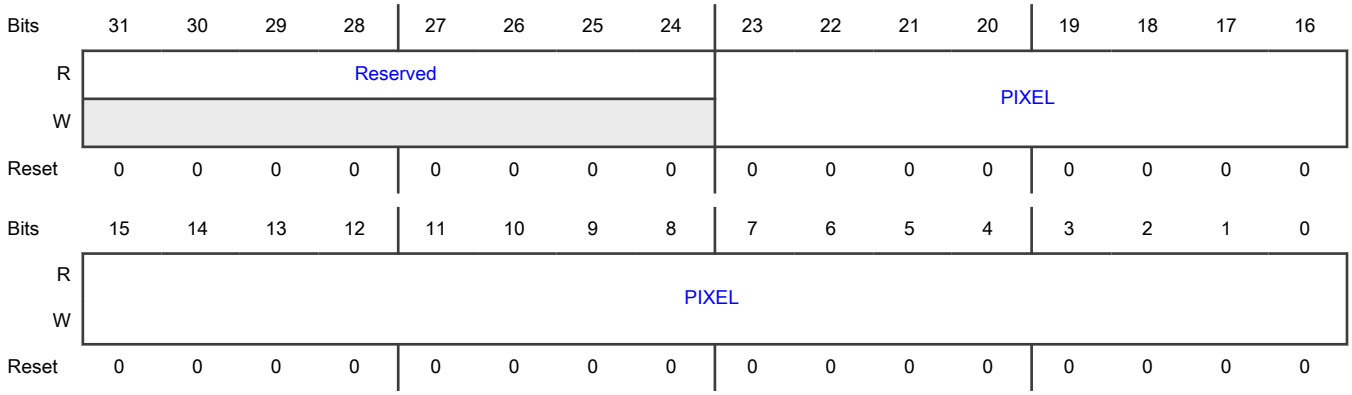
EXAMPLE

```
REG_AS_CLRKEYLOW_WR (0x000000); // colorkey values between black and
REG_AS_CLRKEYHIGH_WR(0x800000); // medium red
```

Offset

Register	Offset
AS_CLRKEYHIGH_1	300h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 PIXEL	PIXEL High range of RGB color key applied to AS buffer. Each overlay has an independent colorkey enable.

56.6.1.46 Control Register 2 (CTRL2)

The CTRL register contains controls for the secondary processing flow in PXP module.

CTRL2: 0x310

CTRL2_SET: 0x314

CTRL2_CLR: 0x318

CTRL2_TOG: 0x31C

The Control register contains the controls for the secondary data flow in PXP block.

EXAMPLE

```
REG_CTRL_SET(BM_PXP_CTRL_SFTRST);
REG_CTRL_CLR(BM_PXP_CTRL_SFTRST | BM_PXP_CTRL_CLKGATE);
```

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

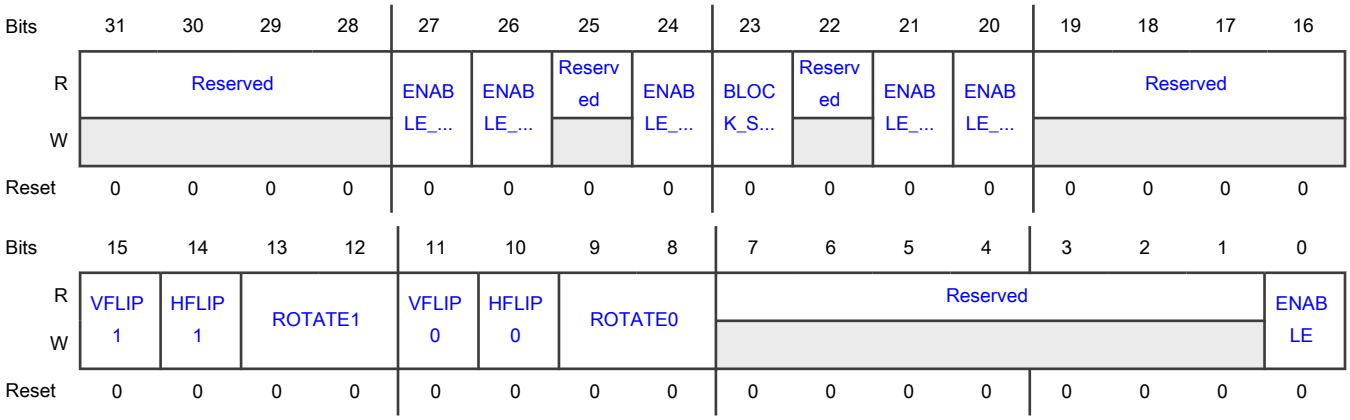
Register	Offset	Description
CTRL2	310h	Control Register 2
CTRL2_SET	314h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL2 is 1

Table continues on the next page...

Table continued from the previous page...

Register	Offset	Description
CTRL2_CLR	318h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL2 is 0
CTRL2_TOG	31Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CTRL2

Diagram



Fields

Field	Description
31-28 —	Reserved
27 ENABLE_ROTATE1	ENABLE_ROTATE1 Enable the ROTATE1 engine in PXP secondary processing flow.
26 ENABLE_ROTATE0	ENABLE_ROTATE0 Enable the ROTATE0 engine in PXP secondary processing flow.
25 —	Reserved
24 ENABLE_CSC2	ENABLE_CSC2 Enable the CSC2 engine in PXP secondary processing flow.
23 BLOCK_SIZE	BLOCK_SIZE Select the block size to process through the Rotate block. 0 - BLK_SIZE_8X8 : Process 8x8 pixel blocks.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - BLK_SIZE_16X16 : Process 16x16 pixel blocks.
22 —	Reserved
21 ENABLE_ALPHA_B	ENABLE_ALPHA_B Enable the Alpha-B engine in PXP secondary processing flow.
20 ENABLE_INPUT_FETCH_STORE	ENABLE_INPUT_FETCH_STORE Enable the Input Fetch and Store engine in PXP secondary processing flow.
19-16 —	Reserved
15 VFLIP1	VFLIP1 Indicates that the input should be flipped vertically (effect applied before rotation).
14 HFLIP1	HFLIP1 Indicates that the input should be flipped horizontally (effect applied before rotation).
13-12 ROTATE1	ROTATE1 Indicates the clockwise rotation to be applied at the input buffer. The rotation effect is defined as occurring after the FLIP_X and FLIP_Y permutation. 00 - ROT_0 : rotate 0 degrees 01 - ROT_90 : rotate 90 degrees 10 - ROT_180 : rotate 180 degrees 11 - ROT_270 : rotate 270 degrees
11 VFLIP0	VFLIP0 Indicates that the output buffer should be flipped vertically (effect applied before rotation).
10 HFLIP0	HFLIP0 Indicates that the output buffer should be flipped horizontally (effect applied before rotation).
9-8 ROTATE0	ROTATE0 Indicates the clockwise rotation to be applied at the output buffer. The rotation effect is defined as occurring after the FLIP_X and FLIP_Y permutation. 00 - ROT_0 : rotate 0 degrees 01 - ROT_90 : rotate 90 degrees

Table continues on the next page...

Table continued from the previous page...

Field	Description
	10 - ROT_180 : rotate 180 degrees 11 - ROT_270 : rotate 270 degrees
7-1 —	Reserved
0 ENABLE	ENABLE Enables PXP secondary data processing flow with specified parameters. The ENABLE bit will remain set while PXP is active and will be cleared once the current operation completes. Software should use the IRQ bit in the STAT when polling for PXP completion.

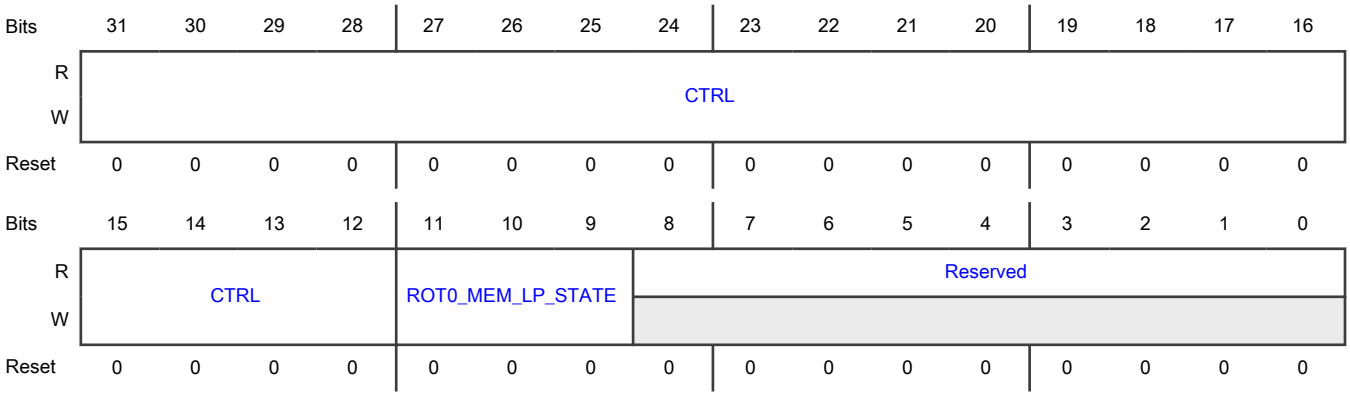
56.6.1.47 Power Control Register 0 (POWER_REG0)

This register controls power states for PXP memories.

Offset

Register	Offset
POWER_REG0	320h

Diagram



Fields

Field	Description
31-12 CTRL	CTRL This register contains power control for PXP module.
11-9	ROT0_MEM_LP_STATE

Table continues on the next page...

Table continued from the previous page...

Field	Description
ROT0_MEM_LP_STATE	Select the low power state of the ROT 0 memory. 000 - NONE : Memory is not in low power state. 001 - LS : Light Sleep Mode. Low leakage mode, maintain memory contents. 010 - DS : Deep Sleep Mode. Low leakage mode, maintain memory contents. 100 - SD : Shut Down Mode. Shut Down periphery and core, no memory retention.
8-0 —	Reserved

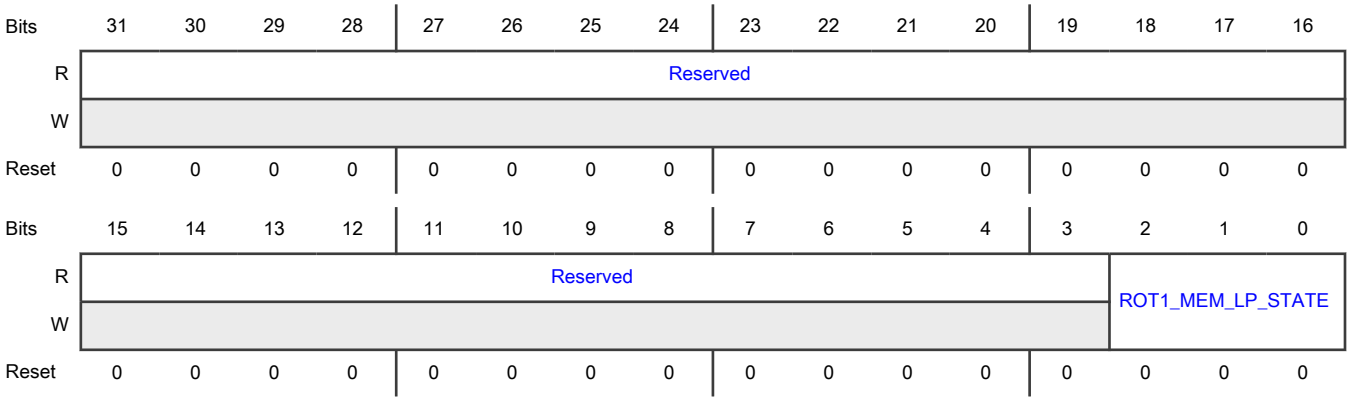
56.6.1.48 Power Control Register 1 (POWER_REG1)

This register controls power states for PXP memories.

Offset

Register	Offset
POWER_REG1	330h

Diagram



Fields

Field	Description
31-3 —	Reserved
2-0	ROT1_MEM_LP_STATE Select the low power state of the ROT 1 memory.

Table continues on the next page...

Table continued from the previous page...

Field	Description
ROT1_MEM_LP_STATE	000 - NONE : Memory is not in low power state. 001 - LS : Light Sleep Mode. Low leakage mode, maintain memory contents. 010 - DS : Deep Sleep Mode. Low leakage mode, maintain memory contents. 100 - SD : Shut Down Mode. Shut Down periphery and core, no memory retention.

56.6.1.49 Data Path Control 0 Register (DATA_PATH_CTRL0)

This register helps decide the data path through PXP.

DATA_PATH_CTRL0: 0x340

DATA_PATH_CTRL0_SET: 0x344

DATA_PATH_CTRL0_CLR: 0x348

DATA_PATH_CTRL0_TOG: 0x34C

The Control register contains the control bits for the data path through PXP.

NOTE

There should be only 1 destination for a pxp processing flow operation, otherwise the output data might be corrupted, i.e. MUX output should not be redirected to 2 paths in single processing flow.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DATA_PATH_CTRL0	340h	Data Path Control 0 Register
DATA_PATH_CTRL0_SET	344h	Writing 1 to a bit in this register ensures that the corresponding bit in DATA_PATH_CTRL0 is 1
DATA_PATH_CTRL0_CLR	348h	Writing 1 to a bit in this register ensures that the corresponding bit in DATA_PATH_CTRL0 is 0
DATA_PATH_CTRL0_TOG	34Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in DATA_PATH_CTRL0

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MUX15_SEL		MUX14_SEL		MUX13_SEL		MUX12_SEL		MUX11_SEL		MUX10_SEL		MUX9_SEL		MUX8_SEL	
Reset	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MUX7_SEL		MUX6_SEL		MUX5_SEL		MUX4_SEL		MUX3_SEL		MUX2_SEL		MUX1_SEL		MUX0_SEL	
Reset	0	1	0	1	1	0	1	0	0	0	1	0	1	0	1	1

Fields

Field	Description
31-30 MUX15_SEL	MUX15_SEL This field chooses the data path through MUX 15. 00 - Output of Input fetch, Channel 0 01 - Output of MUX 10 10 - No output 11 - No Output
29-28 MUX14_SEL	MUX14_SEL This field chooses the data path through MUX 14. 00 - Output of Rotation 0. 01 - Output of MUX 11 10 - No output 11 - No Output
27-26 MUX13_SEL	MUX13_SEL This field chooses the data path through MUX 13. 00 - Input stage Fetch engine, Channel 1 01 - No output 10 - No output 11 - No Output
25-24 MUX12_SEL	MUX12_SEL This field chooses the data path through MUX 12. 00 - Output of MUX 10 01 - Output of MUX 11 10 - No output

Table continues on the next page...

Table continued from the previous page...

Field	Description
	11 - No Output
23-22 MUX11_SEL	MUX11_SEL This field chooses the data path through MUX 11. 00 - No output 01 - Output of MUX 8 10 - No output 11 - No Output
21-20 MUX10_SEL	MUX10_SEL This field chooses the data path through MUX 10. 00 - Output of MUX 7 01 - No output 10 - No output 11 - No Output
19-18 MUX9_SEL	MUX9_SEL reserved
17-16 MUX8_SEL	MUX8_SEL This field chooses the data path through MUX 8. 00 - Output of CSC2 01 - Output of alpha blending / color key 0 10 - No output 11 - No Output
15-14 MUX7_SEL	MUX7_SEL This field chooses the data path through MUX 7. 00 - Output of MUX 5 01 - Output of CSC2 10 - No output 11 - No Output
13-12 MUX6_SEL	MUX6_SEL This field chooses the data path through MUX 6. 00 - Output of alpha blending / color key 1 01 - Output of alpha blending / color key 0 10 - No output

Table continues on the next page...

Table continued from the previous page...

Field	Description
	11 - No Output
11-10 MUX5_SEL	MUX5_SEL This field chooses the data path through MUX 5. 00 - Output of MUX1 01 - Output of alpha blending / color key 1 10 - No output 11 - No Output
9-8 MUX4_SEL	MUX4_SEL reserved.
7-6 MUX3_SEL	MUX3_SEL This field chooses the data path through MUX 3. 00 - Output of the CSC1 engine 01 - Output of the Rotation1 engine 10 - No output 11 - No Output
5-4 MUX2_SEL	MUX2_SEL This field chooses the data path through MUX 2. 00 - Input stage Fetch engine, Channel 1 01 - Output of the Rotation1 engine 10 - No output 11 - No Output
3-2 MUX1_SEL	MUX1_SEL This field chooses the data path through MUX 1. 00 - Input stage Fetch engine, Channel 0 01 - Output of the Rotation1 engine 10 - No output 11 - No Output
1-0 MUX0_SEL	MUX0_SEL This mux chooses the data that will go through the Rotate 2 engine. 00 - Input from Process Surface engine. 01 - Input stage Fetch engine, Channel 0 10 - Input stage Fetch engine, Channel 1

Table continues on the next page...

Table continued from the previous page...

Field	Description
	11 - No output

56.6.1.50 IRQ Mask Register (IRQ_MASK)

Controls masking for all PXP interrupts

IRQ_MASK: 0x390

IRQ_MASK_SET: 0x394

IRQ_MASK_CLR: 0x398

IRQ_MASK_TOG: 0x39C

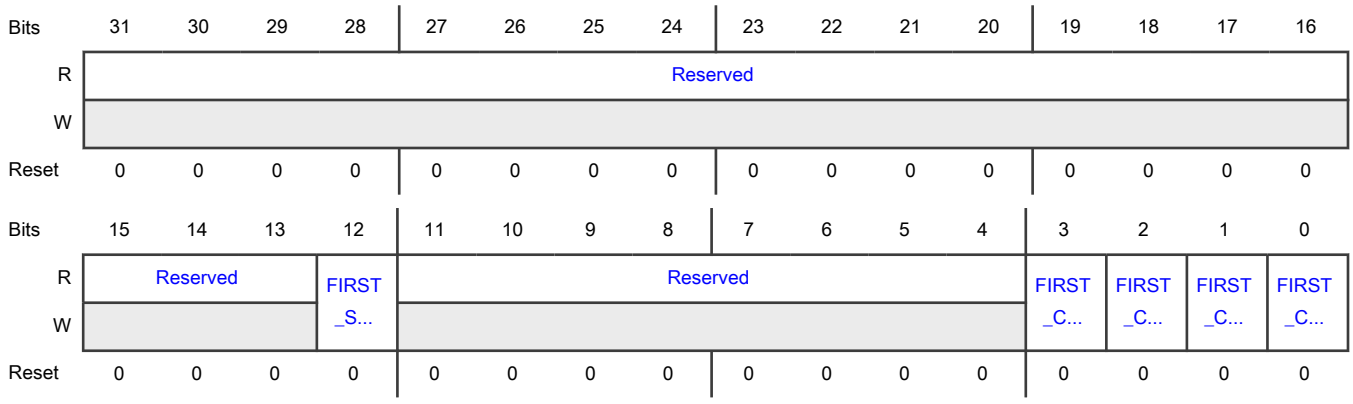
This register controls IRQ masks for all PXP interrupts

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
IRQ_MASK	390h	IRQ Mask Register
IRQ_MASK_SET	394h	Writing 1 to a bit in this register ensures that the corresponding bit in IRQ_MASK is 1
IRQ_MASK_CLR	398h	Writing 1 to a bit in this register ensures that the corresponding bit in IRQ_MASK is 0
IRQ_MASK_TOG	39Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in IRQ_MASK

Diagram



Fields

Field	Description
31-13 —	Reserved
12 FIRST_STORE_IRQ_EN	FIRST_STORE_IRQ_EN Enable First store engine interrupt detection
11-4 —	Reserved
3 FIRST_CH1_STORE_IRQ_EN	FIRST_CH1_STORE_IRQ_EN Enable First ch1 store engine interrupt detection
2 FIRST_CH0_STORE_IRQ_EN	FIRST_CH0_STORE_IRQ_EN Enable First ch0 store engine interrupt detection
1 FIRST_CH1_PREFETCH_IRQ_EN	FIRST_CH1_PREFETCH_IRQ_EN Enable First ch1 prefetch engine interrupt detection
0 FIRST_CH0_PREFETCH_IRQ_EN	FIRST_CH0_PREFETCH_IRQ_EN Enable First ch0 prefetch engine interrupt detection

56.6.1.51 Interrupt Register (IRQ)

PXP Fetch and Store engine IRQs. Each interrupt has a corresponding mask register in HW_EPDC_IRQ_MASK

IRQ: 0x3A0

IRQ_SET: 0x3A4

IRQ_CLR: 0x3a8

IRQ_TOG: 0x3AC

This register houses the interrupt bits for the Prefetch and Store Engines

NOTE

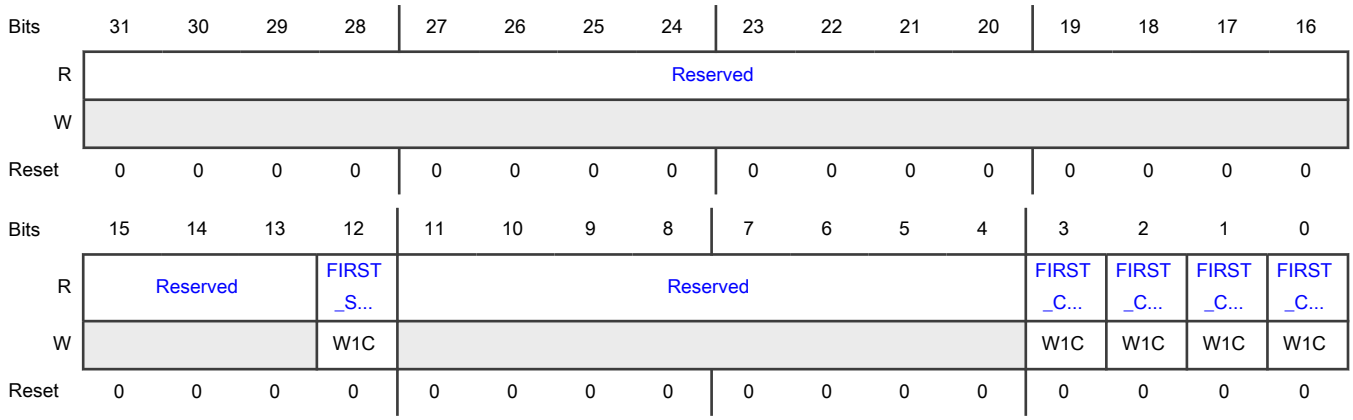
Writing on SET or TOG variants of these registers will have no impact on value.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
IRQ	3A0h	Interrupt Register
IRQ_SET	3A4h	Writing 1 to a bit in this register ensures that the corresponding bit in IRQ is 1
IRQ_CLR	3A8h	Writing 1 to a bit in this register ensures that the corresponding bit in IRQ is 0
IRQ_TOG	3ACh	Writing 1 to a bit in this register inverts the value of the corresponding bit in IRQ

Diagram



Fields

Field	Description
31-13 —	Reserved
12 FIRST_STORE_IRQ	FIRST_STORE_IRQ Initial store engine interrupt
11-4 —	Reserved
3 FIRST_CH1_STORE_IRQ	FIRST_CH1_STORE_IRQ Initial ch1 store engine interrupt
2 FIRST_CH0_STORE_IRQ	FIRST_CH0_STORE_IRQ Initial ch0 store engine interrupt

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 FIRST_CH1_PREFETCH_IRQ	FIRST_CH1_PREFETCH_IRQ Initial ch1 prefetch engine interrupt
0 FIRST_CH0_PREFETCH_IRQ	FIRST_CH0_PREFETCH_IRQ Initial ch0 prefetch engine interrupt

56.6.1.52 Next Frame Pointer Register (NEXT)

This register contains a pointer to a data structure used to reload PXP registers at the end of the current frame.

To enable this functionality, software must write this register while PXP is processing the current data frame (if PXP is currently idle, this will also initiate an immediate load of registers from the pointer). The process of writing this register (WRITE operation) will set a semaphore in hardware to notify the control logic that a register reload operation must be performed when the current frame processing is complete. At the end of a frame, PXP will fetch the register settings from this location, signal an interrupt to software, then proceed with rendering the next frame of data. Software may cancel the reload operation by issuing a CLEAR operation to this register. SET and TOGGLE operations should not be used when addressing this register. All registers will be reloaded with the exception of the following: STAT, CSCCOEFn, NEXT, VERSION. All other registers will be loaded in the order they appear in the register map. Once the pointer's contents have been loaded into PXP's registers, the NEXT_IRQ interrupt will be issued (see the PXP_STATUS register).

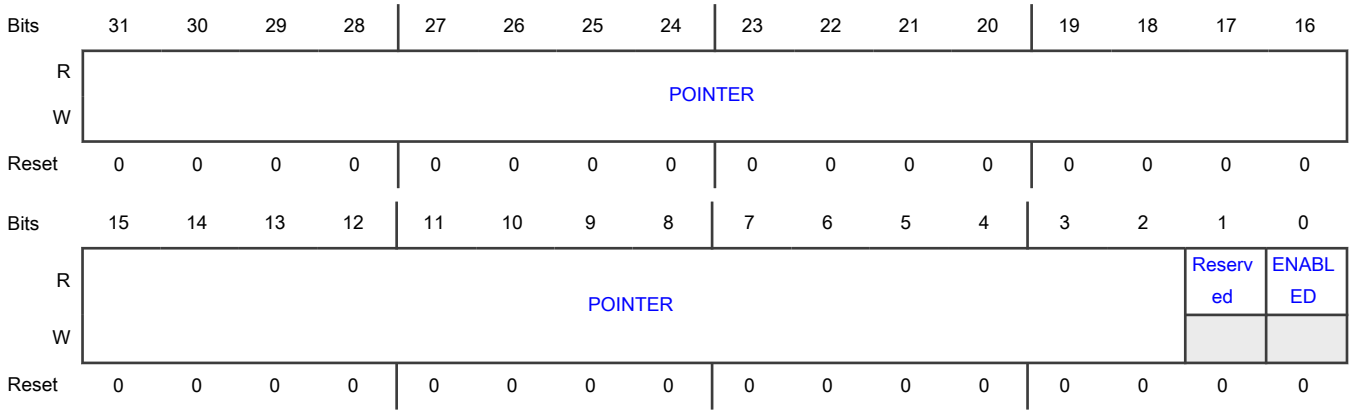
EXAMPLE

```
// create register command structure in memory
u32* pxp_commands0[48], pxp_commands1;
u32 rc;
// initialize control structure for frame 0
pxp_commands0[0] = ...; // CTRL
pxp_commands0[1] = ...; // OUT Buffer ...
pxp_commands0[47] = ..; // Overlay7 param2
// initialize control structure for frame 1
pxp_commands1[0] = ...; // CTRL
pxp_commands1[1] = ...; // OUT Buffer ...
pxp_commands1[47] = ..; // Overlay7 param2
// poll until a command isn't queued while (rc=REG_NEXT_RD() amp; BM_PXP_NEXT_ENABLED );
REG_NEXT_WR(pxp_commands0); // enable PXP operation 0 via command pointer
// poll until first command clears while (rc=REG_NEXT_RD() amp; BM_PXP_NEXT_ENABLED );
REG_NEXT_WR(pxp_commands1); // enable PXP operation 1 via command pointer
```

Offset

Register	Offset
NEXT	400h

Diagram



Fields

Field	Description
31-2 POINTER	POINTER A pointer to a data structure containing register values to be used when processing the next frame. The pointer must be 32-bit aligned and should reside in on-chip or off-chip memory.
1 —	Reserved
0 ENABLED	ENABLED Indicates that the quot;next framequot; functionality has been enabled. This bit reflects the status of the hardware semaphore indicating that a reload operation is pending at the end of the current frame.

56.6.1.53 Input Fetch Control Channel 0 Register (INPUT_FETCH_CTRL_CH0)

This register defines the control bits for the pxp prefetch_engine sub-block.

INPUT_FETCH_CTRL_CH0: 0x450

INPUT_FETCH_CTRL_CH0_SET: 0x454

INPUT_FETCH_CTRL_CH0_CLR: 0x458

INPUT_FETCH_CTRL_CH0_TOG: 0x45C

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

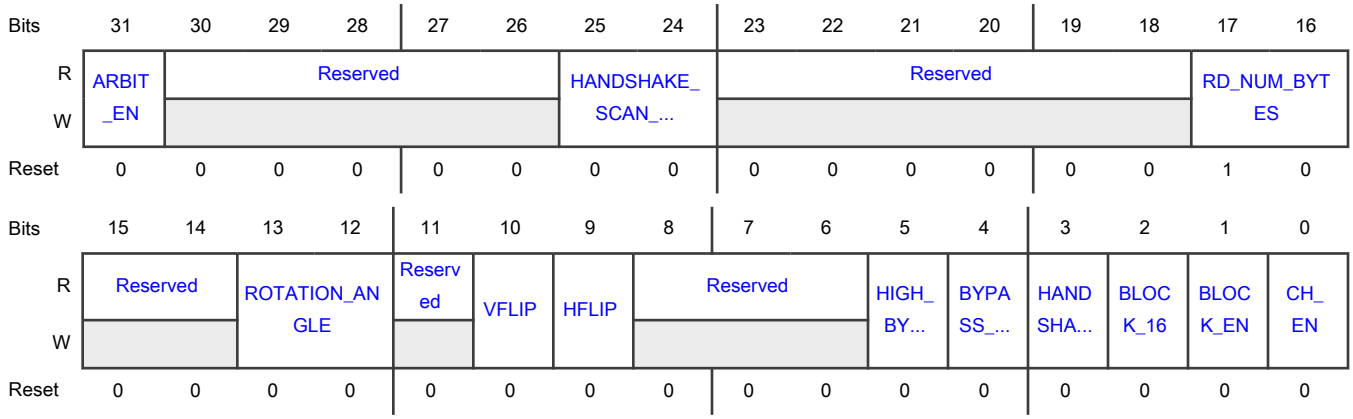
Register	Offset	Description
INPUT_FETCH_CTRL_CH0	450h	Input Fetch Control Channel 0 Register

Table continues on the next page...

Table continued from the previous page...

Register	Offset	Description
INPUT_FETCH_CTRL_CH0_SET	454h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_CTRL_CH0 is 1
INPUT_FETCH_CTRL_CH0_CLR	458h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_CTRL_CH0 is 0
INPUT_FETCH_CTRL_CH0_TOG	45Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in INPUT_FETCH_CTRL_CH0

Diagram



Fields

Field	Description
31 ARBIT_EN	ARBIT_EN Enables Arbitration 0 - Arbitration disable. If using 2 channels, will output 2 axi bus sets. 1 - Arbitration enable. If using 2 channel, will only output 1 axi bus sets
30-26 —	Reserved
25-24 HANDSHAKE_SCAN_LINE_NUM	HANDSHAKE_SCAN_LINE_NUM scan handshake line number 00 - 1 line. 01 - 8 lines 10 - 16 lines 11 - 16 lines

Table continues on the next page...

Table continued from the previous page...

Field	Description
23-18 —	Reserved
17-16 RD_NUM_BYTES	RD_NUM_BYTES Bytes in a read burst 00 - NUM_8_BYTES : 8 bytes 01 - NUM_16_BYTES : 16 bytes 10 - NUM_32_BYTES : 32 bytes 11 - NUM_64_BYTES : 64 bytes
15-14 —	Reserved
13-12 ROTATION_ANGLE	ROTATION_ANGLE 00 - ROT_0 : Rotate image by 0 degrees. 01 - ROT_90 : Rotate image by 90 degrees. 10 - ROT_180 : Rotate image by 180 degrees. 11 - ROT_270 : Rotate image by 270 degrees.
11 —	Reserved
10 VFLIP	VFLIP Enables VFLIP 0 - VFLIP disable 1 - VFLIP enable
9 HFLIP	HFLIP Enables HFLIP. 0 - HFLIP disable 1 - VFLIP enable
8-6 —	Reserved
5 HIGH_BYTE	HIGH_BYTE channel 0 high byte selection 0 - In 64 bit mode, the output high byte will use channel1.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - In 64 bit mode, the output high byte will use channel0
4 BYPASS_PIXEL_EN	BYPASS_PIXEL_EN Selects Channel 0 pixel source 0 - Channel 0 is from memory 1 - Channel 0 is from previous process engine
3 HANDSHAKE_EN	HANDSHAKE_EN Enable bit for handshake with the store engine. 0 - Handshake with the store engine is disabled 1 - Handshake with the store engine is enabled
2 BLOCK_16	BLOCK_16 Determines the block size. 0 - BLK_SIZE_8x8 : Block size is 8x8 1 - BLK_SIZE_16x16 : Block size is 16x16
1 BLOCK_EN	BLOCK_EN Chooses the prefetch mode. 0 - Prefetch in scan mode 1 - Prefetch in block mode
0 CH_EN	CH_EN Channel enable. 0 - Prefetch function is disable 1 - Prefetch function is enable

56.6.1.54 Input Fetch Control Channel 1 Register (INPUT_FETCH_CTRL_CH1)

This register defines the control bits for the pxp prefetch_engine sub-block.

INPUT_FETCH_CTRL_CH1: 0x460

INPUT_FETCH_CTRL_CH1_SET: 0x464

INPUT_FETCH_CTRL_CH1_CLR: 0x468

INPUT_FETCH_CTRL_CH1_TOG: 0x46C

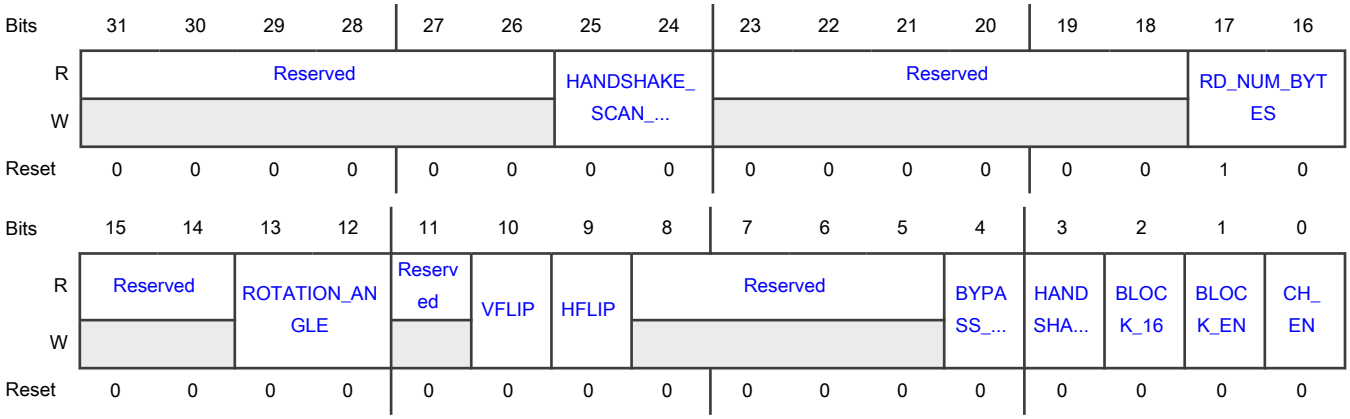
The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
INPUT_FETCH_CTRL_CH1	460h	Input Fetch Control Channel 1 Register
INPUT_FETCH_CTRL_CH1_SET	464h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_CTRL_CH1 is 1
INPUT_FETCH_CTRL_CH1_CLR	468h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_CTRL_CH1 is 0
INPUT_FETCH_CTRL_CH1_TOG	46Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in INPUT_FETCH_CTRL_CH1

Diagram



Fields

Field	Description
31-26 —	Reserved
25-24 HANDSHAKE_SCAN_LINE_NUM	HANDSHAKE_SCAN_LINE_NUM scan handshake line number 00 - 1 line. 01 - 8 lines 10 - 16 lines 11 - 16 lines
23-18 —	Reserved
17-16	RD_NUM_BYTES

Table continues on the next page...

Table continued from the previous page...

Field	Description
RD_NUM_BYTES	Bytes in a read burst 00 - NUM_8_BYTES : 8 bytes 01 - NUM_16_BYTES : 16 bytes 10 - NUM_32_BYTES : 32 bytes 11 - NUM_64_BYTES : 64 bytes
15-14 —	Reserved
13-12 ROTATION_ANGLE	ROTATION_ANGLE 00 - ROT_0 : Rotate image by 0 degrees. 01 - ROT_90 : Rotate image by 90 degrees. 10 - ROT_180 : Rotate image by 180 degrees. 11 - ROT_270 : Rotate image by 270 degrees.
11 —	Reserved
10 VFLIP	VFLIP Enables VFLIP 0 - VFLIP disable 1 - VFLIP enable
9 HFLIP	HFLIP Enables HFLIP. 0 - HFLIP disable 1 - VFLIP enable
8-5 —	Reserved
4 BYPASS_PIXEL_EN	BYPASS_PIXEL_EN Selects Channel 1 pixel source 0 - Channel 1 is from memory 1 - Channel 1 is from previous process engine
3 HANDSHAKE_EN	HANDSHAKE_EN Enable bit for handshake with the store engine. 0 - Handshake with the store engine is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Handshake with the store engine is enabled
2 BLOCK_16	BLOCK_16 Determines the block size. 0 - BLK_SIZE_8x8 : Block size is 8x8 1 - BLK_SIZE_16x16 : Block size is 16x16
1 BLOCK_EN	BLOCK_EN Chooses the prefetch mode. 0 - Prefetch in scan mode 1 - Prefetch in block mode
0 CH_EN	CH_EN Channel enable. 0 - prefetch function is disable 1 - prefetch function is enable

56.6.1.55 Input Fetch Status Channel 0 Register (INPUT_FETCH_STATUS_CH0)

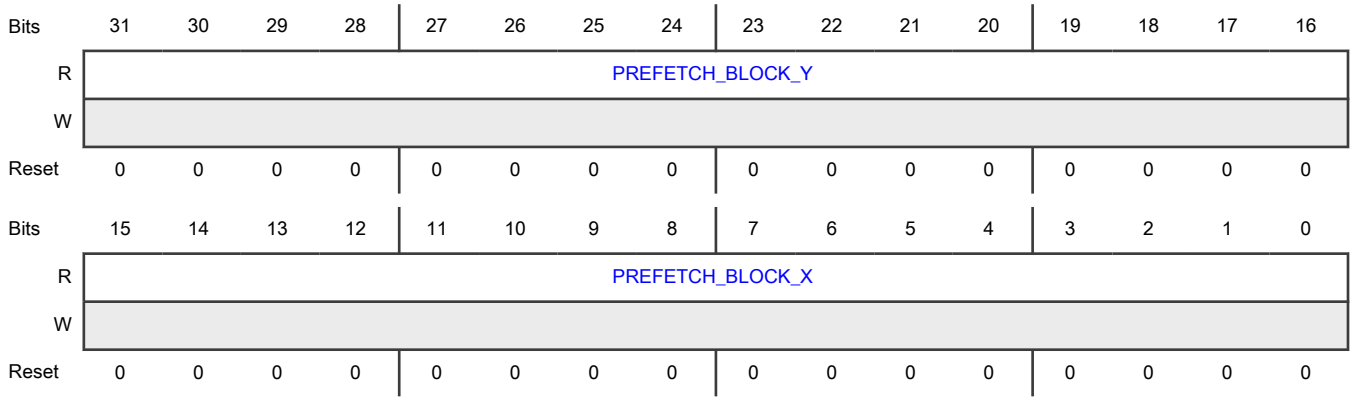
This register defines the status bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_STATUS_CH0	470h

Diagram



Fields

Field	Description
31-16 PREFETCH_BLOCK_Y	PREFETCH_BLOCK_Y When in scan mode, this field indicates the current Y coordinate of the frame. In block mode, it indicates the Y coordinate of the block currently being rendered.
15-0 PREFETCH_BLOCK_X	PREFETCH_BLOCK_X When in scan mode, this field is always 0. In block mode, it indicates the X coordinate of the block currently being rendered.

56.6.1.56 Input Fetch Status Channel 1 Register (INPUT_FETCH_STATUS_CH1)

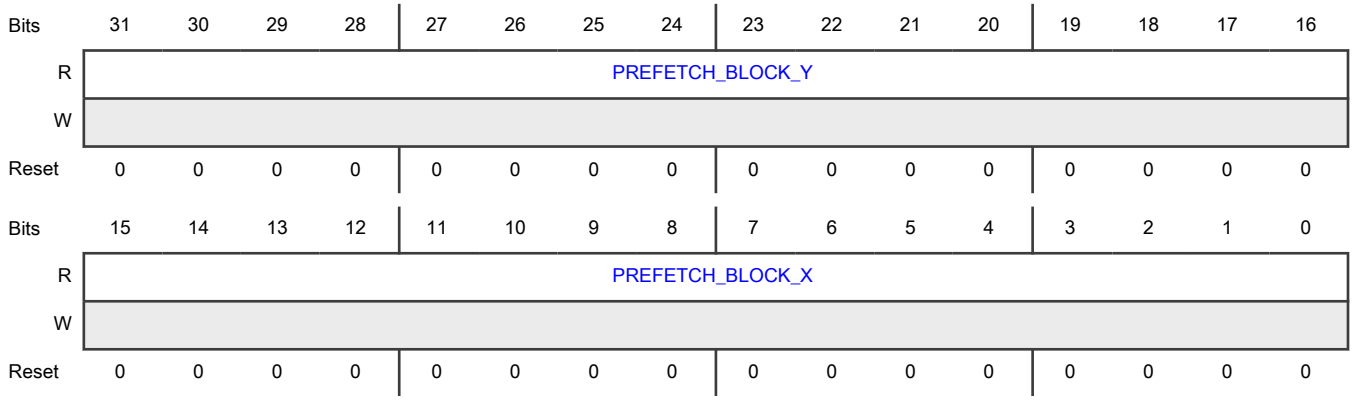
This register defines the status bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_STATUS_CH1	480h

Diagram



Fields

Field	Description
31-16 PREFETCH_BLOCK_Y	PREFETCH_BLOCK_Y When in scan mode, this field indicates the current Y coordinate of the frame. In block mode, it indicates the Y coordinate of the block currently being rendered.
15-0 PREFETCH_BLOCK_X	PREFETCH_BLOCK_X When in scan mode, this field is always 0. In block mode, it indicates the X coordinate of the block currently being rendered.

56.6.1.57 Input Fetch Active Size ULC Channel 0 Register (INPUT_FETCH_ACTIVE_SIZE_ULC_CH0)

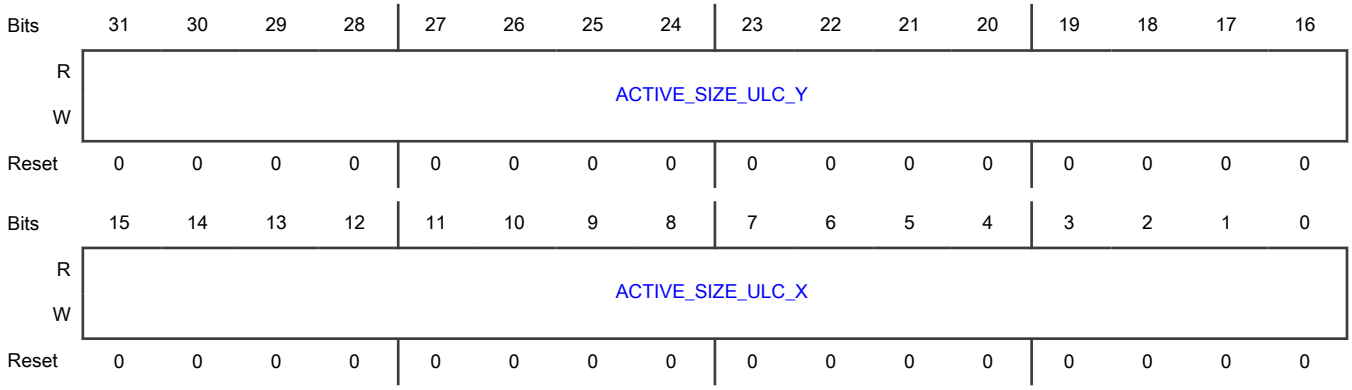
This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_ACTIVE_SIZE_ULC_CH0	490h

Diagram



Fields

Field	Description
31-16 ACTIVE_SIZE_ULC_Y	ACTIVE_SIZE_ULC_Y This field indicates the upper left Y-coordinate(in pixels) of the active surface of the total input memory
15-0 ACTIVE_SIZE_ULC_X	ACTIVE_SIZE_ULC_X This field indicates the upper left X-coordinate(in pixels) of the active surface of the total input memory

56.6.1.58 Input Fetch Active Size LRC Channel 0 Register (INPUT_FETCH_ACTIVE_SIZE_LRC_CH0)

This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_ACTIVE_SIZE_LRC_CH0	4A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ACTIVE_SIZE_LRC_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ACTIVE_SIZE_LRC_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 ACTIVE_SIZE_LRC_Y	ACTIVE_SIZE_LRC_Y This field indicates the lower right Y-coordinate(in pixels) of the active surface of the total input memory
15-0 ACTIVE_SIZE_LRC_X	ACTIVE_SIZE_LRC_X This field indicates the lower right X-coordinate(in pixels) of the active surface of the total input memory

56.6.1.59 Input Fetch Active Size ULC Channel 1 Register (INPUT_FETCH_ACTIVE_SIZE_ULC_CH1)

This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_ACTIVE_SIZE_ULC_CH1	4B0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ACTIVE_SIZE_ULC_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ACTIVE_SIZE_ULC_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 ACTIVE_SIZE_ULC_Y	ACTIVE_SIZE_ULC_Y This field indicates the upper left Y-coordinate(in pixels) of the active surface of the total input memory
15-0 ACTIVE_SIZE_ULC_X	ACTIVE_SIZE_ULC_X This field indicates the upper left X-coordinate(in pixels) of the active surface of the total input memory

56.6.1.60 Input Fetch Active Size LRC Channel 1 Register (INPUT_FETCH_ACTIVE_SIZE_LRC_CH1)

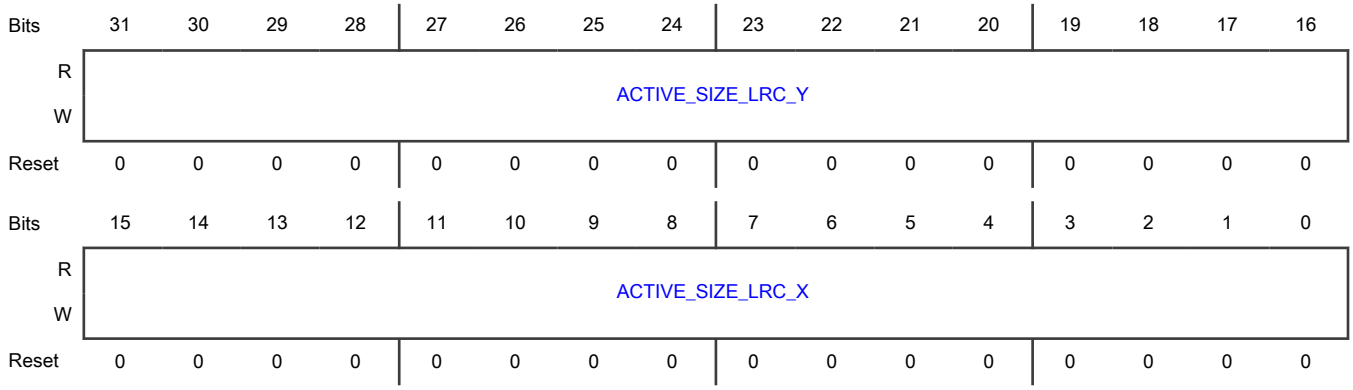
This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_ACTIVE_SIZE_LRC_CH1	4C0h

Diagram



Fields

Field	Description
31-16 ACTIVE_SIZE_LRC_Y	ACTIVE_SIZE_LRC_Y This field indicates the lower right Y-coordinate(in pixels) of the active surface of the total input memory
15-0 ACTIVE_SIZE_LRC_X	ACTIVE_SIZE_LRC_X This field indicates the lower right X-coordinate(in pixels) of the active surface of the total input memory

56.6.1.61 Input Fetch Size Channel 0 Register (INPUT_FETCH_SIZE_CH0)

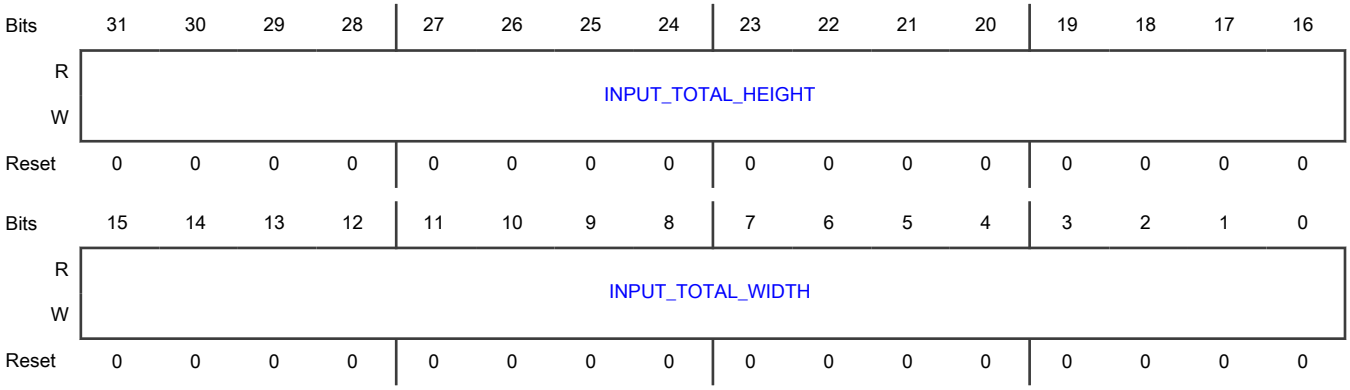
This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_SIZE_CH0	4D0h

Diagram



Fields

Field	Description
31-16 INPUT_TOTAL_HEIGHT	INPUT_TOTAL_HEIGHT actual total height - 1
15-0 INPUT_TOTAL_WIDTH	INPUT_TOTAL_WIDTH actual total width -1

56.6.1.62 Input Fetch Size Channel 1 Register (INPUT_FETCH_SIZE_CH1)

This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_SIZE_CH1	4E0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INPUT_TOTAL_HEIGHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INPUT_TOTAL_WIDTH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 INPUT_TOTAL_HEIGHT	INPUT_TOTAL_HEIGHT actual total height -1
15-0 INPUT_TOTAL_WIDTH	INPUT_TOTAL_WIDTH actual total width -1

56.6.1.63 Input Fetch Background Color Channel 0 Register (INPUT_FETCH_BACKGROUND_COLOR_CH0)

This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_BACKGROUND_COLOR_CH0	4F0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BACKGROUND_COLOR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BACKGROUND_COLOR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 BACKGROUND_COLOR	BACKGROUND_COLOR background color(in 32bpp format) for any pixels not within the buffer range specified by the ULC/LRC

56.6.1.64 Input Fetch Background Color Channel 1 Register (INPUT_FETCH_BACKGROUND_COLOR_CH1)

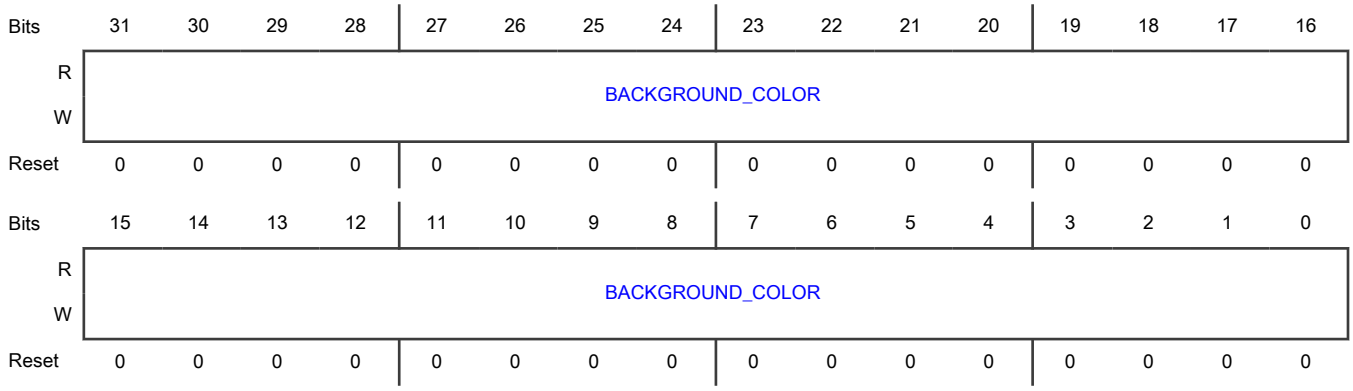
This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_BACKGROUND_COLOR_CH1	500h

Diagram



Fields

Field	Description
31-0 BACKGROUND_COLOR	BACKGROUND_COLOR background color(in 32bpp format) for any pixels not within the buffer range specified by the ULC/LRC

56.6.1.65 Input Fetch Pitch Register (INPUT_FETCH_PITCH)

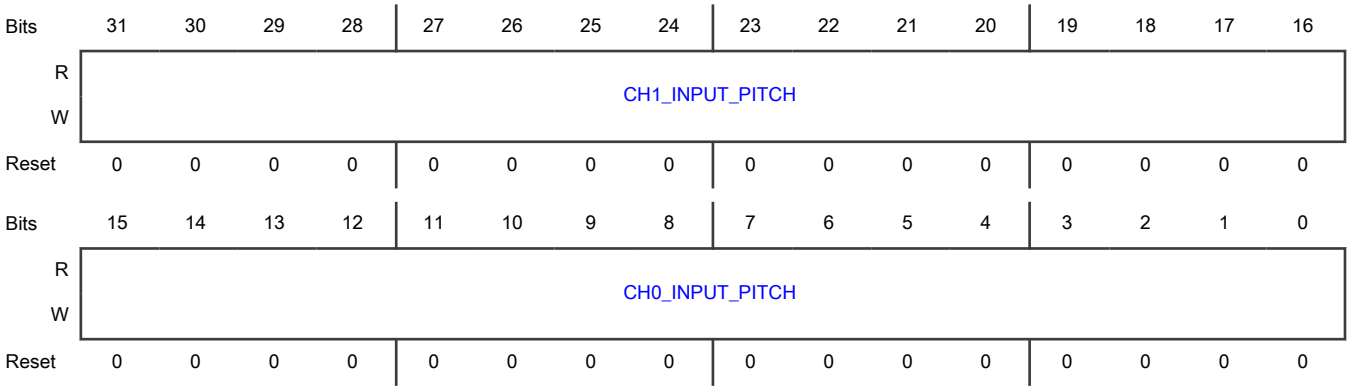
This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_PITCH	510h

Diagram



Fields

Field	Description
31-16 CH1_INPUT_PITCH	CH1_INPUT_PITCH This field indicates the channel 1 input pitch
15-0 CH0_INPUT_PITCH	CH0_INPUT_PITCH This field indicates the channel 0 input pitch

56.6.1.66 Input Fetch Shift Control Channel 0 Register (INPUT_FETCH_SHIFT_CTRL_CH0)

This register defines the control bits for the pxp prefetch_engine sub-block.

INPUT_FETCH_SHIFT_CTRL_CH0: 0x520

INPUT_FETCH_SHIFT_CTRL_CH0_SET: 0x524

INPUT_FETCH_SHIFT_CTRL_CH0_CLR: 0x528

INPUT_FETCH_SHIFT_CTRL_CH0_TOG: 0x52C

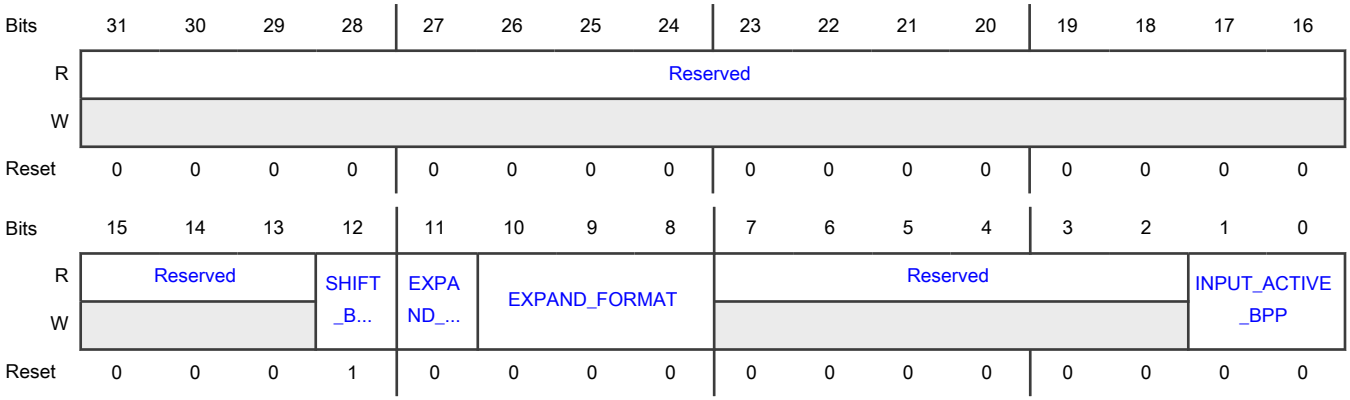
The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
INPUT_FETCH_SHIFT_CTRL_CH0	520h	Input Fetch Shift Control Channel 0 Register
INPUT_FETCH_SHIFT_CTRL_CH0_SET	524h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_SHIFT_CTRL_CH0 is 1
INPUT_FETCH_SHIFT_CTRL_CH0_CLR	528h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_SHIFT_CTRL_CH0 is 0
INPUT_FETCH_SHIFT_CTRL_CH0_TOG	52Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in INPUT_FETCH_SHIFT_CTRL_CH0

Diagram



Fields

Field	Description
31-13 —	Reserved
12 SHIFT_BYPASS	SHIFT_BYPASS 0 - channel0 data will do shift function 1 - channel0 will bypass shift function
11 EXPAND_EN	EXPAND_EN 0 - channel0 format expanding disable 1 - channel0 format expanding enable
10-8	EXPAND_FORMAT Select Pixel format

Table continues on the next page...

Table continued from the previous page...

Field	Description
EXPAND_FORMAT	000 - RGB 565 001 - RGB 555 010 - ARGB 1555 011 - RGB 444 100 - ARGB 4444 101 - YUYV/YVYU 110 - UYVY/VYUY 111 - YUV422_2P
7-2 —	Reserved
1-0 INPUT_ACTIVE_BPP	INPUT_ACTIVE_BPP 00 - 8 bits 01 - 16 bits 10 - 32 bits 11 - 64 bits

56.6.1.67 Input Fetch Shift Control Channel 1 Register (INPUT_FETCH_SHIFT_CTRL_CH1)

This register defines the control bits for the pxp prefetch_engine sub-block.

INPUT_FETCH_SHIFT_CTRL_CH1: 0x530

INPUT_FETCH_SHIFT_CTRL_CH1_SET: 0x534

INPUT_FETCH_SHIFT_CTRL_CH1_CLR: 0x538

INPUT_FETCH_SHIFT_CTRL_CH1_TOG: 0x53C

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

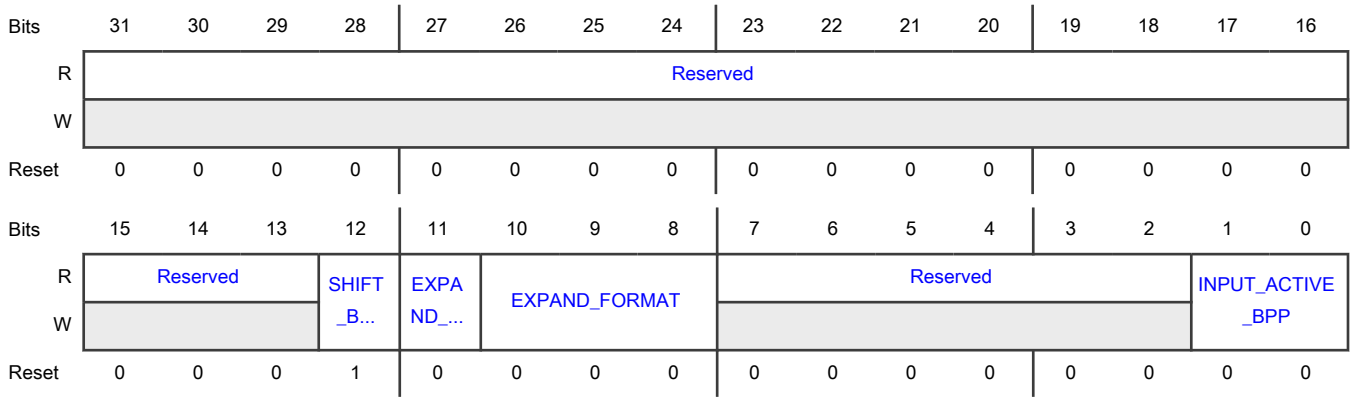
Register	Offset	Description
INPUT_FETCH_SHIFT_CTRL_CH1	530h	Input Fetch Shift Control Channel 1 Register
INPUT_FETCH_SHIFT_CTRL_CH1_SET	534h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_SHIFT_CTRL_CH1 is 1

Table continues on the next page...

Table continued from the previous page...

Register	Offset	Description
INPUT_FETCH_SHIFT_CTRL_CH1_CLR	538h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_SHIFT_CTRL_CH1 is 0
INPUT_FETCH_SHIFT_CTRL_CH1_TOG	53Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in INPUT_FETCH_SHIFT_CTRL_CH1

Diagram



Fields

Field	Description
31-13 —	Reserved
12 SHIFT_BYPASS	SHIFT_BYPASS 0 - channel1 data will do shift function 1 - channel1 will bypass shift function
11 EXPAND_EN	EXPAND_EN 0 - channel1 format expanding disable 1 - channel1 format expanding enable
10-8 EXPAND_FORMAT	EXPAND_FORMAT Select Pixel format 000 - RGB 565 001 - RGB 555 010 - ARGB 1555

Table continues on the next page...

Table continued from the previous page...

Field	Description
	011 - RGB 444 100 - ARGB 4444 101 - YUYV/YVYU 110 - UYVY/VYUY 111 - YUV422_2P
7-2 —	Reserved
1-0 INPUT_ACTIVE_BPP	INPUT_ACTIVE_BPP 00 - 8 bits 01 - 16 bits 10 - 32 bits 11 - 32 bits

56.6.1.68 Input Fetch Shift Offset Channel 0 Register (INPUT_FETCH_SHIFT_OFFSET_CH0)

This register defines the control bits for the pxp prefetch_engine sub-block.

INPUT_FETCH_SHIFT_OFFSET_CH0: 0x540

INPUT_FETCH_SHIFT_OFFSET_CH0_SET: 0x544

INPUT_FETCH_SHIFT_OFFSET_CH0_CLR: 0x548

INPUT_FETCH_SHIFT_OFFSET_CH0_TOG: 0x54C

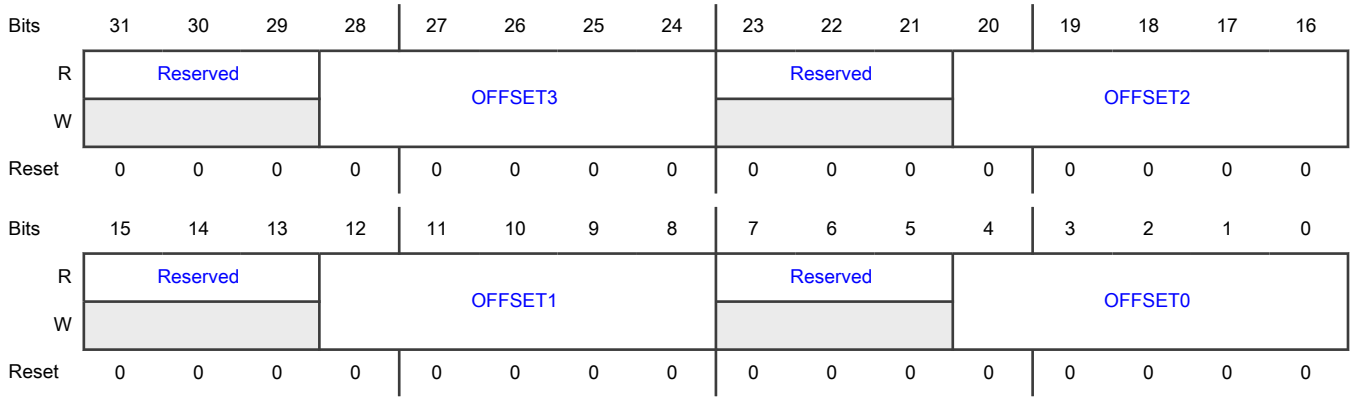
The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
INPUT_FETCH_SHIFT_OFFSET_CH0	540h	Input Fetch Shift Offset Channel 0 Register
INPUT_FETCH_SHIFT_OFFSET_CH0_SET	544h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_SHIFT_OFFSET_CH0 is 1
INPUT_FETCH_SHIFT_OFFSET_CH0_CLR	548h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_SHIFT_OFFSET_CH0 is 0
INPUT_FETCH_SHIFT_OFFSET_CH0_TOG	54Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in INPUT_FETCH_SHIFT_OFFSET_CH0

Diagram



Fields

Field	Description
31-29 —	Reserved
28-24 OFFSET3	OFFSET3 Shift Offset for channel 0 component 3.
23-21 —	Reserved
20-16 OFFSET2	OFFSET2 Shift Offset for channel 0 component 2.
15-13 —	Reserved
12-8 OFFSET1	OFFSET1 Shift Offset for channel 0 component 1.
7-5 —	Reserved
4-0 OFFSET0	OFFSET0 Shift Offset for channel 0 component 0.

56.6.1.69 Input Fetch Shift Offset Channel 1 Register (INPUT_FETCH_SHIFT_OFFSET_CH1)

This register defines the control bits for the pxp prefetch_engine sub-block.

INPUT_FETCH_SHIFT_OFFSET_CH1: 0x550

INPUT_FETCH_SHIFT_OFFSET_CH1_SET: 0x554

INPUT_FETCH_SHIFT_OFFSET_CH1_CLR: 0x558

INPUT_FETCH_SHIFT_OFFSET_CH1_TOG: 0x55C

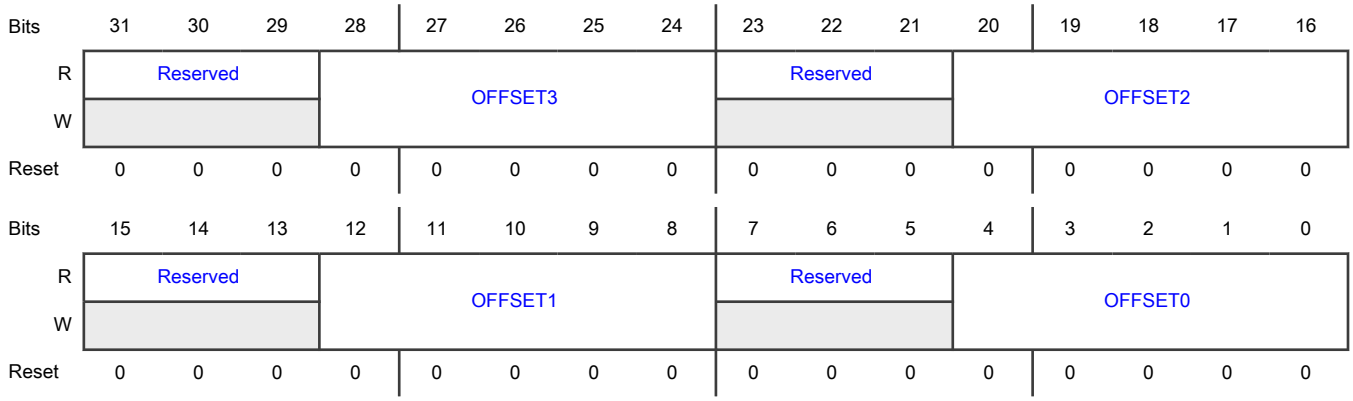
The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
INPUT_FETCH_SHIFT_OFFSET_CH1	550h	Input Fetch Shift Offset Channel 1 Register
INPUT_FETCH_SHIFT_OFFSET_CH1_SET	554h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_SHIFT_OFFSET_CH1 is 1
INPUT_FETCH_SHIFT_OFFSET_CH1_CLR	558h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_SHIFT_OFFSET_CH1 is 0
INPUT_FETCH_SHIFT_OFFSET_CH1_TOG	55Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in INPUT_FETCH_SHIFT_OFFSET_CH1

Diagram



Fields

Field	Description
31-29 —	Reserved
28-24 OFFSET3	OFFSET3 Shift Offset for channel 1 component 3.
23-21 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
20-16 OFFSET2	OFFSET2 Shift Offset for channel 1 component 2.
15-13 —	Reserved
12-8 OFFSET1	OFFSET1 Shift Offset for channel 1 component 1.
7-5 —	Reserved
4-0 OFFSET0	OFFSET0 Shift Offset for channel 1 component 0.

56.6.1.70 Input Fetch Shift Width Channel 0 Register (INPUT_FETCH_SHIFT_WIDTH_CH0)

This register defines the control bits for the pxp prefetch_engine sub-block.

INPUT_FETCH_SHIFT_WIDTH_CH0: 0x560

INPUT_FETCH_SHIFT_WIDTH_CH0_SET: 0x564

INPUT_FETCH_SHIFT_WIDTH_CH0_CLR: 0x568

INPUT_FETCH_SHIFT_WIDTH_CH0_TOG: 0x56C

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
INPUT_FETCH_SHIFT_WIDTH_CH0	560h	Input Fetch Shift Width Channel 0 Register
INPUT_FETCH_SHIFT_WIDTH_CH0_SET	564h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_SHIFT_WIDTH_CH0 is 1
INPUT_FETCH_SHIFT_WIDTH_CH0_CLR	568h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_SHIFT_WIDTH_CH0 is 0
INPUT_FETCH_SHIFT_WIDTH_CH0_TOG	56Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in INPUT_FETCH_SHIFT_WIDTH_CH0

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WIDTH3				WIDTH2				WIDTH1				WIDTH0			
W																
Reset	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-12 WIDTH3	WIDTH3 Shift Width for channel 0 component 3.
11-8 WIDTH2	WIDTH2 Shift Width for channel 0 component 2.
7-4 WIDTH1	WIDTH1 Shift Width for channel 0 component 1.
3-0 WIDTH0	WIDTH0 Shift Width for channel 0 component 0.

56.6.1.71 Input Fetch Shift Width Channel 1 Register (INPUT_FETCH_SHIFT_WIDTH_CH1)

This register defines the control bits for the pxp prefetch_engine sub-block.

INPUT_FETCH_SHIFT_WIDTH_CH1: 0x570

INPUT_FETCH_SHIFT_WIDTH_CH1_SET: 0x574

INPUT_FETCH_SHIFT_WIDTH_CH1_CLR: 0x578

INPUT_FETCH_SHIFT_WIDTH_CH1_TOG: 0x57C

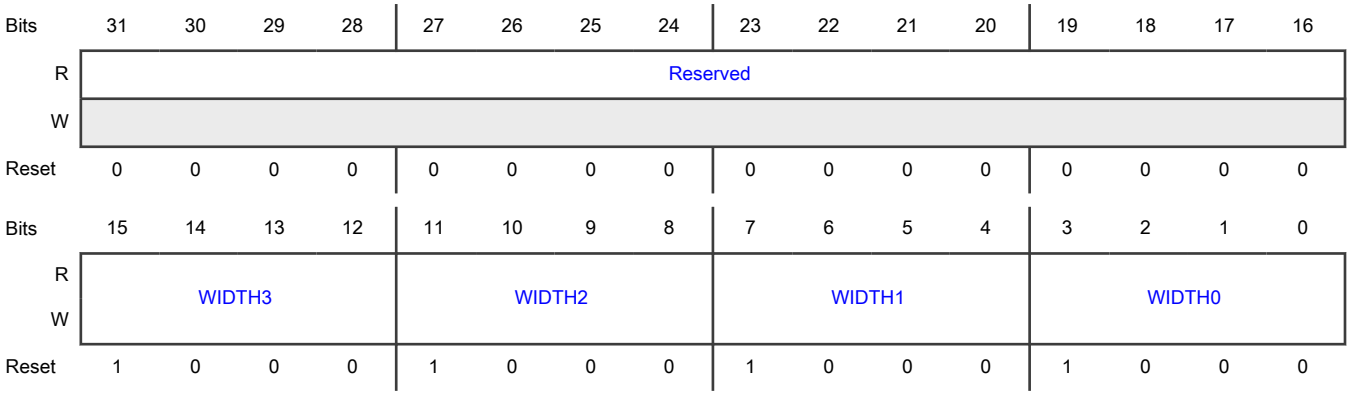
The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
INPUT_FETCH_SHIFT_WIDTH_CH1	570h	Input Fetch Shift Width Channel 1 Register
INPUT_FETCH_SHIFT_WIDTH_CH1_SET	574h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_SHIFT_WIDTH_CH1 is 1
INPUT_FETCH_SHIFT_WIDTH_CH1_CLR	578h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_FETCH_SHIFT_WIDTH_CH1 is 0
INPUT_FETCH_SHIFT_WIDTH_CH1_TOG	57Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in INPUT_FETCH_SHIFT_WIDTH_CH1

Diagram



Fields

Field	Description
31-16 —	Reserved
15-12 WIDTH3	WIDTH3 Shift Width for channel 1 component 3.
11-8 WIDTH2	WIDTH2 Shift Width for channel 1 component 2.
7-4 WIDTH1	WIDTH1 Shift Width for channel 1 component 1.
3-0 WIDTH0	WIDTH0 Shift Width for channel 1 component 0.

56.6.1.72 Input Fetch Address 0 Channel 0 Register (INPUT_FETCH_ADDR_0_CH0)

This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_ADDR_0_CH0	580h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INPUT_BASE_ADDR0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INPUT_BASE_ADDR0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 INPUT_BASE_ADDR0	INPUT_BASE_ADDR0 input base address0. For 2 channel, indicated the channel0 base address. For 1 channel and YUV422 2 plane, indicate the Y base address

56.6.1.73 Input Fetch Address 1 Channel 0 Register (INPUT_FETCH_ADDR_1_CH0)

This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_ADDR_1_CH0	590h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INPUT_BASE_ADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INPUT_BASE_ADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 INPUT_BASE_ADDR1	INPUT_BASE_ADDR1 input base address1. For 2 channel, indicated the channel1 base address. For 1 channel and YUV422 2 plane, indicate the UV base address

56.6.1.74 Input Fetch Address 0 Channel 1 Register (INPUT_FETCH_ADDR_0_CH1)

This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_ADDR_0_CH1	5A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INPUT_BASE_ADDR0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INPUT_BASE_ADDR0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 INPUT_BASE_ADDR0	INPUT_BASE_ADDR0 input base address0. For 2 channel, indicated the channel0 base address. For 1 channel and YUV422 2 plane, indicate the Y base address

56.6.1.75 Input Fetch Address 1 Channel 1 Register (INPUT_FETCH_ADDR_1_CH1)

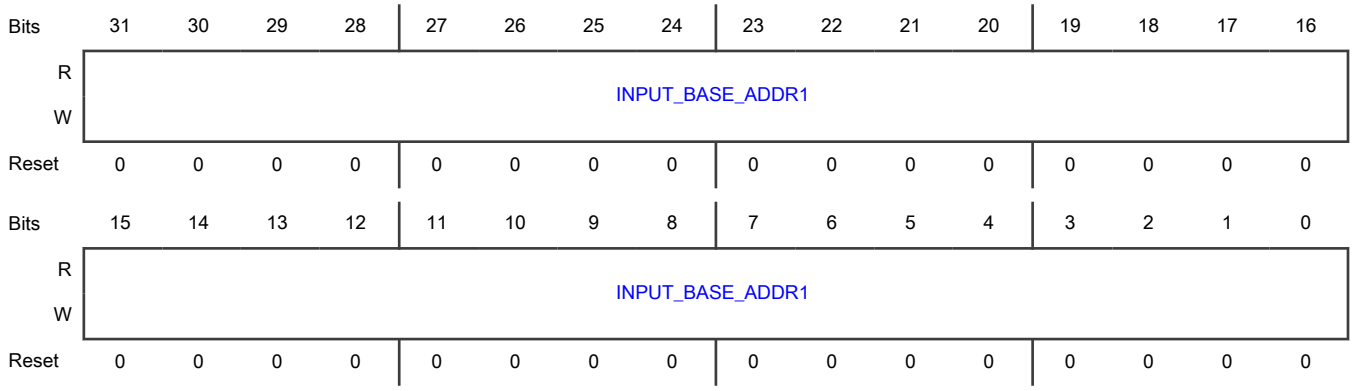
This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

Register	Offset
INPUT_FETCH_ADDR_1_CH1	5B0h

Diagram



Fields

Field	Description
31-0 INPUT_BASE_ADDR1	INPUT_BASE_ADDR1 input base address1. For 2 channel, indicated the channel1 base address. For 1 channel and YUV422 2 plane, indicate the UV base address

56.6.1.76 Input Store Control Channel 0 Register (INPUT_STORE_CTRL_CH0)

This register defines the control bits for the pxp store_engine sub-block.

INPUT_STORE_CTRL_CH0: 0x5C0

INPUT_STORE_CTRL_CH0_SET: 0x5C4

INPUT_STORE_CTRL_CH0_CLR: 0x5C8

INPUT_STORE_CTRL_CH0_TOG: 0x5CC

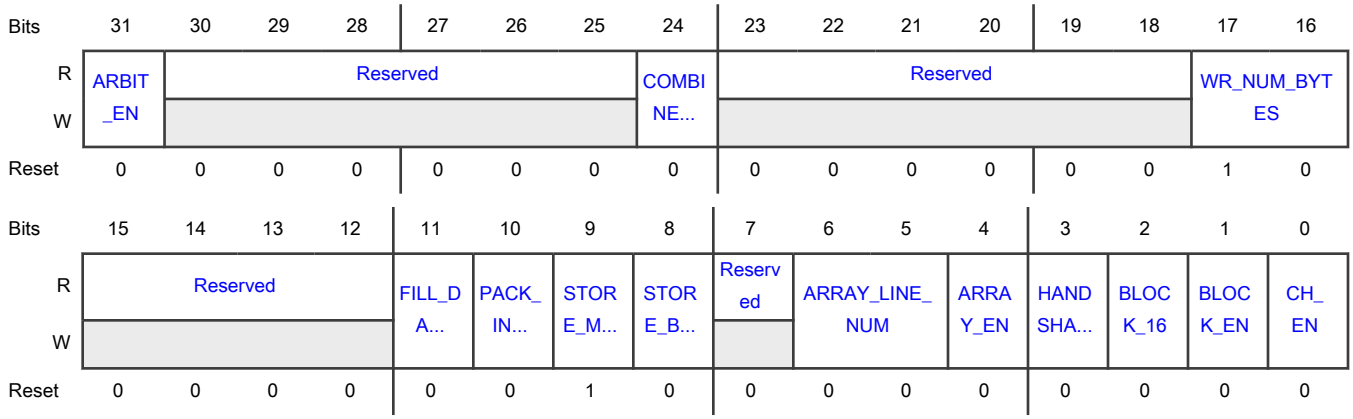
The Control register contains the control bits for the pxp store_engine sub-block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
INPUT_STORE_CTRL_CH0	5C0h	Input Store Control Channel 0 Register
INPUT_STORE_CTRL_CH0_SET	5C4h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_STORE_CTRL_CH0 is 1
INPUT_STORE_CTRL_CH0_CLR	5C8h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_STORE_CTRL_CH0 is 0
INPUT_STORE_CTRL_CH0_TOG	5CCh	Writing 1 to a bit in this register inverts the value of the corresponding bit in INPUT_STORE_CTRL_CH0

Diagram



Fields

Field	Description
31 ARBIT_EN	ARBIT_EN Arbitration Enable 0 - Arbitration disable. If using 2 channels, will output 2 axi bus sets 1 - Arbitration enable. If using 2 channel, will only output 1 axi bus sets
30-25 —	Reserved
24	COMBINE_2CHANNEL

Table continues on the next page...

Table continued from the previous page...

Field	Description
COMBINE_2CHANNEL	Combine 2 channel Enable 0 - combine 2 channel disable 1 - combine 2 channel enable
23-18 —	Reserved
17-16 WR_NUM_BYTES	WR_NUM_BYTES Bytes in a write burst 00 - NUM_8_BYTES : 8 bytes 01 - NUM_16_BYTES : 16 bytes 10 - NUM_32_BYTES : 32 bytes 11 - NUM_64_BYTES : 64 bytes
15-12 —	Reserved
11 FILL_DATA_EN	FILL_DATA_EN fill data enable 0 - Fill data mode disable. 1 - Fill data mode enable. When using fill_data mode, store_engine will store fixed data defined in fill_data register
10 PACK_IN_SEL	PACK_IN_SEL pack_in_sel 0 - select 64 shift out data to pack 1 - select low 32 bit shift out data to pack
9 STORE_MEMORY_EN	STORE_MEMORY_EN store memory enable 0 - store memory mode disable. 1 - store memory mode enable. Data will store to memory
8 STORE_BYPASS_EN	STORE_BYPASS_EN store bypass enable 0 - store bypass mode disable. 1 - store bypass mode enable. Data will bypass to store output.
7	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
6-5 ARRAY_LINE_NUM	ARRAY_LINE_NUM Selects Array Size 00 - Using 1x1 Array 01 - Using 3x3 Array 10 - Using 5x5 Array 11 - Using 5x5 Array
4 ARRAY_EN	ARRAY_EN 0 - Array Handshake Disabled 1 - Array Handshake Enabled
3 HANDSHAKE_EN	HANDSHAKE_EN Enable bit for handshake with the store engine. 0 - Handshake with the prefetch engine is disabled 1 - Handshake with the prefetch engine is enabled
2 BLOCK_16	BLOCK_16 Determines the block size. 0 - BLK_SIZE_8x8 : Block size is 8x8 1 - BLK_SIZE_16x16 : Block size is 16x16
1 BLOCK_EN	BLOCK_EN Chooses the store mode. 0 - Store in scan mode 1 - Store in block mode
0 CH_EN	CH_EN Channel enable. 0 - Store function is disable 1 - Store function is enable

56.6.1.77 Input Store Control Channel 1 Register (INPUT_STORE_CTRL_CH1)

This register defines the control bits for the pxp prefetch_engine sub-block.

INPUT_STORE_CTRL_CH1: 0x5D0

INPUT_STORE_CTRL_CH1_SET: 0x5D4

INPUT_STORE_CTRL_CH1_CLR: 0x5D8

INPUT_STORE_CTRL_CH1_TOG: 0x5DC

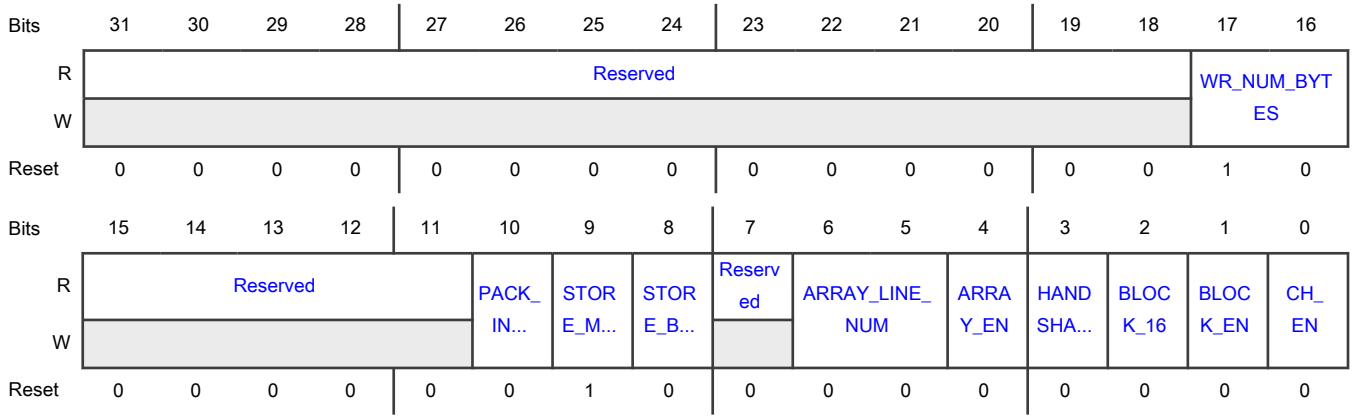
The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
INPUT_STORE_CTRL_CH1	5D0h	Input Store Control Channel 1 Register
INPUT_STORE_CTRL_CH1_SET	5D4h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_STORE_CTRL_CH1 is 1
INPUT_STORE_CTRL_CH1_CLR	5D8h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_STORE_CTRL_CH1 is 0
INPUT_STORE_CTRL_CH1_TOG	5DCh	Writing 1 to a bit in this register inverts the value of the corresponding bit in INPUT_STORE_CTRL_CH1

Diagram



Fields

Field	Description
31-18 —	Reserved
17-16 WR_NUM_BYTES	WR_NUM_BYTES Bytes in a write burst 00 - NUM_8_BYTES : 8 bytes 01 - NUM_16_BYTES : 16 bytes 10 - NUM_32_BYTES : 32 bytes 11 - NUM_64_BYTES : 64 bytes

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-11 —	Reserved
10 PACK_IN_SEL	PACK_IN_SEL pack_in_sel 0 - select 64 shift out data to pack 1 - select channel 0 high 32 bit shift out data to pack
9 STORE_MEMORY_EN	STORE_MEMORY_EN store memory enable 0 - store memory mode disable. 1 - store memory mode enable. Data will store to memory.
8 STORE_BYPASS_EN	STORE_BYPASS_EN enable bit for store bypass 0 - store bypass mode disable. 1 - store bypass mode enable. Data will bypass to store output.
7 —	Reserved
6-5 ARRAY_LINE_NUM	ARRAY_LINE_NUM Selects Array Size 00 - Using 1x1 Array 01 - Using 3x3 Array 10 - Using 5x5 Array 11 - Using 5x5 Array
4 ARRAY_EN	ARRAY_EN 0 - Array Handshake Disabled 1 - Array Handshake Enabled
3 HANDSHAKE_EN	HANDSHAKE_EN Enable bit for handshake with the fetch engine. 0 - Handshake with the fetch engine is disabled 1 - Handshake with the fetch engine is enabled
2	BLOCK_16 Determines the block size.

Table continues on the next page...

Table continued from the previous page...

Field	Description
BLOCK_16	0 - BLK_SIZE_8x8 : Block size is 8x8 1 - BLK_SIZE_16x16 : Block size is 16x16
1 BLOCK_EN	BLOCK_EN Chooses the store mode. 0 - Store in scan mode 1 - Store in block mode
0 CH_EN	CH_EN Channel enable. 0 - Store function is disable 1 - Store function is enable

56.6.1.78 Input Store Status Channel 0 Register (INPUT_STORE_STATUS_CH0)

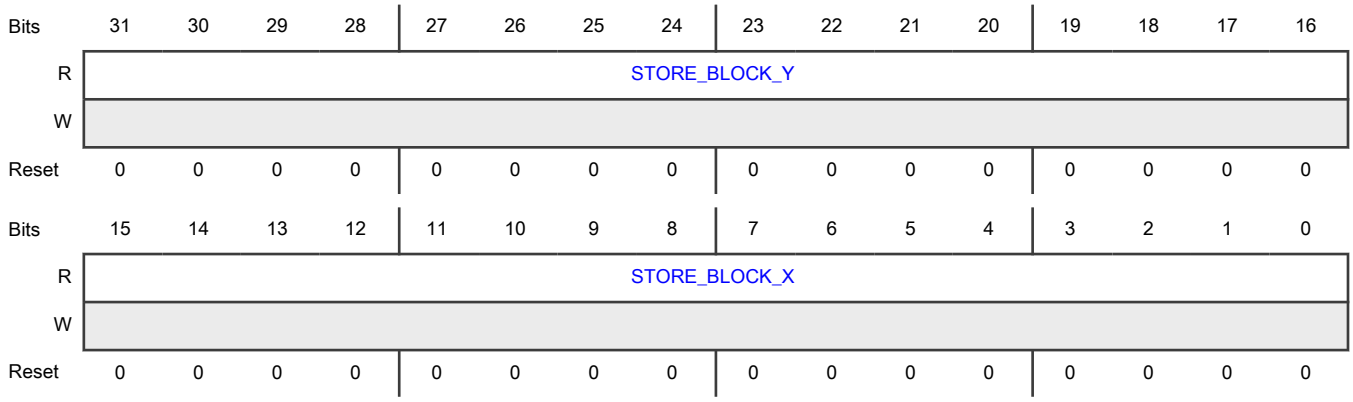
This register defines the status bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_STATUS_CH0	5E0h

Diagram



Fields

Field	Description
31-16 STORE_BLOCK_Y	STORE_BLOCK_Y When in scan mode, this field indicates the current Y coordinate of the frame. In block mode, it indicates the Y coordinate of the block currently being rendered.
15-0 STORE_BLOCK_X	STORE_BLOCK_X When in scan mode, this field is always 0. In block mode, it indicates the X coordinate of the block currently being rendered.

56.6.1.79 Input Store Status Channel 1 Register (INPUT_STORE_STATUS_CH1)

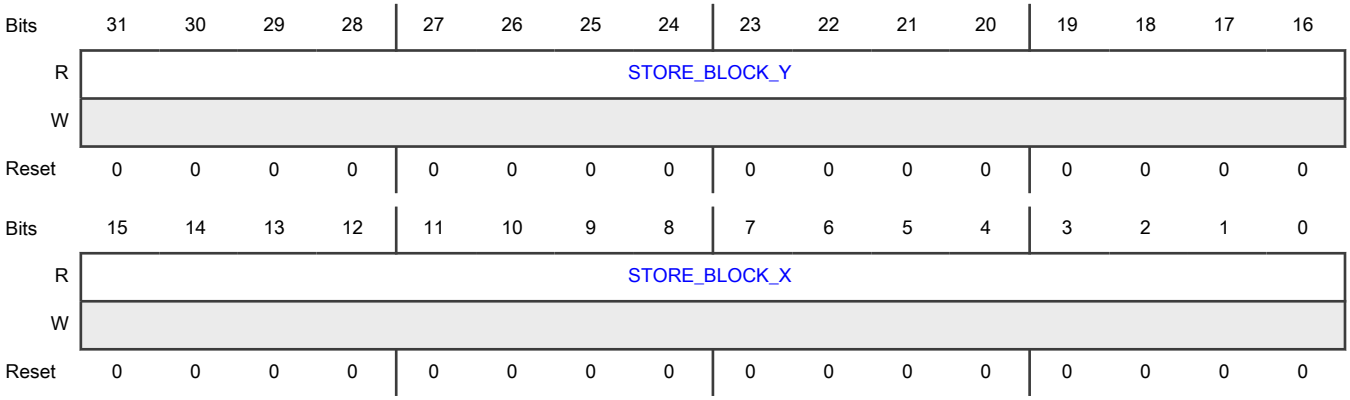
This register defines the status bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_STATUS_CH1	5F0h

Diagram



Fields

Field	Description
31-16 STORE_BLOCK_Y	STORE_BLOCK_Y When in scan mode, this field indicates the current Y coordinate of the frame. In block mode, it indicates the Y coordinate of the block currently being rendered.
15-0 STORE_BLOCK_X	STORE_BLOCK_X When in scan mode, this field is always 0. In block mode, it indicates the X coordinate of the block currently being rendered.

56.6.1.80 Input Store Size Channel 0 Register (INPUT_STORE_SIZE_CH0)

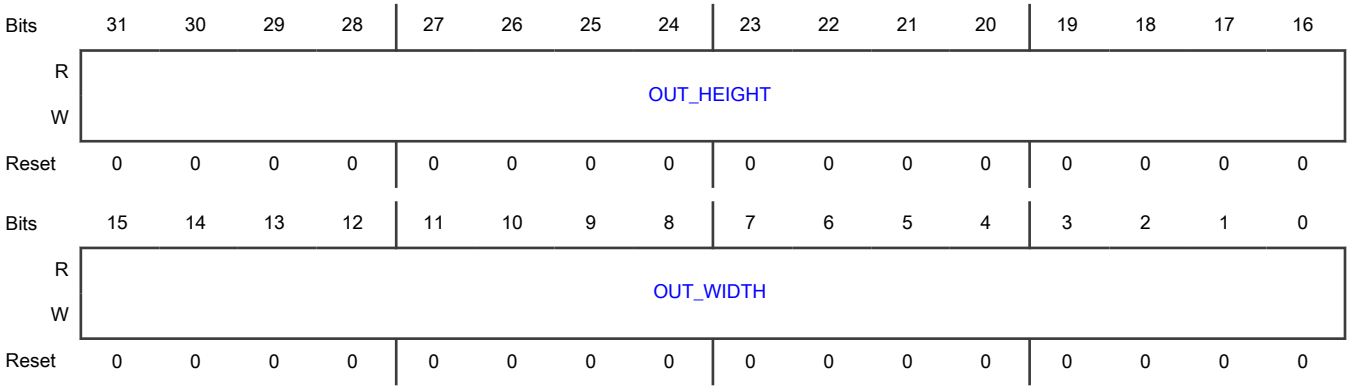
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_SIZE_CH0	600h

Diagram



Fields

Field	Description
31-16 OUT_HEIGHT	OUT_HEIGHT actual output height -1
15-0 OUT_WIDTH	OUT_WIDTH actual output width -1

56.6.1.81 Input Store Size Channel 1 Register (INPUT_STORE_SIZE_CH1)

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_SIZE_CH1	610h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OUT_HEIGHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OUT_WIDTH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 OUT_HEIGHT	OUT_HEIGHT actual output height -1
15-0 OUT_WIDTH	OUT_WIDTH actual output width -1

56.6.1.82 Input Store Pitch Register (INPUT_STORE_PITCH)

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_PITCH	620h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CH1_OUT_PITCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH0_OUT_PITCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 CH1_OUT_PITCH	CH1_OUT_PITCH This field indicates the channel 1 input pitch
15-0 CH0_OUT_PITCH	CH0_OUT_PITCH This field indicates the channel 0 input pitch

56.6.1.83 Input Store Shift Control Channel 0 Register (INPUT_STORE_SHIFT_CTRL_CH0)

This register defines the control bits for the pxp store_engine sub-block.

INPUT_STORE_SHIFT_CTRL_CH0: 0x630

INPUT_STORE_SHIFT_CTRL_CH0_SET: 0x634

INPUT_STORE_SHIFT_CTRL_CH0_CLR: 0x638

INPUT_STORE_SHIFT_CTRL_CH0_TOG: 0x63C

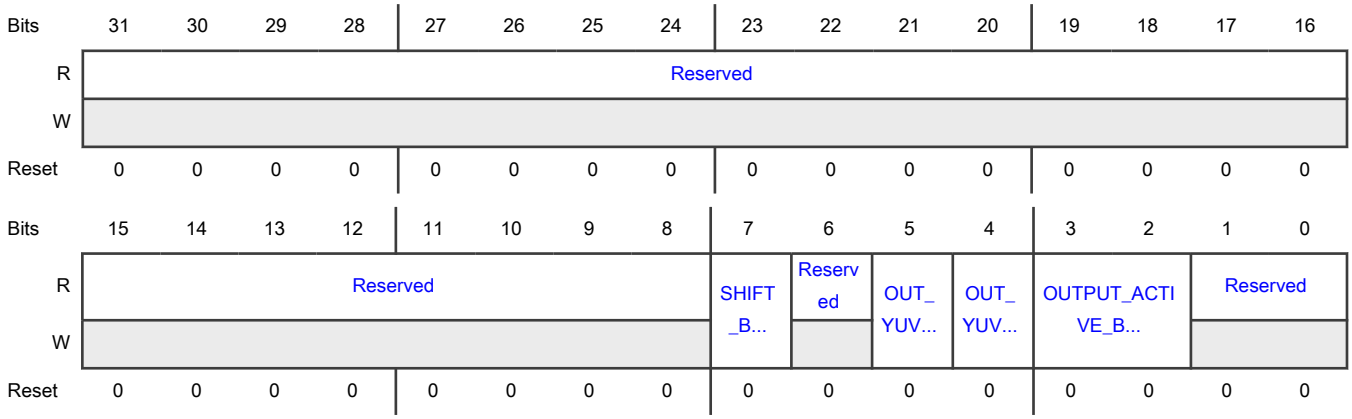
The Control register contains the control bits for the pxp store_engine sub-block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
INPUT_STORE_SHIFT_CTRL_CH0	630h	Input Store Shift Control Channel 0 Register
INPUT_STORE_SHIFT_CTRL_CH0_SET	634h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_STORE_SHIFT_CTRL_CH0 is 1
INPUT_STORE_SHIFT_CTRL_CH0_CLR	638h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_STORE_SHIFT_CTRL_CH0 is 0
INPUT_STORE_SHIFT_CTRL_CH0_TOG	63Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in INPUT_STORE_SHIFT_CTRL_CH0

Diagram



Fields

Field	Description
31-8 —	Reserved
7 SHIFT_BYPASS	SHIFT_BYPASS CH0 shift bypass 0 - data will do shift processing. 1 - data will bypass shift module.
6 —	Reserved
5 OUT_YUV422_2P_EN	OUT_YUV422_2P_EN Enable for YUV422 2 plane 0 - YUYV422 2 plane disabled. 1 - YUYV422 2 plane enabled.
4 OUT_YUV422_1P_EN	OUT_YUV422_1P_EN Enable for YUV422 1 plane 0 - YUYV422 2 plane disabled. 1 - YUYV422 2 plane enabled.
3-2 OUTPUT_ACTIVE_BPP	OUTPUT_ACTIVE_BPP 00 - 8 bits 01 - 16 bits 10 - 32 bits 11 - 64 bits

Table continues on the next page...

Table continued from the previous page...

Field	Description
1-0 —	Reserved

56.6.1.84 Input Store Shift Control Channel 1 Register (INPUT_STORE_SHIFT_CTRL_CH1)

This register defines the control bits for the pxp store_engine sub-block.

INPUT_STORE_SHIFT_CTRL_CH1: 0x640

INPUT_STORE_SHIFT_CTRL_CH1_SET: 0x644

INPUT_STORE_SHIFT_CTRL_CH1_CLR: 0x648

INPUT_STORE_SHIFT_CTRL_CH1_TOG: 0x64C

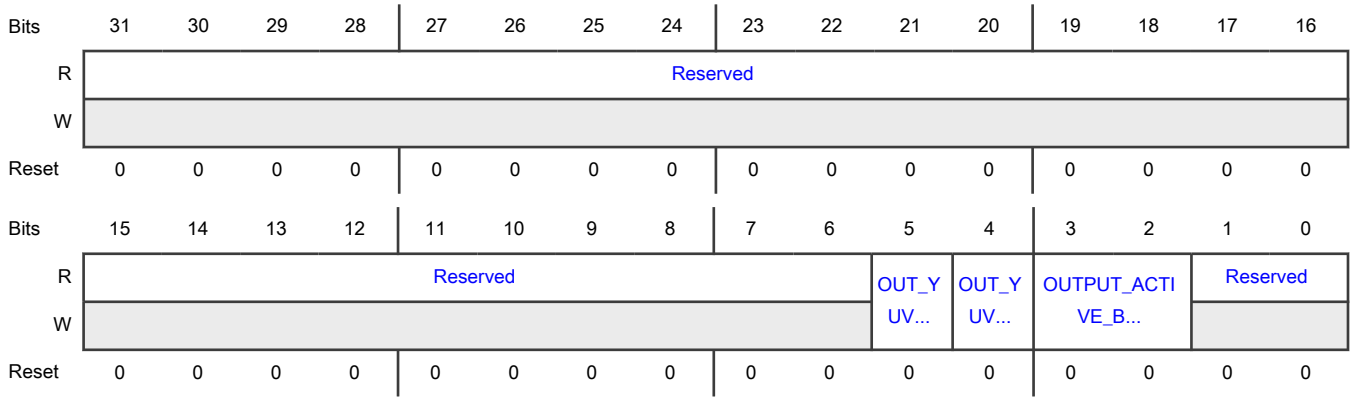
The Control register contains the control bits for the pxp prefetch_engine sub-block.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
INPUT_STORE_SHIFT_CTRL_CH1	640h	Input Store Shift Control Channel 1 Register
INPUT_STORE_SHIFT_CTRL_CH1_SET	644h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_STORE_SHIFT_CTRL_CH1 is 1
INPUT_STORE_SHIFT_CTRL_CH1_CLR	648h	Writing 1 to a bit in this register ensures that the corresponding bit in INPUT_STORE_SHIFT_CTRL_CH1 is 0
INPUT_STORE_SHIFT_CTRL_CH1_TOG	64Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in INPUT_STORE_SHIFT_CTRL_CH1

Diagram



Fields

Field	Description
31-6 —	Reserved
5 OUT_YUV422_2P_EN	OUT_YUV422_2P_EN Enable for YUV422 2 plane 0 - YUYV422 2 plane disabled. 1 - YUYV422 2 plane enabled.
4 OUT_YUV422_1P_EN	OUT_YUV422_1P_EN Enable for YUV422 1 plane 0 - YUYV422 2 plane disabled. 1 - YUYV422 2 plane enabled.
3-2 OUTPUT_ACTIVE_BPP	OUTPUT_ACTIVE_BPP 00 - 8 bits 01 - 16 bits 10 - 32 bits 11 - 32 bits
1-0 —	Reserved

56.6.1.85 Input Store Address 0 Channel 0 Register (INPUT_STORE_ADDR_0_CH0)

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_ADDR_0_CH0	690h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OUT_BASE_ADDR0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OUT_BASE_ADDR0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OUT_BASE_ADDR0	OUT_BASE_ADDR0 input base address0. For 2 channel, indicated the channel0 base address. For 1 channel and YUV422 2 plane, indicate the Y base address

56.6.1.86 Input Store Address 1 Channel 0 Register (INPUT_STORE_ADDR_1_CH0)

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_ADDR_1_CH0	6A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OUT_BASE_ADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OUT_BASE_ADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OUT_BASE_ADDR1	OUT_BASE_ADDR1 input base address1. For 2 channel, indicated the channel 1 base address. For 1 channel and YUV422 2 plane, indicate the UV base address

56.6.1.87 Input Store Fill Data Channel 0 Register (INPUT_STORE_FILL_DATA_CH0)

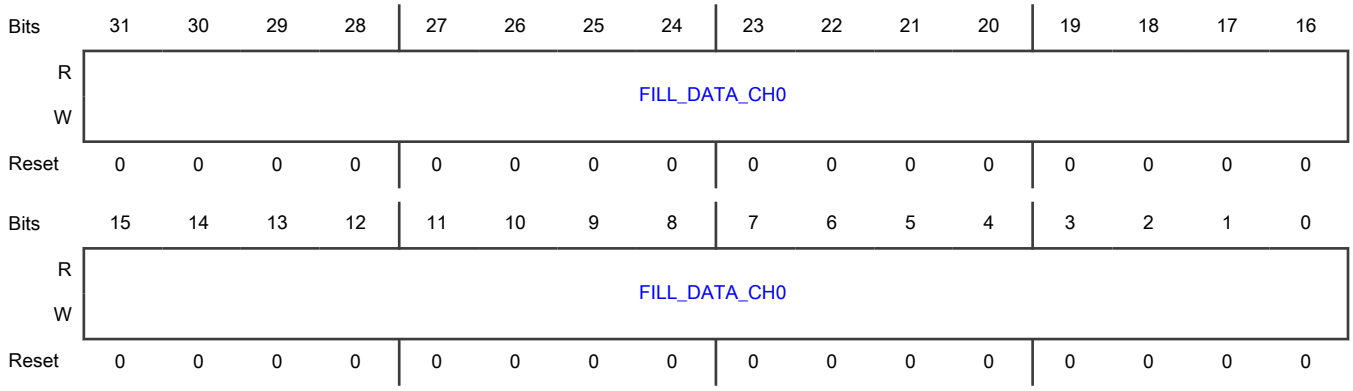
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_FILL_DATA_CH0	6B0h

Diagram



Fields

Field	Description
31-0 FILL_DATA_CH0	FILL_DATA_CH0 when using fill_data mode,store engine channel0 will store the fill_data value defined here.

56.6.1.88 Input Store Address 0 Channel 1 Register (INPUT_STORE_ADDR_0_CH1)

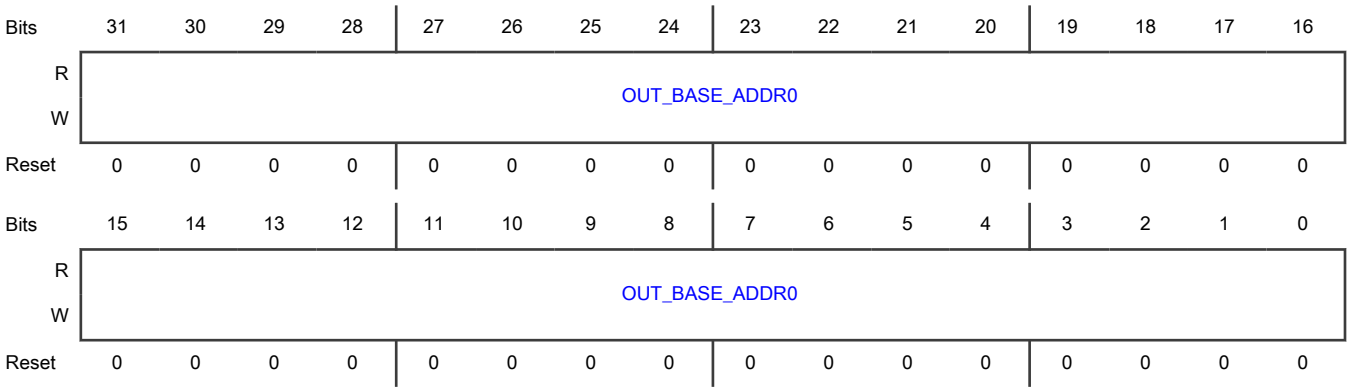
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_ADDR_0_CH1	6C0h

Diagram



Fields

Field	Description
31-0 OUT_BASE_ADDR0	OUT_BASE_ADDR0 input base address0. For 2 channel, indicated the channel0 base address. For 1 channel and YUV422 2 plane, indicate the Y base address

56.6.1.89 Input Store Address 1 Channel 1 Register (INPUT_STORE_ADDR_1_CH1)

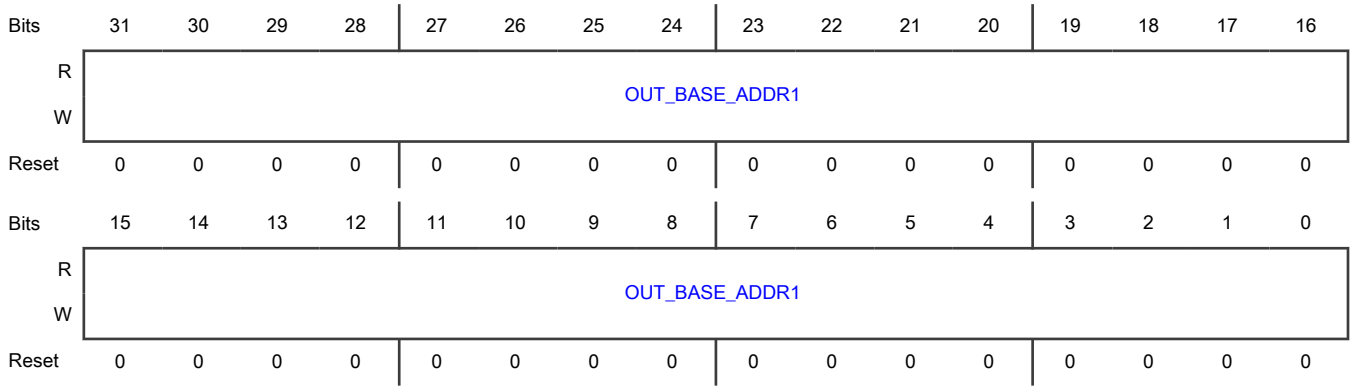
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_ADDR_1_CH1	6D0h

Diagram



Fields

Field	Description
31-0 OUT_BASE_ADDR1	OUT_BASE_ADDR1 input base address1. For 2 channel, indicated the channel 1 base address. For 1 channel and YUV422 2 plane, indicate the UV base address

56.6.1.90 Input Store Data Mask 0 High Byte Channel 0 Register (INPUT_STORE_D_MASK0_H_CH0)

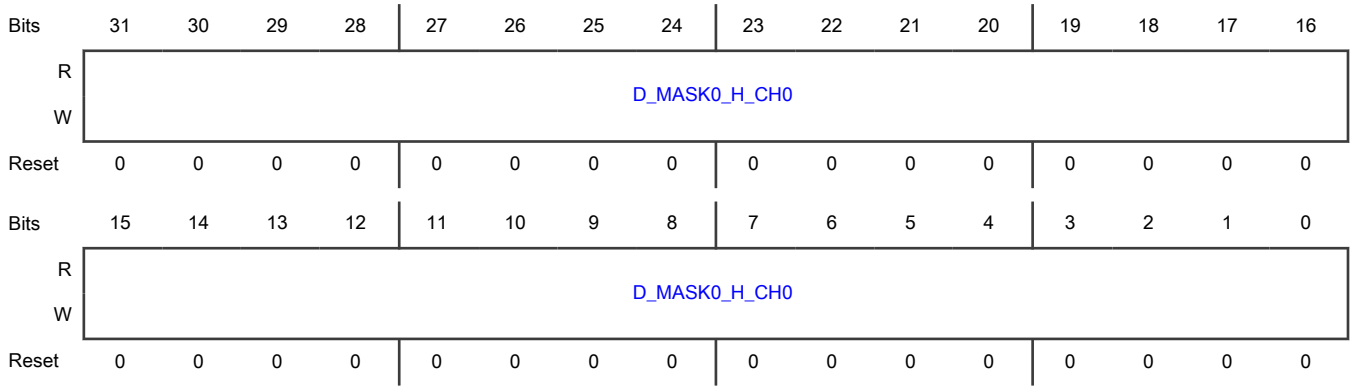
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK0_H_CH0	6E0h

Diagram



Fields

Field	Description
31-0 D_MASK0_H_CH0	D_MASK0_H_CH0 data mask0 high byte

56.6.1.91 Input Store Data Mask 0 Low Byte Channel 0 Register (INPUT_STORE_D_MASK0_L_CH0)

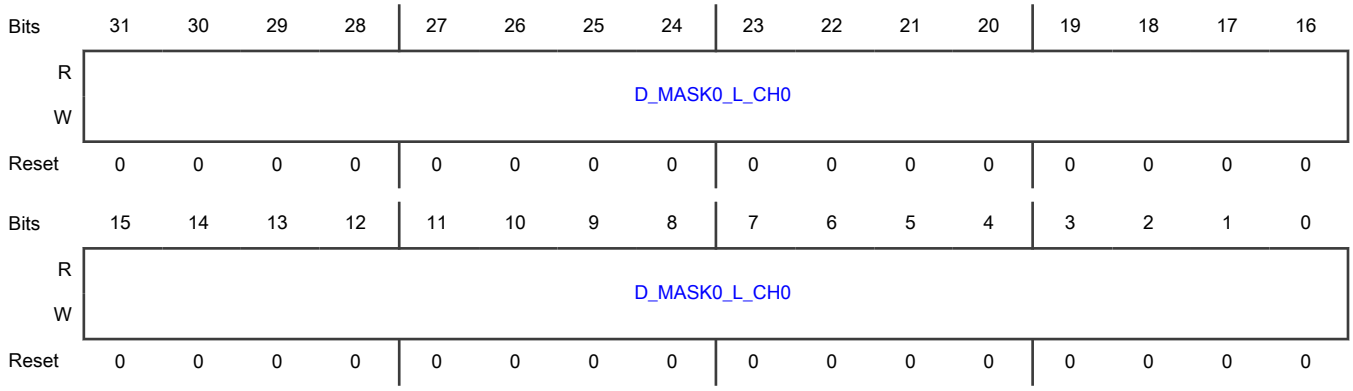
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK0_L_CH0	6F0h

Diagram



Fields

Field	Description
31-0 D_MASK0_L_CH0	D_MASK0_L_CH0 data mask0 low byte

56.6.1.92 Input Store Data Mask 1 High Byte Channel 0 Register (INPUT_STORE_D_MASK1_H_CH0)

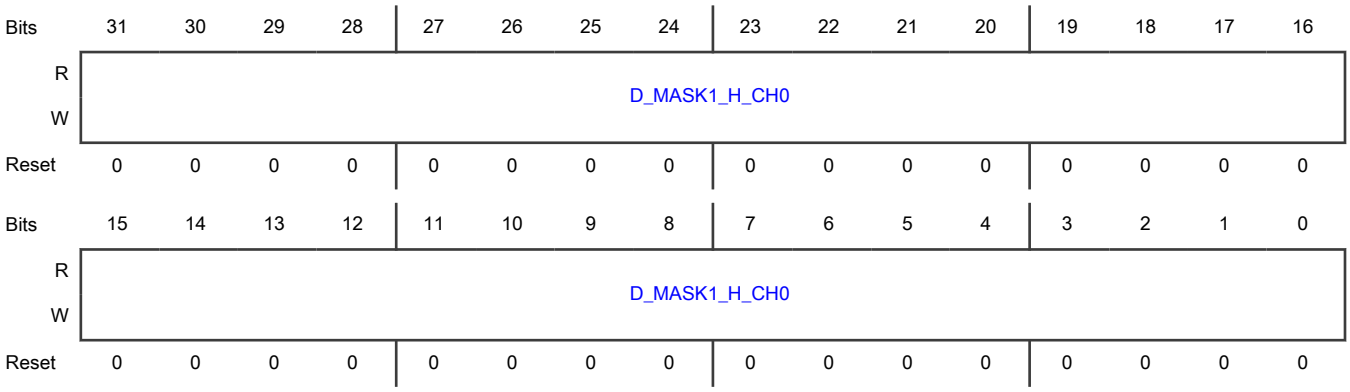
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK1_H_CH0	700h

Diagram



Fields

Field	Description
31-0 D_MASK1_H_CH0	D_MASK1_H_CH0 data mask1 high byte

56.6.1.93 Input Store Data Mask 1 Low Byte Channel 0 Register (INPUT_STORE_D_MASK1_L_CH0)

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK1_L_CH0	710h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	D_MASK1_L_CH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	D_MASK1_L_CH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 D_MASK1_L_CH0	D_MASK1_L_CH0 data mask1 low byte

56.6.1.94 Input Store Data Mask 2 High Byte Channel 0 Register (INPUT_STORE_D_MASK2_H_CH0)

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK2_H_CH0	720h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	D_MASK2_H_CH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	D_MASK2_H_CH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 D_MASK2_H_CH0	D_MASK2_H_CH0 data mask2 high byte

56.6.1.95 Input Store Data Mask 2 Low Byte Channel 0 Register (INPUT_STORE_D_MASK2_L_CH0)

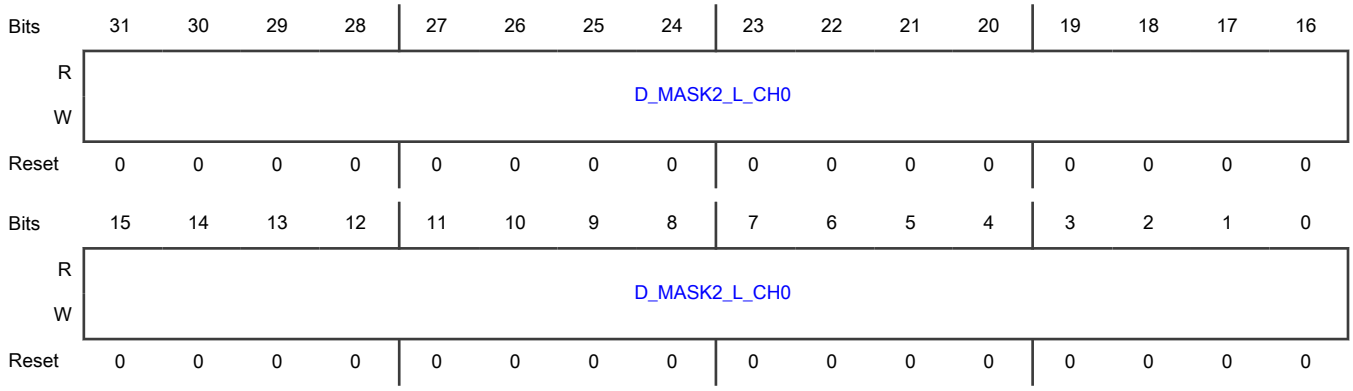
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK2_L_CH0	730h

Diagram



Fields

Field	Description
31-0 D_MASK2_L_CH0	D_MASK2_L_CH0 data mask2 low byte

56.6.1.96 Input Store Data Mask 3 High Byte Channel 0 Register (INPUT_STORE_D_MASK3_H_CH0)

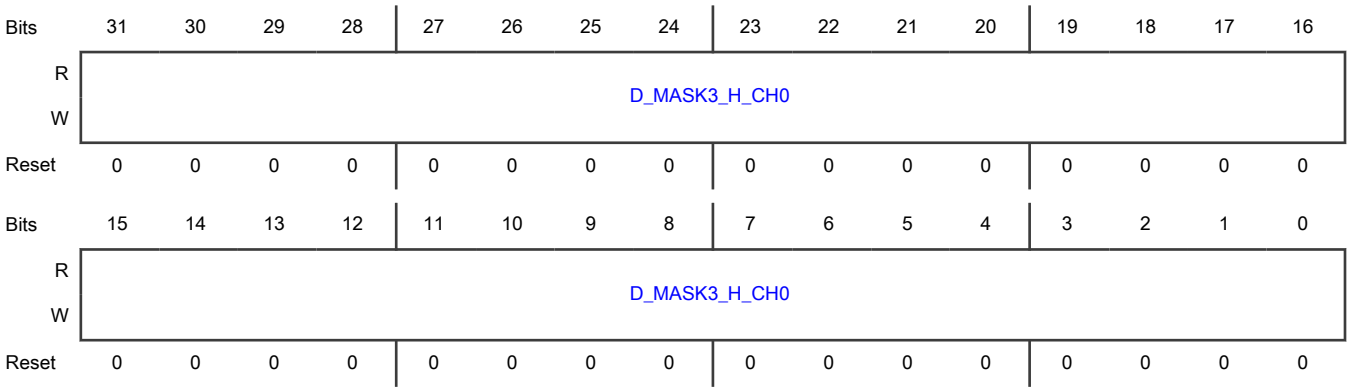
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK3_H_CH0	740h

Diagram



Fields

Field	Description
31-0 D_MASK3_H_CH0	D_MASK3_H_CH0 data mask3 high byte

56.6.1.97 Input Store Data Mask 3 Low Byte Channel 0 Register (INPUT_STORE_D_MASK3_L_CH0)

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK3_L_CH0	750h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	D_MASK3_L_CH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	D_MASK3_L_CH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 D_MASK3_L_CH0	D_MASK3_L_CH0 data mask3 low byte

56.6.1.98 Input Store Data Mask 4 High Byte Channel 0 Register (INPUT_STORE_D_MASK4_H_CH0)

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK4_H_CH0	760h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	D_MASK4_H_CH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	D_MASK4_H_CH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 D_MASK4_H_CH0	D_MASK4_H_CH0 data mask4 high byte

56.6.1.99 Input Store Data Mask 4 Low Byte Channel 0 Register (INPUT_STORE_D_MASK4_L_CH0)

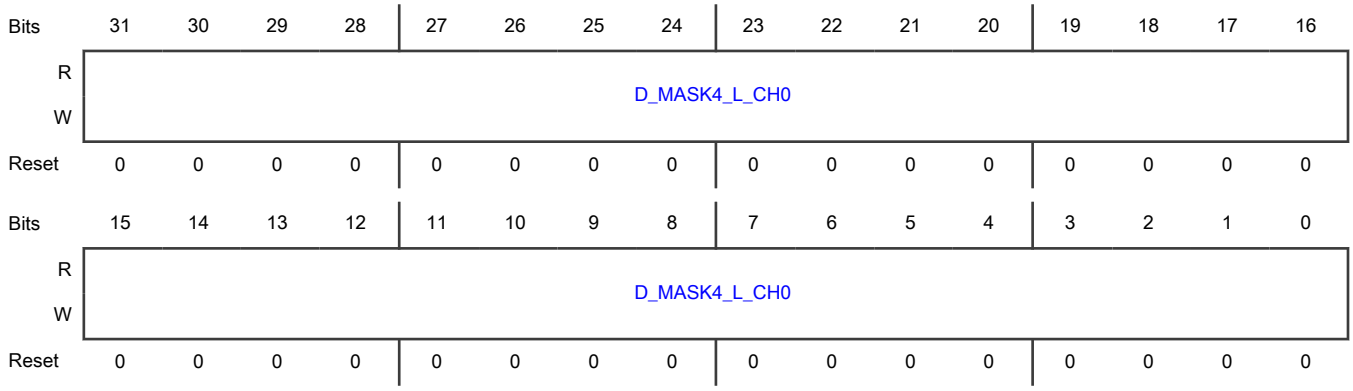
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK4_L_CH0	770h

Diagram



Fields

Field	Description
31-0 D_MASK4_L_CH0	D_MASK4_L_CH0 data mask4 low byte

56.6.1.100 Input Store Data Mask 5 High Byte Channel 0 Register (INPUT_STORE_D_MASK5_H_CH0)

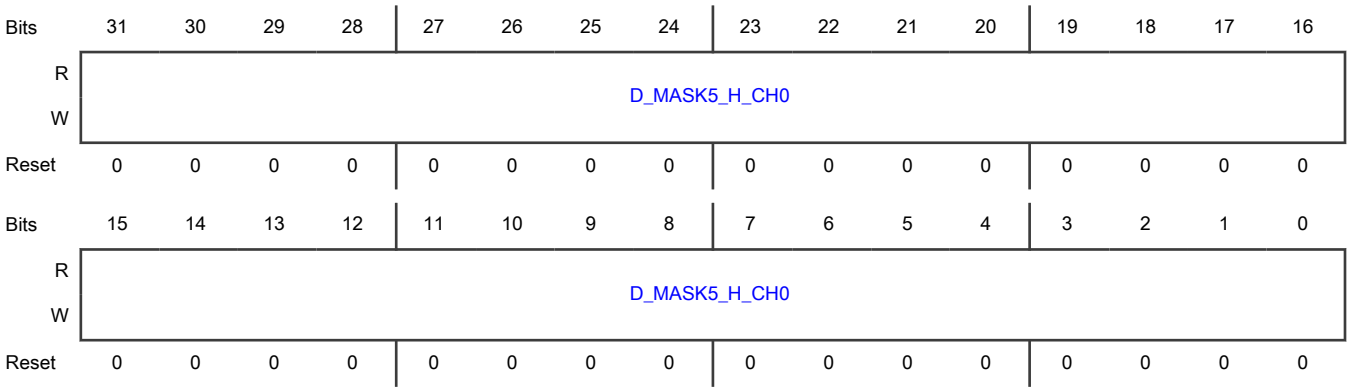
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK5_H_CH0	780h

Diagram



Fields

Field	Description
31-0 D_MASK5_H_CH0	D_MASK5_H_CH0 data mask5 high byte

56.6.1.101 Input Store Data Mask 5 Low Byte Channel 0 Register (INPUT_STORE_D_MASK5_L_CH0)

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK5_L_CH0	790h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	D_MASK5_L_CH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	D_MASK5_L_CH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 D_MASK5_L_CH0	D_MASK5_L_CH0 data mask5 low byte

56.6.1.102 Input Store Data Mask 6 High Byte Channel 0 Register (INPUT_STORE_D_MASK6_H_CH0)

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK6_H_CH0	7A0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	D_MASK6_H_CH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	D_MASK6_H_CH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 D_MASK6_H_CH0	D_MASK6_H_CH0 data mask6 high byte

56.6.1.103 Input Store Data Mask 6 Low Byte Channel 0 Register (INPUT_STORE_D_MASK6_L_CH0)

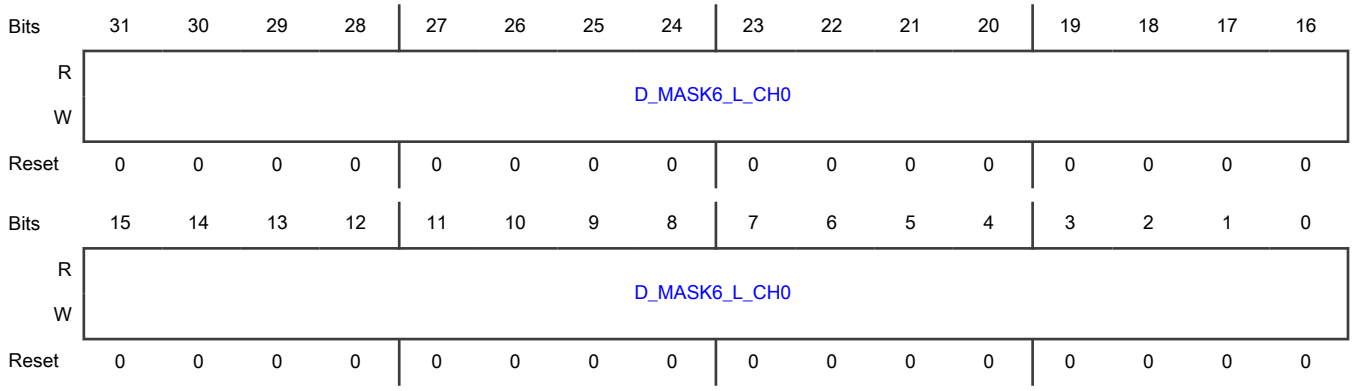
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK6_L_CH0	7B0h

Diagram



Fields

Field	Description
31-0 D_MASK6_L_CH0	D_MASK6_L_CH0 data mask6 low byte

56.6.1.104 Input Store Data Mask 7 High Byte Channel 0 Register (INPUT_STORE_D_MASK7_H_CH0)

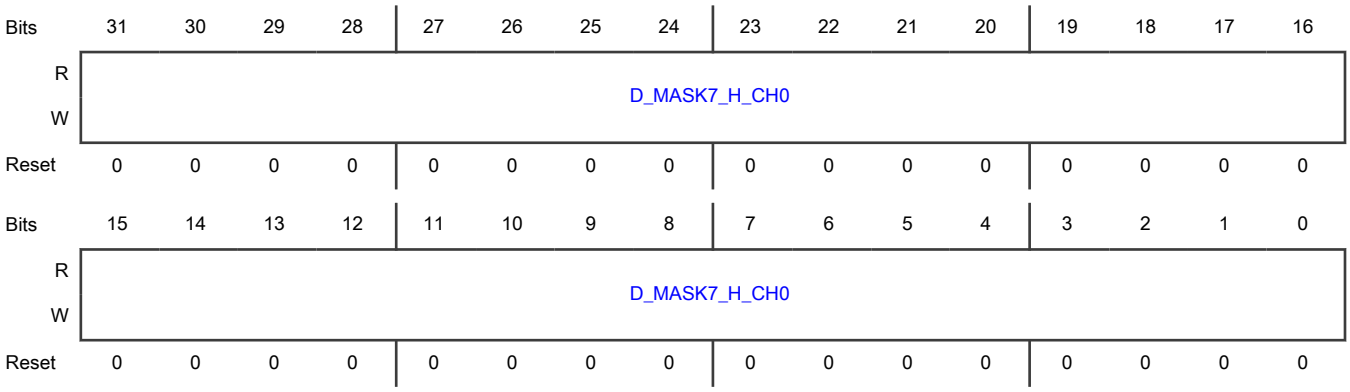
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK7_H_CH0	7C0h

Diagram



Fields

Field	Description
31-0 D_MASK7_H_CH0	D_MASK7_H_CH0 data mask7 high byte

56.6.1.105 Input Store Data Mask 7 Low Byte Channel 0 Register (INPUT_STORE_D_MASK7_L_CH0)

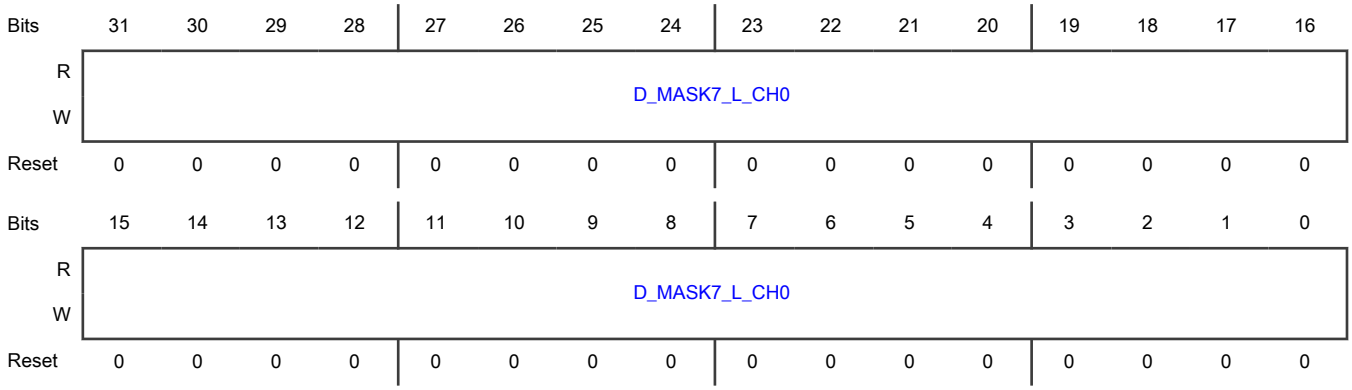
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_MASK7_L_CH0	7E0h

Diagram



Fields

Field	Description
31-0 D_MASK7_L_CH0	D_MASK7_L_CH0 data mask7 low byte

56.6.1.106 Input Store Data Shift Low Channel 0 Register (INPUT_STORE_D_SHIFT_L_CH0)

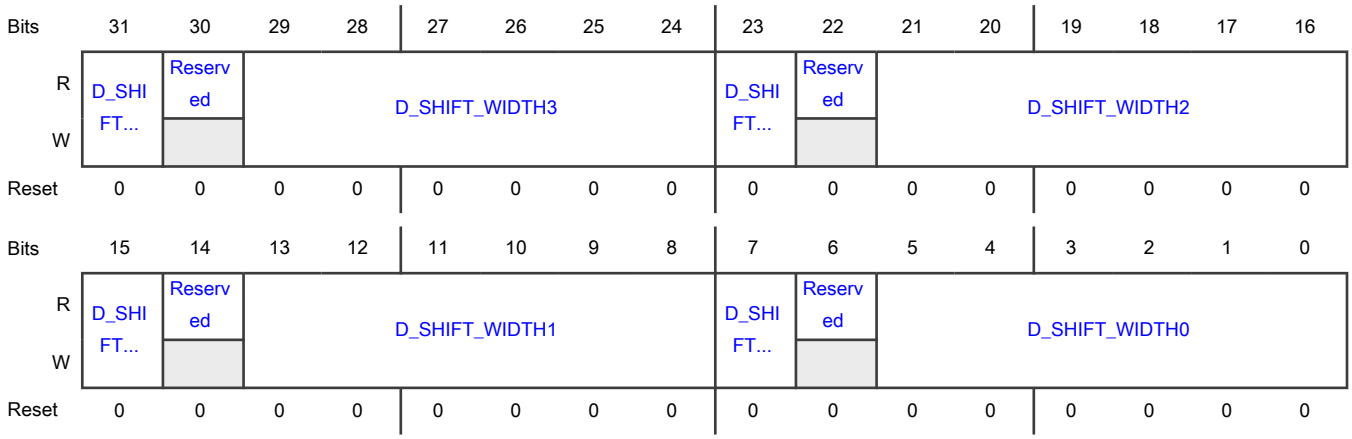
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_SHIFT_L_CH0	7F0h

Diagram



Fields

Field	Description
31 D_SHIFT_FLAG3	D_SHIFT_FLAG3 data shift flag 3
30 —	Reserved
29-24 D_SHIFT_WIDTH3	D_SHIFT_WIDTH3 data shift width 3
23 D_SHIFT_FLAG2	D_SHIFT_FLAG2 data shift flag 2
22 —	Reserved
21-16 D_SHIFT_WIDTH2	D_SHIFT_WIDTH2 data shift width 2
15 D_SHIFT_FLAG1	D_SHIFT_FLAG1 data shift flag 1
14 —	Reserved
13-8 D_SHIFT_WIDTH1	D_SHIFT_WIDTH1 data shift width 1
7 D_SHIFT_FLAG0	D_SHIFT_FLAG0 data shift flag 0
6 —	Reserved
5-0 D_SHIFT_WIDTH0	D_SHIFT_WIDTH0 data shift width 0

56.6.1.107 Input Store Data Shift High Channel 0 Register (INPUT_STORE_D_SHIFT_H_CH0)

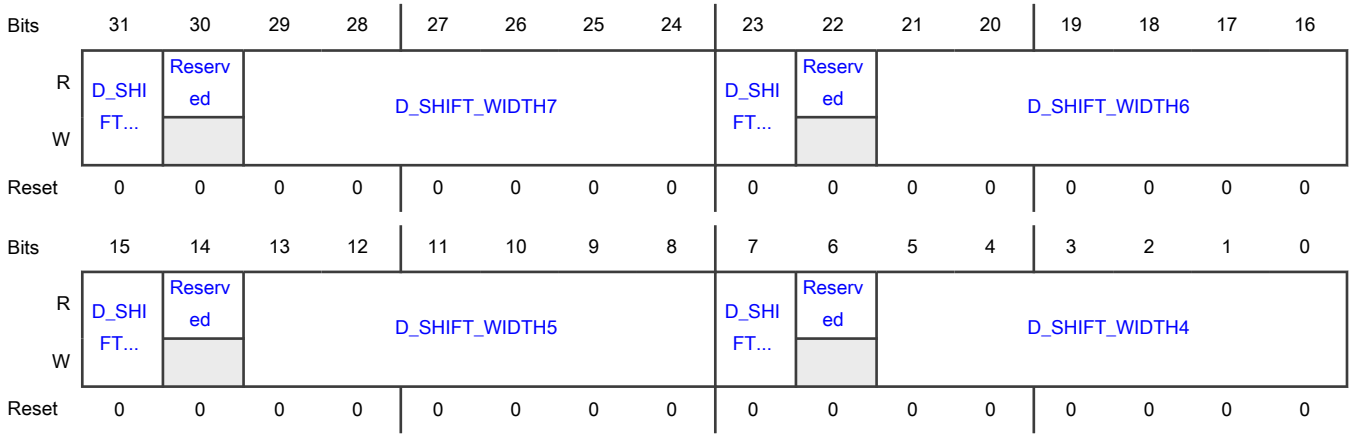
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_D_SHIFT_H_CH0	800h

Diagram



Fields

Field	Description
31 D_SHIFT_FLAG7	D_SHIFT_FLAG7 data shift flag 7
30 —	Reserved
29-24 D_SHIFT_WIDTH7	D_SHIFT_WIDTH7 data shift width 3
23 D_SHIFT_FLAG6	D_SHIFT_FLAG6 data shift flag 6
22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
21-16 D_SHIFT_WIDTH6	D_SHIFT_WIDTH6 data shift width 6
15 D_SHIFT_FLAG5	D_SHIFT_FLAG5 data shift flag 5
14 —	Reserved
13-8 D_SHIFT_WIDTH5	D_SHIFT_WIDTH5 data shift width 5
7 D_SHIFT_FLAG4	D_SHIFT_FLAG4 data shift flag 4
6 —	Reserved
5-0 D_SHIFT_WIDTH4	D_SHIFT_WIDTH4 data shift width 4

56.6.1.108 Input Store Flag Shift Low Channel 0 Register (INPUT_STORE_F_SHIFT_L_CH0)

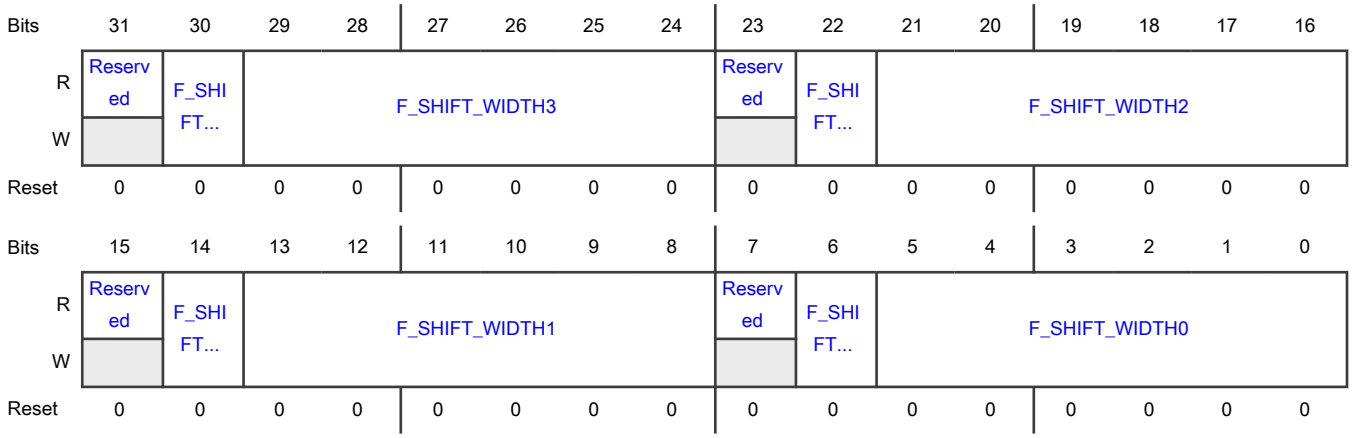
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_F_SHIFT_L_CH0	810h

Diagram



Fields

Field	Description
31 —	Reserved
30 F_SHIFT_FLAG3	F_SHIFT_FLAG3 flag shift flag3
29-24 F_SHIFT_WIDTH3	F_SHIFT_WIDTH3 flag shift width 3
23 —	Reserved
22 F_SHIFT_FLAG2	F_SHIFT_FLAG2 flag shift flag2
21-16 F_SHIFT_WIDTH2	F_SHIFT_WIDTH2 flag shift width 2
15 —	Reserved
14 F_SHIFT_FLAG1	F_SHIFT_FLAG1 flag shift flag1

Table continues on the next page...

Table continued from the previous page...

Field	Description
13-8 F_SHIFT_WIDTH1	F_SHIFT_WIDTH1 flag shift width 1
7 —	Reserved
6 F_SHIFT_FLAG0	F_SHIFT_FLAG0 flag shift flag0
5-0 F_SHIFT_WIDTH0	F_SHIFT_WIDTH0 flag shift width 0

56.6.1.109 Input Store Flag Shift High Channel 0 Register (INPUT_STORE_F_SHIFT_H_CH0)

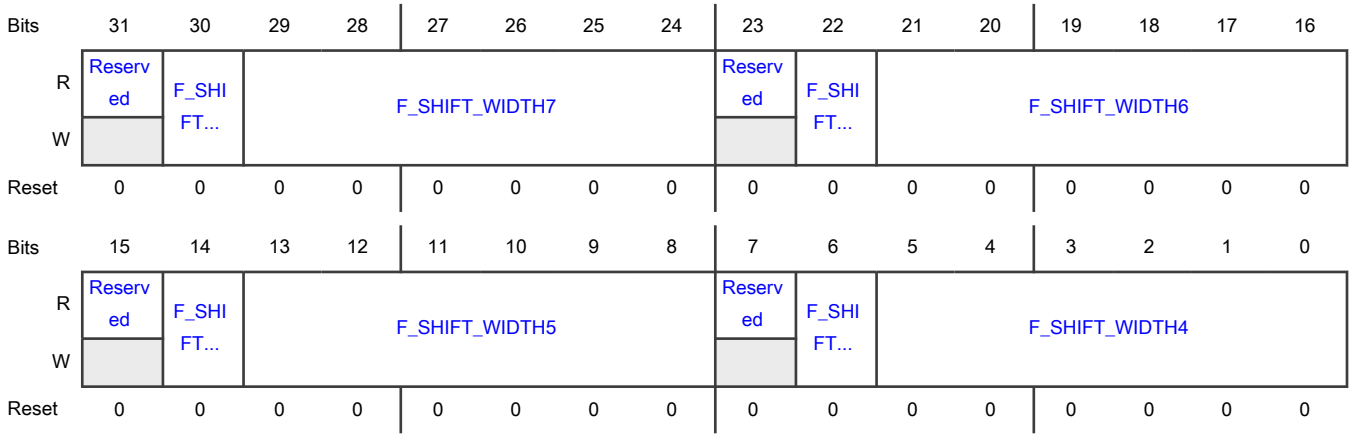
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_F_SHIFT_H_CH0	820h

Diagram



Fields

Field	Description
31 —	Reserved
30 F_SHIFT_FLAG7	F_SHIFT_FLAG7 flag shift flag7
29-24 F_SHIFT_WIDTH7	F_SHIFT_WIDTH7 flag shift width 7
23 —	Reserved
22 F_SHIFT_FLAG6	F_SHIFT_FLAG6 flag shift flag6
21-16 F_SHIFT_WIDTH6	F_SHIFT_WIDTH6 flag shift width 5
15 —	Reserved
14 F_SHIFT_FLAG5	F_SHIFT_FLAG5 flag shift flag5
13-8 F_SHIFT_WIDTH5	F_SHIFT_WIDTH5 flag shift width 5
7 —	Reserved
6 F_SHIFT_FLAG4	F_SHIFT_FLAG4 flag shift flag4
5-0 F_SHIFT_WIDTH4	F_SHIFT_WIDTH4 flag shift width 4

56.6.1.110 Input Store Flag Mask Low Channel 0 Register (INPUT_STORE_F_MASK_L_CH0)

This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_F_MASK_L_CH0	830h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	F_MASK3								F_MASK2							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	F_MASK1								F_MASK0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 F_MASK3	F_MASK3 flag mask3
23-16 F_MASK2	F_MASK2 flag mask2
15-8 F_MASK1	F_MASK1 flag mask1
7-0 F_MASK0	F_MASK0 flag mask0

56.6.1.111 Input Store Flag Mask High Channel 0 Register (INPUT_STORE_F_MASK_H_CH0)

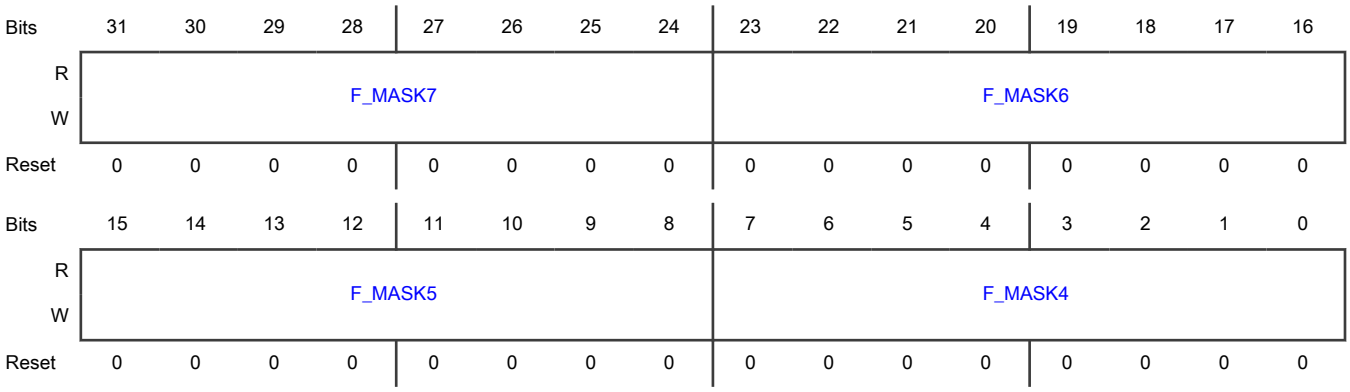
This register defines the control bits for the pxp store_engine sub-block.

The Control register contains the control bits for the pxp store_engine sub-block.

Offset

Register	Offset
INPUT_STORE_F_MASK_H_CH0	840h

Diagram



Fields

Field	Description
31-24 F_MASK7	F_MASK7 flag mask7
23-16 F_MASK6	F_MASK6 flag mask6
15-8 F_MASK5	F_MASK5 flag mask5
7-0 F_MASK4	F_MASK4 flag mask4

56.6.1.112 Bus MUX Register (BUS_MUX)

This register defines the pxp subblock bus mux on top level.

Offset

Register	Offset
BUS_MUX	2CE0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								WR_SEL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RD_SEL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23-16 WR_SEL	WR_SEL Subblock BUS to AXI MUX, setting 0 to axi0 and setting 1 to axi1
15-8 —	Reserved
7-0 RD_SEL	RD_SEL Subblock BUS to AXI MUX, setting 0 to axi0 and setting 1 to axi1

56.6.1.113 Handshake Ready MUX 0 Register (HANDSHAKE_READY_MUX0)

This register defines the pxp subblock handshake signals ready mux on top level.

Offset

Register	Offset
HANDSHAKE_READY_MUX0	2CF0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HSK7				HSK6				HSK5				HSK4			
Reset	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HSK3				HSK2				HSK1				HSK0			
Reset	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Fields

Field	Description
31-28 HSK7	HSK7 Subblock double buffer handshake signals MUX
27-24 HSK6	HSK6 Subblock double buffer handshake signals MUX
23-20 HSK5	HSK5 Subblock double buffer handshake signals MUX
19-16 HSK4	HSK4 Subblock double buffer handshake signals MUX
15-12 HSK3	HSK3 Subblock double buffer handshake signals MUX
11-8 HSK2	HSK2 Subblock double buffer handshake signals MUX
7-4 HSK1	HSK1 Subblock double buffer handshake signals MUX
3-0 HSK0	HSK0 Subblock double buffer handshake signals MUX 0: reserved 1: reserved 2: reserved 3: reserved 4: reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
	5: reserved 6: reserved 7: reserved 8: reserved 9: Ready signal source is from cpu_fetch_sw0_ready A: Ready signal source is from cpu_fetch_sw1_ready B: Ready signal source is from cpu_store_sw0_ready C: Ready signal source is from cpu_store_sw1_ready

56.6.1.114 Handshake Ready MUX 1 Register (HANDSHAKE_READY_MUX1)

This register defines the pxp subblock handshake signals ready mux on top level.

Offset

Register	Offset
HANDSHAKE_READY_MUX1	2D00h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HSK15				HSK14				HSK13				HSK12			
Reset	1	1	1	1	1	1	1	0	1	1	0	1	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HSK11				HSK10				HSK9				HSK8			
Reset	1	0	1	1	1	0	1	0	1	0	0	1	1	0	0	0

Fields

Field	Description
31-28 HSK15	HSK15 Subblock double buffer handshake signals MUX
27-24	HSK14

Table continues on the next page...

Table continued from the previous page...

Field	Description
HSK14	Subblock double buffer handshake signals MUX
23-20 HSK13	HSK13 Subblock double buffer handshake signals MUX
19-16 HSK12	HSK12 Subblock double buffer handshake signals MUX
15-12 HSK11	HSK11 Subblock double buffer handshake signals MUX
11-8 HSK10	HSK10 Subblock double buffer handshake signals MUX
7-4 HSK9	HSK9 Subblock double buffer handshake signals MUX
3-0 HSK8	HSK8 Subblock double buffer handshake signals MUX

56.6.1.115 Handshake Done MUX 0 Register (HANDSHAKE_DONE_MUX0)

This register defines the pxp subblock handshake signals done mux on top level.

Offset

Register	Offset
HANDSHAKE_DONE_MUX0	2D10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HSK7				HSK6				HSK5				HSK4			
Reset	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HSK3				HSK2				HSK1				HSK0			
Reset	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Fields

Field	Description
31-28 HSK7	HSK7 Subblock double buffer handshake signals MUX
27-24 HSK6	HSK6 Subblock double buffer handshake signals MUX
23-20 HSK5	HSK5 Subblock double buffer handshake signals MUX
19-16 HSK4	HSK4 Subblock double buffer handshake signals MUX
15-12 HSK3	HSK3 Subblock double buffer handshake signals MUX
11-8 HSK2	HSK2 Subblock double buffer handshake signals MUX
7-4 HSK1	HSK1 Subblock double buffer handshake signals MUX
3-0 HSK0	HSK0 Subblock double buffer handshake signals MUX 0: reserved 1: reserved 2: reserved 3: reserved 4: reserved 5: reserved 6: reserved 7: reserved 8: reserved 9: Done signal source is from cpu_fetch_sw0_done A: Done signal source is from cpu_fetch_sw1_done B: Done signal source is from cpu_store_sw0_done C: Done signal source is from cpu_store_sw1_done

56.6.1.116 Handshake Done MUX 1 Register (HANDSHAKE_DONE_MUX1)

This register defines the pxp subblock handshake signals done mux on top level.

Offset

Register	Offset
HANDSHAKE_DONE_MUX1	2D20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HSK15				HSK14				HSK13				HSK12			
Reset	1	1	1	1	1	1	1	0	1	1	0	1	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HSK11				HSK10				HSK9				HSK8			
Reset	1	0	1	1	1	0	1	0	1	0	0	1	1	0	0	0

Fields

Field	Description
31-28 HSK15	HSK15 Subblock double buffer handshake signals MUX
27-24 HSK14	HSK14 Subblock double buffer handshake signals MUX
23-20 HSK13	HSK13 Subblock double buffer handshake signals MUX
19-16 HSK12	HSK12 Subblock double buffer handshake signals MUX
15-12 HSK11	HSK11 Subblock double buffer handshake signals MUX
11-8 HSK10	HSK10 Subblock double buffer handshake signals MUX
7-4 HSK9	HSK9 Subblock double buffer handshake signals MUX
3-0 HSK8	HSK8 Subblock double buffer handshake signals MUX

56.6.1.117 Handshake CPU Fetch Register (HANDSHAKE_CPU_FETCH)

This register defines the pxp software handshake signals with CPU.

HANDSHAKE_CPU_FETCH: 0x2D30

HANDSHAKE_CPU_FETCH_SET: 0x2D34

HANDSHAKE_CPU_FETCH_CLR: 0x2D38

HANDSHAKE_CPU_FETCH_TOG: 0x2D3C

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
HANDSHAKE_CPU_FETCH	2D30h	Handshake CPU Fetch Register
HANDSHAKE_CPU_FETCH_SET	2D34h	Writing 1 to a bit in this register ensures that the corresponding bit in HANDSHAKE_CPU_FETCH is 1
HANDSHAKE_CPU_FETCH_CLR	2D38h	Writing 1 to a bit in this register ensures that the corresponding bit in HANDSHAKE_CPU_FETCH is 0
HANDSHAKE_CPU_FETCH_TOG	2D3Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in HANDSHAKE_CPU_FETCH

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW1_HSK...	Reserved									SW1_BUF_LINES		SW1_B1_...	SW1_B0_...	SW1_B1_...	SW1_B0_...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW0_HSK...	Reserved									SW0_BUF_LINES		SW0_B1_...	SW0_B0_...	SW0_B1_...	SW0_B0_...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Fields

Field	Description
31 SW1_HSK_EN	SW1_HSK_EN Enable software handshake 1 with CPU

Table continues on the next page...

Table continued from the previous page...

Field	Description
30-22 —	Reserved
21-20 SW1_BUF_LINES	SW1_BUF_LINES Buffer lines for software handshake 00 - LINE_4 : Buffer lines is 4 lines. 01 - LINE_8 : Buffer lines is 8 lines. 10 - LINE_16 : Buffer lines is 16 lines.
19 SW1_B1_DONE	SW1_B1_DONE CPU b1 buffer done to PXP
18 SW1_B0_DONE	SW1_B0_DONE CPU b0 buffer done to PXP
17 SW1_B1_READY	SW1_B1_READY PXP b1 buffer ready to CPU
16 SW1_B0_READY	SW1_B0_READY PXP b0 buffer ready to CPU
15 SW0_HSK_EN	SW0_HSK_EN Enable software handshake 0 with CPU
14-6 —	Reserved
5-4 SW0_BUF_LINES	SW0_BUF_LINES Buffer lines for software handshake 00 - LINE_4 : Buffer lines is 4 lines. 01 - LINE_8 : Buffer lines is 8 lines. 10 - LINE_16 : Buffer lines is 16 lines.
3 SW0_B1_DONE	SW0_B1_DONE CPU b1 buffer done to PXP
2	SW0_B0_DONE

Table continues on the next page...

Table continued from the previous page...

Field	Description
SW0_B0_DONE	CPU b0 buffer done to PXP
1 SW0_B1_READY	SW0_B1_READY PXP b1 buffer ready to CPU
0 SW0_B0_READY	SW0_B0_READY PXP b0 buffer ready to CPU

56.6.1.118 Handshake CPU Store Register (HANDSHAKE_CPU_STORE)

This register defines the pxp software handshake signals with CPU.

HANDSHAKE_CPU_STORE: 0x2D40

HANDSHAKE_CPU_STORE_SET: 0x2D44

HANDSHAKE_CPU_STORE_CLR: 0x2D48

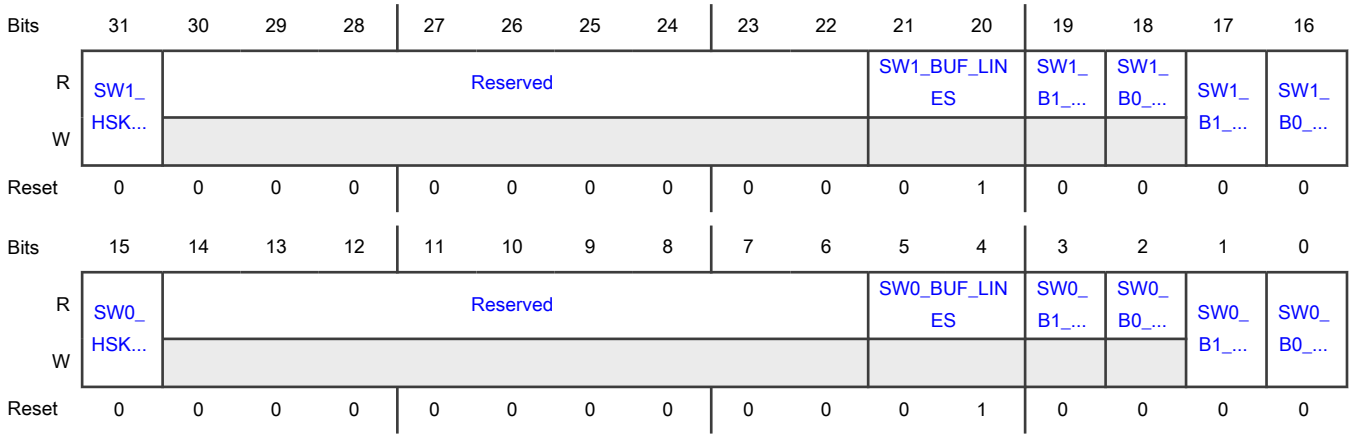
HANDSHAKE_CPU_STORE_TOG: 0x2D4C

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
HANDSHAKE_CPU_STORE	2D40h	Handshake CPU Store Register
HANDSHAKE_CPU_STORE_SET	2D44h	Writing 1 to a bit in this register ensures that the corresponding bit in HANDSHAKE_CPU_STORE is 1
HANDSHAKE_CPU_STORE_CLR	2D48h	Writing 1 to a bit in this register ensures that the corresponding bit in HANDSHAKE_CPU_STORE is 0
HANDSHAKE_CPU_STORE_TOG	2D4Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in HANDSHAKE_CPU_STORE

Diagram



Fields

Field	Description
31 SW1_HSK_EN	SW1_HSK_EN Enable software handshake 1 with CPU
30-22 —	Reserved
21-20 SW1_BUF_LINES	SW1_BUF_LINES Buffer lines for software handshake 00 - LINE_4 : Buffer lines is 4 lines. 01 - LINE_8 : Buffer lines is 8 lines. 10 - LINE_16 : Buffer lines is 16 lines.
19 SW1_B1_DONE	SW1_B1_DONE CPU b1 buffer done to PXP
18 SW1_B0_DONE	SW1_B0_DONE CPU b0 buffer done to PXP
17 SW1_B1_READY	SW1_B1_READY PXP b1 buffer ready to CPU
16 SW1_B0_READY	SW1_B0_READY PXP b0 buffer ready to CPU

Table continues on the next page...

Table continued from the previous page...

Field	Description
15 SW0_HSK_EN	SW0_HSK_EN Enable software handshake 0 with CPU
14-6 —	Reserved
5-4 SW0_BUF_LINES	SW0_BUF_LINES Buffer lines for software handshake 00 - LINE_4 : Buffer lines is 4 lines. 01 - LINE_8 : Buffer lines is 8 lines. 10 - LINE_16 : Buffer lines is 16 lines.
3 SW0_B1_DONE	SW0_B1_DONE CPU b1 buffer done to PXP
2 SW0_B0_DONE	SW0_B0_DONE CPU b0 buffer done to PXP
1 SW0_B1_READY	SW0_B1_READY PXP b1 buffer ready to CPU
0 SW0_B0_READY	SW0_B0_READY PXP b0 buffer ready to CPU

Chapter 57
Image Sensing Interface (ISI)

57.1 Chip-specific ISI information

Table 546. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The ISI module on the chip implements one channel and one camera input. Virtual channels are not supported.

This one camera input can be connected to either of:

- MIPI-CSI camera
- Parallel camera input

57.2 Overview

ISI interfaces to a pixel link source to obtain the image data for processing in its pipeline. The pipeline processes the image line from a configured source and performs one or more functions that you can configure, such as:

- Downscaling
- Color space conversion
- Deinterlacing
- Alpha insertion
- Cropping
- Flipping (horizontal and vertical)

The processed image is stored in programmable memory locations.

This module implements a limited flow control mechanism to control the output from its internal buffer flushing or sourcing an image from memory. Depending on the format type, the module can process and store one line of pixels from the incoming frame.

57.2.1 Block diagram

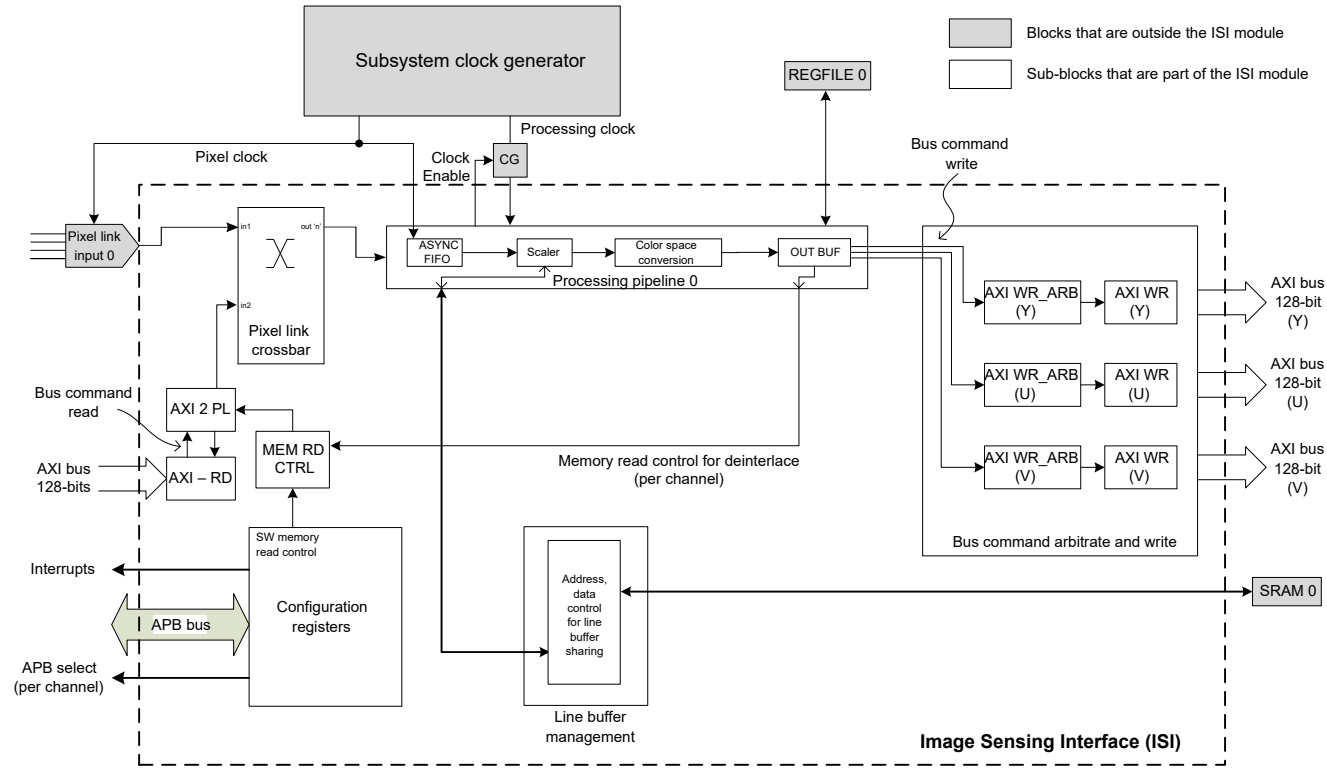


Figure 351. ISI block diagram

57.2.2 Features

- Up to 2K resolution at 30 or 60 fps (24 bpp)
- Input sources:
  - Supports 1 pixel link interface that can interface with 1 camera sensor
  - Supports system memory (AXI manager, internally converted to pixel link interface)
- Ability to assign each processing pipeline or channel to the same or different pixel input source
- Stream multiplexing:
  - Supports simple deinterlacing methods for interlaced input sources:
    - Weaving
- Stream manipulation features
  - Supported pixel formats when storing the image in memory:
    - RAW8, RAW10, RAW12, RAW14, RAW16, and RAW32
    - RGB888, BGR888, RGB565, RGB 10-bit, and BGR 10-bit
    - YUV444, YUV422, YUV420 (8-bit, 10-bit, 12-bit) in planar or semi-planar formats

NOTE

If the input format is YUV420, 10-bit and 12-bit outputs are extended with bits [7:6] or [7:4] based on the highest 8 bits of the input. 2 or 4 bits are lost due to the line buffer (24-bit width for three components of RGB or YUV).

See CHNL_IMG_CTRL[FORMAT] for more formats.

  - Downscaling of input image via decimation and bilinear filtering:
    - Supports decimation by 2, 4, or 8
    - Bilinear filter further downscales by 1.0 to 2.0 (fractional downscaling)
  - Color space conversion (CSC):
    - RGB, YUV, and YCbCr
    - User defined color space matrix-based conversion
  - Alpha channel insertion for RGB formats:
    - Global alpha value
    - Separate rectangular region of interest (ROI) alpha value that has a higher priority than the global alpha value
    - Supports up to four ROI non-overlapping rectangles
  - Mirroring (image flip): horizontal and vertical flips supported
  - Frame awareness and frame skipping:
    - Clean frame start and shutdown based on HSYNC and VSYNC
    - Buffer overrun protection
    - Buffer under run deterministic behavior
- Stream output options
  - Output to memory:
    - Input source is converted to and processed by the processing pipeline as YUV444 or RGB
    - Processed images are taken as an output from the pipeline and stored in a memory location that you specify
    - Full-line storage available at processing channel output before sending data to AXI (this storage can be automatically split or combined depending on the output format being used)
    - Dual buffered addresses used in a ping-pong fashion with an active buffer status indication
    - Line and frame stored interrupt status to software to track the progress of the frame
- Flow control:
  - Provides a panic indication to the chip to increase the priority of its write transactions to avoid potential overflow in output buffers (you can configure thresholds for panic indication)
  - When pixels are sourced from memory or input line buffers are flushed, you can program the rate at which pixels are sent out (by default, one pixel per clock is sent out)
  - Supports back pressure mechanism to stall the channel pipeline during line buffer flushing and sourcing of images from memory, when the AXI bus is unable to accept data, and the output buffers are low on storage
- Metadata processing (the embedded data from the sensor can be processed and written out)

57.3 Functional description

57.3.1 Design overview

This section describes ISI's design architecture and data flow with details of each block.

57.3.1.1 Pixel link crossbar

ISI comprises 1 processing pipeline that you can assign to any of the input image sources via the pixel link crossbar, which is a 2-input (including four pixel link inputs and one input read from memory) and 1-output multiplexer that you can use to configure

each output port to connect to any of the 2 inputs. These 2 inputs are numbered from Pixel link input 1 to 1, and input 2 that is always a memory location.

The image input to the module is through the pixel line subordinate port that provides the following:

- Clock
- Data and control buses
- HSYNC and VSYNC signals

The crossbar connects these subordinate port signals to the processing channel that pull the pixel data out from the asynchronous FIFO in the pixel line subordinate port, the only exception being when the image is read from memory. In this case, ISI converts the lines read from memory into pixel link subordinate interface signals so that the signals seen at the crossbar and processing channels are the same as other sources.

57.3.1.2 Processing pipeline

A processing channel comprises:

- A downscaling engine
- A color space conversion (CSC) block
- An output buffer control block

The scaler performs decimation and bilinear filtering of the input image, and the output buffer control performs various functions such as alpha insertion, cropping, and horizontal or vertical flipping. It generates the AXI write transactions for storing the processed image in memory and supports storing the image as 1/2/3 planar at different memory locations.

57.3.1.3 Output buffer control and AXI output

The output buffer control block of each processing channel performs:

- Image cropping
- Flipping (horizontal and vertical)
- Alpha value insertion

This block collects the processed image pixels in a ping-pong buffer and sends out bursts of 128 bytes or 256 bytes (configurable) of data over the AXI interface to memory. This block also performs the function of deinterlacing the frames while storing them in memory. You can format the final processed image pixels to store them in any of the multiple image formats as mentioned in CHNL_IMG_CTRL[FORMAT].

This block also manages the concatenation of ping-pong buffers per component based on the output image format. The combined storage of the three ping-pong buffers is up to a full line of the image frame depending on the output image format. By using flow control and panic mechanisms, you can ensure that the data is sent out at a rate acceptable to the system and that output buffers do not overflow.

57.3.1.4 AXI memory read

ISI can read an image from memory in different formats as discussed in Channel 0 Memory Read Control (CHNL_MEM_RD_CTRL0). ISI reads the image via AXI read and internally drives the image through a pixel to the processing channel. This operation is also flow-controlled. You can configure the rate at which pixels are processed in the channel and are sent out. By default, one pixel per clock is processed when the read FIFO is not empty.

57.3.1.5 Configuration registers

Each channel has the same set of configuration registers that are aligned to a 64 KB boundary and are programmable via the APB interface. You must ensure that the correct values are being programmed. ISI does not check for the correctness of the values.

57.3.2 Pixel link data formats

A number of pixel link formats are standardized, which consist of different kinds of data, identifying information, and user control bits (commonly the DATA ENABLE bit is included). When HSYNC and VSYNC are asserted (that is, HSYNC and VSYNC are active or equal to zero), the core does not care the entire data field. The control field is more complex where clearly HSYNC and VSYNC are valid. This is intended to cover all of the data types transferred across pixel link.

Control field

Each pixel has an associated control field identifying its data. See chip specific ISI information for more details on control field bits.

NOTE

On the interface to the manager and subordinate modules, HSYNC and VSYNC are separated, so the relevant pixel input bit is different from the control field number (for the actual link).

57.3.3 Scaling operation

The scaling engine works in two parts: it performs pre-decimation of the image followed by bilinear filtering to achieve the desired downscaling factor.

The decimation filter provides a maximum downscaling factor of 8, and the subsequent bilinear filter provides a maximum downscaling factor of 2. Combined, the maximum scaling factor can be up to 16. You can program the decimation and bilinear scaling engines independently. You also program an initial offset to allow more source data to be considered in the bilinear scaling engine. The scaled output image is presented to the CSC module as YUV444 or RGB888 pixels with a single byte for each color channel.

57.3.4 CSC

CSC is a full-duplex color space converter that can convert an image to either RGB or YUV (or YC_bC_r) color spaces depending on the desired output pixel format. You can program all coefficients as two's-complement numbers. If you do not require CSC in the application, then you can bypass the CSC unit individually on a channel. The CSC output is sent to the output buffer control block for further processing and storage in memory.

The following equations indicate the ALU architecture of the CSC module:

(a) RGB to YUV (or YC_bC_r) conversion

- Y = (A1 × R) + (A2 × G) + (A3 × B) + D1
- U = (B1 × R) + (B2 × G) + (B3 × B) + D2
- V = (C1 × R) + (C2 × G) + (C3 × B) + D3

(b) YUV (or YC_bC_r) to RGB conversion

- R = (A1 × (Y + D1)) + (A2 × (U + D2)) + (A3 × (V + D3))
- G = (B1 × (Y + D1)) + (B2 × (U + D2)) + (B3 × (V + D3))
- B = (C1 × (Y + D1)) + (C2 × (U + D2)) + (C3 × (V + D3))

where:

- A1–A3, B1–B3, C1–C3, and D1–D3 are programmable coefficients.
- YUV and RGB refer to pixel values.

You check the saturation of each color channel and correct it for excursions outside the nominal color space. Overflow for the three channels is saturated at 255h and underflow is saturated at 00h.

57.3.5 Image deinterlacing methods

ISI supports image deinterlacing for images that are interlaced across frames. You can use one of the following deinterlace methods to form the deinterlaced image.

57.3.5.1 Weaving

Weaving is performed by combining consecutive fields to form a new frame. It results in artifacts known as combing. The pixels in one frame do not line up with the pixels in the other and form a jagged edge. To perform weaving in an ISI channel, one field's lines are stored in memory with a double line pitch. It leaves one-line space in memory for the second field's lines to be stored. The two interlaced frames are woven while being stored in memory. You can select which frame to have on the top (odd or even), (see Figure 352).

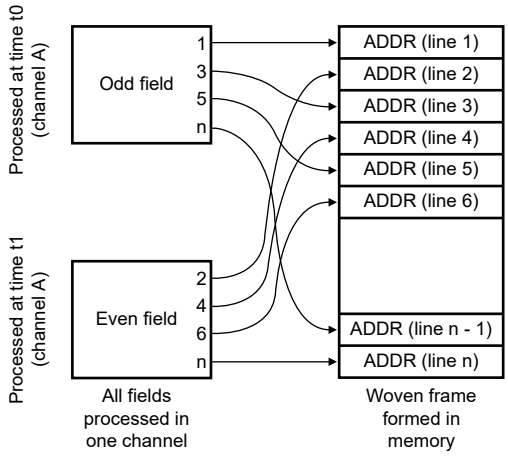


Figure 352. Weave deinterlacing

57.3.6 Output buffer management

The ISI output buffer management provides the following image processing options, and the operations related to them are carried out in this order:

1. Alpha insertion in RGB formats
2. Image cropping
3. Image flipping

You must perform these image manipulation operations before the image is stored in output buffers. The image input to the output buffer block (from scaler or CSC) is in either RGB or YUV444 format. The output buffer logic then formats each pixel from these two formats to the desired output format as specified in CHNL_IMG_CTRL[FORMAT]. The output buffers consolidate the formatted image pixel bytes and send out 128-byte or 256-byte AXI write bursts to store the image in memory.

ISI has an in-built pre-overflow warning mechanism through which it can warn the subsystem and request higher AXI write channel priority to avoid any overflows in its buffers. In any case, if an overflow occurs, ISI stops storing pixels for that line and starts storing pixels from the next line. This causes the frames to be corrupted. You must take appropriate action for such frames. For images sourced from memory, ISI can stall its internal pipeline or control the rate at which pixels are processed in the pipeline. See Flow control mechanisms for more details.

There are two destination addresses that you must program:

- CHNL_OUT_BUF1_ADDR_*n*
  - Channel 0 RGB or Luma (Y) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_Y0)
  - Channel 0 Chroma (U/Cb/UV/CbCr) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_U0)
  - Channel 0 Chroma (V/Cr) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_V0)
- CHNL_OUT_BUF2_ADDR_*n*
  - Channel 0 RGB or Luma (Y) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_Y0)

- Channel 0 Chroma (U/Cb/UV/CbCr) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_U0)
- Channel 0 Chroma (V/Cr) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_V0)

The BUF1 address is used when you enable the channel for the first time. This address increments internally by a line_pitch (Channel 0 Output Buffer Pitch (CHNL_OUT_BUF_PITCH0)) at the end of each line, except the last line, to compute the address for each line storage. ISI generates a line_stored interrupt (CHNL_STS[LINE_STRD]) when a full line is completely stored and a frame_stored interrupt (CHNL_STS[FRM_STRD]) after the last line is completely stored in memory. The next frame is stored at the BUF2 address in the same way. You can update CHNL_OUT_BUF1_ADDR_*n* at this point if desired. On every frame_stored interrupt, ISI toggles between the two buffer addresses. Status fields indicate the buffer that is currently active. You can update the buffer address registers at any time, but they take effect only when the corresponding load address fields are toggled. The new address gets loaded when the current frame is complete.

ISI automatically manages the output buffer size depending on the output image format. For 2- and 3-planar formats, the output buffer is divided accordingly among each component based on the number of bytes that the component uses. For 1-planar, RGB, and RAW, the output buffer functions as a unified buffer. This allows efficient utilization of output buffers even when they are concatenated.

57.3.6.1 Alpha insertion in RGB formats

ISI supports global alpha value insertion to RGB images in which one alpha value is inserted into all pixels in the image. ISI supports up to four non-overlapping regions of interest (ROI) in which you can insert different alpha values. If both ROI alpha and global alpha values are enabled, ROI alpha values take priority over the global alpha value. Pixels that do not lie in any of the ROI can have a global alpha value (if enabled) or do not have any alpha value inserted (if global alpha insertion is disabled). You can configure the upper-left coordinate (ULC) and the lower-right coordinate (LRC) individually for each ROI. ULC and LRC must lie within the dimensions that Channel 0 Scaled Image Configuration (CHNL_SCL_IMG_CFG0) specifies.

57.3.6.2 Image cropping

ISI supports cropping in both X and Y dimensions based on the programmed ULC and LRC, which must lie within the dimensions that Channel 0 Scaled Image Configuration (CHNL_SCL_IMG_CFG0) specifies.

57.3.6.3 Image flipping

ISI supports horizontal and vertical image flipping. You can enable either one or both of these.

57.3.7 Sourcing image from memory

This section discusses the configuration required for software-controlled memory reads. Perform the following steps in any order to configure the channel for software-controlled memory reads (you must initialize the channel before configuring the memory image read on a channel):

- Configure the memory address from where the image is to be sourced in .
- Configure the line and frame pitch in Channel 0 Input Buffer Pitch (CHNL_IN_BUF_PITCH0).
- Configure the read image format in CHNL_MEM_RD_CTRL[IMG_TYPE].
- Complete the initialization of other blocks and interrupts for channel 0 and write 1 to CHNL_CTRL[CHNL_EN].
- Write 1 to CHNL_MEM_RD_CTRL[READ_MEM] to enable reading of the frame from memory.
- Wait for CHNL_STS[MEM_RD_DONE] to become 1 before initiating a new image read. When this interrupt occurs, write 0 to CHNL_MEM_RD_CTRL[READ_MEM] followed by writing 1 to it to start the next read.

57.3.8 Flow control mechanisms

ISI implements the following flow control mechanisms to compensate for low bandwidth on the chip's AXI bus:

57.3.8.1 Panic and overflow

ISI provides a "panic" signal to the subsystem to request for a higher priority on the AXI write channel to empty the ping-pong buffers before they overflow. You can configure the assertion of a panic signal when the output ping-pong buffers are filled to a certain configurable threshold.

Depending on the reaction time from the subsystem, you can configure the appropriate threshold in CHNL_OUT_BUF_CTRL[PANIC_SET_THD_n], where *n* = Y, U, V. The panic signal deasserts when the amount of data inside the ping-pong buffers goes below the configurable threshold. CHNL_STS[PANIC_n_BUF] indicates the panic status, where *n* = Y, U, V. The status fields do not clear if the amount of data inside the ping-pong buffers goes below the configurable threshold.

In case an overflow occurs, ISI stops storing pixels from the current line. ISI then resyncs to the next line and starts storing pixels thereon. CHNL_STS[OFLW_n_BUF] indicates the status of overflow in buffers, where *n* = Y, U, V. You can enable overflow and panic interrupts for the channel in Channel 0 Interrupt Enable (CHNL_IER0).

57.3.8.2 Back pressure to stall internal pipeline

ISI implements an automatic back pressure to stall the channel's pipeline in case the AXI bus is not ready to accept data and the ping-pong buffers have 128 bytes of space left. The back pressure to the channel pipeline is asserted when the scaler line buffers are flushed between frames, or the image is sourced from memory.

However, data can still be lost during scaler line buffer flushing if the back pressure stalls the pipeline for too long. This could happen when a new frame starts, and the flush is not completed because of back pressure. In this case, ISI moves to process the next frame and indicates an overflow error.

57.3.8.3 Adjustable flow rate

ISI provides registers to configure the rate at which pixels are processed in the pipeline when the image is sourced from memory, the scaler line buffers are flushed, or both. By default, the rate is one pixel per clock, assuming read buffers are full with data. Normally, the camera rate is less than 1 pixel per clock and when flushing, the high rate can peak the bandwidth demand on the AXI bus.

You can program the processing rate to match the camera rate as desired, by using Channel 0 Flow Control (CHNL_FLOW_CTRL0). Program the processing rate as a ratio in CHNL_FLOW_CTRL[FC_NUMER] and CHNL_FLOW_CTRL[FC_DENOM]. ISI sends pixels for the FC_NUMER number of clocks out of FC_DENOM clocks.

Return to Initialization and channel configuration steps to complete the initialization of other blocks and enable the channel.

57.3.9 Error resiliency

ISI implements resiliency against possible error frames and recovery from them. An error frame can have one of the following conditions:

- HSYNC at the wrong time; thereby causing fewer or more pixels in the frame. ISI recovers silently and there is no indication of this error event.
- VSYNC at the wrong time; thereby causing fewer or more lines in the frame. ISI writes 1 to CHNL_STS[EARLY_VSYNC_ERR] or CHNL_STS[LATE_VSYNC_ERR] to indicate you of this condition. Next, ISI resyncs to the next frame. You must enable these interrupts by writing 1 to the corresponding interrupt enable field in Channel 0 Interrupt Enable (CHNL_IER0).
- Unequal lines between interlaced fields; ISI expects both frames to have an equal number of lines. ISI asserts CHNL_STS[EARLY_VSYNC_ERR] for the interlaced frame that has fewer lines than the other frame.
- A missing or an extra interlaced field (odd or even); ISI masks out the unexpected field and does not flag any error in this event.
- Frame with no pixels; ISI recovers silently and there is no indication of this error event.
- Frame line with no pixels; ISI recovers silently and there is no indication of this error event.

All error resiliency mechanisms ensure that the frame is not stored outside its given memory range in case ISI receives an error frame.

57.3.10 Clocking

The processing clock must be greater than or equal to the incoming pixel link clock for ISI. The reason for this is that ISI can process one pixel per processing clock while the incoming pixel link can input one pixel per pixel link clock. If the incoming pixel link is faster than the processing clock, ISI can lose data because the speed of incoming data is faster than the speed at which ISI can process it.

Table 547. Clock domains

Clock domain	Description
Pixel clock	Incoming pixel link clock from the camera via pixel link interface
APB clock	Clock used for register accesses
AXI clock	Clock on the AXI bus at the output

57.3.11 Reset

ISI has software reset control for each processing channel. For more information, see CHNL_CTRL[SW_RST].

57.3.12 Interrupts

Each processing channel has an interrupt output. The interrupt is enabled by Channel 0 Interrupt Enable (CHNL_IER0). Interrupt source comes from Channel 0 Status (CHNL_STS0).

57.4 External signals

This module has no external signals.

57.5 Initialization and channel configuration

This section describes the suggested steps to configure individual channels in ISI and handle interrupts for proper and efficient image processing. For the correct operation of the module, you must follow these initialization steps for every channel that needs to be configured.

First, perform the general channel configuration as follows:

1. Write 1 to CNHL_CTRL[CLK_EN] to enable the clock to the channel. This is required to access other registers in the channel and also for image processing by the channel.
2. Write 1 to CHNL_CTRL[CHNL_BYPASS] to enable channel bypass, else write 0 to it. Enable channel bypass if no scaling and no color conversion are required. If the channel is bypassed, the image coming into the channel is directly stored in the memory without any kind of processing. However, you can perform image flipping (horizontal, vertical, or both), image cropping, alpha insertion (RGB format only), and format conversion (YUV444 to YUV422 or YUV420) while channel bypass is enabled. Disabling the CSC and scaler has no impact on these operations.
3. Select the source of input image. The input is Pixel 0.. You can select an additional pixel link (pixel link 1) as a source. Pixel link 1 always reads an image from a programmed location in memory. Configure the source in CHNL_CTRL[SRC].

   NOTE

   - See the chip-specific ISI information for actual source type connected to pixel link 0.

4. Select the type of source for the channel between:
   - Non-memory pixel links (image sourced from camera sensor, CHNL_CTRL[SRC_TYPE] = 0)
   - Memory pixel link (image sourced from memory, CHNL_CTRL[SRC_TYPE] = 1).

   The memory pixel link refers to pixel link 1 that sources the image from a programmed memory location only. This is useful for deinterlacing and processing the image directly from memory. For the channels that do not support sourcing image

from memory, the source type is always a non-memory pixel link. You must configure the source type for the channel in CHNL_CTRL[SRC_TYPE].

NOTE

You can find the actual source type connected to pixel links 0 and 1 in the chip configuration.

5. Configure the image width and height by using CHNL_IMG_CFG[WIDTH] and CHNL_IMG_CFG[HEIGHT]. For example, a 1920x1080 image has width = 0x780 and height = 0x438.
6. Configure the memory address where the processed or bypassed data from the channel has to be stored. The image can be stored as:
   - 1-planar (all color components stored together).
   - 2-planar (Y component stored in one memory address and UV stored in a different memory address).
   - 3-planar (each component stored in a different memory address).

   Configure the 32-bit address as follows:
   - For 1-planar addressing, use Channel 0 RGB or Luma (Y) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_Y0) and Channel 0 RGB or Luma (Y) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_Y0).
   - For 2-planar and 3-planar addressing, use Channel 0 Chroma (U/Cb/UV/CbCr) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_U0) and Channel 0 Chroma (U/Cb/UV/CbCr) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_U0) additionally.
   - For 3-planar addressing, use Channel 0 Chroma (V/Cr) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_V0) and Channel 0 Chroma (V/Cr) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_V0) additionally.

   See Output buffer management for more information on how the two buffer addresses are used.

   NOTE

   RGB and RAW images are always stored as 1-planar while other YUV formats can be stored as 1, 2, or 3-planar.
7. Configure the output buffer line pitch in CHNL_OUT_BUF_PITCH[LINE_PITCH]. The line pitch is defined as the number of bytes between the start location of two lines. It is mainly used to have line addresses start at an aligned boundary.

   NOTE

   - The line pitch is the same for all image components irrespective of whether they are being stored in 1-planar, 2-planar, or 3-planar. In 2- and 3-planar outputs, the U and V components have the same line pitch as the Y component.
   - You must program the line pitch to be no less than the number of bytes in one line.
8. Toggle the address load fields, that is, CHNL_OUT_BUF_CTRL[LOAD_BUF1_ADDR] and CHNL_OUT_BUF_CTRL[LOAD_BUF2_ADDR] to allow new configured addresses to be picked and used in the channel, when the output buffer is enabled later.

See the following sections for configuration information on various functionalities:

- Scaling operation for information about the scaler and configuring the scaler for use.
- CSC for information about CSC and configuring the block for use.
- Image deinterlacing methods section for information about different deinterlacing methods and configuration for enabling each deinterlacing method for the channels.
- Sourcing image from memory for information on how to configure channel 0 to read and process an image that is sourced from memory.
- Output buffer management for information on how to configure controls for image cropping, flipping, alpha insertion, handle buffer overflows, and request higher AXI channel priority for emptying the buffers.
- Flow control mechanisms for information on how to configure the rate of pixel output from this module onto the AXI bus.

After completing the configuration provided in the aforementioned sections, you can now enable and monitor the channel using the following steps:

1. Write 1 to appropriate interrupt enable fields in Channel 0 Interrupt Enable (CHNL_IER0) to enable the necessary interrupts.
2. Write 1 to CHNL_CTRL[CHNL_EN] to enable the channel. This enables the channel to start accepting images from the selected source.
3. Monitor the status of the channel through Channel 0 Status (CHNL_STS0). Each field provides you with a status or error indication to take appropriate action. Interrupt status fields become 0 by writing 1 to the respective fields.
4. Write 1 to CHNL_CTRL[SW_RST], when required, to reset the complete channel to its default reset state. You must retain the value of this field as 1 for at least 5 to 6 NOP cycles to allow the reset to propagate. Then, write 0 to it to complete the reset cycle. After the software reset, the channel returns to the same default state as it would have been after a hardware reset. No configuration register resets during the software reset. You can additionally write 0 to CHNL_CTRL[CHNL_EN] while the software reset is asserted to control the restarting of the channel after the software reset is complete.

When configuring multiple channels, you must perform the aforementioned steps for all channels.

57.6 Application information

57.6.1 Configuring the scaler

Channel initialization is a prerequisite to configuring the scaler. Perform the following steps in any order to configure the scaler:

1. Configure the scale offset values for the X and Y dimensions in Channel 0 Scale Offset (CHNL_SCALE_OFFSET0). This is the initial offset from the (0,0) coordinate, using which the first output scaled pixel is computed.
2. Configure the scale factor values for the X and Y dimensions in Channel 0 Scale Factor (CHNL_SCALE_FACTOR0). This is the step factor through which the image is sampled for downscaling.
3. Configure the decimation factor values for the X and Y dimensions in CHNL_IMG_CTRL[DEC_X] and CHNL_IMG_CTRL[DEC_Y]. The final scaling factor that the scaler uses is = decimation factor × scale factor.
4. Write 1 to CHNL_IMG_CTRL[YCBCR_MODE] if the incoming or converted image is in the YC_bC_r format.
5. Configure Channel 0 Scaled Image Configuration (CHNL_SCL_IMG_CFG0) according to the dimension of the image after scaling. You must configure this register whenever the scaler is enabled, even with 1:1 scaling.

Return to Initialization and channel configuration to complete the initialization of other blocks and enable the channel.

Example: If the image needs to be downscaled by a factor of 4.5 in the X and Y dimension, you can program the scaler by:

- Writing 10b to CHNL_IMG_CTRL[DEC_X] and CHNL_IMG_CTRL[DEC_Y] indicating "decimate by 4."
- Writing 1200h to CHNL_SCALE_FACTOR[X_SCALE] and CHNL_SCALE_FACTOR[Y_SCALE] indicating a scaling factor of 1.125.

The overall downscaling is 4 x 1.125 = 4.5.

57.6.2 Configuring CSC

Channel initialization is a prerequisite to configuring CSC. Perform the following steps in any order to configure CSC:

- Write 1 to CHNL_IMG_CTRL[CSC_BYP] to bypass the CSC block, if you do not require to run CSC and the channel is not bypassed.
- Configure the coefficients in Channel 0 Color Space Conversion Coefficient 0 (CHNL_CSC_COEFF00) to Channel 0 Color Space Conversion Coefficient 5 (CHNL_CSC_COEFF50), if CSC is required.
- Write 1 to CHNL_IMG_CTRL[YCBCR_MODE] if the incoming image or the image after color space conversion is in YC_bC_r mode.
- Configure CHNL_IMG_CTRL[CSC_MODE] according to the required conversion method.

Return to Initialization and channel configuration to complete the initialization of other blocks and enable the channel.

57.6.3 Configuring deinterlacing

Channel initialization is a prerequisite to configuring the deinterlace block. You can configure the deinterlace block using any one of the following deinterlace methods:

- **Weaving -** Write 10b or 11b to CHNL_IMG_CTRL[DEINT] to configure the weaving deinterlace method to enable weave deinterlacing with the odd frame first or even frame first, respectively.

Return to Initialization and channel configuration to complete the initialization of other blocks and enable the channel.

NOTE

- Channels for which you configure CHNL_IMG_CTRL[DEINT] with values other than 00xb, the scaler and CSC are automatically bypassed.

57.6.4 Configuring additional image processing options

You must initialize the channel before configuring additional image processing controls. To configure the image processing options, perform alpha insertion, image cropping, and image flipping in an order:

1. Configure CHNL_IMG_CTRL[GBL_ALPHA_VAL] to program the global alpha value. Write 1 to CHNL_IMG_CTRL[GBL_ALPHA_EN] if a global alpha value is to be inserted in the image. You must configure this field when both the global and regional alpha values are to be inserted.
2. Write 1 to CHNL_ROI_n_ALPHA[ALPHA] and CHNL_ROI_n_ALPHA[ALPHA_EN] (where *n* = 0, 1, 2, or 3) to have region-specific alpha values.
3. Configure the ROI coordinates in CHNL_ROI_n_ULC and CHNL_ROI_n_LRC (where *n* = 0, 1, 2, 3). The regions should neither overlap nor lie outside the scaled image dimensions.
4. Configure the cropped region of the image in Channel 0 Crop Upper Left Corner Coordinate (CHNL_CROP_ULC0) and Channel 0 Crop Lower Right Corner Coordinate (CHNL_CROP_LRC0). This is the region that is stored in memory.
5. Enable cropping by writing 1 to CHNL_IMG_CTRL[CROP_EN].
6. Write 1 to CHNL_IMG_CTRL[HFLIP_EN] to configure horizontal flipping of the output image.
7. Write 1 to CHNL_IMG_CTRL[VFLIP_EN] to configure vertical flipping of the output image.

NOTE

You must configure ULC and LRC (both for alpha insertion and cropping) based on the image resolution output from the scaler and not on the original image resolution. The output buffer block works on the output from CSC (when no bypass happens) or scaler (if CSC is bypassed). The original image resolution is used for determining ULC and LRC only during a channel bypass or 1:1 scaling.

57.6.5 Configuring output buffer controls

Perform the following steps in any order, involving output memory addressing, to configure the output buffer controls (you must perform channel initialization before configuring these controls):

- Configure the destination memory address in CHNL_OUT_BUF1_ADDR_*n* and CHNL_OUT_BUF2_ADDR_*n* (where *n* = Y, U, V).
- Configure the output image line pitch in CHNL_OUT_BUF_PITCH[LINE_PITCH]. The line pitch is the spacing between two lines in terms of the number of bytes.
- Toggle the load address fields, that is, CHNL_OUT_BUF_CTRL[LOAD_BUF1_ADDR] and CHNL_OUT_BUF_CTRL[LOAD_BUF2_ADDR] to allow new configured addresses to be picked and used in the channel.
- Update the appropriate buffer 1 or buffer 2 address on every CHNL_STS[FRM_STRD] interrupt assertion. If this is not done, the same address is used, and the frame is overwritten.
- Write 1 to the appropriate address load fields on every update to the CHNL_OUT_BUF1_ADDR_*n* or CHNL_OUT_BUF2_ADDR_*n* (where *n* = Y, U, V) register. Without configuring the load field, the new address is not picked.

The status fields CHNL_STS[BUF1_ACTIVE] and CHNL_STS[BUF2_ACTIVE] indicate the active buffer to which the current frame is stored.

NOTE

— When performing deinterlacing, the output memory addressing changes for that channel. See Image deinterlacing methods for configuring the destination memory addresses for that channel.

— You must program the line_pitch to be greater than the number of bytes in one line.

Return to Initialization and channel configuration to complete the initialization of other blocks and enable the channel.

57.7 ISI register descriptions

The memory map comprises 32-bit aligned registers that you can access via 32-bit accesses only. Write accesses to reserved locations generate a transfer error. Read accesses to reserved locations also generate a transfer error and the read data bus shows all 0s.

NOTE

- Ensure that the correct values are programmed for a smooth operation. The module does not check for correctness.
- The ISI registers are readable but cannot be configured or written successfully when CHNL_CTRL[CLK_EN] = 0.
- The following registers are dedicated and are spaced apart by 64 KB boundary.

57.7.1 ISI memory map

ISI base address: 4AE4_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Channel 0 Control (CHNL_CTRL0)	32	RW	0000_0000
4	Channel 0 Image Control (CHNL_IMG_CTRL0)	32	RW	0000_0000
8	Channel 0 Output Buffer Control (CHNL_OUT_BUF_CTRL0)	32	RW	0000_0000
C	Channel 0 Image Configuration (CHNL_IMG_CFG0)	32	RW	0438_0780
10	Channel 0 Interrupt Enable (CHNL_IER0)	32	RW	0000_0000
14	Channel 0 Status (CHNL_STS0)	32	RW	0000_0000
18	Channel 0 Scale Factor (CHNL_SCALE_FACTOR0)	32	RW	1000_1000
1C	Channel 0 Scale Offset (CHNL_SCALE_OFFSET0)	32	RW	0000_0000
20	Channel 0 Crop Upper Left Corner Coordinate (CHNL_CROP_ULC0)	32	RW	0000_0000
24	Channel 0 Crop Lower Right Corner Coordinate (CHNL_CROP_LRC0)	32	RW	0000_0000
28	Channel 0 Color Space Conversion Coefficient 0 (CHNL_CSC_COEFF00)	32	RW	0000_0000
2C	Channel 0 Color Space Conversion Coefficient 1 (CHNL_CSC_COEFF10)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
30	Channel 0 Color Space Conversion Coefficient 2 (CHNL_CSC_COEFF20)	32	RW	0000_0000
34	Channel 0 Color Space Conversion Coefficient 3 (CHNL_CSC_COEFF30)	32	RW	0000_0000
38	Channel 0 Color Space Conversion Coefficient 4 (CHNL_CSC_COEFF40)	32	RW	0000_0000
3C	Channel 0 Color Space Conversion Coefficient 5 (CHNL_CSC_COEFF50)	32	RW	0000_0000
40	Channel 0 Alpha Value for ROI 0 (CHNL_ROI_0_ALPHA0)	32	RW	0000_0000
44	Channel 0 Upper Left Coordinate for ROI 0 (CHNL_ROI_0_ULC0)	32	RW	0000_0000
48	Channel 0 Lower Right Coordinate for ROI 0 (CHNL_ROI_0_LRC0)	32	RW	0000_0000
4C	Channel 0 Alpha Value for ROI 1 (CHNL_ROI_1_ALPHA0)	32	RW	0000_0000
50	Channel 0 Upper Left Coordinate for ROI 1 (CHNL_ROI_1_ULC0)	32	RW	0000_0000
54	Channel 0 Lower Right Coordinate for ROI 1 (CHNL_ROI_1_LRC0)	32	RW	0000_0000
58	Channel 0 Alpha Value for ROI 2 (CHNL_ROI_2_ALPHA0)	32	RW	0000_0000
5C	Channel 0 Upper Left Coordinate for ROI 2 (CHNL_ROI_2_ULC0)	32	RW	0000_0000
60	Channel 0 Lower Right Coordinate for ROI 2 (CHNL_ROI_2_LRC0)	32	RW	0000_0000
64	Channel 0 Alpha Value for ROI 3 (CHNL_ROI_3_ALPHA0)	32	RW	0000_0000
68	Channel 0 Upper Left Coordinate for ROI 3 (CHNL_ROI_3_ULC0)	32	RW	0000_0000
6C	Channel 0 Lower Right Coordinate for ROI 3 (CHNL_ROI_3_LRC0)	32	RW	0000_0000
70	Channel 0 RGB or Luma (Y) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_Y0)	32	RW	0000_0000
74	Channel 0 Chroma (U/Cb/UV/CbCr) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_U0)	32	RW	0000_0000
78	Channel 0 Chroma (V/Cr) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_V0)	32	RW	0000_0000
7C	Channel 0 Output Buffer Pitch (CHNL_OUT_BUF_PITCH0)	32	RW	0000_0000
80	Channel 0 Input Buffer Address (CHNL_IN_BUF_ADDR0)	32	RW	0000_0000
84	Channel 0 Input Buffer Pitch (CHNL_IN_BUF_PITCH0)	32	RW	0000_0000
88	Channel 0 Memory Read Control (CHNL_MEM_RD_CTRL0)	32	RW	0000_0000
8C	Channel 0 RGB or Luma (Y) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_Y0)	32	RW	0000_0000
90	Channel 0 Chroma (U/Cb/UV/CbCr) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_U0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
94	Channel 0 Chroma (V/Cr) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_V0)	32	RW	0000_0000
98	Channel 0 Scaled Image Configuration (CHNL_SCL_IMG_CFG0)	32	RW	0438_0780
9C	Channel 0 Flow Control (CHNL_FLOW_CTRL0)	32	RW	0000_0000

57.7.2 Channel 0 Control (CHNL_CTRL0)

Provides global settings for the channel pipeline, and controls how the internal blocks and line buffer perform the image processing.

Offset

Register	Offset
CHNL_CTRL0	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CHNL_EN	CLK_EN	CHNL_BY...	0	0	0		SW_RST	0	RAW_MSB...	VER_ID				0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0			0	0			0		0	SRC_TYPE	0			SRC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 CHNL_EN	Enable Channel Processing Enables image processing in the channel. You must configure all other channels before writing 1 to this field. 0 - Disable 1 - Enable
30	Channel Clock Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
CLK_EN	Enables the channel processing clock. Writing 1 to this field specifies that the clock gating (outside the ISI block) is disabled and the clock to the channel is enabled. **NOTE** When this field is 0, the ISI registers are readable but cannot be configured or written successfully. 0 - Disable 1 - Enable
29 CHNL_BYPASS	Channel Bypass Enable Enables you to bypass a channel if scaling and CSC are not required. For certain applications, the ISI module might not be required to perform any scaling or CSC on the image. For such cases, you must write 1 to this field to bypass the channel pipeline and store the image in memory. Configure the output image format (CHNLn_IMG_CTRL[FORMAT]) in the same way as the input image format. You can still carry out image flipping, cropping, and alpha insertion. **NOTE** • For deinterlacing purposes, the channel is automatically bypassed for one frame. Do not write 1 to this field for deinterlacing. • If this field is 1, CSC is bypassed even if CHNLn_IMG_CTRL[CSC_BYP] = 0. • If input is in RAW format, this bit is ignored, and the channel is always bypassed by hardware. • If channel is bypassed, line buffer is not used except for input format YUV420. YUV420 input data always uses line buffer even if channel is bypassed (if scaler is enabled for YUV420 input, it needs 2 line buffers, which requires to chain the line buffer of another channel). 0 - Disables 1 - Enables
28 —	Reserved
27 —	Reserved
26-25 —	Reserved
24 SW_RST	Software Reset Asserts a software reset to channel pipeline. Writing 1 to this field resets the processing pipeline similar to a hardware reset. This field does not become 0 itself; therefore, when writing 1, you must allow sufficient delay (four to six NOP cycles) for reset propagation before writing 0 to disable the software reset.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE Asserting the software reset does not clear the registers. It affects the processing pipeline and memory read control. Software reset allows any ongoing AXI transactions to complete and does not interrupt. 0 - No reset 1 - Software reset
23 —	Reserved
22 RAW_MSB_ALIGN	RAW to MSB Align Aligns the data selection from the pixel link to MSB. Data from Pixel link is aligned to MSB[15] when RAW size is smaller than 16 bits, otherwise it is aligned to MSB[31]. 0 - LSB aligned selection 1 - MSB aligned selection
21-18 VER_ID	Version ID Specifies a version ID for the module.
17 —	Reserved
16 —	Reserved
15 —	Reserved
14-12 —	Reserved
11 —	Reserved
10-8 —	Reserved
7-6 —	Reserved
5 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 SRC_TYPE	Type of Selected Input Image Source Defines the type of image input source. Write 1 to this field only when the source of the image is from memory. For all other sources, this field is 0. 0 - Pixel link 1 - Memory
3-1 —	Reserved
0 SRC	Input Image Source Port Selection Allows the source to select the desired input port of the pixel link crossbar from which the image is sourced. For example, if this field value is 000b, the image is sourced from input port 0 of the pixel link crossbar, and so on. Configure this field before enabling the channel processing. Do not change this field while the channel pipeline is working. **NOTE** The highest port number is for memory. 0 - Port 0 1 - Port 1

57.7.3 Channel 0 Image Control (CHNL_IMG_CTRL0)

Contains the fields that control how the input image is processed in the processing pipeline or channel.

All the output image formats that the FORMAT field defines are described in Table 548.

Table 548. Output image formats

Bit field value	Format	Description
000_0000b	RGBA8888	RGB format with alpha in LSB; 8 bits per component. 'A' indicates alpha value.
000_0001b	ABGR8888	BGR format with alpha in MSB; 8 bits per component. 'A' indicates alpha value.
000_0010b	ARGB8888	RGB format with alpha in MSB; 8 bits per component. 'A' indicates alpha value.
000_0011b	RGBX888	RGB format with 8 bits per color component (unpacked and MSB aligned in 32-bit DWORD). 'X' indicates the waste bits.
000_0100b	XBGR888	BGR format with 8 bits per color component (unpacked and LSB aligned in 32-bit DWORD). 'X' indicates the waste bits.

Table continues on the next page...

Table 548. Output image formats (continued)

Bit field value	Format	Description
000_0101b	XRGB888	RGB format with 8 bits per color component (unpacked and LSB aligned in 32-bit DWORD). 'X' indicates the waste bits.
000_0110b	RGB888P	RGB format with 8 bits per color component (packed into 24 bits). No waste bits.
000_0111b	BGR888P	BGR format with 8 bits per color component (packed into 24 bits). No waste bits.
000_1000b	A2BGR10	BGR format with 2 bits alpha in MSB; 10 bits per color component. 'A' indicates alpha value.
000_1001b	A2RGB10	RGB format with 2-bit alpha in MSB; 10 bits per color component. 'A' indicates alpha value.
000_1010b	RGB565	RGB format with 5 bits of R, B; 6 bits of G (packed into 16-bit WORD). No waste bits.
000_1011b	RAW8	8-bit RAW data packed into 32-bit DWORD See Figure 353.
000_1100b	RAW10	10-bit RAW data packed into 16-bit WORD See Figure 353.
000_1101b	RAW10P	10-bit RAW data packed into 32-bit DWORD See Figure 353.
000_1110b	RAW12	12-bit RAW data packed into 16-bit DWORD See Figure 353.
000_1111b	RAW16	16-bit RAW data packed into 32-bit DWORD See Figure 353.
001_0000b	YUV444_1P8P	8 bits per color component; 1-plane, YUV interleaved packed bytes
001_0001b	YUV444_2P8P	8 bits per color component; 2-plane, UV interleaved packed bytes
001_0010b	YUV444_3P8P	8 bits per color component; 3-plane, non-interleaved packed bytes
001_0011b	YUV444_1P8	8 bits per color component; 1-plane YUV interleaved unpacked bytes (8 MSB waste bits in 32-bit DWORD)
001_0100b	YUV444_1P10	10 bits per color component; 1-plane, YUV interleaved unpacked bytes (6 LSB waste bits in 16-bit WORD)
001_0101b	YUV444_2P10	10 bits per color component; 2-plane, UV interleaved unpacked bytes (6 LSB waste bits in 16-bit WORD)

Table continues on the next page...

Table 548. Output image formats (continued)

Bit field value	Format	Description
001_0110b	YUV444_3P10	10 bits per color component; 3-plane, non-interleaved unpacked bytes (6 LSB waste bits in 16-bit WORD)
001_0111b		Reserved
001_1000b	YUV444_1P10P	10 bits per color component; 1-plane, YUV interleaved packed bytes (2 MSB waste bits in 32-bit DWORD)
001_1001b	YUV444_2P10P	10 bits per color component; 2-plane, UV interleaved packed bytes (2 MSB waste bits in 32-bit DWORD)
001_1010b	YUV444_3P10P	10 bits per color component; 3-plane, non-interleaved packed bytes (2 MSB waste bits in 32-bit DWORD)
001_1011b		Reserved
001_1100b	YUV444_1P12	12 bits per color component; 1-plane, YUV interleaved unpacked bytes (4 LSB waste bits in 16-bit WORD)
001_1101b	YUV444_2P12	12 bits per color component; 2-plane, UV interleaved unpacked bytes (4 LSB waste bits in 16-bit WORD)
001_1110b	YUV444_3P12	12 bits per color component; 3-plane, non-interleaved unpacked bytes (4 LSB waste bits in 16-bit WORD)
001_1111b		Reserved
010_0000b	YUV422_1P8P	8 bits per color component; 1-plane, YUV interleaved packed bytes
010_0001b	YUV422_2P8P	8 bits per color component; 2-plane, UV interleaved packed bytes
010_0010b	YUV422_3P8P	8 bits per color component; 3-plane, non-interleaved packed bytes
010_0011b		Reserved
010_0100b	YUV422_1P10	10-bits per color component; 1-plane, YUV interleaved unpacked bytes (6 LSB waste bits in 16-bit WORD)
010_0101b	YUV422_2P10	10-bits per color component; 2-plane, UV interleaved unpacked bytes (6 LSB waste bits in 16-bit WORD)
010_0110b	YUV422_3P10	10 bits per color component; 3-plane, non-interleaved unpacked bytes (6 LSB waste bits in 16-bit WORD)
010_0111b		Reserved
010_1000b	YUV422_1P10P	10 bits per color component; 1-plane, YUV interleaved packed bytes (2 MSB waste bits in 32-bit DWORD)

Table continues on the next page...

Table 548. Output image formats (continued)

Bit field value	Format	Description
010_1001b	YUV422_2P10P	10 bits per color component; 2-plane, UV interleaved packed bytes (2 MSB waste bits in 32-bit DWORD)
010_1010b	YUV422_3P10P	10 bits per color component; 3-plane, non-interleaved packed bytes (2 MSB waste bits in 32-bit DWORD)
010_1011b		Reserved
010_1100b	YUV422_1P12	12 bits per color component; 1-plane, YUV interleaved unpacked bytes (4 LSB waste bits in 16-bit WORD)
010_1101b	YUV422_2P12	12 bits per color component; 2-plane, UV interleaved unpacked bytes (4 LSB waste bits in 16-bit WORD)
010_1110b	YUV422_3P12	12 bits per color component; 3-plane, non-interleaved unpacked bytes (4 LSB waste bits in 16-bit WORD)
010_1111b		Reserved
011_0000b		Reserved
011_0001b	YUV420_2P8P	8 bits per color component; 2-plane, UV interleaved packed bytes **NOTE** When pixel link input is of legacy YUV420 format, for example: Line 1: {Y0,Y1,U0}, {Y2,Y3,U1}… Line 2: {Y0,Y1,V0}, {Y2,Y3,V1}… The output pixel data stored in the memory is as below: **MSB LSB** Y_n, Y_{n-1}, ... Y_0 V_m, U_m, ... V_0, U_0
011_0010b	YUV420_3P8P	8 bits per color component; 3-plane, non-interleaved packed bytes
011_0011b		Reserved
011_0100b		Reserved
011_0101b	YUV420_2P10	10 bits per color component; 2-plane, UV interleaved unpacked bytes (6 LSB waste bits in 16-bit WORD)

Table continues on the next page...

Table 548. Output image formats (continued)

Bit field value	Format	Description
		NOTE When pixel link input is of legacy YUV420 format, for example: Line 1: {Y0,Y1,U0}, {Y2,Y3,U1}… Line 2: {Y0,Y1,V0}, {Y2,Y3,V1}… The output pixel data stored in the memory is as below: **MSB LSB** Y_n, Y_{n-1}, ... Y_0 V_m, U_m, ... V_0, U_0
011_0110b	YUV420_3P10	10 bits per color component; 3-plane, non-interleaved unpacked bytes (6 LSB waste bits in 16-bit WORD)
011_0111b		Reserved
011_1000b		Reserved
011_1001b	YUV420_2P10P	10 bits per color component; 2-plane, UV interleaved packed bytes (2 MSB waste bits in 32-bit DWORD) **NOTE** When pixel link input is of legacy YUV420 format, for example: Line 1: {Y0,Y1,U0}, {Y2,Y3,U1}… Line 2: {Y0,Y1,V0}, {Y2,Y3,V1}… The output pixel data stored in the memory is as below: **MSB LSB** Y_n, Y_{n-1}, ... Y_0 V_m, U_m, ... V_0, U_0
011_1010b	YUV420_3P10P	10 bits per color component; 3-plane, non-interleaved packed bytes (2 MSB waste bits in 32-bit DWORD)
011_1011b		Reserved
011_1100b		Reserved
011_1101b	YUV420_2P12	12 bits per color component; 2-plane, UV interleaved unpacked bytes (4 LSB waste bits in 16-bit WORD)

Table continues on the next page...

Table 548. Output image formats (continued)

Bit field value	Format	Description
		NOTE When pixel link input is of legacy YUV420 format, for example: Line 1: {Y0,Y1,U0}, {Y2,Y3,U1}… Line 2: {Y0,Y1,V0}, {Y2,Y3,V1}… The output pixel data stored in the memory is as below: **MSB LSB** Y_n, Y_{n-1}, ... Y_0 V_m, U_m, ... V_0, U_0
011_1110b	YUV420_3P12	12 bits per color component; 3-plane, non-interleaved unpacked bytes (4 LSB waste bits in 16-bit WORD)
011_1111b		Reserved
100_0000b	RAW32	32-bit RAW data packed into 32-bit DWORD
100_0001b	RAW14	14-bit RAW data packed into 32-bit DWORD See Figure 353.

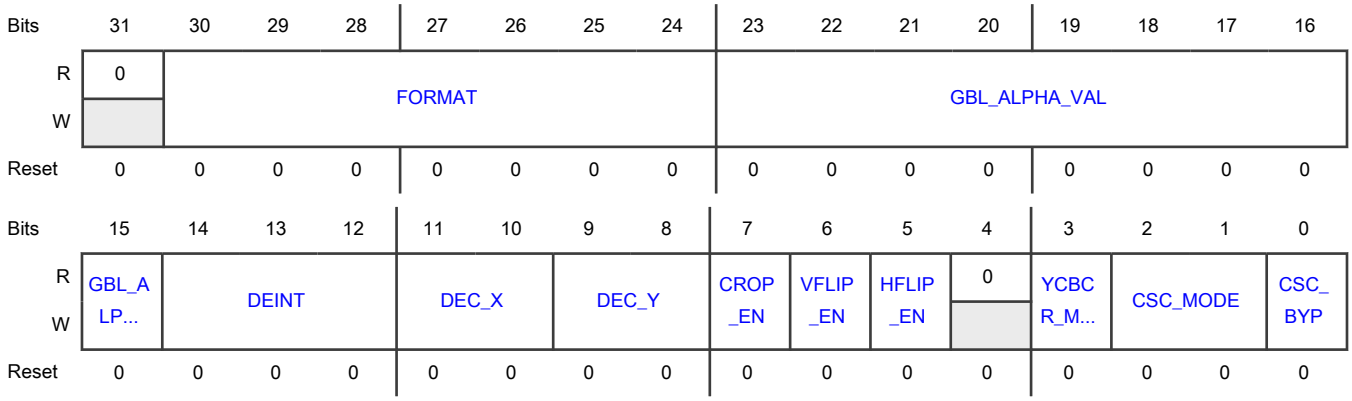
	Bytes	Byte3								Byte2								Byte1								Byte0							
	Bits	31								23								15								7							
Raw8	Pixel data	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
Raw10	Pixel data	2	2	2	2	2	2	2	2	2	2							1	1	1	1	1	1	1	1	1	1						
Raw10P	Pixel data			3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
Raw12	Pixel data	2	2	2	2	2	2	2	2	2	2	2	2					1	1	1	1	1	1	1	1	1	1	1	1				
Raw14	Pixel data	2	2	2	2	2	2	2	2	2	2	2	2	2	2			1	1	1	1	1	1	1	1	1	1	1	1	1	1		
Raw16	Pixel data	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 353. RAW format output

Offset

Register	Offset
CHNL_IMG_CTRL0	4h

Diagram



Fields

Field	Description
31 —	Reserved
30-24 FORMAT	Output Image Format Specifies the image format to be used when storing the image output from channel pipeline to memory. The UV byte lanes are synonymous with CbCr byte lanes for YUV output pixel formats. For example, you must select the YUV420_2P* formats when the output is YCbCr 2-plane 420 output format. See all output image formats in Table 548. **NOTE** For RAW pixel formats, if the output pixel format has few bits per pixel than the input pixel format, ISP truncates the MSB of each pixel. 000_0000-100_0001 - See Table 548.
23-16 GBL_ALPHA_VAL	Global Alpha Value Specifies the alpha value that is inserted with all RGB pixels in the image when being stored into the memory. For RGB 10-bit formats, only 2 MSBs are used as alpha value for insertion. When different alpha values are required to be inserted in up to four regions of interest, use the ROI_x_ALPHA, ROI_x_ULC, and ROI_x_LRC (where x = 0, 1, 2, 3) registers. 0000_0000-1111_1111 - Alpha value to be inserted with all RGB pixels
15 GBL_ALPHA_EN	Global Alpha Value Insertion Enable Enables you to insert the global alpha value in the RGB image. If you need to insert different alpha values in up to four regions of interest, use the ROI_x_ALPHA, ROI_x_ULC, and ROI_x_LRC (where x = 0, 1, 2, 3) registers. 0 - Disable 1 - Enable
14-12	Deinterlace Control

Table continues on the next page...

Table continued from the previous page...

Field	Description
DEINT	Defines the basic deinterlacing method that you use to combine the interlaced frames. Deinterlacing is required to send the interlaced frames over the pixel link. See Image deinterlacing methods for details on deinterlacing techniques and how to configure the channels to perform deinterlacing. 000,001 - No deinterlacing 010 - Weave deinterlacing (odd, even) 011 - Weave deinterlacing (even, odd) All other values are reserved.
11-10 DEC_X	Horizontal Pre-Decimation Control Controls the pre-decimation filter for down scaling of the input image in the horizontal direction. For field value 00b, pre-decimation filter is disabled but the bilinear scaling filter is still operational. 00 - Disabled 01 - Decimate by 2 10 - Decimate by 4 11 - Decimate by 8
9-8 DEC_Y	Vertical Pre-Decimation Control Controls the pre-decimation filter for down scaling of the input image in the vertical direction. For field value 00b, pre-decimation filter is disabled but the bilinear scaling filter is still operational. 00 - Disabled 01 - Decimate by 2 10 - Decimate by 4 11 - Decimate by 8
7 CROP_EN	Output Image Cropping Enable Enables cropping of the image when you write 1 to this field. Cropping of the image is carried out after down scaling. Therefore, you must program the cropping coordinates according to the scaled image. You can configure the cropping coordinates in Channel 0 Crop Upper Left Corner Coordinate (CHNL_CROP_ULC0) and Channel 0 Crop Lower Right Corner Coordinate (CHNL_CROP_LRC0). 0 - Disable 1 - Enable
6 VFLIP_EN	Vertical Flip Control Enables vertical image flipping when you write 1 to this field. 0 - Disable 1 - Enable
5 HFLIP_EN	Horizontal Flip Control Enables horizontal image flipping when you write 1 to this field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Disable 1 - Enable
4 —	Reserved
3 YCBCR_MODE	YCbCr Mode Enables YCbCr mode. Write 1 to this field when performing YCbCr to RCB conversion and write 0 to this field when performing RGB to YUV conversion. This field changes the behavior of the scaler when performing U/V scaling. 0 - Disable 1 - Enable
2-1 CSC_MODE	Color Space Conversion Operating Mode Controls how the CSC unit operates on pixels when CSC is not bypassed. 00 - Convert from YUV to RGB 01 - Convert from YCbCr to RGB 10 - Convert from RGB to YUV 11 - Convert from RGB to YCbCr
0 CSC_BYP	Color Space Conversion Bypass Control Bypasses the CSC block that means no operation occurs on the pixels when you write 1 to this field. When you write 0 to this field, color conversion selected by CSC_MODE takes place. 0 - CSC operational 1 - CSC bypassed

57.7.4 Channel 0 Output Buffer Control (CHNL_OUT_BUF_CTRL0)

Controls how the output buffer loads the destination memory address and configures the overflow panic controls.

Offset

Register	Offset
CHNL_OUT_BUF_CTRL0	8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAX_WR_...	MAX_WR_...	0										PANIC_SET_THD_V			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LOAD_BU...	LOAD_BU...	0		PANIC_SET_THD_U				0				PANIC_SET_THD_Y			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 MAX_WR_BEATS_Y	Maximum AXI Write Beats for Y-Buffer Specifies the maximum write beats per write request for the Y-buffer. When Y-buffer is concatenated with other buffers (Y+U+V or Y+V), you can increase throughput by increasing the maximum write beats to 16 (that is, 256 bytes) per write request. When buffer is not combined, this field has no effect, and a maximum of 8 beats per write request are supported. 0 - 8 beats per write (128 bytes) 1 - 16 beats per write (256 bytes)
30 MAX_WR_BEATS_UV	Maximum AXI Write Beats for U- and V-Buffers Specifies the maximum write beats per write request for the U- and V-buffers. When U-buffer and V-buffer are concatenated, you can increase throughput by increasing the maximum write beats to 16 (that is, 256 bytes) per write request. When buffers are not combined, this field has no effect, and a maximum of 8 beats per write request are supported. 0 - 8 beats per write (128 bytes) 1 - 16 beats per write (256 bytes)
29-20 —	Reserved
19-16 PANIC_SET_THD_V	Overflow Panic Set Threshold Value for V Output Buffer Contains the threshold value after which the potential overflow panic alert interrupt (CHNL_STS[PANIC_V_BUF]) and the panic signal to the subsystem are asserted to request a higher priority on AXI to drain the output buffers of the channel. Panic is asserted when the buffer is *n*×6.25%full (where *n* = 1 to 15). Panic signal deasserts when the number of filled rows in the ping-pong buffer fall below the programmed threshold. CHNL_STS[PANIC_V_BUF] becomes 0 when you write 1 to it. 0000 - No panic alert 0001-1111 - Panic asserts

Table continues on the next page...

Table continued from the previous page...

Field	Description
15 LOAD_BUF2_ADDR	Load Buffer 2 Address Specifies the load buffer 2 address from the CHNL_OUT_BUF2_ADDR_*n* (where *n* = Y, U, V) registers. When you require to use a new value, you must toggle this field to indicate that a new buffer address has been programmed for use. On every toggle of this field, the new address values are shadowed in design.
14 LOAD_BUF1_ADDR	Load Buffer 1 Address Specifies the load buffer 1 address from the CHNL_OUT_BUF1_ADDR_*n* (where *n* = Y, U, V) registers. When you require to use a new value, you must toggle this field to indicate that a new buffer address has been programmed for use. On every toggle of this field, the new address values are shadowed in design.
13-12 —	Reserved
11-8 PANIC_SET_THD_U	Overflow Panic Set Threshold Value for U Output Buffer Contains the threshold value after which the potential overflow panic alert interrupt (CHNL_STS[PANIC_U_BUF]) and the panic signal to the subsystem are asserted to request a higher priority on AXI to drain the output buffers of the channel. Panic is asserted when the buffer is *n*×6.25%full (where *n* = 1 to 15). Panic signal deasserts when the number of filled rows in the ping-pong buffer fall below the programmed threshold. CHNL_STS[PANIC_U_BUF] becomes 0 when you write 1 to it. 0000 - No panic alert 0001-1111 - Panic asserts
7-4 —	Reserved
3-0 PANIC_SET_THD_Y	Overflow Panic Set Threshold Value for Y or RGB Output Buffer Contains the threshold value after which the potential overflow panic alert interrupt (CHNL_STS[PANIC_Y_BUF]) and the panic signal to the subsystem are asserted to request a higher priority on AXI to drain the output buffers of the channel. Panic is asserted when the buffer is *n*×6.25%full (where *n* = 1 to 15). Panic signal deasserts when the number of filled rows in the ping-pong buffer fall below the programmed threshold. CHNL_STS[PANIC_Y_BUF] becomes 0 when you write 1 to it. 0000 - No panic alert 0001-1111 - Panic asserts

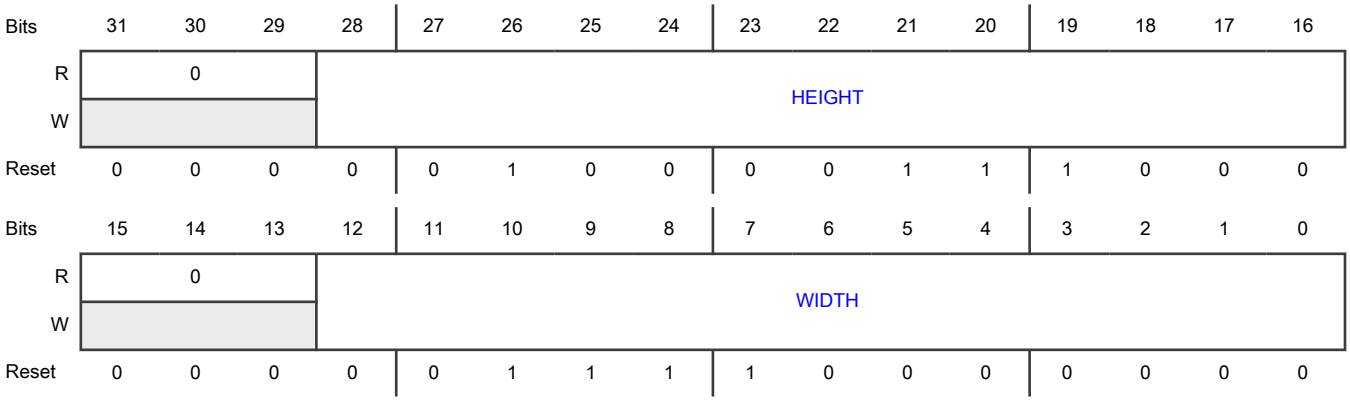
57.7.5 Channel 0 Image Configuration (CHNL_IMG_CFG0)

Specifies details related to the image width (in pixels) and height (in lines). This register specifies the image resolution for any type of source for the processing channel.

Offset

Register	Offset
CHNL_IMG_CFG0	Ch

Diagram



Fields

Field	Description
31-29 —	Reserved
28-16 HEIGHT	Input Image Height Contains the height of the input image in terms of the number of lines for both cases when the image is input from the pixel link or from memory. The default height is 1080 lines.
15-13 —	Reserved
12-0 WIDTH	Input Image Width Contains the width of the input image in terms of the number of pixels for both cases when the image is input from the pixel link or from memory. The default width is 1920 pixels.

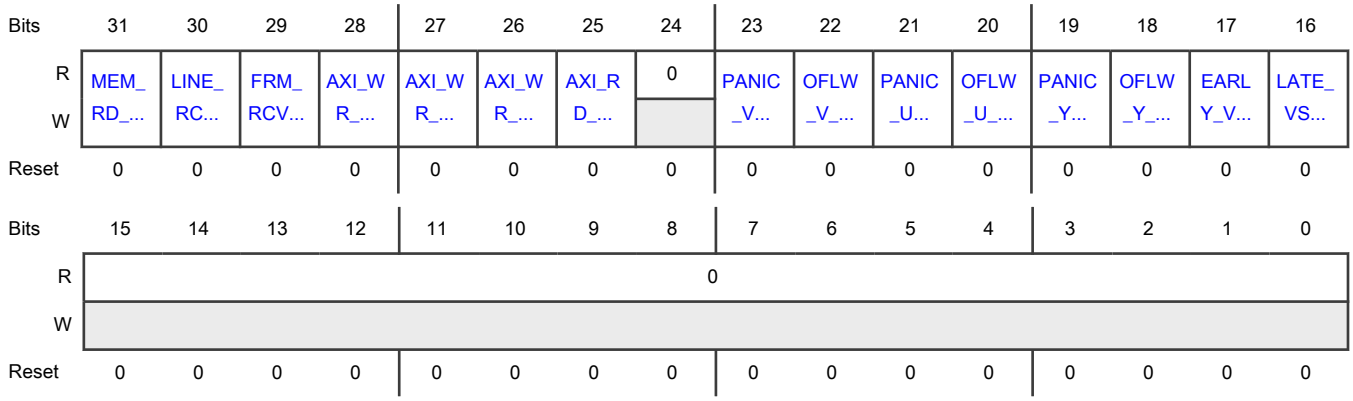
57.7.6 Channel 0 Interrupt Enable (CHNL_IER0)

Enables the generation of an interrupt to the processor.

Offset

Register	Offset
CHNL_IER0	10h

Diagram



Fields

Field	Description
31 MEM_RD_DONE_EN	Memory Read Complete Interrupt Enable Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
30 LINE_RCVD_EN	Line Received Interrupt Enable Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
29 FRM_RCVD_EN	Frame Received Interrupt Enable Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
28 AXI_WR_ERR_V_EN	AXI Bus Read Error Interrupt Enable for V Data Buffer Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
27 AXI_WR_ERR_U_EN	AXI Bus Read Error Interrupt Enable for U Data Buffer Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
26	AXI Bus Read Error Interrupt Enable for Y and RGB Data Buffer Enables the assertion of an interrupt from the channel when the corresponding status field is 1.

Table continues on the next page...

Table continued from the previous page...

Field	Description
AXI_WR_ERR_Y_EN	0 - Disable 1 - Enable
25 AXI_RD_ERR_EN	AXI Bus Read Error Interrupt Enable Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
24 —	Reserved
23 PANIC_V_BUF_EN	V Output Buffer Potential Overflow Panic Interrupt Enable Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
22 OFLW_V_BUF_EN	V Output Buffer Overflow Interrupt Enable Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
21 PANIC_U_BUF_EN	U Output Buffer Potential Overflow Panic Interrupt Enable Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
20 OFLW_U_BUF_EN	U Output Buffer Overflow Interrupt Enable Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
19 PANIC_Y_BUF_EN	Y Output Buffer Potential Overflow Panic Interrupt Enable Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
18 OFLW_Y_BUF_EN	Y Output Buffer Overflow Interrupt Enable Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable
17 EARLY_VSYNC_ERR_EN	VSYNC Timing (Early) Error Interrupt Enable Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
16 LATE_VSYNC_ERR_EN	VSYNC Timing (Late) Error Interrupt Enable Enables the assertion of an interrupt from the channel when the corresponding status field is 1. 0 - Disable 1 - Enable
15-0 —	Reserved

57.7.7 Channel 0 Status (CHNL_STS0)

Specifies the module status.

Offset

Register	Offset
CHNL_STS0	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MEM_RD_...	LINE_ST...	FRM_STRD	AXI_WR_...	AXI_WR_...	AXI_WR_...	AXI_RD_...	0	PANIC_V...	OFLW_V_...	PANIC_U...	OFLW_U_...	PANIC_Y...	OFLW_Y_...	EARLY_V...	LATE_VS...
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C		W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					MEM_RD_...	BUF2_AC...	BUF1_AC...	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 MEM_RD_DONE	Memory Read Complete Interrupt Flag Indicates that the channel has read the full image from the input memory location (). Configure the system to read the next image from the memory. 0 - Not complete or not started 1 - Completed
30 LINE_STRD	Line Received and Stored Interrupt Flag Indicates that a line is received and stored in the memory. This interrupt flag sets for every line received and stored in memory. 0 - Not received 1 - Received and stored
29 FRM_STRD	Frame Stored Successfully Interrupt Flag Indicates that a full frame is received and stored in the memory. This interrupt flag sets for every frame received and stored in memory. 0 - Not received or in progress 1 - Received and stored
28 AXI_WR_ERR_V	AXI Bus Write Error Interrupt Flag for V Data Buffer Indicates that an error has occurred on the AXI bus when storing V data into memory. 0 - No error 1 - Error occurred
27 AXI_WR_ERR_U	AXI Bus Write Error Interrupt Flag for U Data Buffer Indicates that an error has occurred on the AXI bus when storing U data into memory. 0 - No error 1 - Error occurred
26 AXI_WR_ERR_Y	AXI Bus Write Error Interrupt Flag for Y/RGB Data Buffer Indicates that an error has occurred on the AXI bus when storing Y/RGB data into memory. 0 - No error 1 - Error occurred
25 AXI_RD_ERR	AXI Bus Read Error Interrupt Flag Indicates that an error has occurred on the AXI bus when reading the image data from memory. 0 - No error 1 - Error occurred
24	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
23 PANIC_V_BUF	V Output Buffer Potential Overflow Panic Alert Interrupt Flag Indicates whether the buffer has crossed the panic threshold limit. The channel requires you to increase the AXI bus priority for the V channel so that it can empty its output buffers; otherwise, the buffer might overflow and loose valid pixel data. You must perform an action on this interrupt in a way to empty out the buffers before any overflow occurs. You can configure the threshold for this panic interrupt in Channel 0 Output Buffer Control (CHNL_OUT_BUF_CTRL0) 0 - Threshold limit not crossed 1 - Threshold limit crossed
22 OFLW_V_BUF	Overflow in V Output Buffer Interrupt Flag Indicates that an overflow has occurred in the channel because of internal buffers not being read via AXI and are now full and not accepting new valid pixels in the output buffer control stage. The number of bytes lost during the overflow event is also reported in Channel 0 Status (CHNL_STS0). 0 - No overflow 1 - Overflow
21 PANIC_U_BUF	U Output Buffer Potential Overflow Panic Alert Interrupt Flag Indicates whether the buffer has crossed the panic threshold limit. The channel requires you to increase the AXI bus priority for the U channel so that it can empty its output buffers; otherwise, the buffer might overflow and loose valid pixel data. You must perform an action on this interrupt in a way that buffers are emptied out before any overflow occurs. You can configure the threshold for this panic interrupt in Channel 0 Output Buffer Control (CHNL_OUT_BUF_CTRL0). 0 - Threshold limit not crossed 1 - Threshold limit crossed
20 OFLW_U_BUF	Overflow in U Output Buffer Interrupt Flag Indicates that an overflow has occurred in the channel because of internal buffers not being read via AXI and are now full and not accepting new valid pixels in the output buffer control stage. The number of bytes lost during the overflow event is also reported in Channel 0 Status (CHNL_STS0). 0 - No overflow 1 - Overflow
19 PANIC_Y_BUF	Y or RGB Output Buffer Potential Overflow Panic Alert Interrupt Flag Indicates whether the buffer has crossed the panic threshold limit. The channel requires you to increase the AXI bus priority for the Y or RGB channel so that it can empty its output buffers; otherwise, the buffer might overflow and loose valid pixel data.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You must perform an action on this interrupt in a way that buffers are emptied out before any overflow occurs. You can configure the threshold for this panic interrupt in Channel 0 Output Buffer Control (CHNL_OUT_BUF_CTRL0). 0 - Threshold limit not crossed 1 - Threshold limit crossed
18 OFLW_Y_BUF	Overflow in Y or RGB Output Buffer Interrupt Flag Indicates that an overflow has occurred in the channel because of internal buffers not being read via AXI and are now full and not accepting new valid pixels in the output buffer control stage. The number of bytes lost during the overflow event is also reported in Channel 0 Status (CHNL_STS0). 0 - No overflow 1 - Overflow
17 EARLY_VSYNC_ERR	VSYNC Timing (Early) Error Interrupt flag Indicates that the VSYNC for the frame was detected earlier than expected. The resultant frame has corrupted data. 0 - No error 1 - VSYNC detected earlier
16 LATE_VSYNC_ERR	VSYNC Timing (Late) Error Interrupt flag Indicates that the VSYNC for the frame was detected later than expected. The resultant frame has corrected data. The extra lines in frame are ignored. 0 - No error 1 - VSYNC detected later
15-11 —	Reserved
10 MEM_RD_OFLOW	Memory Read FIFO Overflow Error Status Indicates that the read from memory has caused an internal FIFO to overflow. Data loss has occurred during memory read. Status is valid only when a memory read has been initiated. 0 - No overflow 1 - FIFO overflow
9 BUF2_ACTIVE	Current Frame Stored in Buffer 2 Address Indicates that the buffer 2 address is being used to store current frame. 0 - Buffer 2 address inactive 1 - Buffer 2 address in use
8	Current Frame Stored in Buffer 1 Address

Table continues on the next page...

Table continued from the previous page...

Field	Description
BUF1_ACTIVE	Indicates that the buffer 1 address is being used to store current frame. 0 - Buffer 1 address inactive 1 - Buffer 1 address in use
7-0 —	Reserved

57.7.8 Channel 0 Scale Factor (CHNL_SCALE_FACTOR0)

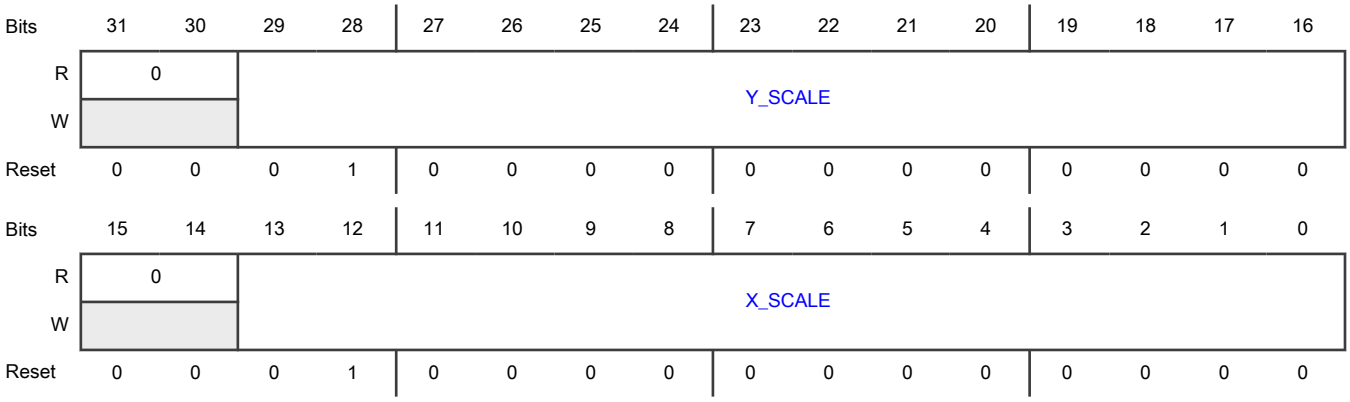
Provides the scale factor for the bilinear filter in the scaling engine. The maximum downscaling factor is 1/2 in a way that the output image in either axis is 1/2 the size of the source. The scale-down factor can range from 1.0 to 2.0. Use decimation and the bilinear filter together to achieve scaling by more than a factor of 2.

Write a value of 10.0000_0000_0000 to this register to reduce the image by a factor of two in the output frame buffer.

Offset

Register	Offset
CHNL_SCALE_FACTOR0	18h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-16	Vertical Scaling Factor

Table continues on the next page...

Table continued from the previous page...

Field	Description
Y_SCALE	Represents the Y-dimension scaling factor for the bilinear filter as a two-bit integer and 12-bit fractional number (##.####_####_####). The maximum value you can write to this field is 2 because scaling down by a factor greater than 2 is not supported with the bilinear filter. **NOTE** ISI actual output line value will be rounded up to an integer. For example, if the input line value = 800 and the output line value = 720, you must set [Y_SCALE] to a value such that the theoretical output line value (800/[Y_SCALE]) is equal to or less than 720, which means: • If [Y_SCALE] is set to 1C8h, the theoretical output line value = 719.859375 which will be rounded up to 720 → The actual output line is correct. • If [Y_SCALE] is set to 1C7h, the theoretical output line value = 720.017578 which will be rounded up to 721 → There will be one excessive line in the actual output.
15-14 —	Reserved
13-0 X_SCALE	Horizontal Scaling Factor Represents the X-dimension scaling factor for the bilinear filter as a two-bit integer and 12-bit fractional number (##.####_####_####). The maximum value you can write to this field is 2 because scaling down by a factor greater than 2 is not supported with the bilinear filter. **NOTE** ISI actual output width will be rounded up to an integer. For example, if the input width = 800 and the output width = 720, you must set [X_SCALE] to a value such that the theoretical output width (800/[X_SCALE]) is equal to or less than 720, which means: • If [X_SCALE] is set to 1C8h, the theoretical output width = 719.859375 which will be rounded up to 720 → The actual output width is correct. • If [X_SCALE] is set to 1C7h, the theoretical output width = 720.017578 which will be rounded up to 721 → There will be one excessive pixel in the actual output.

57.7.9 Channel 0 Scale Offset (CHNL_SCALE_OFFSET0)

Provides the initial scale offset for the bilinear filter. The X and Y offset provide the ability to access the source image with a per sub-pixel granularity. This provides the capability to use all source pixels to affect the output image. The fixed offset values can be used for sub-pixel adjustments in the bilinear scaling filter.

For example, when scaling an image down by a factor of 2, an initial offset of 0 results in subsampling every other pixel. In case of a fixed offset of 800h (1/2), all pixels are used in scaling the final output pixel value. In this case, the first output pixel is the sum of (1/2*P0) + (1/2*P1). This fixed offset is applied after the decimation filter stage, and before the bilinear filter stage.

Offset

Register	Offset
CHNL_SCALE_OFFSET0	1Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				Y_OFFSET											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				X_OFFSET											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 Y_OFFSET	Vertical Scaling Offset Represents the Y-dimension scaling offset as a 12-bit fractional number (0.####_####_####). This represents a fixed pixel offset that gets added to the scaled address to determine source data for the scaling engine.
15-12 —	Reserved
11-0 X_OFFSET	Horizontal Scaling Offset Represents the X-dimension scaling offset as a 12-bit fractional number (0.####_####_####). This represents a fixed pixel offset that gets added to the scaled address to determine source data for the scaling engine.

57.7.10 Channel 0 Crop Upper Left Corner Coordinate (CHNL_CROP_ULC0)

Provides the upper left coordinate of the source image that must be cropped. Configure ULC as per:

- The scaled image resolution (if scaler is enabled)
- The input image resolution (if scaler is bypassed)

The output image, after cropping, comprises pixels of the source image that are within the region that CHNL_ROI_n_ULC and CHNL_ROI_n_LRC (where *n* = 0, 1, 2, 3) specify. Write 1 to CHNL_IMG_CTRL[CROP_EN] to perform the cropping operation.

Offset

Register	Offset
CHNL_CROP_ULC0	20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				X											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				Y											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 X	Upper Left X-Coordinate Contains the upper left X-coordinate (in pixels) of the scaled image. The range of valid values includes 0 to (WIDTH - 1).
15-12 —	Reserved
11-0 Y	Upper Left Y-Coordinate Indicates the upper left Y-coordinate (in pixels) of the scaled image. The range of valid values includes 0 to (HEIGHT - 1).

57.7.11 Channel 0 Crop Lower Right Corner Coordinate (CHNL_CROP_LRC0)

Provides the lower right coordinate of the source image that must be cropped. Configure LRC as per:

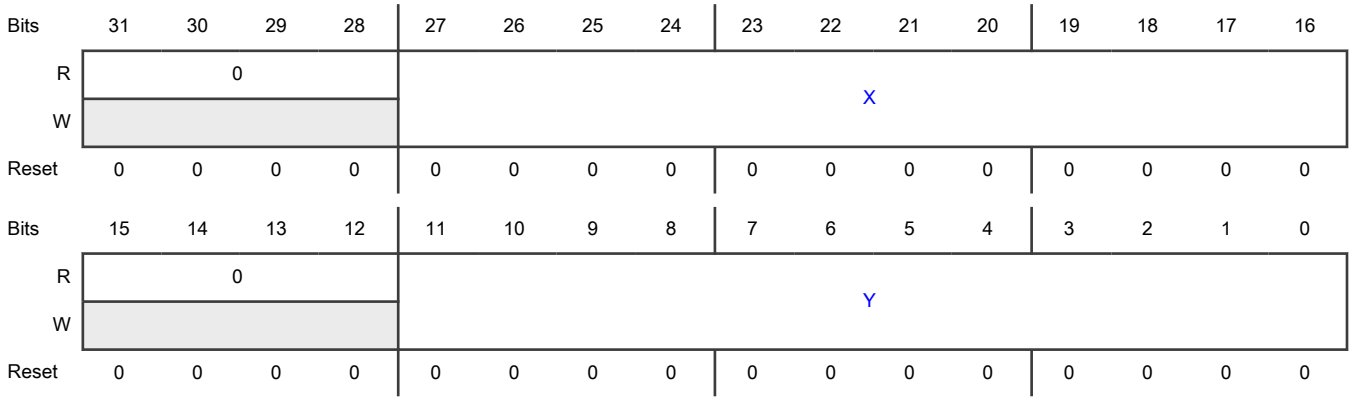
- The scaled image resolution (if scaler is enabled)
- The input image resolution (if scaler is bypassed)

The output image, after cropping, comprises pixels of the source image that are within the region that CHNL_ROI_n_ULC and CHNL_ROI_n_LRC (where *n* = 0, 1, 2, 3) specify. Write 1 to CHNL_IMG_CTRL[CROP_EN] to perform the cropping operation.

Offset

Register	Offset
CHNL_CROP_LRC0	24h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 X	Lower Right X-Coordinate Contains the lower right X-coordinate (in pixels) of the source image. The range of valid values includes 0 to (WIDTH - 1).
15-12 —	Reserved
11-0 Y	Lower Right Y-Coordinate Indicates the lower right Y-coordinate (in pixels) of the source image. The range of valid values includes 0 to (HEIGHT - 1).

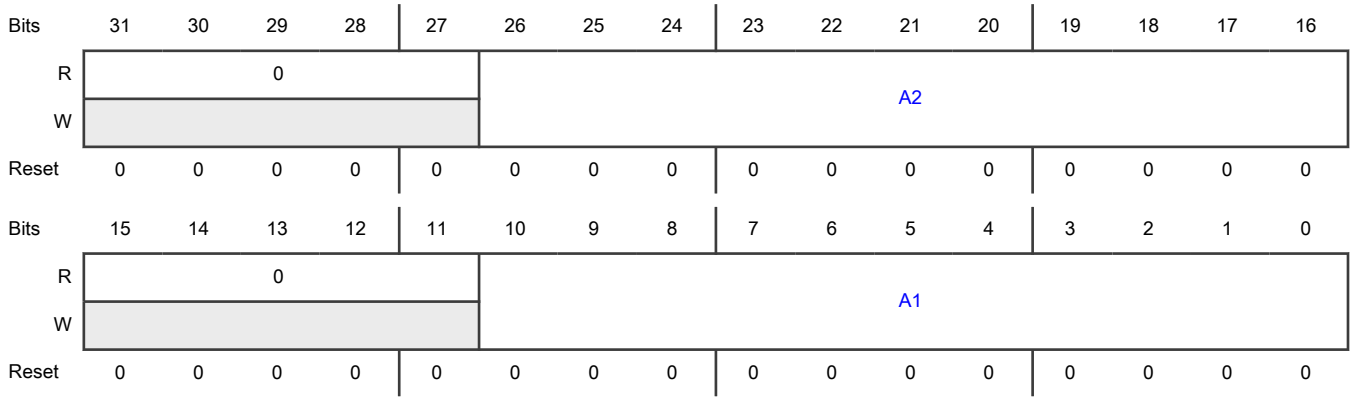
57.7.12 Channel 0 Color Space Conversion Coefficient 0 (CHNL_CSC_COEFF00)

Contains color space conversion coefficients in the two's-complement notation.

Offset

Register	Offset
CHNL_CSC_COEFF00	28h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 A2	CSC Coefficient A2 Value Specifies the two's-complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 A1	CSC Coefficient A1 Value Specifies the two's-complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

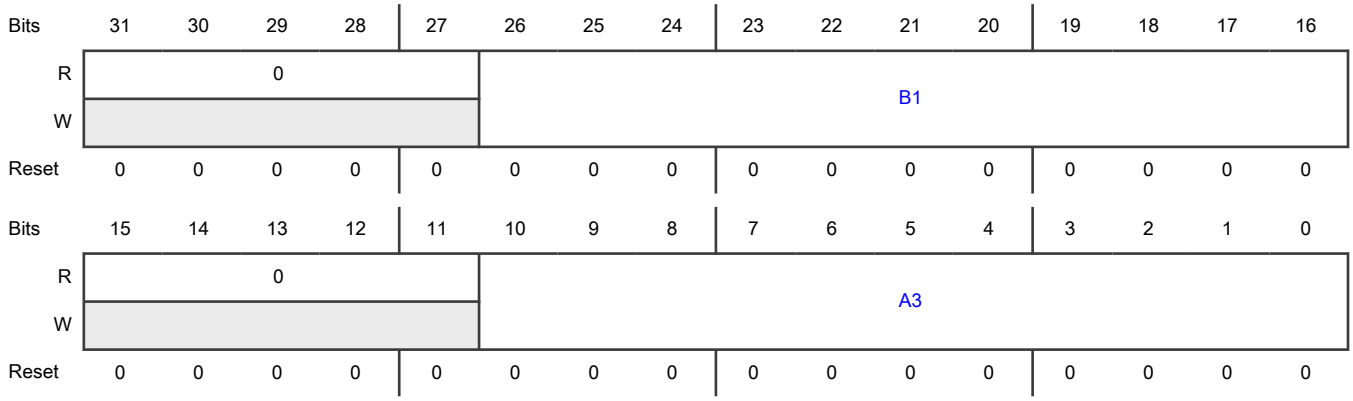
57.7.13 Channel 0 Color Space Conversion Coefficient 1 (CHNL_CSC_COEFF10)

Contains color space conversion coefficients in the two's-complement notation.

Offset

Register	Offset
CHNL_CSC_COEFF10	2Ch

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 B1	CSC Coefficient B1 Value Specifies the two's-complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 A3	CSC Coefficient A3 Value Specifies the two's-complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

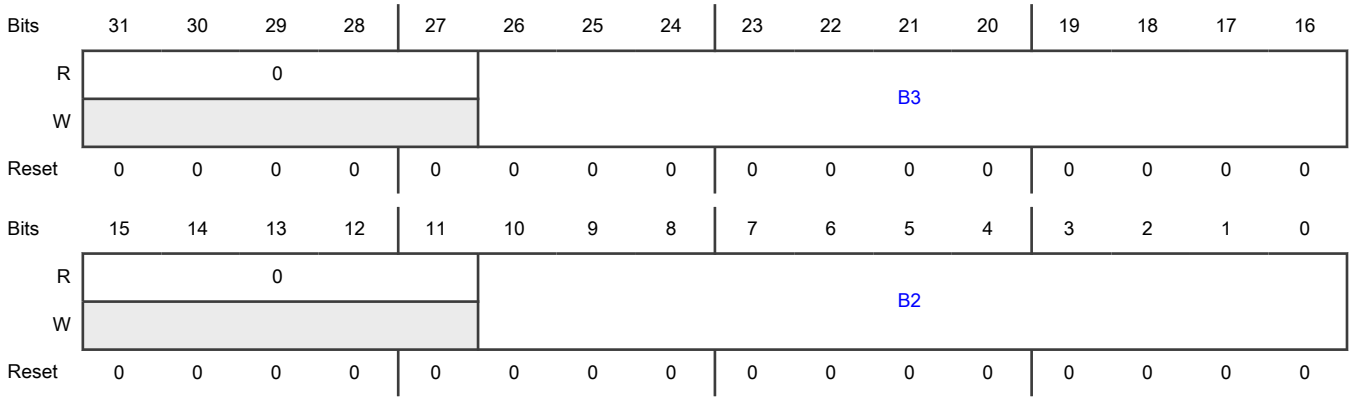
57.7.14 Channel 0 Color Space Conversion Coefficient 2 (CHNL_CSC_COEFF20)

Contains color space conversion coefficients in the two's-complement notation.

Offset

Register	Offset
CHNL_CSC_COEFF20	30h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 B3	CSC Coefficient B3 Value Specifies the two's-complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 B2	CSC Coefficient B2 Value Specifies the two's-complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

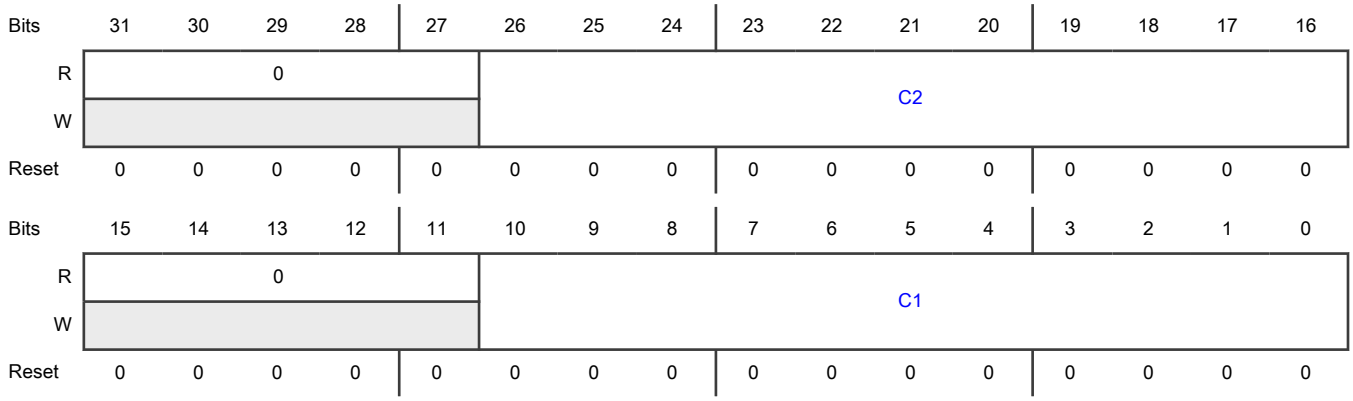
57.7.15 Channel 0 Color Space Conversion Coefficient 3 (CHNL_CSC_COEFF30)

Contains color space conversion coefficients in the two's-complement notation.

Offset

Register	Offset
CHNL_CSC_COEFF30	34h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-16 C2	CSC Coefficient C2 Value Specifies the two's-complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 C1	CSC Coefficient C1 Value Specifies the two's-complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

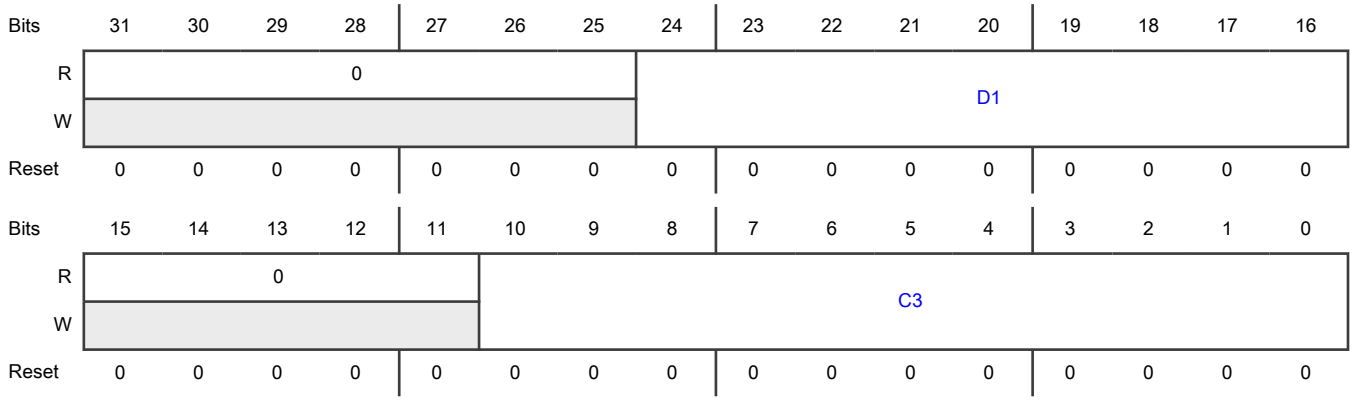
57.7.16 Channel 0 Color Space Conversion Coefficient 4 (CHNL_CSC_COEFF40)

Contains color space conversion coefficients in the two's-complement notation.

Offset

Register	Offset
CHNL_CSC_COEFF40	38h

Diagram



Fields

Field	Description
31-25 —	Reserved
24-16 D1	CSC Coefficient D1 Value Specifies the two's-complement coefficient integer offset that is to be added.
15-11 —	Reserved
10-0 C3	CSC Coefficient C3 Value Specifies the two's-complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

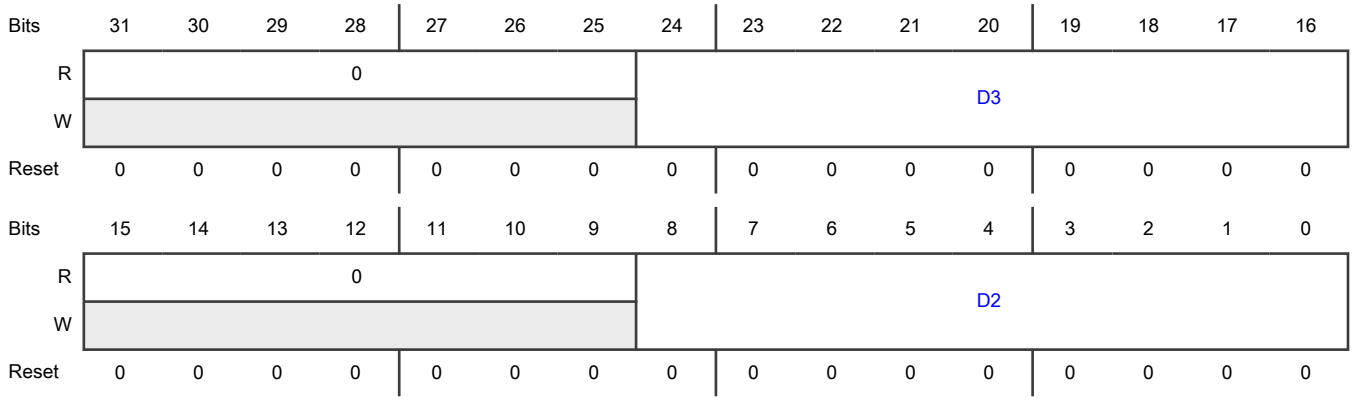
57.7.17 Channel 0 Color Space Conversion Coefficient 5 (CHNL_CSC_COEFF50)

Contains color space conversion coefficients in the two's-complement notation.

Offset

Register	Offset
CHNL_CSC_COEFF50	3Ch

Diagram



Fields

Field	Description
31-25 —	Reserved
24-16 D3	CSC Coefficient D3 Value Specifies the two's-complement coefficient integer offset that is to be added.
15-9 —	Reserved
8-0 D2	CSC Coefficient D2 Value Specifies the two's-complement coefficient integer offset that is to be added.

57.7.18 Channel 0 Alpha Value for ROI 0 (CHNL_ROI_0_ALPHA0)

Provides the alpha value and control to enable alpha value insertion into ROI 0 of an image.

The region is defined by Channel 0 Upper Left Coordinate for ROI 0 (CHNL_ROI_0_ULC0) and Channel 0 Lower Right Coordinate for ROI 0 (CHNL_ROI_0_LRC0). The alpha insertion happens after you perform the scaling, so that ULC and LRC correspond to the scaled image dimensions.

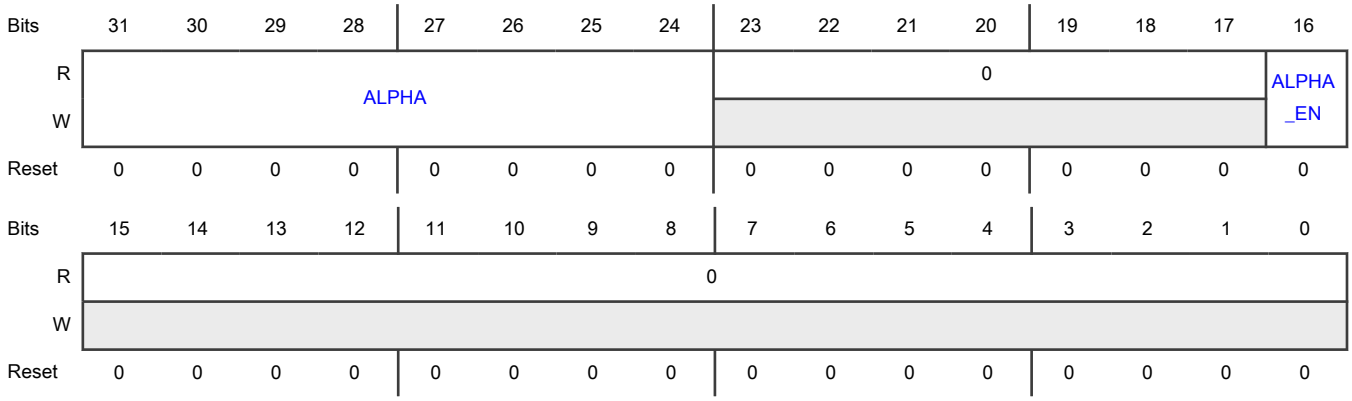
NOTE

If one value of alpha is to be inserted globally for the whole image, use Channel 0 Image Control (CHNL_IMG_CTRL0) to provide the global alpha value and enable control. The ROI_* registers are not used.

Offset

Register	Offset
CHNL_ROI_0_ALPHA0	40h

Diagram



Fields

Field	Description
31-24 ALPHA	Alpha Value Specifies the alpha value to be inserted into a ROI of an image if you write 1 to ALPHA_EN. ROI is defined by Channel 0 Upper Left Coordinate for ROI 0 (CHNL_ROI_0_ULC0) and Channel 0 Lower Right Coordinate for ROI 0 (CHNL_ROI_0_LRC0).
23-17 —	Reserved
16 ALPHA_EN	Alpha Value Insertion Enable Controls whether the alpha value is inserted into a ROI of an image. ROI is defined by Channel 0 Upper Left Coordinate for ROI 0 (CHNL_ROI_0_ULC0) and Channel 0 Lower Right Coordinate for ROI 0 (CHNL_ROI_0_LRC0). 0 - Disable 1 - Enable
15-0 —	Reserved

57.7.19 Channel 0 Upper Left Coordinate for ROI 0 (CHNL_ROI_0_ULC0)

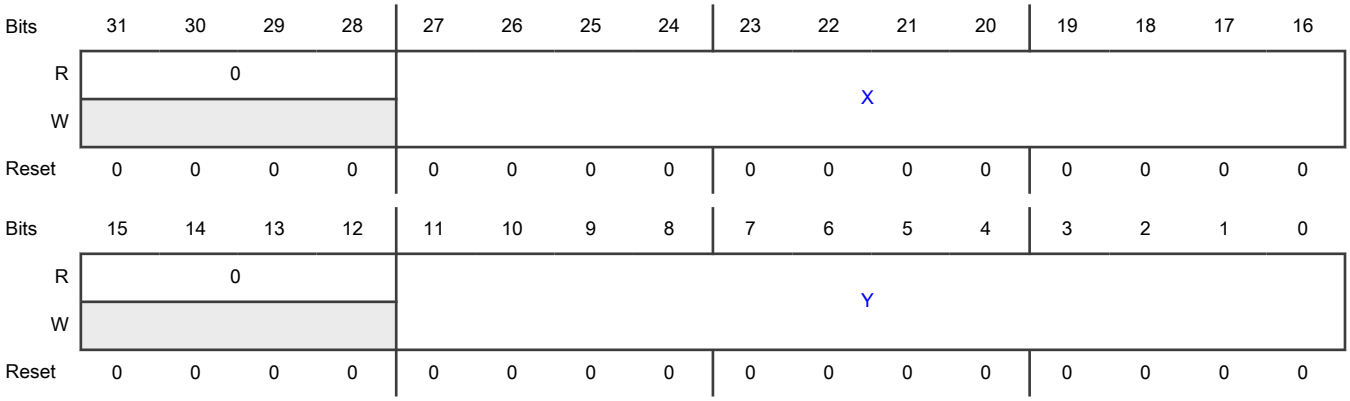
Defines the upper-left coordinate for ROI 0. Channel 0 Upper Left Coordinate for ROI 0 (CHNL_ROI_0_ULC0) and Channel 0 Lower Right Coordinate for ROI 0 (CHNL_ROI_0_LRC0) must correspond to the scaled image dimensions because the alpha insertion happens after scaling is performed.

Pixel locations that are greater than or equal to the upper-left coordinates and lesser than or equal to the lower-right coordinates define ROI.

Offset

Register	Offset
CHNL_ROI_0_ULC0	44h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 X	Upper Left X-Coordinate Indicates the upper-left X-coordinate (in pixels) of ROI 0.
15-12 —	Reserved
11-0 Y	Upper Left Y-Coordinate Indicates the upper-left Y-coordinate (in pixels) of ROI 0.

57.7.20 Channel 0 Lower Right Coordinate for ROI 0 (CHNL_ROI_0_LRC0)

Defines the lower-right coordinate for ROI 0. Channel 0 Upper Left Coordinate for ROI 0 (CHNL_ROI_0_ULC0) and Channel 0 Lower Right Coordinate for ROI 0 (CHNL_ROI_0_LRC0) must correspond to the scaled image dimensions because the alpha insertion happens after scaling is performed.

Pixel locations that are greater than or equal to the upper-left coordinates and less than or equal to the lower-right coordinates define ROI.

Offset

Register	Offset
CHNL_ROI_0_LRC0	48h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				X											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				Y											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 X	Lower Right X-Coordinate Indicates the lower-right X-coordinate (in pixels) of ROI 0.
15-12 —	Reserved
11-0 Y	Lower Right Y-Coordinate Indicates the lower-right Y-coordinate (in pixels) of ROI 0.

57.7.21 Channel 0 Alpha Value for ROI 1 (CHNL_ROI_1_ALPHA0)

Provides the alpha value and control to enable alpha value insertion into ROI 1 of an image.

The region is defined by Channel 0 Upper Left Coordinate for ROI 1 (CHNL_ROI_1_ULC0) and Channel 0 Lower Right Coordinate for ROI 1 (CHNL_ROI_1_LRC0). The alpha insertion happens after you perform the scaling, so that ULC and LRC correspond to the scaled image dimensions.

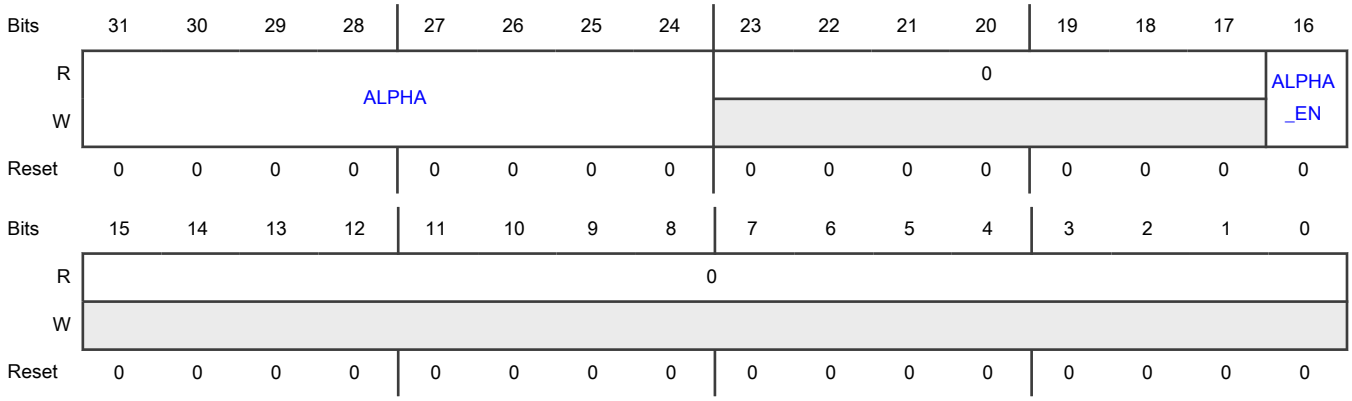
NOTE

If one particular value of alpha is to be inserted globally for the whole image, use Channel 0 Image Control (CHNL_IMG_CTRL0) to provide the global alpha value and enable control. The ROI_* registers are not used.

Offset

Register	Offset
CHNL_ROI_1_ALPHA0	4Ch

Diagram



Fields

Field	Description
31-24 ALPHA	Alpha Value Specifies the alpha value to be inserted into the ROI of an image if ALPHA_EN = 1. The ROI is defined by Channel 0 Upper Left Coordinate for ROI 1 (CHNL_ROI_1_ULC0) and Channel 0 Lower Right Coordinate for ROI 1 (CHNL_ROI_1_LRC0).
23-17 —	Reserved
16 ALPHA_EN	Alpha Value Insertion Enable Controls whether alpha value is inserted into the ROI of an image. The ROI is defined by Channel 0 Upper Left Coordinate for ROI 1 (CHNL_ROI_1_ULC0) and Channel 0 Lower Right Coordinate for ROI 1 (CHNL_ROI_1_LRC0). 0 - Disable 1 - Enable
15-0 —	Reserved

57.7.22 Channel 0 Upper Left Coordinate for ROI 1 (CHNL_ROI_1_ULC0)

Defines the upper-left coordinate for ROI 1. Channel 0 Upper Left Coordinate for ROI 1 (CHNL_ROI_1_ULC0) and Channel 0 Lower Right Coordinate for ROI 1 (CHNL_ROI_1_LRC0) must correspond to the scaled image dimensions because the alpha insertion happens after scaling is performed.

Pixel locations that are greater than or equal to the upper-left coordinates and lesser than or equal to the lower-right coordinates define ROI.

Offset

Register	Offset
CHNL_ROI_1_ULC0	50h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				X											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				Y											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 X	Upper Left X-Coordinate Indicates the upper-left X-coordinate (in pixels) of ROI 1.
15-12 —	Reserved
11-0 Y	Upper Left Y-Coordinate Indicates the upper-left Y-coordinate (in pixels) of ROI 1.

57.7.23 Channel 0 Lower Right Coordinate for ROI 1 (CHNL_ROI_1_LRC0)

Defines the lower-right coordinate for ROI 1. Channel 0 Upper Left Coordinate for ROI 1 (CHNL_ROI_1_ULC0) and Channel 0 Lower Right Coordinate for ROI 1 (CHNL_ROI_1_LRC0) must correspond to the scaled image dimensions because the alpha insertion happens after scaling is performed.

Pixel locations that are greater than or equal to the upper-left coordinates and lesser than or equal to the lower-right coordinates define ROI.

Offset

Register	Offset
CHNL_ROI_1_LRC0	54h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				X											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				Y											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 X	Lower Right X-Coordinate Indicates the lower-right X-coordinate (in pixels) of ROI 1.
15-12 —	Reserved
11-0 Y	Lower Right Y-Coordinate Indicates the lower-right Y-coordinate (in pixels) of ROI 1.

57.7.24 Channel 0 Alpha Value for ROI 2 (CHNL_ROI_2_ALPHA0)

Provides the alpha value and control to enable alpha value insertion into ROI 2 of an image. The alpha insertion happens after you perform the scaling, so that ULC and LRC correspond to the scaled image dimensions.

The region is defined by Channel 0 Upper Left Coordinate for ROI 2 (CHNL_ROI_2_ULC0) and Channel 0 Lower Right Coordinate for ROI 2 (CHNL_ROI_2_LRC0).

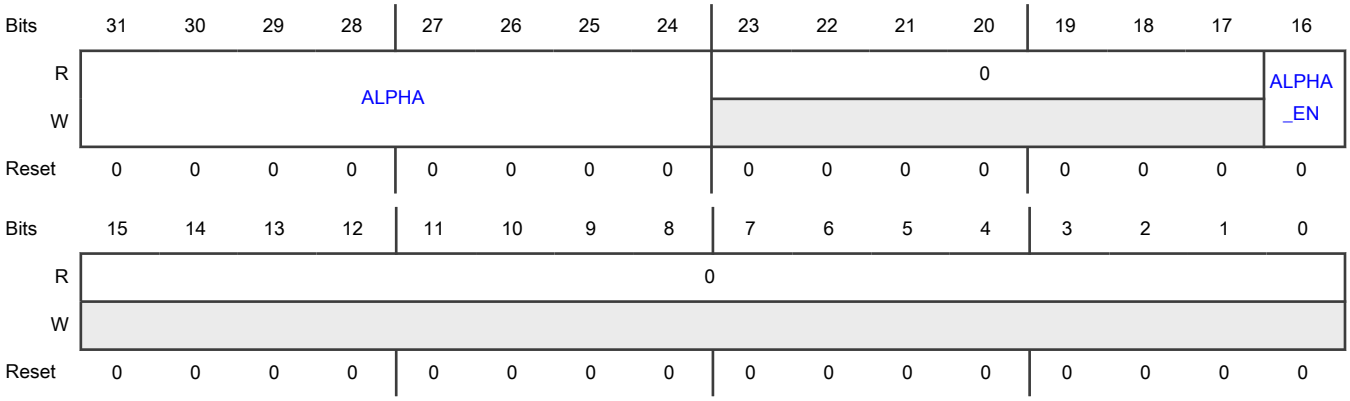
NOTE

If one particular value of alpha is to be inserted globally for the whole image, use Channel 0 Image Control (CHNL_IMG_CTRL0) to provide the global alpha value and enable control. The ROI_* registers are not used.

Offset

Register	Offset
CHNL_ROI_2_ALPHA0	58h

Diagram



Fields

Field	Description
31-24 ALPHA	Alpha Value Specifies the alpha value to be inserted into the ROI of an image if ALPHA_EN = 1. ROI is defined by Channel 0 Upper Left Coordinate for ROI 2 (CHNL_ROI_2_ULC0) and Channel 0 Lower Right Coordinate for ROI 2 (CHNL_ROI_2_LRC0).
23-17 —	Reserved
16 ALPHA_EN	Alpha Value Insertion Enable Controls whether alpha value is inserted or not into the ROI of an image. ROI is defined by Channel 0 Upper Left Coordinate for ROI 2 (CHNL_ROI_2_ULC0) and Channel 0 Lower Right Coordinate for ROI 2 (CHNL_ROI_2_LRC0). 0 - Disable 1 - Enable
15-0 —	Reserved

57.7.25 Channel 0 Upper Left Coordinate for ROI 2 (CHNL_ROI_2_ULC0)

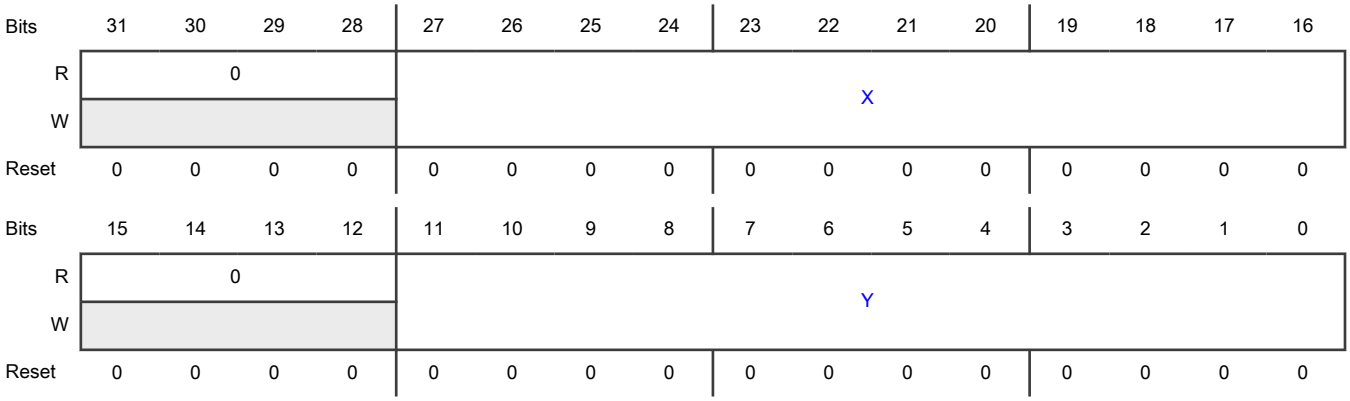
Defines the upper-left coordinate for ROI 2. Channel 0 Upper Left Coordinate for ROI 2 (CHNL_ROI_2_ULC0) and Channel 0 Lower Right Coordinate for ROI 2 (CHNL_ROI_2_LRC0) must correspond to the scaled image dimensions because the alpha insertion happens after scaling is performed.

Pixel locations that are greater than or equal to the upper-left coordinates and lesser than or equal to the lower-right coordinates define ROI.

Offset

Register	Offset
CHNL_ROI_2_ULC0	5Ch

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 X	Upper Left X-Coordinate Indicates the upper-left X-coordinate (in pixels) of ROI 2.
15-12 —	Reserved
11-0 Y	Upper Left Y-Coordinate Indicates the upper-left Y-coordinate (in pixels) of ROI 2.

57.7.26 Channel 0 Lower Right Coordinate for ROI 2 (CHNL_ROI_2_LRC0)

Defines the lower-right coordinate for ROI 2. Channel 0 Upper Left Coordinate for ROI 2 (CHNL_ROI_2_ULC0) and Channel 0 Lower Right Coordinate for ROI 2 (CHNL_ROI_2_LRC0) must correspond to the scaled image dimensions because the alpha insertion happens after scaling is performed.

Pixel locations that are greater than or equal to the upper-left coordinates and less than or equal to the lower-right coordinates define ROI.

Offset

Register	Offset
CHNL_ROI_2_LRC0	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				X											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				Y											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 X	Lower Right X-Coordinate Indicates the lower-right X-coordinate (in pixels) of ROI 2.
15-12 —	Reserved
11-0 Y	Lower Right Y-Coordinate Indicates the lower-right Y-coordinate (in pixels) of ROI 2.

57.7.27 Channel 0 Alpha Value for ROI 3 (CHNL_ROI_3_ALPHA0)

Provides the alpha value and control to enable alpha value insertion into ROI 3 of an image. The region is defined by Channel 0 Upper Left Coordinate for ROI 3 (CHNL_ROI_3_ULC0) and Channel 0 Lower Right Coordinate for ROI 3 (CHNL_ROI_3_LRC0). The alpha insertion happens after you perform the scaling, so that ULC and LRC correspond to the scaled image dimensions.

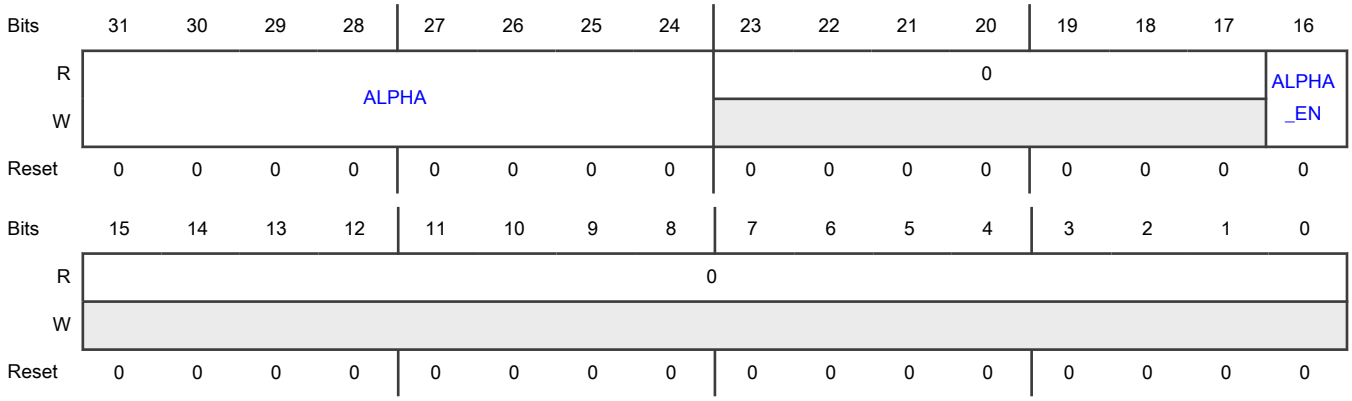
NOTE

If one particular value of alpha is to be inserted globally for the whole image, use Channel 0 Image Control (CHNL_IMG_CTRL0) to provide the global alpha value and enable control. The ROI_* registers are not used

Offset

Register	Offset
CHNL_ROI_3_ALPHA0	64h

Diagram



Fields

Field	Description
31-24 ALPHA	Alpha Value Specifies the alpha value to be inserted into the ROI of an image if ALPHA_EN = 1. ROI is defined by Channel 0 Upper Left Coordinate for ROI 3 (CHNL_ROI_3_ULC0) and Channel 0 Lower Right Coordinate for ROI 3 (CHNL_ROI_3_LRC0).
23-17 —	Reserved
16 ALPHA_EN	Alpha Value Insertion Enable Controls whether alpha value is inserted into the ROI of an image. ROI is defined by Channel 0 Upper Left Coordinate for ROI 3 (CHNL_ROI_3_ULC0) and Channel 0 Lower Right Coordinate for ROI 3 (CHNL_ROI_3_LRC0). 0 - Disable 1 - Enable
15-0 —	Reserved

57.7.28 Channel 0 Upper Left Coordinate for ROI 3 (CHNL_ROI_3_ULC0)

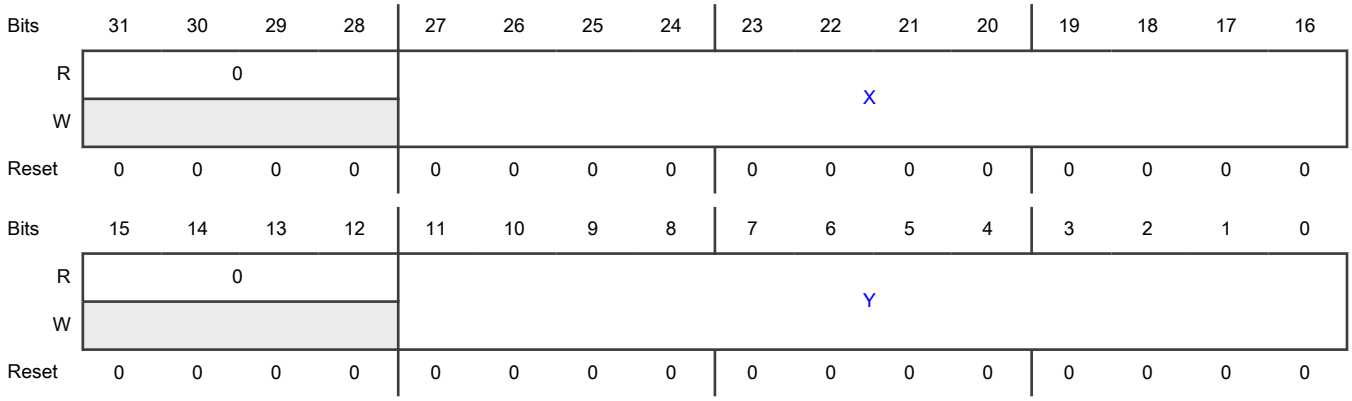
Defines the upper-left coordinate for ROI 3. Channel 0 Upper Left Coordinate for ROI 3 (CHNL_ROI_3_ULC0) and Channel 0 Lower Right Coordinate for ROI 3 (CHNL_ROI_3_LRC0) must correspond to the scaled image dimensions because the alpha insertion happens after scaling is performed.

Pixel locations that are greater than or equal to the upper-left coordinates and lesser than or equal to the lower-right coordinates define ROI.

Offset

Register	Offset
CHNL_ROI_3_ULC0	68h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 X	Upper Left X-Coordinate Indicates the upper-left X-coordinate (in pixels) of ROI 3.
15-12 —	Reserved
11-0 Y	Upper Left Y-Coordinate Indicates the upper-left Y-coordinate (in pixels) of ROI 3.

57.7.29 Channel 0 Lower Right Coordinate for ROI 3 (CHNL_ROI_3_LRC0)

Defines the lower-right coordinate for ROI 3. Channel 0 Upper Left Coordinate for ROI 3 (CHNL_ROI_3_ULC0) and Channel 0 Lower Right Coordinate for ROI 3 (CHNL_ROI_3_LRC0) must correspond to the scaled image dimensions because the alpha insertion happens after scaling is performed.

Pixel locations that are greater than or equal to the upper-left coordinates and lesser than or equal to the lower-right coordinates define ROI.

Offset

Register	Offset
CHNL_ROI_3_LRC0	6Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				X											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				Y											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 X	Lower Right X-Coordinate Indicates the lower-right X-coordinate (in pixels) of ROI 3.
15-12 —	Reserved
11-0 Y	Lower Right Y-Coordinate Indicates the lower-right Y-coordinate (in pixels) of ROI 3.

57.7.30 Channel 0 RGB or Luma (Y) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_Y0)

Specifies the starting memory address of buffer 1, where the RGB or Y (Luma) data of the processed image is stored.

Offset

Register	Offset
CHNL_OUT_BUF1_ADDR_Y0	70h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ADDR	Address for RGB or Y (Luma) Specifies the starting address for the RGB or Y (luma) memory location.

57.7.31 Channel 0 Chroma (U/Cb/UV/CbCr) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_U0)

Specifies the starting memory address of buffer 1, where the U or Cb data of the processed image is stored. In a two-plane operation, this register points to the starting memory address where the UV or CbCr chroma of the processed image is stored. You do not use this register in a one-plane operation.

Offset

Register	Offset
CHNL_OUT_BUF1_ADDR_U0	74h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ADDR	Address for U/Cb/UV/CbCr Specifies the starting address for the (U or Cb) or 2-plane (UV or CbCr) chroma memory location.

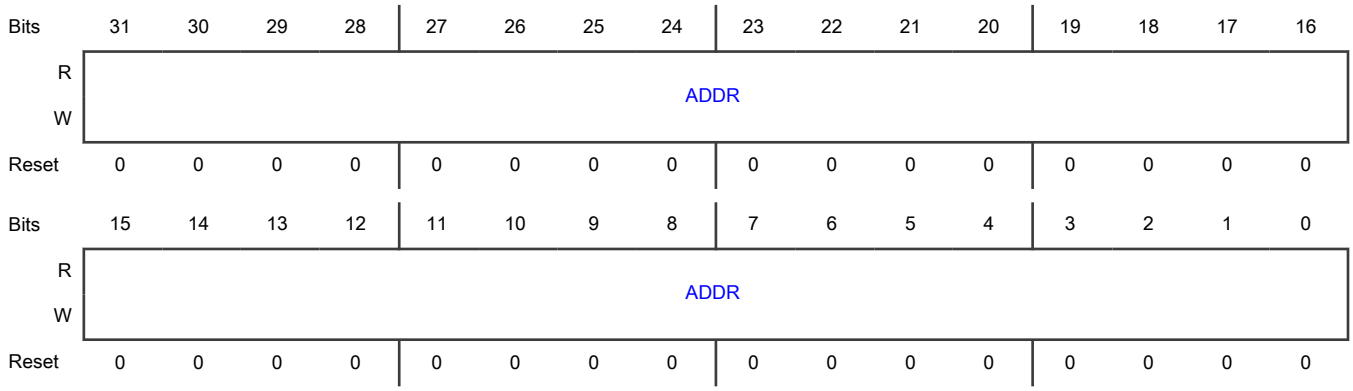
57.7.32 Channel 0 Chroma (V/Cr) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_V0)

Specifies the starting memory address of buffer 1, where the V or Cr data of the processed image is stored. You do not use this register in one or two-plane operation.

Offset

Register	Offset
CHNL_OUT_BUF1_ADDR_V0	78h

Diagram



Fields

Field	Description
31-0 ADDR	Address for V or Cr Specifies the starting address for the V or Cr memory location.

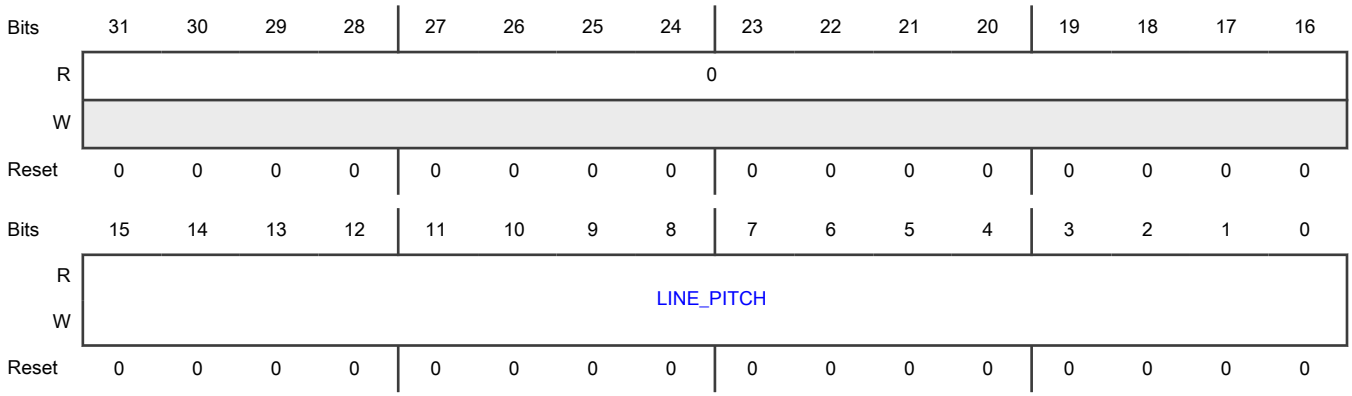
57.7.33 Channel 0 Output Buffer Pitch (CHNL_OUT_BUF_PITCH0)

Contains the output buffer pitch in bytes. Any byte value indicates the vertical pitch. You use this value for output pixel address calculations.

Offset

Register	Offset
CHNL_OUT_BUF_PITCH0	7Ch

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 LINE_PITCH	Output Buffer Line Pitch Indicates the number of bytes in memory between two vertically adjacent pixels.

57.7.34 Channel 0 Input Buffer Address (CHNL_IN_BUF_ADDR0)

Provides the start address for reading the standalone or deinterlaced image frames from memory. Blending (or averaging) deinterlacing stores the odd frame image into the memory address that this register specifies. The channel reads back the image and sends it to the deinterlacing logic when SRC is set as memory (CHNL_CTRL[SRC_TYPE] = 1).

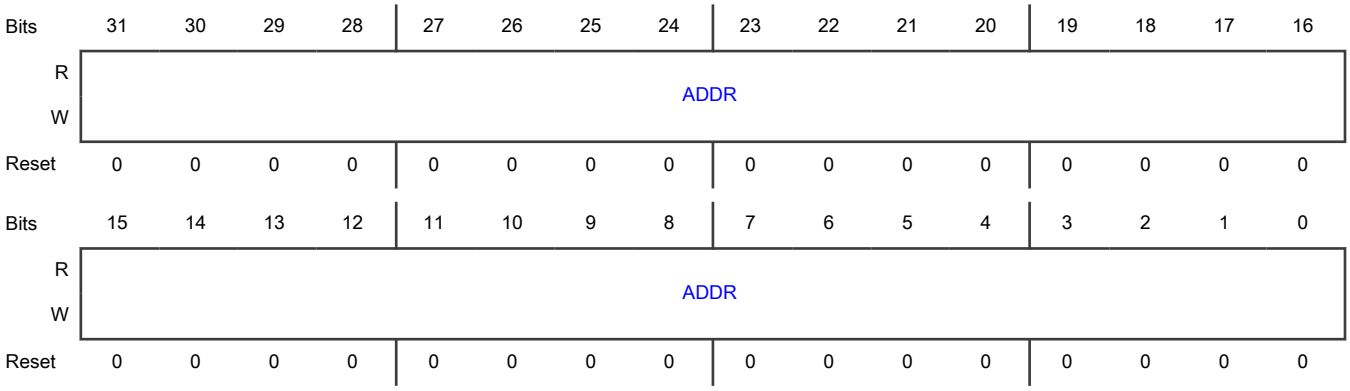
The storage location allows for two frames to be stored. When reading the image frames from memory, the module switches between the two locations automatically. The read operation starts automatically when you need to perform blend deinterlacing. When processing standalone images from memory, you must trigger the processing of images. The format of the image that can be stored in this location can be RGB, YUV444, or YUV422 only. The image format is set automatically when one channel is configured to perform blend deinterlacing.

For deinterlacing, configure the deinterlace method and source type in one channel and configure only the source type as memory in another channel. The module can pair the deinterlace and memory channels automatically to perform blend deinterlace. For other deinterlace methods, read back of frames from memory is not required. See Image deinterlacing methods for more information on deinterlacing.

Offset

Register	Offset
CHNL_IN_BUF_ADDR0	80h

Diagram



Fields

Field	Description
31-0 ADDR	Starting Address Specifies the starting address from where the input image is read. You can update this field during direct memory reads. For deinterlacing of the intermediate frames, you must reserve address space of two frame size + pitch range. The hardware logic determines the addresses for the intermediate frames during deinterlacing because the software is not concerned about intermediate frames.

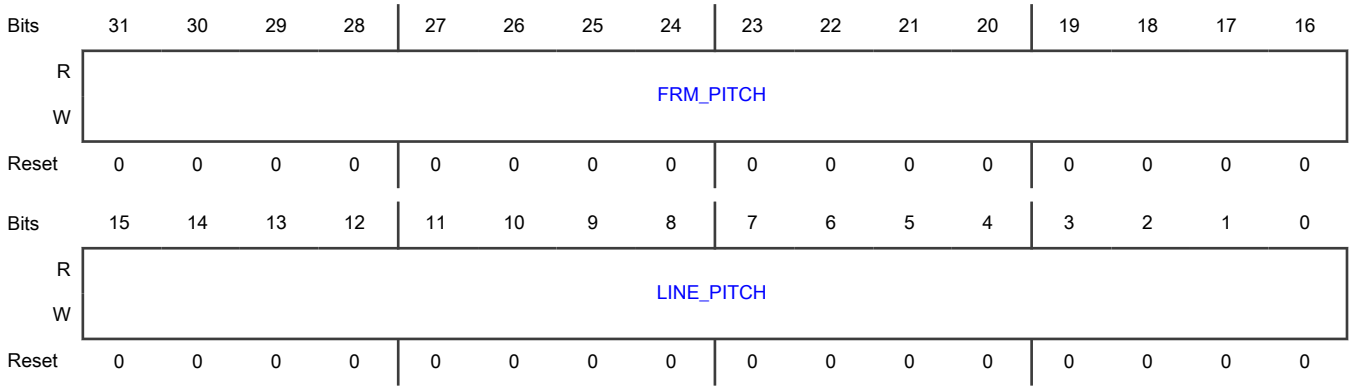
57.7.35 Channel 0 Input Buffer Pitch (CHNL_IN_BUF_PITCH0)

Provides the line and frame pitch for the image to be read out from the memory location that you configure in .

Offset

Register	Offset
CHNL_IN_BUF_PITCH0	84h

Diagram



Fields

Field	Description
31-16 FRM_PITCH	Frame Pitch Indicates the number of bytes in memory between the two frames that are read from the memory. You can alternatively keep the frame pitch as zero and update during direct memory reads. For deinterlacing, the frame pitch is required to store the intermediate frames. You must reserve two frame size + pitch range of addresses for the intermediate frames because the software is not concerned about them.
15-0 LINE_PITCH	Line Pitch Indicates the number of bytes in memory between two vertically adjacent pixels.

57.7.36 Channel 0 Memory Read Control (CHNL_MEM_RD_CTRL0)

Provides the image format information and control to start reading image from memory. This register is used when the image source type is memory (CHNL_CTRL[SRC_TYPE] = 1) and standalone image is being read from memory (with no deinterlacing performed). This register is not used during deinterlacing where the deinterlace logic controls the image read back from memory.

Both deinterlace image read and standalone image read cannot happen together, even on different channels.

All input image formats that IMG_TYPE defines are described in Table 549.

Table 549. Input image formats

Bit field value	Format	Description
0000b	BGR8P	BGR format with 8 bits per color component (packed into 32-bit DWORD)
0001b	RGB8P	RGB format with 8 bits per color component (packed into 32-bit DWORD)
0010b	XRGB8	RGB format with 8 bits per color component (unpacked and LSB aligned in 32-bit DWORD)
0011b	RGBX8	RGB format with 8 bits per color component (unpacked and MSB aligned in 32-bit DWORD)

Table continues on the next page...

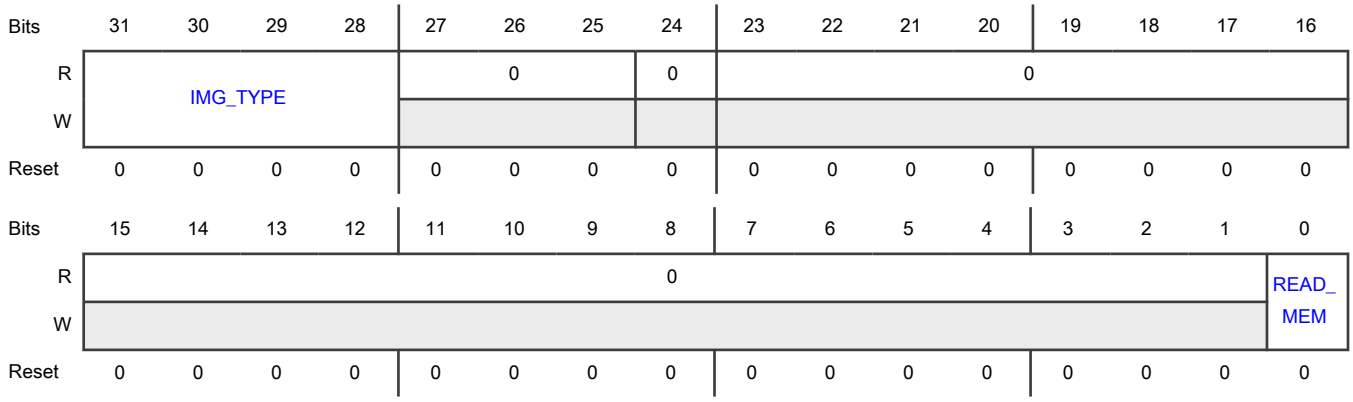
Table 549. Input image formats (continued)

Bit field value	Format	Description
0100b	XBGR8	BGR format with 8 bits per color component (unpacked and LSB aligned in 32-bit DWORD)
0101b	RGB565	RGB format with 5 bits of R, B; 6 bits of G (packed into 32-bit DWORD)
0110b	A2BGR10	BGR format with 2 bits alpha in MSB; 10 bits per color component
0111b	A2RGB10	RGB format with 2 bits alpha in MSB; 10 bits per color component
1000b	YUV444_1P8P	8 bits per color component; 1-plane, YUV interleaved packed bytes
1001b	YUV444_1P10	10 bits per color component; 1-plane, YUV interleaved unpacked bytes (6 LSB waste bits in 16-bit WORD)
1010b	YUV444_1P10P	10 bits per color component; 1-plane, YUV interleaved packed bytes (2 MSB waste bits in 32-bit WORD)
1011b	YUV444_1P12	12 bits per color component; 1-plane, YUV interleaved unpacked bytes (4 LSB waste bits in 16-bit WORD)
1100b	YUV444_1P8	8 bits per color component; 1-plane YUV interleaved unpacked bytes (8 MSB waste bits in 32-bit DWORD)
1101b	YUV422_1P8P	8 bits per color component; 1-plane YUV interleaved packed bytes
1110b	YUV422_1P10	10 bits per color component; 1-plane, YUV interleaved unpacked bytes (6 LSB waste bits in 16-bit WORD)
1111b	YUV422_1P12	12 bits per color component; 1-plane, YUV interleaved packed bytes (4 LSB waste bits in 16-bit WORD)

Offset

Register	Offset
CHNL_MEM_RD_CTRL0	88h

Diagram



Fields

Field	Description
31-28 IMG_TYPE	Input Image Format Specifies the format of the image when an image is read from memory (even during deinterlacing). See all the input image formats in Table 549.
27-25 —	Reserved
24 —	Reserved
23-1 —	Reserved
0 READ_MEM	Initiate Read from Memory Initiates direct read of image from memory when you write 1 to this field. You can read one full image frame at a time. To read another image frame from memory, write 0 and then write 1 to this field. You can write 0 to this field after writing 1 while read from memory is in progress; this will not stop the read operation. The read operation stops when one full frame is read. This field is not used during deinterlacing. 0 - No reads 1 - Reads initiated

57.7.37 Channel 0 RGB or Luma (Y) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_Y0)

Specifies the starting memory address of buffer 2 where the RGB or Y (Luma) data of the processed image is stored.

Offset

Register	Offset
CHNL_OUT_BUF2_ADDR_Y0	8Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ADDR	Starting Address for RGB or Y Specifies the starting address for RGB or Y (luma) memory location.

57.7.38 Channel 0 Chroma (U/Cb/UV/CbCr) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_U0)

Specifies the starting memory address of buffer 2 where the U or Cb data of the processed image is stored. In a two-plane operation, this register points to the starting memory address where the UV or CbCr chroma of the processed image is stored. You do not use this register for a one-plane operation.

Offset

Register	Offset
CHNL_OUT_BUF2_ADDR_U0	90h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 ADDR	Starting Address for U/Cb or 2-plane UV/CbCr Specifies the starting address for the (U or Cb) or two-plane (UV or CbCr) memory location.

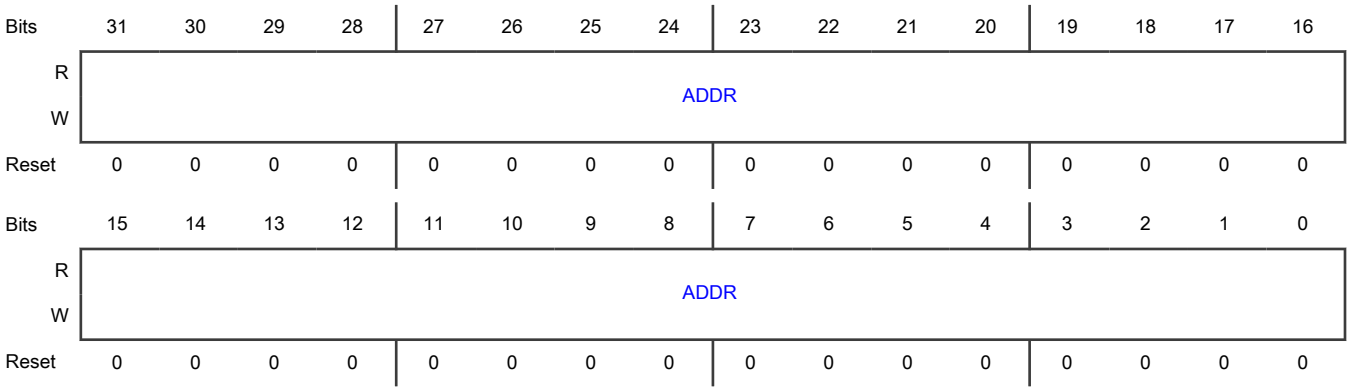
57.7.39 Channel 0 Chroma (V/Cr) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_V0)

Specifies the starting memory address of buffer 2 where the V or Cr data of the processed image is stored. You do not use this register in one- or two-plane operations.

Offset

Register	Offset
CHNL_OUT_BUF2_ADDR_V0	94h

Diagram



Fields

Field	Description
31-0 ADDR	Starting Address for V or Cr Specifies the starting address for the V or Cr memory location.

57.7.40 Channel 0 Scaled Image Configuration (CHNL_SCL_IMG_CFG0)

Provides details related to the image width (in pixels) and height (in lines) at the scaler output. Because the image after scaling can be flipped, cropped, or both; the dimension of the image after scaling becomes an important parameter for the output buffer control logic.

The scaled image dimension is computed as: dimension in Channel 0 Image Configuration (CHNL_IMG_CFG0) divided by the scale factor in Channel 0 Scale Factor (CHNL_SCALE_FACTOR0) and rounded to the closest integer.

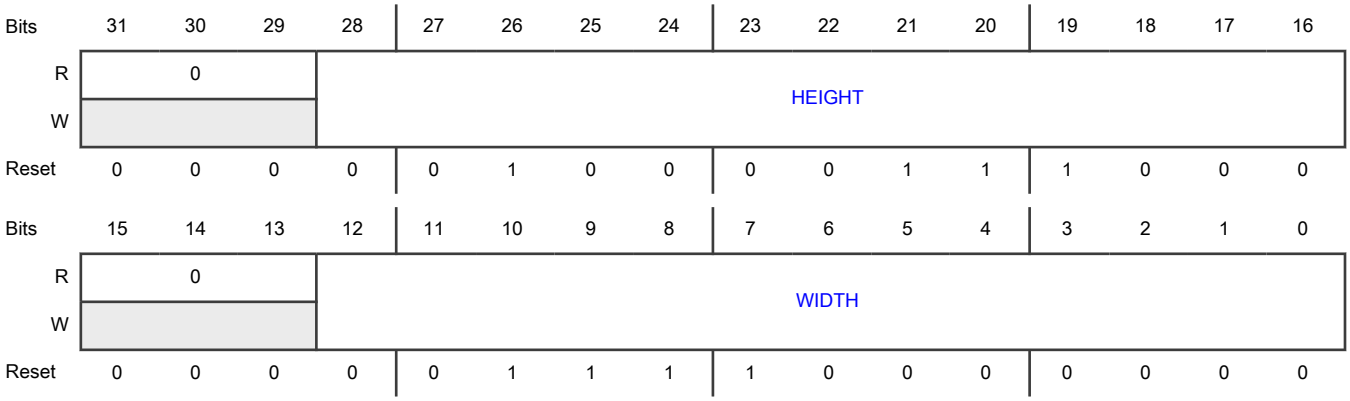
NOTE

You must program this register with the correct image dimensions even if the scaler is bypassed.

Offset

Register	Offset
CHNL_SCL_IMG_CFG0	98h

Diagram



Fields

Field	Description
31-29 —	Reserved
28-16 HEIGHT	Scaled Image Height (Lines) Provides the height of the scaled image in terms of number of lines. Default height is 1080 lines for 1:1 scaling.
15-13 —	Reserved
12-0 WIDTH	Scaled Image Width (Pixels) Provides the width of the scaled image in terms of number of pixels. Default width is 1920 pixels for 1:1 scaling.

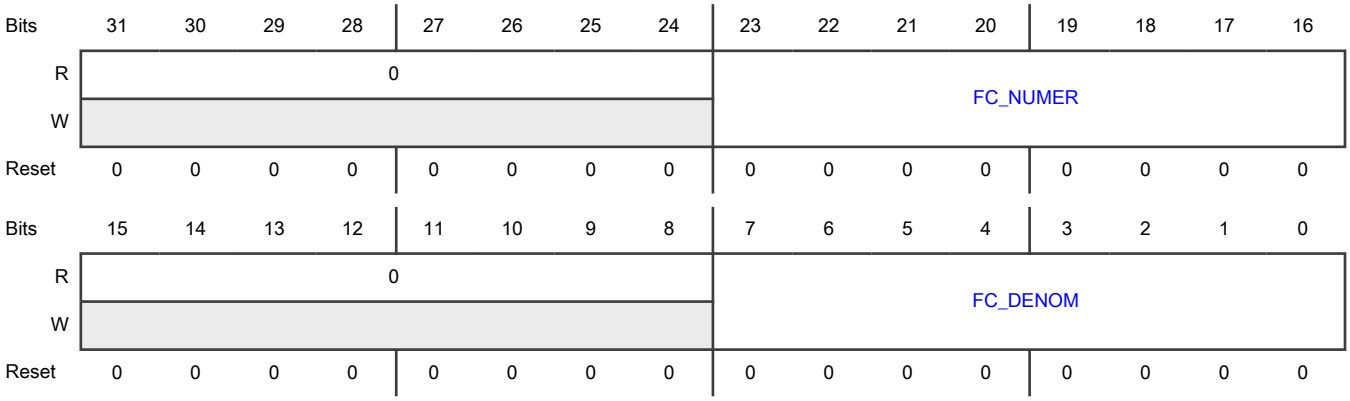
57.7.41 Channel 0 Flow Control (CHNL_FLOW_CTRL0)

Provides the amount of usable AXI bus bandwidth for writing out the pixel data from the channel's output buffers (Y, U, or V). The usable bandwidth is specified as a fraction (numerator/denominator) where denominator >= numerator. This flow control kicks in for the whole frame when the source of frame is memory or for only the last line when the source is a pixel link input and the line buffer inside the scaler is being used (that is, when scaler is enabled or YUV420 data is being input).

Offset

Register	Offset
CHNL_FLOW_CTRL0	9Ch

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 FC_NUMER	Numerator Value of Fraction of Usable Bandwidth Determines the clock cycles out of the FC_DENOM number of clock cycles during which pixels can be written out. FC_NUMER cannot be greater than FC_DENOM. Flow control is disabled if you write 0 to this field. The AXI write request consumes the full available bandwidth if you configure FC_NUMER = FC_DENOM. 0000_0000 - Invalid value (flow control disabled)
15-8 —	Reserved
7-0 FC_DENOM	Denominator Value of Fraction of Usable Bandwidth Determines the total cycles within which the FC_NUMER clock cycles are used to write out the data on the AXI write bus. FC_DENOM cannot be less than FC_NUMER. Flow control is disabled if you write 0 to this field. The AXI write request consumes the full available bandwidth if you configure FC_NUMER = FC_DENOM. 0000_0000 - Invalid value (flow control disabled)

Chapter 58
Parallel Camera Interface

58.1 Overview

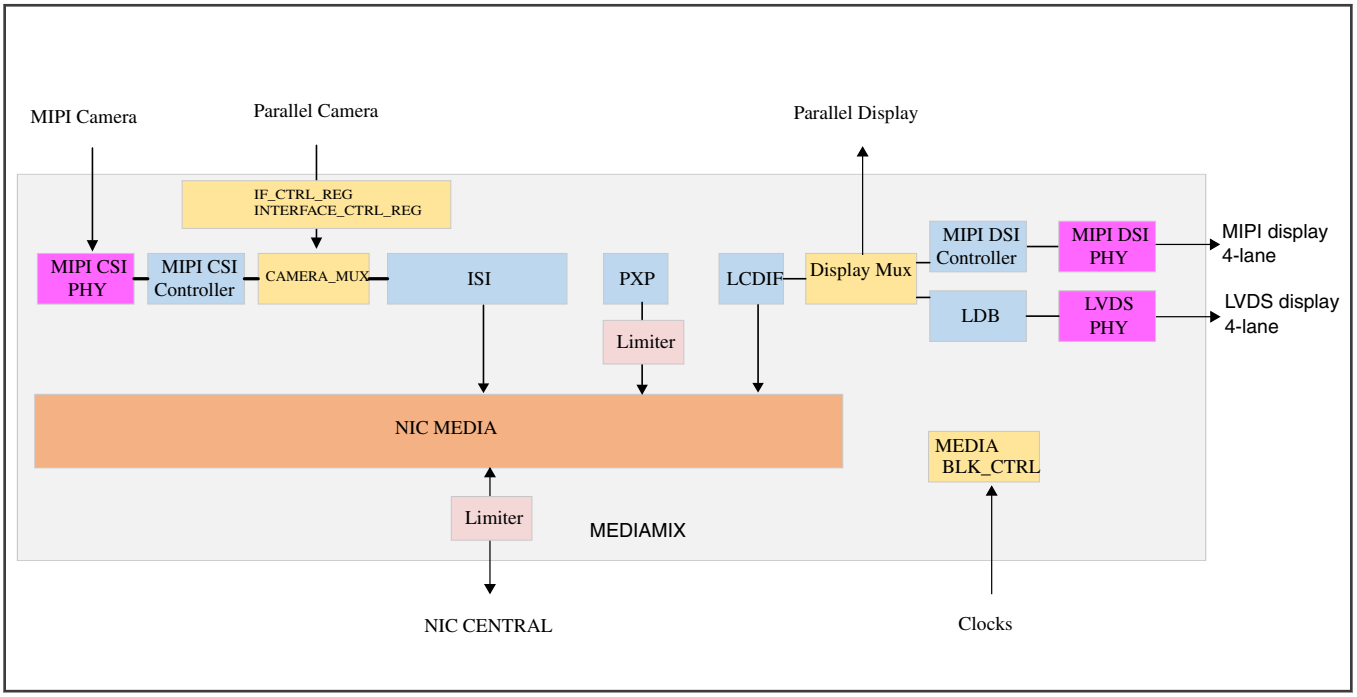
This chapter describes the Parallel Capture Interface. It interfaces the chip to the Parallel CSI sensor.

58.1.1 Features

The parallel camera interface provides the following features.

- Configurable interface logic to support most commonly used parallel CMOS sensors
- Supports up to 80MHz input clock from sensor
- Supports the following input data formats
  - 8-bit/10-bit Camera Sensor Interface (CSI)
  - 8-bit data port for RGB, YCbCr, and YUV data input
  - 8-bit/10-bit data ports for Bayer data input
- Parallel Camera Interface is hooked to the Imaging subsystem via the Pixel Link

58.1.2 Block Diagram



NOTE

For more information on Parallel Camera Interface Register (IF_CTRL_REG), Interface Control Register (INTERFACE_CTRL_REG), Camera Mux and Display Mux please refer Media Mix Domain Block Control.

58.2 Functional Description

This section provides the functional description of the Parallel Camera Interface. The parallel camera interface is capable of forming a standard synchronous link between pixel sources (camera and/or image capture devices or the imaging subsystem) and

a pixel consumer (ISI). The parallel camera interface provides a transport mechanism to deliver pixel data and bus connections to ISI.

Various types of data are supported by the interface and the format selection can be done by the INTERFACE_CTRL_REG and IF_CTRL_REG in the BLK_CTRL_MEDIAMIX. Please see the Media Mix Domain Block Control (BLK_CTRL_MEDIAMIX) for more feature information. The pixel transfer function is configured to support 36 data bits and up to 12 control bits standard pixel link interface. The control signals for this interface originate from the parallel port and are fed along with data at the same clock rate.

Data_type_in (BLK)	Clk gate (BLK)	Pixel_width (BLK)	DEINT (ISI)	YCBCR_MODE (ISI)	CSC_MODE (ISI)	CSC_BYP (ISI)	COEFF (ISI)
0x0 UYVY bt656 8-bit	1	Image width -1	Depend on camera	0x0	0x0	0x0	
0x5 YUV422 YVYU 8-bit	0	Image width -1	Depend on camera	0x1	0x1	0x0	

```
#ifdef __YUV422__
//YCrCb to RGB
#define COEFF0 0x12a | (0x0 << 16)// 1.164 / 0.
#define COEFF1 (0x198) | (0x12a << 16) // 1.596 / 1.164
#define COEFF2 (0x79c) | (0x730 << 16) // -0.392 / -0.813
#define COEFF3 0x12a | (0x204 << 16)// 1.164 / 2.017
#define COEFF4 (0x0) | (0x1f0 << 16)// 0 / -16
#define COEFF5 (0x180) | (0x180 << 16) // -128 / -128
#endif
```

```
//YUV to RGB
#define COEFF0 (0x100   | (0 << 16)    ) // 1.0 / 0
#define COEFF1 ((0x124) | (0x100 << 16)) // 1.14 / 1
#define COEFF2 ((0x79b) | (0x76b << 16)) // -0.395 /- 0.58
#define COEFF3 (0x100   | (0x208 << 16)) // 1 / 2.032
#define COEFF4 ((0x0)   | (0x00 << 16) ) // 0.00 / 0
#define COEFF5 ((0x00)  | (0x00 << 16) ) // 0 / 0
```

1. When doing YUV to RGB, use COEFF of “YCrCb to RGB” and set the CSC_MODE to 1, which means YCrCb to RGB. The YUV mode of parallel camera module is recognized as YCbCr.

```
output_control &= ~(0b1 << 3);
output_control |= (0b0 << 3);  //0=yuv,1=ycbcr
```

NOTE

2. When doing YCrCb to RGB, use COEFF of “YUV to RGB” and set the CSC_MODE to 0, which means YUV to RGB. The YCbCr mode of parallel camera module is recognized as YUV.

```
output_control &= ~(0b1 << 3);
output_control |= (0b1 << 3);  //0=yuv,1=ycbcr
```

58.3 External Signals

Table 550. Signals for Parallel Camera interface

IO PAD	Alt2
GPIO_IO00	PCLK
GPIO_IO01	D[0]
GPIO_IO02	FRAME_VALID
GPIO_IO03	LINE_VALID
GPIO_IO07	D[1]
GPIO_IO08	D[2]
GPIO_IO09	D[3]
GPIO_IO10	D[4]
GPIO_IO11	D[5]

Table continues on the next page...

Table 550. Signals for Parallel Camera interface (continued)

GPIO_IO14	D[6]
GPIO_IO15	D[7]
GPIO_IO17	D[8]
GPIO_IO18	D[9]

Chapter 59
LVDS Display Bridge and Pixel Mapper

59.1 Chip-specific LVDS information

Table 551. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The chip includes one instance of LVDS in MEDIAMIX.

This module does not have any programmable registers. Please refer BLK_CTRL_MEDIAMIX for LVDS and LDB configuration parameters described in LVDS Control and LDB Control registers.

59.2 Overview

The LVDS Display Bridge (LDB) connects to an External LVDS Display Interface.

The purpose of the LDB is to support flow of synchronous RGB data to external display devices through the LVDS interface.

59.2.1 Block Diagram

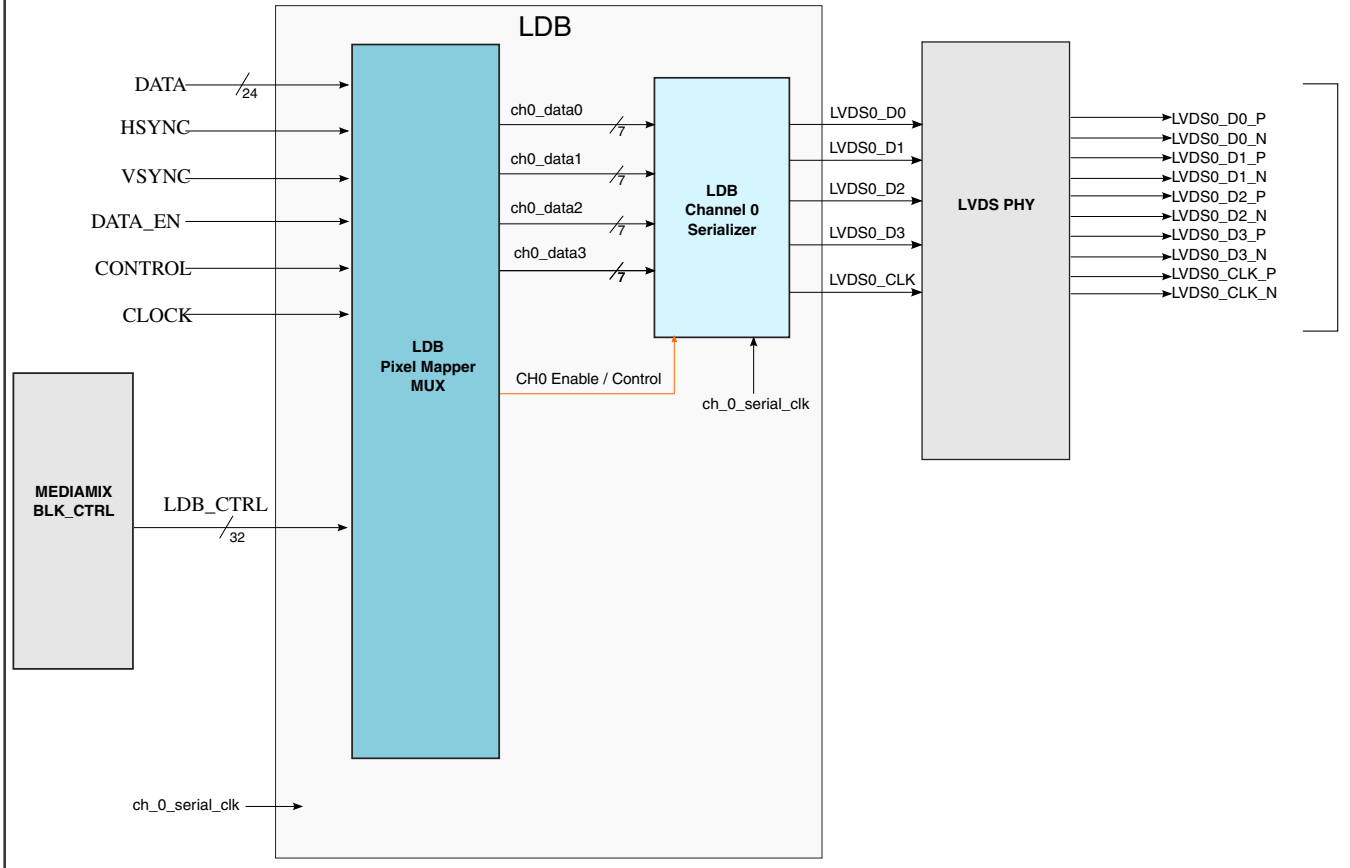


Figure 354. LDB Block Diagram

59.2.2 Features

LVDS Display Bridge block provides the following functionalities:

- Connectivity to relevant devices - Displays with LVDS receivers.
- Arranging the data as required by the external display receiver and by LVDS display standards.
- Synchronization and control capabilities.

59.3 Functional Description

The following sections describe functional details of the LDB module.

Table 552. LDB - Block Description

Block	Description
LDB Mux and Control	Gets control signals from SoC and determines parameters of LDB, mapping input data and serializer clock to channel serializers
Channel Serializers	LDB has 1 channel serializer. The serializer does parallel to serial conversion 7:1 to 4 data lines

59.3.1 Bit Mapping

LDB supports two mapping standards:

- SPWG mapping
- JEIDA mapping

Table 553. SPWG/PSWG/VESA 24 bpp Data Mapping

Serializer input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
data0	G0	R5	R4	R3	R2	R1	R0
data1	B1	B0	G5	G4	G3	G2	G1
data2	DE	VSYNC	HSYNC	B5	B4	B3	B2
data3	CTL	B7	B6	G7	G6	R7	R6

Table 554. JEIDA 24bpp Data Mapping

Serializer input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
data0	G2	R7	R6	R5	R4	R3	R2
data1	B3	B2	G7	G6	G5	G4	G3
data2	DE	VSYNC	HSYNC	B7	B6	B5	B4
data3	CTL	B1	B0	G1	G0	R1	R0

NOTE

Several options of control usage can be available. some display devices use only DE, some others use all 3 controls, some use only HSYNC, VSYNC. "CTL" is an optional general purpose control. It is always 0 for this SoC.

59.3.2 Input Parallel Display Ports

One RGB input port is supported.

Only synchronous access mode is supported. Each RGB data interface contains the following:

- Pixel value(24-bit data)
- Pixel clock
- Control signals: HSYNC, VSYNC, DE, CTL.

Total of up to 28 bits per data interface are transferred per pixel clock cycle.

59.3.3 Output LVDS Ports

There is 1 LVDS channel. The output is used to communicate RGB data and control external LCD displays.

The output LVDS port complies to the EIA-644-A standard.

59.3.4 Processing

LDB data processing stages are as follows:

- Receive input data from parallel input interface: 24 RGB data + up to 4 controls and map them to LVDS channels.
- Re-arrange the input data according to channel configuration, and muxing scheme.
- Serialize the 28 bit input bus (per channel) on 4 output serial data lines (7:1)

59.3.5 Relevant Standards

Below are the relevant standards to LDB:

1. ANSI EIA-644-A. Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits.
2. SPWG Notebook Panel Specification (V3.8 from 03/2007)
3. PSWG standards (Panel Standardization Working Group) - set of standards for panels using LVDS.
4. DISM Standard JEIDA-59-1999

59.3.6 Clocks

A table with the LDB Clock Sources can be found here.

Table 555. LDB Clocks

Clock name	Description	Comments
DI0_CLK	DI0 interface pixel clock	
CH0_SERIAL_CLK	CH0 interface serializer clock	This is x 7 the rate of the DI_CLK

59.3.7 External Signals

The following table describes the external signals for LVDS PHY:

Table 556. External Signals

Signal	Description	Direction
CLK_N	Negative Clock Signal	O
CLK_P	Positive Clock Signal	O
DATA[3:0]_N	Negative Data Signals	O
DATA[3:0]_P	Positive Data Signals	O

Chapter 60
Low Power Inter-Integrated Circuit (LPI2C)

60.1 Chip-specific LPI2C information

Table 557. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing

The chip includes eight instances of LPI2C Modules. LPI2C1 and LPI2C2 are in AONMIX. LPI2C3-8 in WAKEUPMIX. SPI1 and SPI2 are only supported in Master Mode, not Slave Mode.

The LPI2C feature of a bridge mode (separate SCL/SDA for I2C master and SCLS/SDAS for I2C slave on the one LPI2C instance) is not supported. The 4-pin unidirectional mode (separate SCL_IN, SCL_OUT, SDA_IN, SDA_OUT) is not supported.

NOTE

HREQ host request feature to allow HW triggers to initiate the start of an I2C request is not supported.

NOTE

Peripheral triggers are not supported on this chip. References to LPI2C triggers in Peripheral triggers do not apply.

60.2 Overview

LPI2C supports an efficient interface to an I2C bus as a controller and target:

- Implements logic support for Standard, Fast, Fast+, HS-mode (target only) and Ultra-Fast modes of operation
- Uses little CPU overhead, with DMA offloading of FIFO register accesses
- Continues operating in Stop modes if an appropriate clock is available

LPI2C also complies with the System Management Bus (SMBus) Specification, version 3. The SMBus is a single-ended simple two-wire bus, which is typically used for low-bandwidth communications.

The Inter-Integrated Circuit (I^2C) serial bus is multi-controller, multi-target, packet-switched, and single-ended, and is often used to attach microcontroller ICs to lower-speed peripheral ICs.

NOTE

Terminology in this chapter has been updated to align with I^2C-bus specification, Rev. 7.0, as shown in Table 558.

Table 558. Updated terms

Updated term	Deprecated term
Controller	Master
Target	Slave

60.2.1 Block diagram

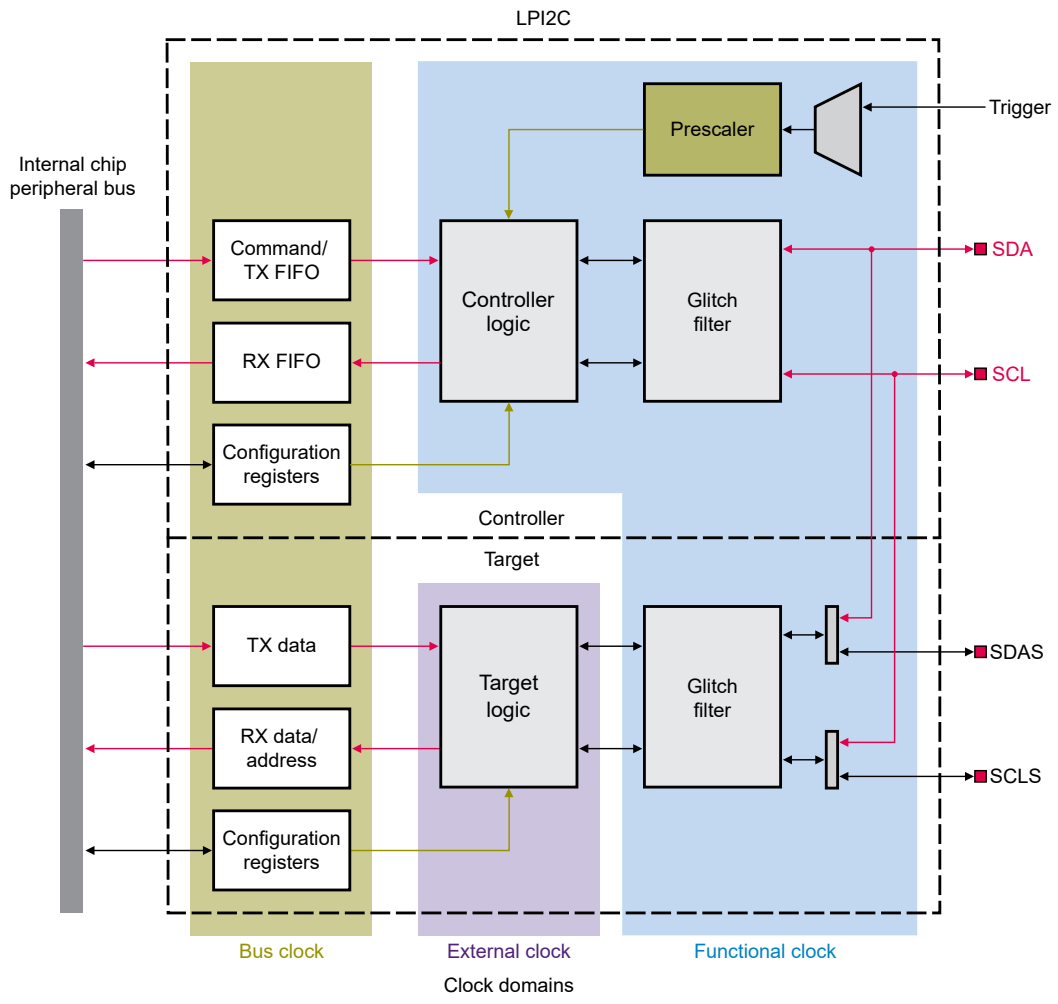


Figure 355. Block diagram

60.2.2 Features

LPI2C supports:

- Standard, Fast, Fast+, HS (target only), and Ultra Fast modes
- Multicontroller, including synchronization and arbitration, means that any number of controller nodes can be present. Also, controller and target roles can be changed between messages (after a Stop signal is sent).
- Clock stretching. Used on the SCL line, as an I2C flow control mechanism.
- Arbitration for when the system has more than one controller. When used on the SDA line, ensures that there is only one I2C transmitter at a time.
- General call, seven-bit addressing, and ten-bit addressing
- Software reset, Start byte, and device ID (also require software support)

The LPI2C controller supports:

- Command and transmit FIFO of 8 words (8-bit transmit data + 3-bit command)
- Receive FIFO of 8 words (8-bit receive data).

- Command FIFO waiting for an I2C idle bus before initiating a transfer
- Initiation of repeated Start and Stop conditions and one or more controller-receiver transfers by command FIFO
- Stop condition generation from command FIFO, or automatic generation of Stop condition when the transmit FIFO is empty
- Interrupt generation on data match and unwanted data rejection, via flexible receive data match
- Flags and optional interrupt signals at repeated Start condition, Stop condition, loss of arbitration, unexpected NACK, and command word errors
- Configurable bus idle timeout and pin-stuck-low timeout

The LPI2C target supports:

- Separate I2C target registers to minimize software overhead because of controller or target switching
- 7-bit or 10-bit addressing, address range, SMBus alert, and general call address.
- Transmit data register that supports interrupt or DMA requests
- Receive data register that supports interrupt or DMA requests
- Software-controllable ACK or NACK, with optional clock stretching on ACK or NACK field
- Configurable clock stretching, to avoid transmit-FIFO-underrun and receive-FIFO-overrun errors
- Flags and optional interrupt at end of packet, Stop condition, or bit error detection

60.3 Functional description

60.3.1 Controller mode

The LPI2C controller logic operates independently from the target logic to perform all controller-mode transfers on the I2C bus.

60.3.1.1 Transmit and Command FIFO commands

The transmit FIFO stores command data to initiate various I2C operations. The following operations can be initiated through commands in the transmit FIFO:

- Start or repeated Start condition with address byte, expecting ACK or NACK.
- Transmit data. This operation is the default for zero-extended-byte writes to the transmit FIFO.
- Receive 1-256 bytes of data. You can configure this operation to discard received data and not to store it in the receive FIFO.
- Stop condition. You can configure this operation to send a Stop condition when the transmit FIFO is empty.

Multiple transmit and receive commands can be inserted between the Start and Stop conditions. To comply with the I2C specification, transmit and receive commands must not be interleaved. The receive data command and the receive data and discard commands can be interleaved. This interleaving ensures that only the desired received data is stored in the receive FIFO (or compared with the data match logic).

The LPI2C controller automatically transmits a NACK on the last byte of a receive data command. It transmits the NACK unless the next command in the FIFO is also a receive data command. If the transmit FIFO is empty when a receive data command completes, a NACK is also automatically transmitted.

The LPI2C controller supports 10-bit addressing via a (repeated) Start condition, followed by a transmit data byte containing the second address byte, followed by any number of data bytes with the controller transmit data.

A Start or repeated Start condition expecting a NACK (for example, HS mode controller code) must be followed by a Stop or (repeated) Start condition.

60.3.1.2 Controller operations

When LPI2C is enabled, it monitors the I2C bus to detect when the I2C is idle (MSR[BBF]). If either SCL or SDA are low, the I2C bus is no longer considered idle. The bus becomes idle if a Stop condition is detected or if a bus idle timeout is detected (as configured by MCFGR2[BUSIDLE]). After the bus is idle, if the transmit FIFO is not empty and the host request is asserted or disabled, the LPI2C controller initiates a transfer on the bus. This transfer involves the following steps:

1. Wait the bus idle time equal to (MCCR0[CLKLO] + 1) multiplied by the prescaler (MCFGR1[PRESCALE]).
2. Transmit a Start condition and address byte using the timing configuration in Controller Clock Configuration 0 (MCCR0). If an HS mode transfer is configured, the timing configuration from Controller Clock Configuration 1 (MCCR1) is used instead.
3. Perform controller transmit or controller receive transfers, as configured by the transmit FIFO.
4. Transmit NACK on the last byte of a controller receive transfer. This action is performed unless the next command in the transmit FIFO is also a receive data command and the transmit FIFO is not empty.
5. Transmit a repeated Start or Stop condition as configured by the transmit FIFO or MCFGR1[AUTOSTOP]. A repeated Start can change which timing configuration register is used.

When MCFGR0[RELAX] is 1, the LPI2C controller does not wait for the I2C bus to be idle before starting a transfer. It also relaxes some restrictions on the order of FIFO commands. For example, it allows a Stop command when it is idle to generate a Start condition, followed by one SCL clock pulse, and then a Stop condition.

When the LPI2C controller is disabled, LPI2C continues emptying the transmit FIFO until a Stop condition is transmitted. (The controller could be disabled due to MCR[MEN] being 0, or automatically due to mode entry.) However, LPI2C no longer stalls the I2C bus by waiting for the transmit or receive FIFO. After the transmit FIFO is empty, LPI2C generates a Stop condition automatically.

The LPI2C controller can stall the I2C bus under certain conditions. This stalling results in SCL pulled low continuously on the first bit of a byte, until these conditions change:

- The LPI2C controller is enabled and busy, the transmit FIFO is empty, and MCFGR1[AUTOSTOP] is 0. The LPI2C controller continues to stall the bus until the transmit FIFO is loaded with more data.
- The LPI2C controller is enabled and receiving data, receive data is not being discarded (due to command or receive data match), and the receive FIFO is full. The LPI2C controller continues to stall the I2C bus until the receive FIFO is emptied.

The LPI2C controller aborts the existing transfer when MCFGR0[ABORT] becomes 1. This action causes the LPI2C controller to complete the existing word and then transmit a Stop condition. If the receive FIFO is full when the receive operation is aborted, the last received data is discarded. A new transfer does not start while MCFGR0[ABORT] is 1.

60.3.1.3 Receive FIFO and data matching

The receive FIFO stores receive data during controller-receiver transfers. You can configure the LPI2C controller to discard received data instead of storing it in the receive FIFO. This option is configured via the command word in the transmit FIFO.

Received data supports a receive data match function that can match received data against one of two bytes, or against a masked data byte. You can configure the data match function to compare only the first one or two data words received since the last (repeated) Start condition. Received data that is already discarded due to the command word cannot cause a data match. It delays the match on the first data word received until after the discarded data is received.

You can configure the receiver match function to discard all received data until a data match is detected, using MCFGR0[RDMO]. Following a data match, write 0 to MCFGR0[RDMO] before writing 0 to MSR[DMF] to allow all subsequent data to be received.

60.3.1.4 Timing parameters

The LPI2C controller can configure the following timing parameters. Parameters are configured separately for HS mode (Controller Clock Configuration 1 (MCCR1)) and other modes (Controller Clock Configuration 0 (MCCR0)). This separation allows the HS mode controller code to be sent using regular timing parameters. Then it allows a switch to HS mode timing (following a repeated Start) until the next STOP condition.

Configure the LPI2C controller timing parameters, measured in LPI2C functional clock cycles, as shown in Table 559. You must configure these parameters to meet the I2C timing specification for the required mode.

Table 559. Timing parameters

I2C specification timing parameter	I2C specification timing symbol	LPI2C timing parameter (in LPI2C functional clock cycles)
SCL clock period	tSCL	(CLKHI + CLKLO + 2 + SCL_LATENCY) × (2 ^ PRESCALE)
Hold time (repeated) Start condition	tHD:STA	(SETHOLD +1) × (2 ^ PRESCALE)
Low period of the SCL clock	tLOW	(CLKLO + 1) × (2 ^ PRESCALE)
High period of the SCL clock	tHIGH	(CLKHI + 1 + SCL_LATENCY) × (2 ^ PRESCALE)
Setup time for a repeated Start condition or Stop condition	tSU:STA, tSU:STO	(SETHOLD + 1 + SCL_LATENCY) × (2 ^ PRESCALE)
Data hold time	tHD:DAT	(DATAVD + 1) × (2 ^ PRESCALE)
Data setup time	tSU:DAT	(SDA_LATENCY + 1) × (2 ^ PRESCALE)
Bus free time between a Stop and Start condition	tBUF	(CLKLO + 1 + SDA_LATENCY) × (2 ^ PRESCALE)
Data valid time, data valid acknowledge time	tVD:DAT, tVD:ACK	(DATAVD + 1) × (2 ^ PRESCALE)

Table 560 defines the latency parameters. These parameters assume that the rise time is less than one LPI2C functional clock cycle. The rise time depends on a number of factors, including the I/O propagation delay, the I2C bus loading, and the external pullup resistor sizing. A larger rise time increases the number of cycles that the signal takes to propagate through the synchronizer (and glitch filter), which increases the latency.

Table 560. Synchronization latency

Timing parameter	Timing definition
SCL_LATENCY	ROUNDDOWN ((2 + FILTSCL + SCL_RISETIME) ÷ (2 ^ PRESCALE))
SDA_LATENCY	ROUNDDOWN ((2 + FILTSDA + SDA_RISETIME) ÷ (2 ^ PRESCALE))

The following timing restrictions must be enforced to avoid unexpected Start or Stop conditions on the I2C bus. These restrictions also avoid unexpected Start or Stop conditions detected by the LPI2C controller. The timing restrictions can be summarized as "SDA cannot change when SCL is high outside a transmitted (repeated) Start or Stop condition."

Table 561. LPI2C timing parameter restrictions

Timing parameter	Minimum	Maximum	Comment
CLKLO	03h	—	CLKLO x (2 ^ PRESCALE) > SCL_LATENCY
CLKHI	01h	—	Configure CLKHI to meet the duty cycle requirements in the I2C specification
SETHOLD	02h	—	SETHOLD × (2 ^ PRESCALE) > SDA_LATENCY

Table continues on the next page...

Table 561. LPI2C timing parameter restrictions (continued)

Timing parameter	Minimum	Maximum	Comment
DATAVD	01h	CLKLO – SDA_LATENCY – 1	Configure DATAVD to meet the data hold requirement in the I2C specification
FILTSCL	00h	[CLKLO × (2 ^ PRESCALE)] – 3	FILTSCL and FILTSDA are the only parameters not multiplied by (2 ^ PRESCALE)
FILTSDA	FILTSCL	[CLKLO × (2 ^ PRESCALE)] – 3	Configuring FILTSDA greater than FILTSCL can delay the SDA input to compensate for board level skew
BUSIDLE	(CLKLO + SETHOLD + 2) × 2	—	Must also be greater than (CLKHI + 1)

See the UM10204, *I2C-bus specification and user manual*.

See Application information for example LPI2C timing configurations.

60.3.1.5 Error conditions

The LPI2C controller monitors errors while it is active. The following conditions generate an error flag and block a new Start condition from being sent, until the flag is cleared by software:

- A Start or Stop condition is detected and is not generated by the LPI2C controller (MSR[ALF] becomes 1).
- Transmitting data on SDA and different values are received (MSR[ALF] becomes 1). LPI2C controller stops driving SDA immediately, but continues to drive SCL for the remainder of the byte.
- NACK is detected when transmitting data, and MCFGR1[IGNACK] is 0 (MSR[NDF] becomes 1).
- NACK is detected and is expecting ACK for the address byte, and MCFGR1[IGNACK] is 0 (MSR[NDF] becomes 1).
- ACK is detected and is expecting NACK for the address byte, and MCFGR1[IGNACK] is 0 (MSR[NDF] becomes 1).
- Transmit FIFO is requesting to transmit or receive data without a Start condition (MSR[FEF] becomes 1). This restriction is ignored when MCFGR0[RELAX] is 1.
- SCL (or SDA if MCFGR1[TIMECFG] is 1) is low for (MCFGR2[TIMELOW] × 256) prescaler cycles without a pin transition (MSR[PLTF] becomes 1).

You must respond to MSR[PLTF] to terminate the existing command. You can terminate the command cleanly by writing 0 to MCR[MEN], or you can terminate it abruptly by writing 1 to MCR[RST].

You can use MCFGR0[RELAX] to attempt to recover a target with SDA stuck low. If MCFGR0[RELAX] is 1, the LPI2C controller does not wait for the I2C bus to be idle before starting a transfer. Initiating a Start command with address FFh, then the Stop condition, should generate sufficient SCL clock edges to cause the target to release SDA.

You can use MCFGR2[BUSIDLE] to force the I2C bus to be considered idle when SCL and SDA remain high for (BUSIDLE + 1) prescaler cycles. The bus is considered idle when the LPI2C controller is first enabled. When BUSIDLE is configured greater than zero, then SCL or SDA must be high for (BUSIDLE + 1) prescaler cycles before the I2C bus is considered idle.

60.3.1.6 DMA support

For efficient DMA transfers to the transmit and command FIFO, two alias regions support incrementing 8-bit, 16-bit, or 32-bit write accesses to that FIFO.

- Controller Transmit Command Burst (MTCBR0 - MTCBR127) is a 512-byte region that supports pushing CMD and DATA into the transmit FIFO.

- Transmit Data Burst (MTDBR0 - MTDBR255) is a 1024-byte region that supports pushing zero extended DATA into the transmit FIFO.

The two regions are contiguous. A DMA transfer can start in the MTCBR region to initialize the transfer including START condition with address. The same transfer can complete in the MTDBR region with the data to transmit without changing the transfer size.

The transmit FIFO prevents writes that overflow the FIFO, but this prevention does not signal an error. Do not perform 32-bit writes to the MTDBR unless there are four empty slots in the transmit FIFO. Do not perform 16-bit writes to the MTDBR unless there are two empty slots in the transmit FIFO.

60.3.1.7 Pin configuration

Configuration	Description
Open-drain support	The LPI2C controller defaults to open-drain configuration of the SDA and SCL pins. Support for true open drain depends on the specific device, and requires the pins where LPI2C pins are muxed to support true open drain.
HS mode support	Support for HS mode depends on the specific device. This mode requires the SCL pin to support the current source pullup required in the I2C specification.
Ultra-Fast mode support	The LPI2C controller supports the output-only push-pull function required for I2C Ultra-Fast mode using the SDA and SCL pins. Support for Ultra-Fast mode also requires MCFGR1[IGNACK] to be 1.
Push-pull two-wire support	A push-pull two-wire configuration is available to the LPI2C controller. If LPI2C is the only controller and all I2C pins on the bus are at the same voltage, this configuration may support a partial HS mode. A partial HS mode, not a full HS mode, because this configuration actively drives high rather than enabling a current service pull-up. This configuration sets the SCL pin as push-pull for every clock except the first clock pulse, to allow HS-mode-compatible targets to perform clock stretching. In this mode, the SDA pin is tristated for controller-receive data bits and controller-transmit ACK/NACK bits, and is configured as push-pull at other times. To avoid the risk of contention when SDA is push-pull, configure the pin for open-drain operation, as part of the device-specific configuration.
Push-pull four-wire support	The push-pull four-wire configuration separates the SCL input data and output data into separate pins. It also separates the SDA input data and output data into separate pins. The SCL/SDA pins are used for input data; the SCLS/SDAS pins are used for output data, with configurable polarity. This configuration simplifies external connections when connecting the LPI2C to the I2C bus through external level shifters or discrete components. When using this four-wire configuration, the LPI2C controller logic and LPI2C target logic cannot connect to separate I2C buses.

60.3.2 Target mode

To perform all target mode transfers on the I2C bus, the LPI2C target logic operates independently from the LPI2C controller logic.

60.3.2.1 Address matching

You can configure the LPI2C target:

- To match one of two addresses, using either 7-bit or 10-bit addressing modes for each address.
- To match a range of addresses in either 7-bit or 10-bit addressing modes.
- To match the general call address and generate appropriate flags.
- To match the SMBus alert address and generate appropriate flags.
- To detect the HS mode controller code address, and to disable the digital filters and output valid delay time until the next Stop condition is detected.

After a valid address is matched, the LPI2C target automatically performs target-transmit or target-receive transfers until:

- A NACK is detected (unless SCFGR1[IGNACK] becomes 1).
- A bit error is detected (the LPI2C target is driving SDA, but a different value is sampled).
- A (repeated) Start or Stop condition is detected.

When SCFGR1[RXALL] is 1, the LPI2C target receives all addresses and data on the I2C bus. Unless the address is matched, it never drives the SDA pin. Configure the LPI2C target to match only on 7-bit address mode when this feature is enabled. Software can support 10-bit address mode. This option is intended for debugging the contents of an I2C transfer between another controller and target. Likewise, SCFGR1[SDCFG] and SCFGR1[RSCFG] configure the Stop or repeated Start flags to assert on all Stop or repeated Start conditions, even when there is no address match.

60.3.2.2 Transmit and receive data

Target Transmit Data (STDR) and Target Receive Data (SRDR) are double-buffered and only update during a target-transmit and target-receive transfer, respectively.

You can configure the target address that was received to be read from SRDR (for example, when using DMA to transfer data) or from Target Address Status (SASR).

You can configure STDR to request data only after a target-transmit transfer is detected. You can also configure it to request new data whenever STDR is empty.

Write to STDR only when SSR[TDF] is set.

Read SRDR only when SSR[RDF] is set, or when SSR[AVF] is set and SCFGR1[RXCFG] = 1.

Read SASR only when SSR[AVF] is set.

60.3.2.3 Read request

The LPI2C target generates a read request when the I2C bus is idle and you write 1 to SCFGR0[RDREQ]. This occurrence causes the LPI2C target to pull the SDA pin low until either the first SCL falling edge or SCFGR0[RDREQ] becomes 0. Following the next Stop condition or a software timeout, you can proceed as follows:

- If the LPI2C target is accessed at the correct address, the read request is acknowledged, and you must write 0 to SCFGR0[RDREQ].
- If SCFGR0[RDACK] is 1, the read request is acknowledged by a Start followed by one SCL pulse followed by a STOP condition. You must write 0 to SCFGR0[RDREQ].
- If neither of the first two cases occurred, the read request is not acknowledged. You must write 0 and then 1 to SCFGR0[RDREQ] after an appropriate delay, provided the I2C bus is idle.
- If a software timeout occurs, the read request is not acknowledged, and you should write 0 to SCFGR0[RDREQ]. Another request can be generated after an appropriate delay.

Writing to SCFGR0[RDREQ] when the I2C bus is busy results in a logic 0 being written. Write 1 to SCFGR0[RDREQ], then confirm that the register is written to correctly. If the register does not update correctly, the I2C bus is no longer idle and the read request must be attempted later.

60.3.2.4 Clock stretching

The LPI2C target supports many configurable options for clock stretching. You can configure these conditions to perform clock stretching:

- SSR[AVF] is set during the ninth clock pulse of the address byte.
- SSR[TDF] is set during the ninth clock pulse of a target-transmit transfer.
- SSR[RDF] is set during the ninth clock pulse of a target-receive transfer.
- SSR[TAF] is set during the eighth clock pulse of an address byte or a target-receive transfer. In HS mode, this option is disabled.

- Clock stretching can be extended for a number of cycles equal to the value of SCFGR2[CLKHOLD] cycles. This stretching allows additional setup time to sample the SDA pin externally. In HS mode, this option is disabled.

When clock stretching is enabled, clock stretching extends for one peripheral bus clock cycle after SDA updates, unless extended by the SCFGR2[CLKHOLD] configuration.

60.3.2.5 Timing parameters

The LPI2C target can configure the following timing parameters:

- SDA data valid time from SCL negation to SDA update
- SCL hold time when clock stretching is enabled to increase setup time when sampling SDA externally
- SCL glitch filter time
- SDA glitch filter time

These parameters are disabled when SCR[FILTEN] is 0, when SCR[FILTDZ] is 1 in Doze mode, and when LPI2C target detects HS mode. When disabled, the LPI2C target is clocked directly from the I2C bus. In this case, the target may not satisfy all timing requirements of the I2C specification (such as SDA minimum hold time in Standard/Fast mode).

The LPI2C target places the following restrictions on the timing parameters:

- You must configure SCFGR2[FILTSDA] to be greater than or equal to SCFGR2[FILTSCL] (unless compensating for board level skew between SDA and SCL).
- You must configure SCFGR2[DATAVD] to be less than the minimum SCL low period.

60.3.2.6 Error conditions

The LPI2C target can flag the following error conditions:

- SSR[BEF] is set when the LPI2C target is driving SDA but it samples a different value than what is expected.
- SSR[FEF] is set due to a transmit data underrun or a receive data overrun. To eliminate the possibility of underrun and overrun, enable clock stretching. When SCFGR1[RXNACK] is 1, the LPI2C target generates a NACK on a receive data overrun.
- SSR[FEF] is also set due to an address overrun, but only when SCFGR1[RXCFG] is 1. To eliminate the possibility of overrun, enable clock stretching. When SCFGR1[RXNACK] is 1, the LPI2C target generates a NACK on an address overrun regardless of the value of SCFGR1[RXCFG].

The LPI2C target does not implement a timeout due to SCL or SDA being stuck low. If this detection is required, use the LPI2C controller logic so you can reset the LPI2C target when this condition is detected.

60.3.3 Low-power modes

LPI2C remains functional during low-power modes, if MCR[DOZEN] = 0 and LPI2C uses an external or internal clock source that remains enabled. LPI2C can generate an interrupt or DMA request to cause a wake-up from low-power modes.

You can configure LPI2C to be disabled in low-power modes when MCR[DOZEN] = 1. In this case, LPI2C waits for the current transfer to complete any pending operation.

NOTE

See the chip-specific information for low-power modes available on your chip.

60.3.4 Debug mode

Table 562. Debug mode

Mode	LPI2C operation
Debug	If MCR[DBGEN] = 1, can continue operating in Debug mode.

60.3.5 Peripheral triggers

The connection of the LPI2C peripheral triggers to other peripherals depends upon the specific device being used.

Table 563. LPI2C triggers

Trigger	Description
Controller output trigger	Generates an output trigger that can be connected to other peripherals on the device. The controller output trigger asserts on either a repeated Start or a Stop condition. The trigger remains asserted for one cycle of the LPI2C functional clock divided by MCFGR1[PRESCALE].
Target output trigger	Generates an output trigger that can be connected to other peripherals on the device. The target output trigger asserts on either a repeated Start or a Stop condition that occurs after a target address match. The target output trigger remains asserted until the next target SCL pin negation.
Input trigger	Controls the start of a LPI2C bus transfer. The input trigger is synchronized. To be detected, the input trigger must assert for at least two cycles of the LPI2C functional clock divided by the value of MCFGR1[PRESCALE].

60.3.6 Clocking

Table 564. LPI2C clocks

Clock	Description
LPI2C functional clock	The LPI2C functional clock is asynchronous to the bus clock. It can remain enabled in low-power modes to support I2C bus transfers by the LPI2C controller. The functional clock is also used by the LPI2C target to support digital filter and data hold time configurations. The LPI2C controller divides the functional clock by a prescaler (MCFGR1[PRESCALE]) and the resulting frequency must be at least eight times faster than the I2C bus bandwidth.
External clock	The LPI2C target logic is clocked directly from the external pins. These pins are SCL and SDA, or SCLS and SDAS if the controller and target are implemented on separate pins). This clocking allows the LPI2C target to remain operational, even when the LPI2C functional clock is disabled. **NOTE** If the LPI2C functional clock is disabled, the LPI2C target digital filter must be disabled. This condition can affect compliance with some timing parameters of the I2C specification, such as data hold time.
Bus clock	The bus clock is only used for bus accesses to the control and configuration registers. The bus clock frequency must be sufficient to support the data bandwidth requirements of the LPI2C controller and target registers.

For chip-specific clocking information, see the Clocking chapter.

60.3.7 Reset

Table 565. LPI2C resets

Reset	Description
Chip reset	The logic and registers for the LPI2C controller and target are reset to their default states after a chip reset.
Software reset	The LPI2C controller implements a software reset field in its control register. MCR[RST] resets all controller logic and registers to their default states, except for Controller Control (MCR) itself. The LPI2C target implements a software reset field in its control register. SCR[RST] resets all target logic and registers to their default states, except for Target Control (SCR) itself.
FIFO reset	The LPI2C controller implements write-only control fields that reset the transmit FIFO (MCR[RTF]) and receive FIFO (MCR[RRF]). After a FIFO is reset, that FIFO is empty. The LPI2C target implements write-only control fields that reset the transmit data register (SCR[RTF]) and receive data register (SCR[RRF]). After a data register is reset, that data register is empty.

60.3.8 Interrupts and DMA requests

Depending on the configuration, interrupts and DMA requests can be combined:

- LPI2C controller and target interrupts
- LPI2C controller and target transmit DMA requests
- LPI2C controller and target receive DMA requests

60.3.8.1 Controller mode

Table 566 lists the Controller mode sources that can generate LPI2C controller interrupts and LPI2C controller transmit and receive DMA requests.

Table 566. Controller interrupts and DMA requests

Status flag	Description	Can generate		
		Interrupt?	DMA request?	Low-power wake-up?
Transmit Data Flag (MSR[TDF])	Data can be written to transmit FIFO, as configured by MFCR[TXWATER].	Y	TX	Y
Receive Data Flag (MSR[RDF])	Data can be read from the receive FIFO, as configured by MFCR[RXWATER].	Y	RX	Y
End Packet Flag (MSR[EPF])	Controller has transmitted a repeated Start or Stop condition.	Y	N	Y
Stop Detect Flag (MSR[SDF])	Controller has transmitted a Stop condition (and the LPI2C controller is idle, if MCFGR1[STOPCFG] = 1) .	Y	N	Y
NACK Detect Flag (MSR[NDF])	During an address byte, the controller expects an ACK but detects a NACK. During an address byte, the controller expects a NACK but detects an ACK.	Y	N	Y

Table continues on the next page...

Table 566. Controller interrupts and DMA requests (continued)

Status flag	Description	Can generate		
		Interrupt?	DMA request?	Low-power wake-up?
	During a controller-transmitter data byte, the controller detects a NACK.			
Arbitration Lost Flag (MSR[ALF])	The controller lost arbitration due to a Start or Stop condition detected at the wrong time, or the controller was transmitting data but received data different from the data that was transmitted.	Y	N	Y
FIFO Error Flag (MSR[FEF])	The controller expects a Start condition in the command FIFO, but the next entry in the command FIFO is not a Start condition.	Y	N	Y
Pin Low Timeout Flag (MSR[PLTF])	Pin low timeout is enabled and SCL (or SDA, if configured) is low for longer than the configured timeout.	Y	N	Y
Data Match Flag (MSR[DMF])	The received data matches the configured data match, but the received data is not discarded due to a command FIFO entry.	Y	N	Y
Start Detect Flag (MSR[STF])	A Start condition is detected on the I2C bus (and the LPI2C controller is idle, if MCFGR1[STARTCFG] = 1).	Y	N	Y
Controller Busy Flag (MSR[MBF])	LPI2C controller is busy transmitting or receiving data.	N	N	N
Bus Busy Flag (MSR[BBF])	LPI2C controller is enabled and activity is detected on the I2C bus, but no Stop condition is detected and no bus idle timeout (if enabled) occurred.	N	N	N

60.3.8.2 Target mode

Table 567 lists the target mode sources that can generate LPI2C target interrupts and LPI2C target transmit and receive DMA requests.

Table 567. Target interrupts and DMA requests

Status flag	Description	Can generate		
		Interrupt?	DMA request?	Low-power wake-up?
Transmit Data Flag (SSR[TDF])	Data can be written to Target Transmit Data (STDR).	Y	TX	Y
Receive Data Flag (SSR[RDF])	Data can be read from Target Receive Data (SRDR).	Y	RX	Y

Table continues on the next page...

Table 567. Target interrupts and DMA requests (continued)

Status flag	Description	Can generate		
		Interrupt?	DMA request?	Low-power wake-up?
Address Valid Flag (SSR[AVF])	Address can be read from Target Address Status (SASR).	Y	RX	Y
Transmit ACK Flag (SSR[TAF])	ACK or NACK can be written to Target Transmit ACK (STAR).	Y	N	Y
Repeated Start Flag (SSR[RSF])	Target has detected an address match followed by a repeated Start condition.	Y	RX	Y
Stop Detect Flag (SSR[SDF])	Target has detected an address match followed by a Stop condition.	Y	RX	Y
Bit Error Flag (SSR[BEF])	Target was transmitting data, but received data is different from what was transmitted.	Y	N	Y
FIFO Error Flag (SSR[FEF])	This flag is set by: • Transmit data underrun • Receive data overrun • Address status overrun when SCFGR1[RXCFG] = 1 This flag can only be set when clock stretching is disabled.	Y	N	Y
Address Match 0 Flag (SSR[AM0F])	Target detected an address match SAMR[ADDR0].	Y	N	N
Address Match 1 Flag (SSR[AM1F])	Target detected an address match with SAMR[ADDR1] or using an address range.	Y	N	N
General Call Flag (SSR[GCF])	Target detected an address match with the general call address.	Y	N	N
SMBus Alert Response Flag (SSR[SARF])	Target detected an address match with the SMBus alert address.	Y	N	N
Target Busy Flag (SSR[SBF])	LPI2C target is busy receiving an address byte or is transmitting or receiving data.	N	N	N
Bus Busy Flag (SSR[BBF])	LPI2C target is enabled and a Start condition is detected on I2C bus, but no Stop condition detected.	N	N	N

60.3.8.3 End-of-packet DMA transfer

End-of-packet functionality serves serial interfaces where the transfer size may not be known in advance and the data is pushed by an external device. Examples include UART receive, I2C Target mode, and SPI Target mode. End-of-packet processing is intended to ensure that data does not become stranded in either the receive FIFO or DMA receive buffer. Support for end-of-packet processing must be implemented in both the serial interfaces and DMA controller.

The condition that signals the end of packet differs for each serial interface but is processed by the serial peripheral and DMA in the same way. For example:

- UART end of packet is signaled by an idle line condition.

- I2C end of packet is signaled by Stop or repeated Start condition.
- SPI end of packet is signaled by PCS negation.

When the serial peripheral is configured to signal an end-of-packet condition to the DMA and the serial interface detects an end-of-packet condition, it asserts the DMA request for the receive FIFO regardless of the watermark configuration. For larger watermark configurations, this behavior ensures that the last few words of the transfer are first flushed from the receive FIFO.

The DMA then reads the contents of the receive FIFO, depending on the first word in the FIFO:

- If the receive FIFO is empty, the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, but the first word in the FIFO starts a new packet, data is not pulled from the receive FIFO. Also, the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, and the first word in the FIFO does not start a new packet, the DMA transfers data as normal.

Because the DMA may transfer multiple words per request, the end-of-packet condition persists until the DMA minor loop finishes and no additional data is pulled from the FIFO. The status flag that triggered the end-of-packet condition is cleared when the minor loop completes after the end of packet is signaled to the DMA controller.

When the DMA detects the end-of-packet condition, it writes all received words up to the end of packet into system memory. It also saves the destination address for the word after the last valid data. The DMA then terminates the channel as if the major loop completed, including final offsets and optional interrupts, channel linking, and scatter-gather. The final destination address can be saved to system memory or is available in the destination address register.

Because the DMA terminates the major loop, the receive FIFO is not serviced until the DMA is reconfigured. You can perform this reconfiguration through software or hardware (for example, channel linking or scatter-gather). Minimize this delay to avoid receiver FIFO overrun. Automatic DMA end-of-packet processing is not recommended when there are only a few words transferred between end-of-packet conditions. In this case, the DMA spends more time processing the end of packet than transferring the data. For example, to avoid an excessive number of idle conditions, increase the UART idle line length as needed.

60.4 External signals

Table 568. External signals

Signal	Name	Two-wire scheme	Four-wire scheme	Direction
SCL	LPI2C clock line	SCL	In Four-Wire mode, this pin is the SCL input pin.	Input or output
SDA	LPI2C data line	SDA	In Four-Wire mode, this pin is the SDA input pin.	Input or output
SCLS	Secondary I2C clock line	Not used	In Four-Wire mode, this pin is the SCLS output pin. If LPI2C controller/target are configured to use separate pins, then this pin is the LPI2C target SCL pin.	Input or output
SDAS	Secondary I2C data line	Not used	In Four-Wire mode, this pin is the SDAS output pin. If LPI2C controller/target are configured to use separate pins, then this pin is the LPI2C target SDA pin.	Input or output

Figure 356 shows the two-signal connection.

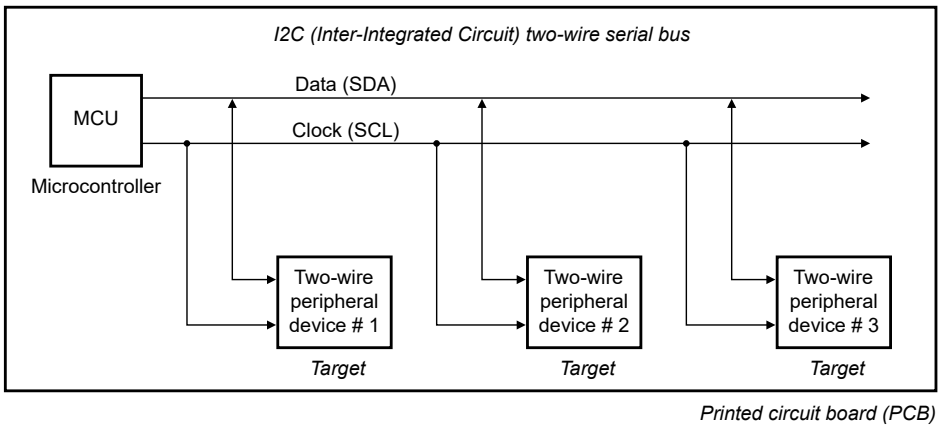


Figure 356. I²C two-wire serial bus

Figure 357 shows a possible four-signal connection.

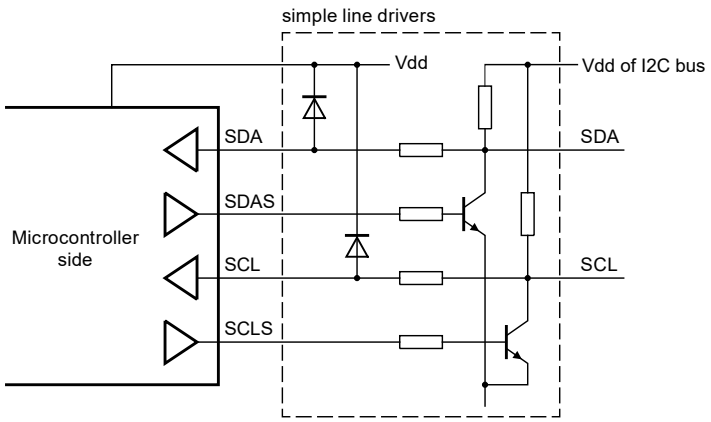


Figure 357. I²C four-wire serial bus

60.5 Initialization

To initialize the LPI2C controller:

1. Configure Controller Configuration 0 (MCFGR0)–Controller Configuration 3 (MCFGR3) as required by the application.
2. Configure Controller Clock Configuration 0 (MCCR0) and Controller Clock Configuration 1 (MCCR1) to satisfy the timing requirements of the I2C mode supported by the application.
3. Enable controller interrupts and DMA requests as required by the application.
4. Enable the LPI2C controller by writing 1 to MCR[MEN].

To initialize the LPI2C target:

1. Configure Target Address Match (SAMR) with the I2C address of the target location on the I2C bus.
2. Configure Target Configuration 0 (SCFGR0) and Target Configuration 1 (SCFGR1) as required by the application.
3. Configure Target Configuration 2 (SCFGR2) to satisfy the timing requirements of the I2C mode supported by the application.
4. Enable target interrupts and DMA requests as required by the application.
5. Enable the LPI2C target by writing 1 to SCR[SEN].

60.6 Application information

Configure the I2C timing parameters to meet the requirements of the I2C specification. This configuration depends on the supported mode and LPI2C functional clock frequency. When switching between two modes using different clock configuration registers (for example, Fast mode and HS mode), MCFGR1[PRESCALE] must remain constant between the modes.

Table 569. Example timing configurations

I2C mode	Clock frequency	Baud rate	PRESCALE	FILTSCL / FILTSDA	SETHOLD	CLKLO	CLKHI	DATAVD
Standard	8 MHz	100 kbit/s	0h	0h/0h	24h	28h	24h	02h
Standard	48 MHz	100 kbit/s	2h	1h/1h	37h	3Fh	37h	03h
Standard	60 MHz	100 kbit/s	2h	1h/1h	45h	50h	44h	04h
Fast	8 MHz	400 kbit/s	0h	0h/0h	04h	0Bh	05h	02h
Fast+	8 MHz	1 Mbit/s	0h	0h/0h	02h	03h	01h	01h
Fast	48 MHz	400 kbit/s	0h	1h/1h	1Dh	3Eh	35h	0Fh
Fast	48 MHz	400 kbit/s	2h	1h/1h	07h	11h	0Bh	03h
Fast+	48 MHz	1 Mbit/s	2h	1h/1h	03h	06h	04h	04h
HS	48 MHz	3.2 Mbit/s	0h	0h/0h	07h	08h	03h	01h
Fast	60 MHz	400 kbit/s	1h	2h/2h	11h	28h	1Fh	08h
Fast+	60 MHz	1 Mbit/s	1h	2h/2h	07h	0Fh	0Bh	01h
HS	60 MHz	3.33 Mbit/s	1h	0h/0h	04h	04h	02h	01h

60.7 Memory map and registers

60.7.1 LPI2C register descriptions

Writing to a read-only register or reading from a write-only register can cause bus errors. This module does not check whether programmed values in the registers are correct; you must ensure that valid programmed values are written to the registers.

60.7.1.1 LPI2C memory map

LPI2C1 base address: 4434_0000h

LPI2C2 base address: 4435_0000h

LPI2C3 base address: 4253_0000h

LPI2C4 base address: 4254_0000h

LPI2C5 base address: 426B_0000h

LPI2C6 base address: 426C_0000h

LPI2C7 base address: 426D_0000h

LPI2C8 base address: 426E_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VERID)	32	R	0103_0003
4	Parameter (PARAM)	32	R	0000_0303
10	Controller Control (MCR)	32	RW	0000_0000
14	Controller Status (MSR)	32	RW	0000_0001
18	Controller Interrupt Enable (MIER)	32	RW	0000_0000
1C	Controller DMA Enable (MDER)	32	RW	0000_0000
20	Controller Configuration 0 (MCFGR0)	32	RW	0000_0000
24	Controller Configuration 1 (MCFGR1)	32	RW	0000_0000
28	Controller Configuration 2 (MCFGR2)	32	RW	0000_0000
2C	Controller Configuration 3 (MCFGR3)	32	RW	0000_0000
40	Controller Data Match (MDMR)	32	RW	0000_0000
48	Controller Clock Configuration 0 (MCCR0)	32	RW	0000_0000
50	Controller Clock Configuration 1 (MCCR1)	32	RW	0000_0000
58	Controller FIFO Control (MFCR)	32	RW	0000_0000
5C	Controller FIFO Status (MFSR)	32	R	0000_0000
60	Controller Transmit Data (MTDR)	32	W	0000_0000
70	Controller Receive Data (MRDR)	32	R	0000_4000
78	Controller Receive Data Read Only (MRDROR)	32	R	0000_4000
110	Target Control (SCR)	32	RW	0000_0000
114	Target Status (SSR)	32	RW	0000_0000
118	Target Interrupt Enable (SIER)	32	RW	0000_0000
11C	Target DMA Enable (SDER)	32	RW	0000_0000
120	Target Configuration 0 (SCFGR0)	32	RW	0000_0000
124	Target Configuration 1 (SCFGR1)	32	RW	0000_0000
128	Target Configuration 2 (SCFGR2)	32	RW	0000_0000
140	Target Address Match (SAMR)	32	RW	0000_0000
150	Target Address Status (SASR)	32	R	0000_4000
154	Target Transmit ACK (STAR)	32	RW	0000_0000
160	Target Transmit Data (STDR)	32	W	0000_0000
170	Target Receive Data (SRDR)	32	R	0000_4000
178	Target Receive Data Read Only (SRDROR)	32	R	0000_4000
200 - 3FC	Controller Transmit Command Burst (MTCBR0 - MTCBR127)	32	W	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
400 - 7FC	Transmit Data Burst (MTDBR0 - MTDBR255)	32	W	0000_0000

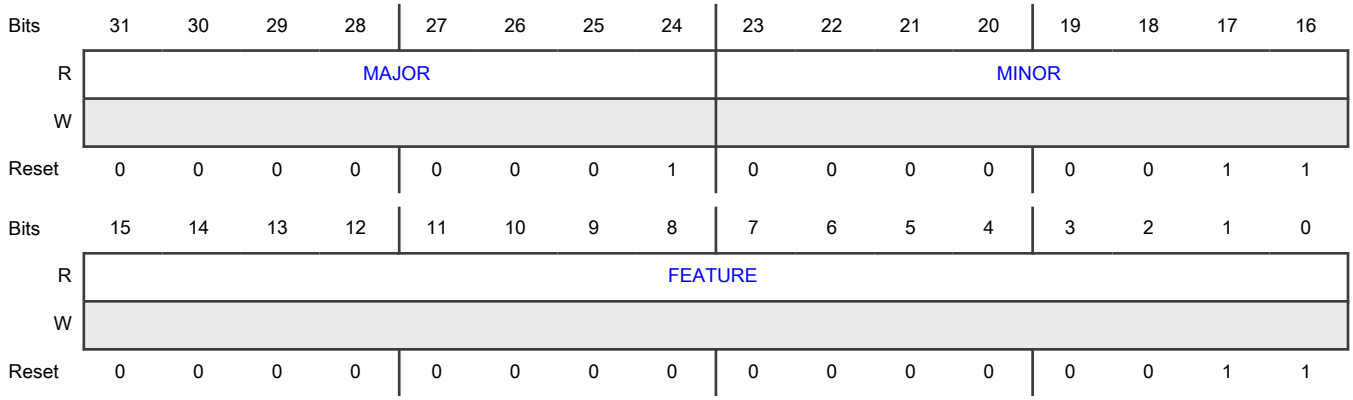
60.7.1.2 Version ID (VERID)

Contains version numbers for the module design and feature set.

Offset

Register	Offset
VERID	0h

Diagram



Fields

Field	Description
31-24 MAJOR	Major Version Number Returns the major version number for the module design specification.
23-16 MINOR	Minor Version Number Returns the minor version number for the module design specification.
15-0 FEATURE	Feature Specification Number Returns the feature set number. 0000_0000_0000_0010 - Controller only, with standard feature set 0000_0000_0000_0011 - Controller and target, with standard feature set

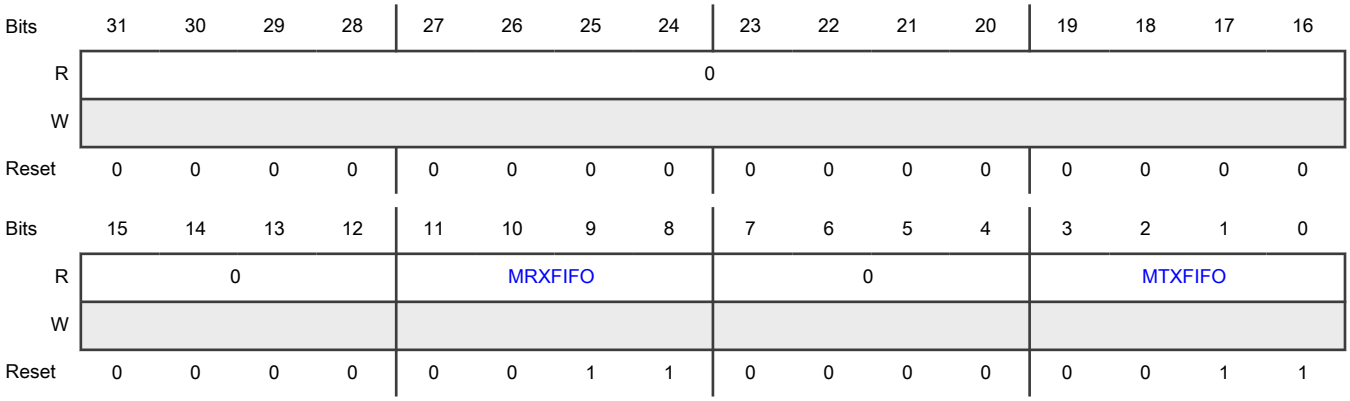
60.7.1.3 Parameter (PARAM)

Contains parameter values implemented in the module.

Offset

Register	Offset
PARAM	4h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-12 —	Reserved
11-8 MRXFIFO	Controller Receive FIFO Size Configures the number of words in the controller receive FIFO to $2^{MRXFIFO}$.
7-4 —	Reserved
3-0 MTXFIFO	Controller Transmit FIFO Size Configures the number of words in the controller transmit FIFO to $2^{MTXFIFO}$.

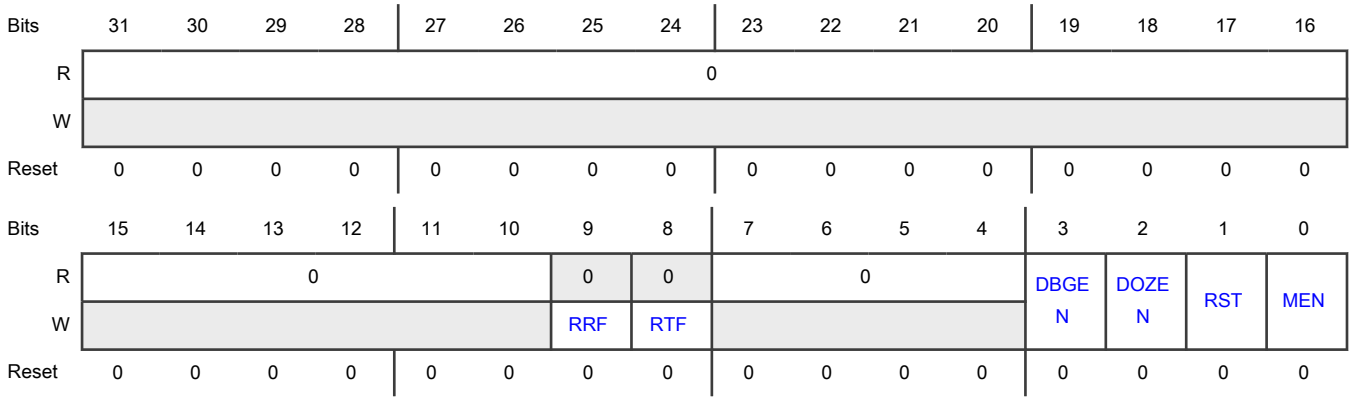
60.7.1.4 Controller Control (MCR)

Contains resets, debug enable, and other controller control settings.

Offset

Register	Offset
MCR	10h

Diagram



Fields

Field	Description
31-10 —	Reserved
9 RRF	Reset Receive FIFO Resets the receive FIFO in Controller Receive Data (MRDR). 0 - No effect 1 - Reset receive FIFO
8 RTF	Reset Transmit FIFO Resets the transmit FIFO in Controller Transmit Data (MTDR). 0 - No effect 1 - Reset transmit FIFO
7-4 —	Reserved
3 DBGEN	Debug Enable Enables the controller in Debug mode. 0 - Disable 1 - Enable
2 DOZEN	Doze Mode Enable Enables the controller in Doze mode. 0 - Enable 1 - Disable
1	Software Reset

Table continues on the next page...

Table continued from the previous page...

Field	Description
RST	Resets all internal controller logic and registers except Controller Control (MCR). This field remains 1 (enabled) until you write 0 to it. The reset takes effect immediately and remains asserted until negated by software. There is no minimum delay required before clearing the software reset. 0 - No effect 1 - Reset
0 MEN	Controller Enable Enables the controller logic. 0 - Disable 1 - Enable

60.7.1.5 Controller Status (MSR)

Contains status flags for transmit and receive data, for start and stop conditions, and for bus and controller busy or idle status.

Offset

Register	Offset
MSR	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BBF	MBF	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STF	DMF	PLTF	FEF	ALF	NDF	SDF	EPF	0						RDF	TDF
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31-26 —	Reserved
25	Bus Busy Flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
BBF	Specifies whether the I2C bus is busy. 0 - Idle 1 - Busy
24 MBF	Controller Busy Flag Specifies whether the I2C controller is busy. 0 - Idle 1 - Busy
23-16 —	Reserved
15 STF	Start Flag Specifies whether a Start condition is detected on the bus when the bus is idle. When MCFGR1[STARTCFG] is 0, this field becomes 1 only if the LPI2C controller is also idle. When MCFGR1[STARTCFG] is 1, STF becomes 1 for any Start condition when the I2C bus is idle. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Start condition not detected 1 - Start condition detected When writing 0 - No effect 1 - Clear the flag
14 DMF	Data Match Flag Indicates whether the received data matches MDMR[MATCH0] or MDMR[MATCH1] (as configured by MCFGR1[MATCFG]). Received data discarded due to MTDR[CMD] does not cause this flag to set. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Matching data not received 1 - Matching data received When writing 0 - No effect 1 - Clear the flag
13	Pin Low Timeout Flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
PLTF	Indicates whether pin low timeout has occurred. Sets when the SCL or SDA input is low for more than the number of PINLOW cycles defined by MCFGR3[PINLOW], even when the LPI2C controller is idle. You must resolve the pin low condition via software. PLTF cannot be cleared as long as the pin low timeout continues. Before LPI2C can initiate a Start condition, you must clear this flag. See MCFGR1[TIMECFG] for the SCL and/or SDA timeout settings. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Pin low timeout did not occur 1 - Pin low timeout occurred When writing 0 - No effect 1 - Clear the flag
12 FEF	FIFO Error Flag Detects the LPI2C controller's attempt to send or receive data without first generating a (repeated) Start condition. This error can occur when the transmit FIFO underflows when MCFGR1[AUTOSTOP] = 1. When this flag is set, the LPI2C controller sends a Stop condition (if busy). The controller does not initiate a new Start condition until the flag is cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No FIFO error 1 - FIFO error When writing 0 - No effect 1 - Clear the flag
11 ALF	Arbitration Lost Flag Indicates whether arbitration is lost. Either of these conditions sets this flag: • The LPI2C controller transmits a logic 1 and detects a logic 0 on the I2C bus. • The LPI2C controller detects a Start or Stop condition when the LPI2C controller is transmitting data. When ALF is set, the LPI2C controller releases the I2C bus (goes idle), and the LPI2C controller does not initiate a new Start condition until the ALF is cleared. **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When reading 0 - Controller did not lose arbitration 1 - Controller lost arbitration When writing 0 - No effect 1 - Clear the flag
10 NDF	NACK Detect Flag Indicates whether an unexpected NACK has been detected. This flag is set when the LPI2C controller detects a NACK it was not expecting when transmitting an address or data. When set, the controller does not initiate a new Start condition until this flag is cleared. If a NACK is expected for a given address (as configured by the command word), this flag is set if a NACK is not generated. When this flag is set, the LPI2C controller automatically transmits a Stop condition if MCFGR1[AUTOSTOP] = 1, or if the transmit FIFO is not empty. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No unexpected NACK detected 1 - Unexpected NACK detected When writing 0 - No effect 1 - Clear the flag
9 SDF	Stop Detect Flag Indicates whether the LPI2C controller has generated a Stop condition. When MCFGR1[STOPCFG] = 0, this flag is set for any Stop condition. When MCFGR1[STOPCFG] = 1, this flag is set only when the LPI2C controller is also idle (transmit FIFO is empty). **NOTE** This field behaves differently for register reads and writes. When reading 0 - No Stop condition generated 1 - Stop condition generated When writing 0 - No effect 1 - Clear the flag
8	End Packet Flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
EPF	Indicates whether the LPI2C controller has generated a repeated Start condition or a Stop condition. When the controller first generates a Start condition, this flag is not set. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No Stop or repeated Start generated 1 - Stop or repeated Start generated When writing 0 - No effect 1 - Clear the flag
7-2 —	Reserved
1 RDF	Receive Data Flag Indicates whether the receive data is ready. This flag is set when the number of words in the receive FIFO is greater than MFCR[RXWATER]. 0 - Receive data not ready 1 - Receive data ready
0 TDF	Transmit Data Flag Indicates whether transmit data is requested. This flag is set when the number of words in the transmit FIFO is equal or less than MFCR[TXWATER]. 0 - Transmit data not requested 1 - Transmit data requested

60.7.1.6 Controller Interrupt Enable (MIER)

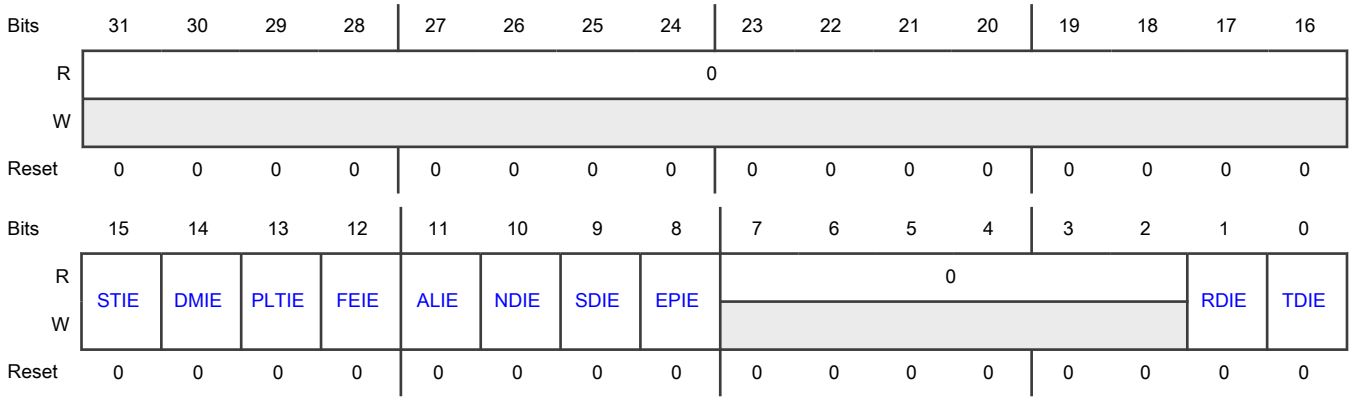
Contains enables for:

- Transmit and receive data interrupts
- Start, Stop, and NACK detection interrupts
- DMA interrupts

Offset

Register	Offset
MIER	18h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 STIE	Start Interrupt Enable Enables interrupt for Start condition. 0 - Disable 1 - Enable
14 DMIE	Data Match Interrupt Enable Enables interrupt for data match. 0 - Disable 1 - Enable
13 PLTIE	Pin Low Timeout Interrupt Enable Enables interrupt for pin-low timeout. 0 - Disable 1 - Enable
12 FEIE	FIFO Error Interrupt Enable Enables interrupt for FIFO error. 0 - Disable 1 - Enable
11 ALIE	Arbitration Lost Interrupt Enable Enables interrupt for arbitration lost. 0 - Disable 1 - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
10 NDIE	NACK Detect Interrupt Enable Enables interrupt for NACK detection. 0 - Disable 1 - Enable
9 SDIE	Stop Detect Interrupt Enable Enables interrupt for Stop detection. 0 - Disable 1 - Enable
8 EPIE	End Packet Interrupt Enable Enables interrupt for end packet. 0 - Disable 1 - Enable
7-2 —	Reserved
1 RDIE	Receive Data Interrupt Enable Enables interrupt for receive data. 0 - Disable 1 - Enable
0 TDIE	Transmit Data Interrupt Enable Enables interrupt for transmit data. 0 - Disable 1 - Enable

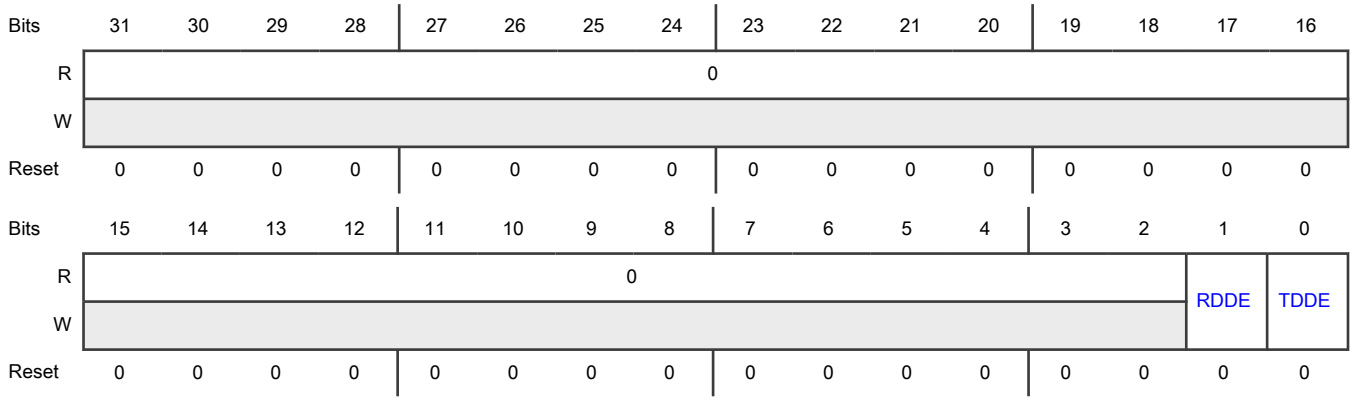
60.7.1.7 Controller DMA Enable (MDER)

Contains DMA transmit, request, and receive enables.

Offset

Register	Offset
MDER	1Ch

Diagram



Fields

Field	Description
31-2 —	Reserved
1 RDDE	Receive Data DMA Enable Enables DMA receive data. 0 - Disable 1 - Enable
0 TDDE	Transmit Data DMA Enable Enables DMA transmit data. 0 - Disable 1 - Enable

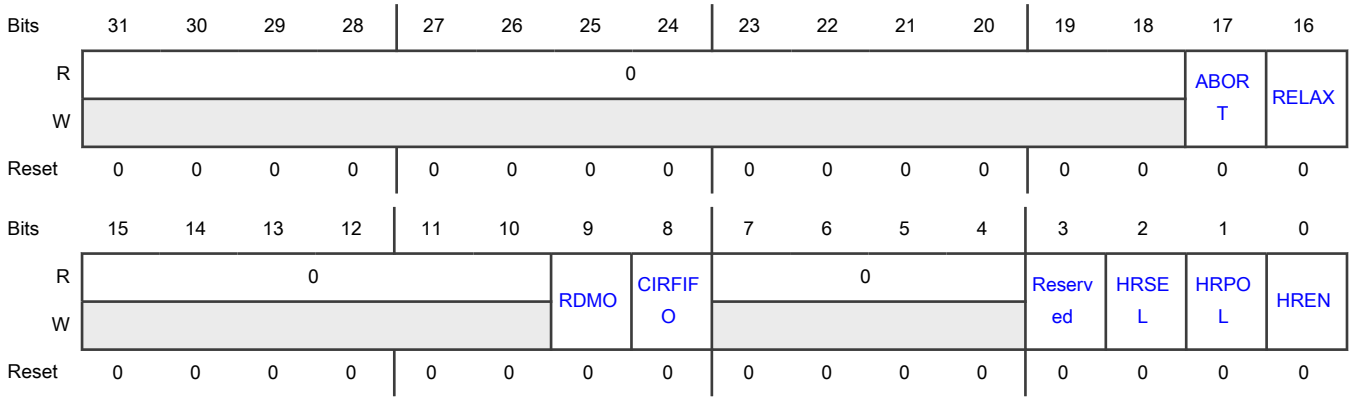
60.7.1.8 Controller Configuration 0 (MCFGR0)

Contains host settings and other receive and transfer settings.

Offset

Register	Offset
MCFGR0	20h

Diagram



Fields

Field	Description
31-18 —	Reserved
17 ABORT	Abort Transfer Determines whether the LPI2C controller completes the existing byte and then sends a Stop condition. A new transfer does not start while this field is 1. 0 - Normal transfer 1 - Abort existing transfer and do not start a new one
16 RELAX	Relaxed Mode Specifies the type of transfer. If this field is 1, the LPI2C controller relaxes several transfer restrictions. Transfers start when the bus is busy and FIFO commands are not checked for errors. You can use this mode to attempt recovery of a stuck I2C target by toggling the SCL pin. 0 - Normal transfer 1 - Relaxed transfer
15-10 —	Reserved
9 RDMO	Receive Data Match Only Determines whether all received data that does not set MSR[DMF] is discarded. After MSR[DMF] is set, the RDMO configuration is ignored. When disabling RDMO, write 0 to this field before writing 0 to MSR[DMF] to ensure that no receive data is lost. 0 - Received data is stored in the receive FIFO 1 - Received data is discarded unless MSR[DMF] is set
8 CIRFIFO	Circular FIFO Enable Enables the transmit FIFO read pointer to be saved to a temporary register. The transmit FIFO empties as normal. After the LPI2C controller is idle and the transmit FIFO is empty, the read pointer value is

Table continues on the next page...

Table continued from the previous page...

Field	Description
	restored from the temporary register. This setting causes the contents of the transmit FIFO to be cycled through repeatedly. If MCFGR1[AUTOSTOP] is 1, then a Stop condition is sent whenever the transmit FIFO is empty and the read pointer is restored. 0 - Disable 1 - Enable
7-4 —	Reserved
3 —	Reserved
2 HRSEL	Host Request Select Selects the source of the host request input. When host request is enabled, this field must not change. 0 - Reserved 1 - Host request input is input trigger
1 HRPOL	Host Request Polarity Configures the polarity of the host request input. When host request is enabled, this field must not change. HRPOL sets the polarity for both the HREQ pin and the input trigger. • When HRPOL=0, the polarity is configured for active low, so host request is asserted if the HREQ pin or input trigger are logic 0. • When HRPOL=1, the polarity is configured for active high, so host request is asserted if the HREQ pin or input trigger are logic 1. 0 - Active low 1 - Active high
0 HREN	Host Request Enable Enables host request. When enabled, the LPI2C controller only initiates a Start condition if the host request input is asserted and the bus is idle. A repeated Start condition is not affected by the host request. 0 - Disable 1 - Enable

60.7.1.9 Controller Configuration 1 (MCFGR1)

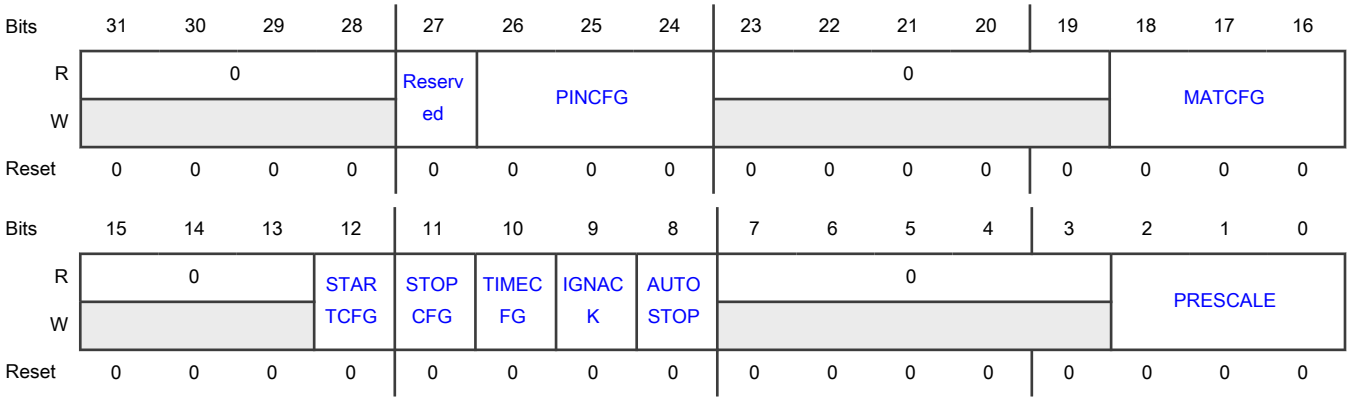
Contains controls for pin configuration, clock prescaler, and various other control settings.

Write to this register only when the I2C controller is disabled.

Offset

Register	Offset
MCFGR1	24h

Diagram



Fields

Field	Description
31-28 —	Reserved
27 —	Reserved
26-24 PINCFG	Pin Configuration Configures the pin mode for LPI2C. 000 - Two-pin open drain mode. SCL/SDA pins: Bidirectional open drain for controller and target. SCLS/SDAS pins: Not used. 001 - Two-pin output only mode (Ultra-Fast mode). SCL/SDA pins: Output-only (Ultra-Fast mode) open drain for controller and target. SCLS/SDAS pins: Not used. 010 - Two-pin push-pull mode. SCL/SDA pins: Bidirectional push-pull for controller and target. SCLS/SDAS pins: Not used. 011 - Four-pin push-pull mode. SCL/SDA pins: Input only for controller and target. SCLS/SDAS pins: Output-only push-pull for controller and target. 100 - Two-pin open-drain mode with separate LPI2C target. SCL/SDA pins: Bidirectional open drain for controller. SCLS/SDAS pins: Bidirectional open drain for target. 101 - Two-pin output only mode (Ultra-Fast mode) with separate LPI2C target. SCL/SDA pins: Output-only (Ultra-Fast mode) open drain for controller. SCLS/SDAS pins: Output-only open drain for target. 110 - Two-pin push-pull mode with separate LPI2C target. SCL/SDA pins: Bidirectional push-pull for controller. SCLS/SDAS pins: Bidirectional push-pull for target.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	111 - Four-pin push-pull mode (inverted outputs). SCL/SDA pins: Input only for controller and target. SCLS/SDAS pins: Inverted output-only push-pull for controller and target.
23-19 —	Reserved
18-16 MATCFG	Match Configuration Configures the condition that sets MSR[DMF]. See Controller Data Match (MDMR). 000 - Match is disabled 001 - Reserved 010 - Match is enabled: first data word equals MDMR[MATCH0] OR MDMR[MATCH1] 011 - Match is enabled: any data word equals MDMR[MATCH0] OR MDMR[MATCH1] 100 - Match is enabled: (first data word equals MDMR[MATCH0]) AND (second data word equals MDMR[MATCH1) 101 - Match is enabled: (any data word equals MDMR[MATCH0]) AND (next data word equals MDMR[MATCH1) 110 - Match is enabled: (first data word AND MDMR[MATCH1]) equals (MDMR[MATCH0] AND MDMR[MATCH1]) 111 - Match is enabled: (any data word AND MDMR[MATCH1]) equals (MDMR[MATCH0] AND MDMR[MATCH1])
15-13 —	Reserved
12 STARTCFG	Start Configuration Configures the conditions that set MSR[STF] when a Start condition occurs. When this field is 0, MSR[STF] is set on a Start condition when both the I2C bus and LPI2C are idle. In other words, any nonrepeated Start condition is initiated by any controller on the bus except the LPI2C controller. When this field is 1, MSR[STF] is set on a Start condition when the I2C bus is idle. In other words, any nonrepeated Start condition is initiated by any controller on the bus including the LPI2C controller. 0 - Sets when both I2C bus and LPI2C controller are idle 1 - Sets when I2C bus is idle
11 STOPCFG	Stop Configuration Configures the conditions that set MSR[SDF]. When this field is 0, any Stop condition generated by the LPI2C controller sets SDF. When this field is 1, the last Stop condition before the LPI2C controller is idle sets SDF. In this case, the transmit FIFO is empty at the time of the Stop condition. 0 - Any Stop condition

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Last Stop condition
10 TIMECFG	Timeout Configuration Configures which signals must be low for longer than the configured timeout to set MSR[PLTF]. When this field is 0, MSR[PLTF] is set when SCL is low for longer than the configured timeout. 0 - SCL 1 - SCL or SDA
9 IGNACK	Ignore NACK Determines whether the LPI2C controller ignores a received NACK and treats it as an ACK. In this case, MSR[NDF] is never set. This field must be 1 in Ultra-Fast mode. 0 - No effect 1 - Treat a received NACK as an ACK
8 AUTOSTOP	Automatic Stop Generation Determines whether a Stop condition is generated when the LPI2C controller is busy and the transmit FIFO is empty. A Stop condition can also be generated using a transmit FIFO command. When this field is 1, a Stop condition is automatically generated when the transmit FIFO is empty and the LPI2C controller is busy. 0 - No effect 1 - Stop automatically generated
7-3 —	Reserved
2-0 PRESCALE	Prescaler Configures the clock prescaler used for all LPI2C controller logic except the digital glitch filters. 000 - Divide by 1 001 - Divide by 2 010 - Divide by 4 011 - Divide by 8 100 - Divide by 16 101 - Divide by 32 110 - Divide by 64 111 - Divide by 128

60.7.1.10 Controller Configuration 2 (MCFGR2)

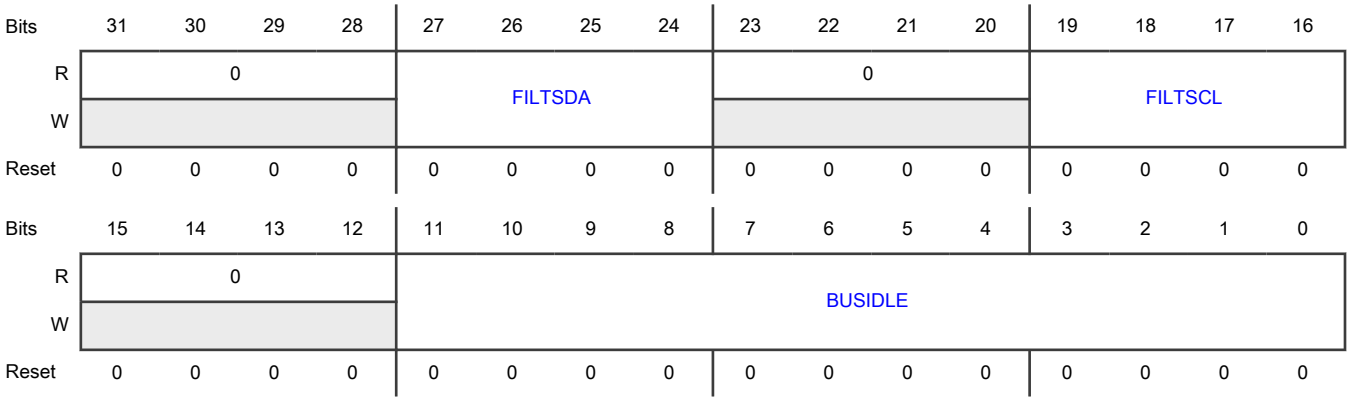
Contains the configuration for the bus idle timeout and glitch filters for SDA and SCL.

Write to this register only when the I2C controller is disabled.

Offset

Register	Offset
MCFGR2	28h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-24 FILTSDA	Glitch Filter SDA Configures the I2C controller digital glitch filters for the SDA input. The latency through the glitch filter is equal to the number of cycles defined by this field. The value of this field must be less than the minimum SCL low or high period. Glitches equal to or less than the number of cycles defined by this field are filtered out and ignored. Writing 0 to this field disables the glitch filter. MCFGR1[PRESCALE] does not affect the glitch filter cycle count. It is automatically bypassed in HS mode.
23-20 —	Reserved
19-16 FILTSCL	Glitch Filter SCL Configures the I2C controller digital glitch filters for SCL input. The latency through the glitch filter is equal to the number of cycles defined by this field. The value of this field must be less than the minimum SCL low or high period. Glitches equal to or less than the number of cycles defined by this field are filtered out and ignored. These cycles are based on the functional clock. Writing 0 to this field disables the glitch filter. MCFGR1[PRESCALE] does not affect the glitch filter cycle count. It is automatically bypassed in HS mode.
15-12	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
11-0 BUSIDLE	Bus Idle Timeout Configures the bus idle timeout period, in clock cycles. If both SCL and SDA are high for longer than the number of cycles defined by this field, the I2C bus is assumed to be idle and the controller can generate a Start condition. Writing 0 to this field disables the bus idle timeout.

60.7.1.11 Controller Configuration 3 (MCFGR3)

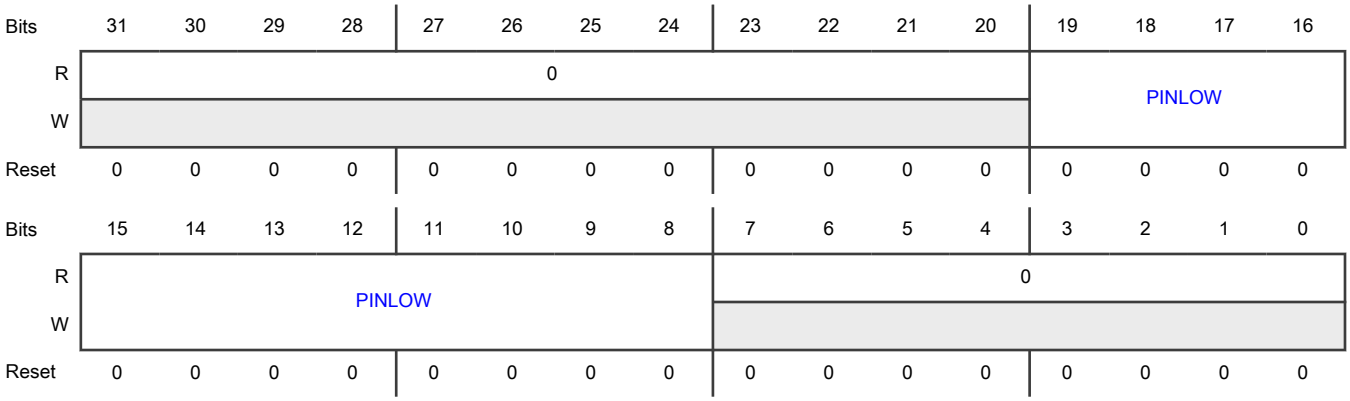
Configures the threshold value for the pin low timeout flag.

Write to this register only when the I2C controller is disabled.

Offset

Register	Offset
MCFGR3	2Ch

Diagram



Fields

Field	Description
31-20 —	Reserved
19-8 PINLOW	Pin Low Timeout Configures the threshold value, in clock cycles, that sets MSR[PLTF].

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If SCL or SDA (selected by MCFGR1[TIMECFG]) is low for longer than (PINLOW × 256) cycles, MSR[PLTF] is set. When this field is 0, the pin low timeout feature is disabled.
7-0 —	Reserved

60.7.1.12 Controller Data Match (MDMR)

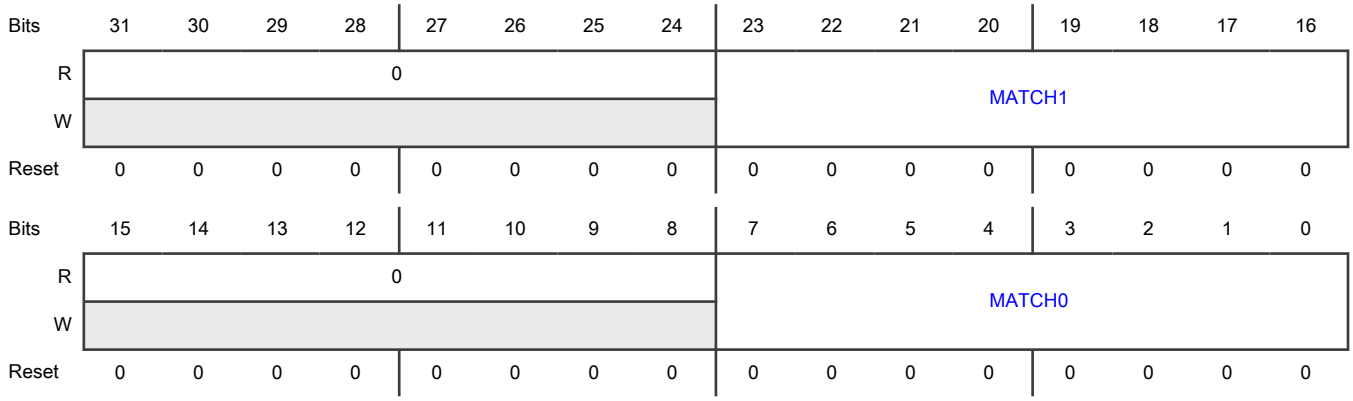
Contains data match values.

Write to this register only when the I2C controller is disabled or idle.

Offset

Register	Offset
MDMR	40h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 MATCH1	Match 1 Value Specifies match 1 value that is compared to the received data when receive data match is enabled.
15-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7-0 MATCH0	Match 0 Value Specifies match 0 value that is compared to the received data when receive data match is enabled.

60.7.1.13 Controller Clock Configuration 0 (MCCR0)

Configures various clock controls.

You cannot make changes to this register when the I2C controller is enabled and is used for standard, fast, fast-mode plus, and ultra-fast transfers.

Offset

Register	Offset
MCCR0	48h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		DATAVD						0		SETHOLD					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		CLKHI						0		CLKLO					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-30 —	Reserved
29-24 DATAVD	Data Valid Delay Specifies the minimum number of cycles (minus one) used as the data hold time for SDA. This value must be less than the minimum SCL low period.
23-22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
21-16 SETHOLD	Setup Hold Delay Specifies the minimum number of cycles (minus one) used by the controller for these conditions: • Hold time for a Start • Setup and hold time for a repeated Start • Setup time for a Stop The setup time is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.
15-14 —	Reserved
13-8 CLKHI	Clock High Period Specifies the minimum number of cycles (minus one) that the controller drives the SCL clock high. The SCL high time is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.
7-6 —	Reserved
5-0 CLKLO	Clock Low Period Specifies the minimum number of cycles (minus one) that the controller drives the SCL clock low. This value is also used for the minimum bus free time between a Stop and a Start condition. This period is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.

60.7.1.14 Controller Clock Configuration 1 (MCCR1)

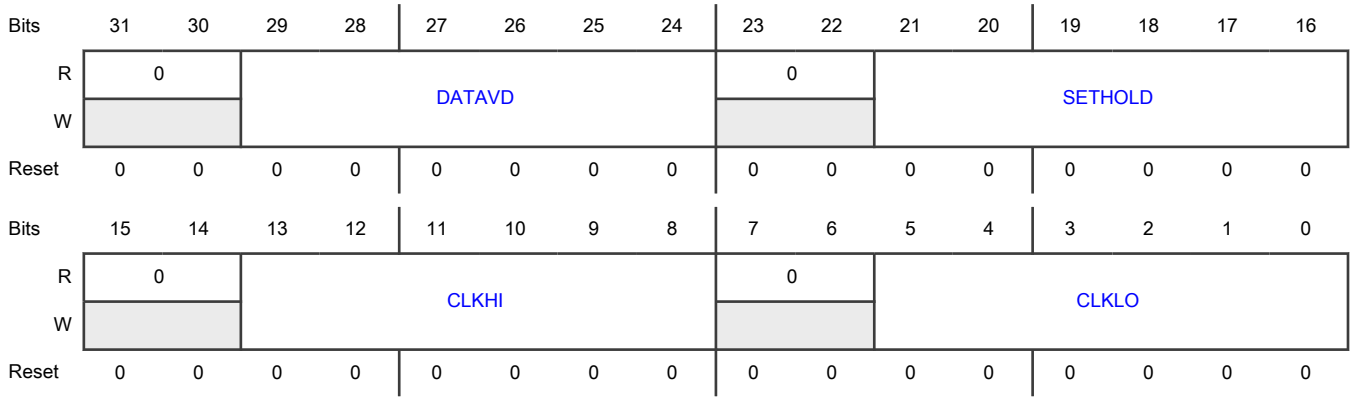
Configures various clock controls.

You cannot makes changes to this register when the I2C controller is enabled and is used for HS mode transfers. The separate clock configuration for HS mode allows arbitration to take place in Fast mode (with timing configured by Controller Clock Configuration 0 (MCCR0)), before switching to HS mode (with timing configured by MCCR1).

Offset

Register	Offset
MCCR1	50h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-24 DATAVD	Data Valid Delay Specifies the minimum number of cycles (minus one) used as the data hold time for SDA. This value must be less than the minimum SCL low period.
23-22 —	Reserved
21-16 SETHOLD	Setup Hold Delay Specifies the minimum number of cycles (minus one) used by the controller for these conditions: • Hold time for a Start condition • Setup and hold time for a repeated Start condition • Setup time for a Stop condition The setup time is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.
15-14 —	Reserved
13-8 CLKHI	Clock High Period Specifies the minimum number of cycles (minus one) that the controller drives the SCL clock high. The SCL high time is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.
7-6 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
5-0 CLKLO	Clock Low Period Specifies the minimum number of cycles (minus one) that the controller drives the SCL clock low. This value is also used for the minimum bus free time between a Stop and a Start condition. This period is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.

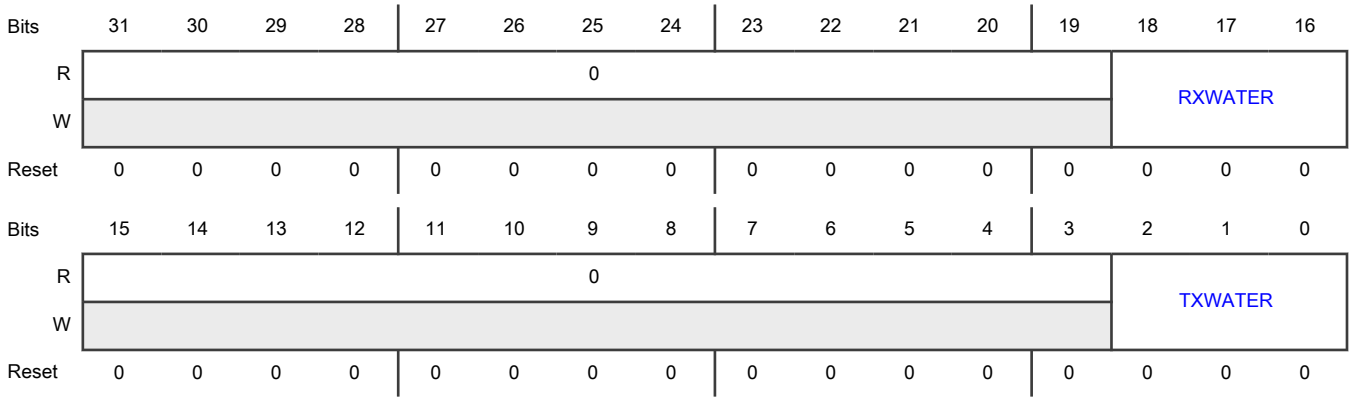
60.7.1.15 Controller FIFO Control (MFCR)

Controls the receive and transmit FIFO watermark values.

Offset

Register	Offset
MFCR	58h

Diagram



Fields

Field	Description
31-19 —	Reserved
18-16 RXWATER	Receive FIFO Watermark Determines the watermark for setting MSR[RDF]. That flag is set when the number of words in the receive FIFO is greater than the value of this field. Writing a value equal to or greater than the FIFO size truncates the value.
15-3 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2-0 TXWATER	Transmit FIFO Watermark Determines the watermark for setting MSR[TDF]. That flag is set when the number of words in the transmit FIFO is equal or less than the value of this field. Writing a value equal to or greater than the FIFO size truncates the value.

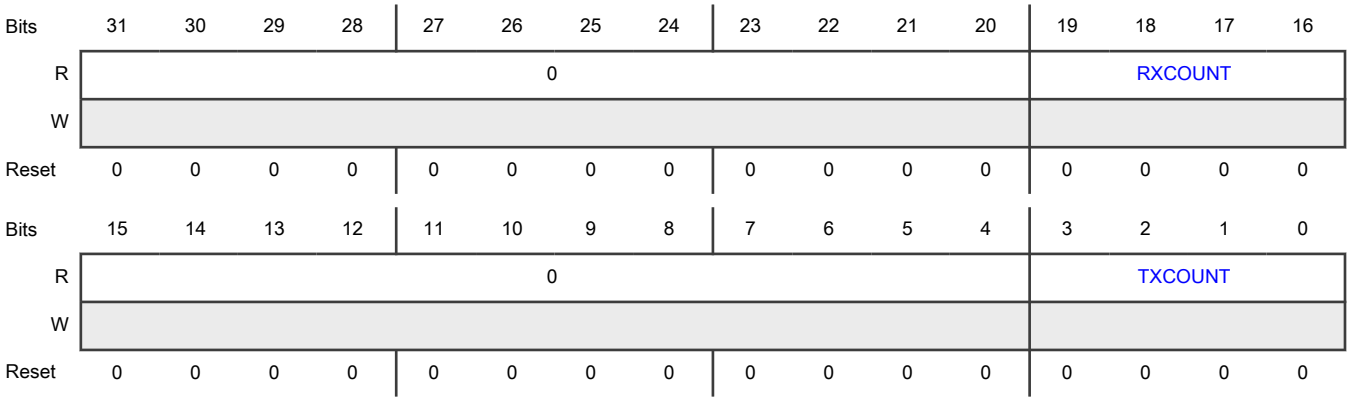
60.7.1.16 Controller FIFO Status (MFSR)

Specifies the number of words in the transmit and receive FIFOs.

Offset

Register	Offset
MFSR	5Ch

Diagram



Fields

Field	Description
31-20 —	Reserved
19-16 RXCOUNT	Receive FIFO Count Specifies the number of words in the receive FIFO.
15-4 —	Reserved
3-0 TXCOUNT	Transmit FIFO Count Specifies the number of words in the transmit FIFO.

60.7.1.17 Controller Transmit Data (MTDR)

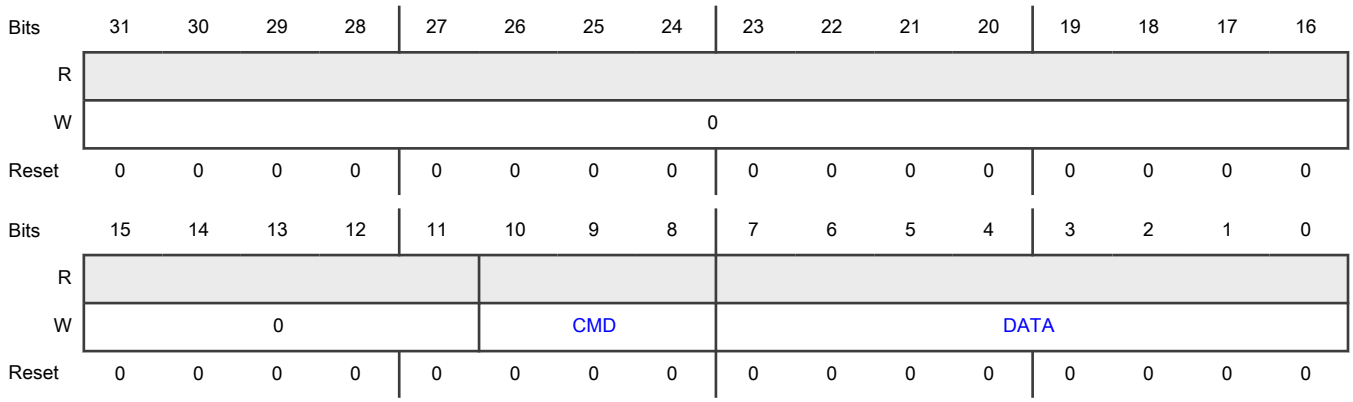
Configures transmit data:

- An 8-bit write to MTDR[CMD] is ignored and does not increment the FIFO write pointer.
- An 8-bit write to MTDR[DATA] zero-extends the value of MTDR[CMD] and increments the FIFO write pointer.
- A 16-bit or 32-bit write operation writes to both MTDR[CMD] and MTDR[DATA] and increments the FIFO write pointer.

Offset

Register	Offset
MTDR	60h

Diagram



Fields

Field	Description
31-11 —	Reserved
10-8 CMD	Command Data Selects command transmitted by controller. 000 - Transmit the value in DATA[7:0] 001 - Receive (DATA[7:0] + 1) bytes. DATA[7:0] is used as a byte counter. Receive that many bytes and check each for a data match (if configured) before storing the received data in the receive FIFO. 010 - Generate Stop condition on I2C bus 011 - Receive and discard (DATA[7:0] + 1) bytes. DATA[7:0] is used as a byte counter. Receive that many bytes but do not check for a data match or store those bytes in the receive FIFO. 100 - Generate (repeated) Start on the I2C bus and transmit the address in DATA[7:0] 101 - Generate (repeated) Start on the I2C bus and transmit the address in DATA[7:0] (this transfer expects a NACK to be returned) 110 - Generate (repeated) Start on the I2C bus and transmit the address in DATA[7:0] using HS mode

Table continues on the next page...

Table continued from the previous page...

Field	Description
	111 - Generate (repeated) Start on the I2C bus and transmit the address in DATA[7:0] using HS mode (this transfer expects a NACK to be returned)
7-0 DATA	Transmit Data Contains data used by the commands listed in MTDR[CMD]. Performing an 8-bit write to this field zero-extends the value of MTDR[CMD].

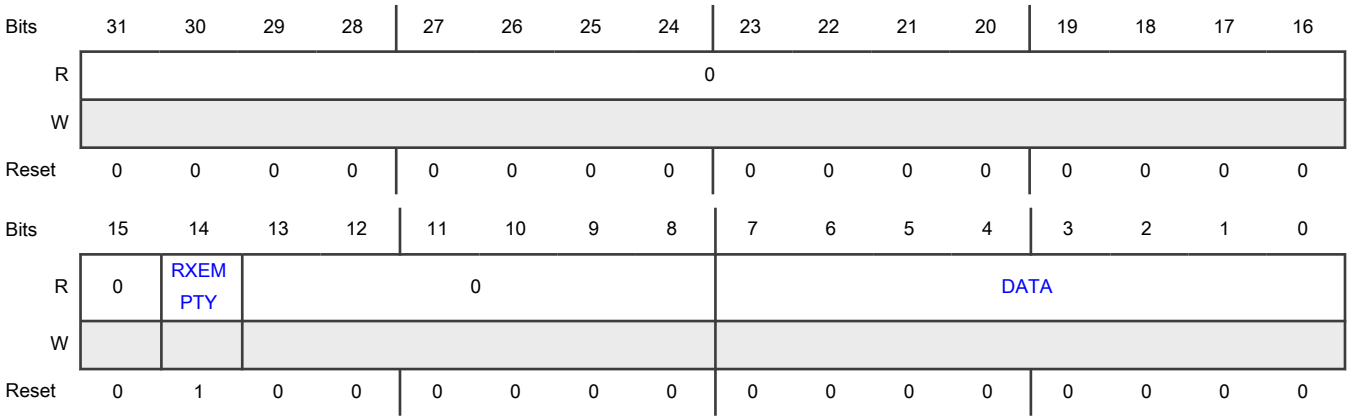
60.7.1.18 Controller Receive Data (MRDR)

Contains the status of the receive FIFO and the data received by the I2C controller that has not been discarded.

Offset

Register	Offset
MRDR	70h

Diagram



Fields

Field	Description
31-15 —	Reserved
14 RXEMPTY	Receive Empty Indicates whether the controller receive data FIFO is empty. 0 - Not empty 1 - Empty

Table continues on the next page...

Table continued from the previous page...

Field	Description
13-8 —	Reserved
7-0 DATA	Receive Data Contains data received by the I2C controller that has not been discarded. Received data can be discarded due to the command in MTDR[CMD], or the controller can be configured to discard nonmatching data.

60.7.1.19 Controller Receive Data Read Only (MRDROR)

Contains the status of the receive FIFO and returns the data received by the I2C controller, but does not pull the data from the FIFO.

Offset

Register	Offset
MRDROR	78h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RXEMPTY	0						DATA							
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-15 —	Reserved
14 RXEMPTY	RX Empty Indicates whether the receive data FIFO is empty. 0 - Not empty

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Empty
13-8 —	Reserved
7-0 DATA	Receive Data

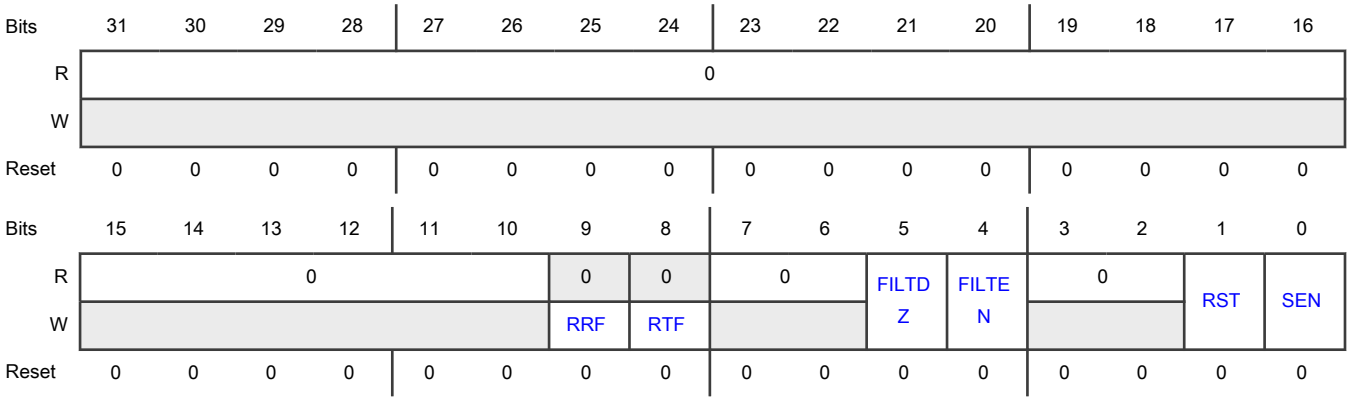
60.7.1.20 Target Control (SCR)

Contains resets and other target control settings.

Offset

Register	Offset
SCR	110h

Diagram



Fields

Field	Description
31-10 —	Reserved
9 RRF	Reset Receive FIFO Empties the receive FIFO in Target Receive Data (SRDR). 0 - No effect 1 - SRDR is now empty

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 RTF	Reset Transmit FIFO Empties the transmit FIFO in Target Transmit Data (STDR). 0 - No effect 1 - STDR is now empty
7-6 —	Reserved
5 FILTDZ	Filter Doze Enable Enables filter in Doze mode. Update this field only when the I2C target is disabled. 0 - Enable 1 - Disable
4 FILTEN	Filter Enable Enables digital filter and output delay counter for target mode. Update this field only when the I2C target is disabled. 0 - Disable 1 - Enable
3-2 —	Reserved
1 RST	Software Reset Resets target mode logic. The reset takes effect immediately. The value of this field remains 1 until you write 0 to it. There is no minimum delay required before clearing the software reset. 0 - Not reset 1 - Reset
0 SEN	Target Enable Enables I2C Target mode. 0 - Disable 1 - Enable

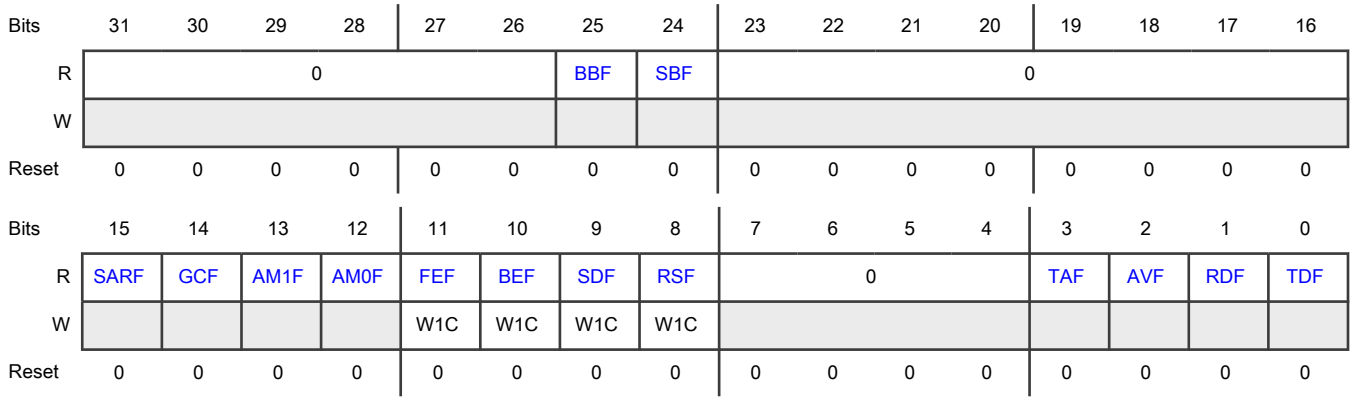
60.7.1.21 Target Status (SSR)

Contains status flags for transmit and receive data, for error conditions, and for bus and target busy or idle status.

Offset

Register	Offset
SSR	114h

Diagram



Fields

Field	Description
31-26 —	Reserved
25 BBF	Bus Busy Flag Indicates whether an I2C bus is idle or busy. 0 - Idle 1 - Busy
24 SBF	Target Busy Flag Indicates whether an I2C target is idle or busy. 0 - Idle 1 - Busy
23-16 —	Reserved
15 SARF	SMBus Alert Response Flag Indicates whether an SMBus alert response has been detected. You can clear this flag by reading Target Address Status (SASR). This flag cannot generate an asynchronous wakeup. 0 - Disabled or not detected 1 - Enabled and detected
14 GCF	General Call Flag Indicates whether a target has detected the general call address. You can clear this flag by reading Target Address Status (SASR). This flag cannot generate an asynchronous wakeup.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - General call address disabled or not detected 1 - General call address detected
13 AM1F	Address Match 1 Flag Indicates whether the received address matches the value in ADDR1, or it falls within the ADDR0 to ADDR1 range as configured by SCFGR1[ADDRCFG]. This flag is cleared by reading Target Address Status (SASR). This flag cannot generate an asynchronous wakeup. 0 - Matching address not received 1 - Matching address received
12 AM0F	Address Match 0 Flag Indicates whether the received address matches the ADDR0 field, as configured by SCFGR1[ADDRCFG]. This flag is cleared by reading Target Address Status (SASR). This flag cannot generate an asynchronous wakeup. 0 - ADDR0 matching address not received 1 - ADDR0 matching address received
11 FEF	FIFO Error Flag Indicates whether there is a FIFO error. This flag can only be set when clock stretching is disabled. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No FIFO error 1 - FIFO error When writing 0 - No effect 1 - Clear the flag
10 BEF	Bit Error Flag Indicates whether the LPI2C target has transmitted a logic 1 and detects a logic 0 on the I2C bus. The target ignores the rest of the transfer until the next (repeated) Start condition. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No bit error occurred

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Bit error occurred When writing 0 - No effect 1 - Clear the flag
9 SDF	Stop Detect Flag Indicates whether the LPI2C target detects a Stop condition, and if the LPI2C target matched the last address byte. When SCFGR1[SDCFG] = 1, this flag is set on any Stop condition. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No Stop detected 1 - Stop detected When writing 0 - No effect 1 - Clear the flag
8 RSF	Repeated Start Flag Indicates whether the LPI2C target detects a repeated Start condition and if the LPI2C target matched the last address byte. When SCFGR1[RSCFG] = 1, this flag is set on any repeated Start condition. This flag is not set when the target first detects a Start condition. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No repeated Start detected 1 - Repeated Start detected When writing 0 - No effect 1 - Clear the flag
7-4 —	Reserved
3 TAF	Transmit ACK Flag Indicates whether a transmit ACK or NACK is required. You can clear this flag by writing to Target Transmit ACK (STAR). 0 - Not required

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Required
2 AVF	Address Valid Flag Indicates whether the contents of Target Address Status (SASR) are valid. You can clear this flag by reading SASR. When SCFGR1[RXCFG] = 1, this flag is also cleared by reading Target Receive Data (SRDR). 0 - Not valid 1 - Valid
1 RDF	Receive Data Flag Indicates whether receive data is ready. You can clear this flag by reading Target Receive Data (SRDR). When SCFGR1[RXCFG] = 1, this flag is not cleared when reading Target Receive Data (SRDR) if SSR[AVF] = 1. 0 - Not ready 1 - Ready
0 TDF	Transmit Data Flag Indicates whether transmit data has been requested. This flag is cleared by writing to Target Transmit Data (STDR). When SCFGR1[TXCFG] = 0, if a NACK, repeated Start, or Stop condition is detected, this flag is also cleared. 0 - Transmit data not requested 1 - Transmit data is requested

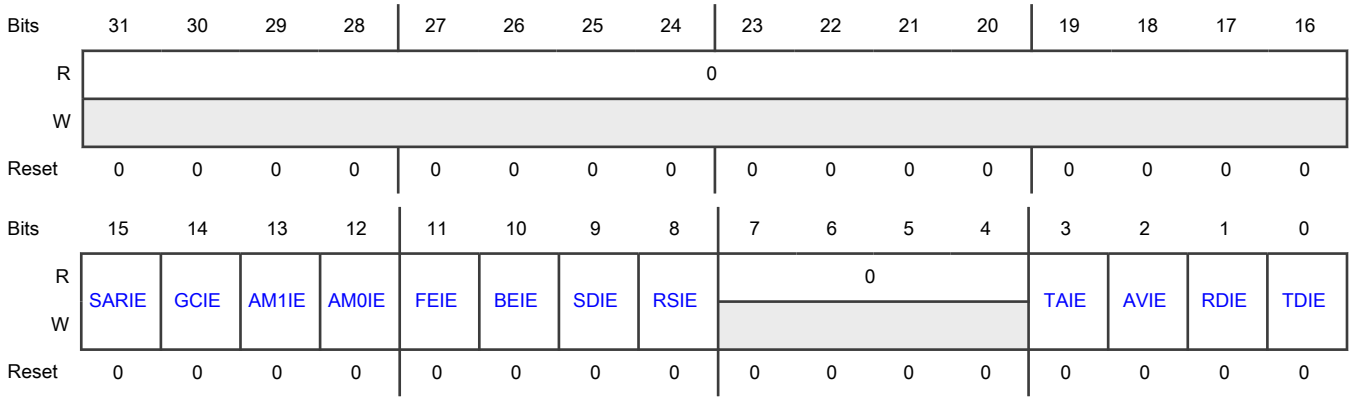
60.7.1.22 Target Interrupt Enable (SIER)

Contains transmit and receive data interrupt enables, start and stop detect interrupt enables, and other target interrupt enables.

Offset

Register	Offset
SIER	118h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 SARIE	SMBus Alert Response Interrupt Enable Enables interrupt for SMBus alert response. 0 - Disable 1 - Enable
14 GCIE	General Call Interrupt Enable Enables interrupt for general call. 0 - Disabled 1 - Enabled
13 AM1IE	Address Match 1 Interrupt Enable Enables interrupt for address match 1. 0 - Disable 1 - Enable
12 AM0IE	Address Match 0 Interrupt Enable Enables interrupt for address match 0. 0 - Disable 1 - Enable
11 FEIE	FIFO Error Interrupt Enable Enables interrupt for FIFO error. 0 - Disable 1 - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
10 BEIE	Bit Error Interrupt Enable Enables interrupt for bit error. 0 - Disable 1 - Enable
9 SDIE	Stop Detect Interrupt Enable Enables interrupt for Stop detection. 0 - Disable 1 - Enable
8 RSIE	Repeated Start Interrupt Enable Enables interrupt for repeated start. 0 - Disable 1 - Enable
7-4 —	Reserved
3 TAIE	Transmit ACK Interrupt Enable Enables interrupt for transmit ACK. 0 - Disable 1 - Enable
2 AVIE	Address Valid Interrupt Enable Enables interrupt for valid address. 0 - Disable 1 - Enable
1 RDIE	Receive Data Interrupt Enable Enables interrupt for receive data. 0 - Disable 1 - Enable
0 TDIE	Transmit Data Interrupt Enable Enables interrupt for transmit data. 0 - Disable 1 - Enable

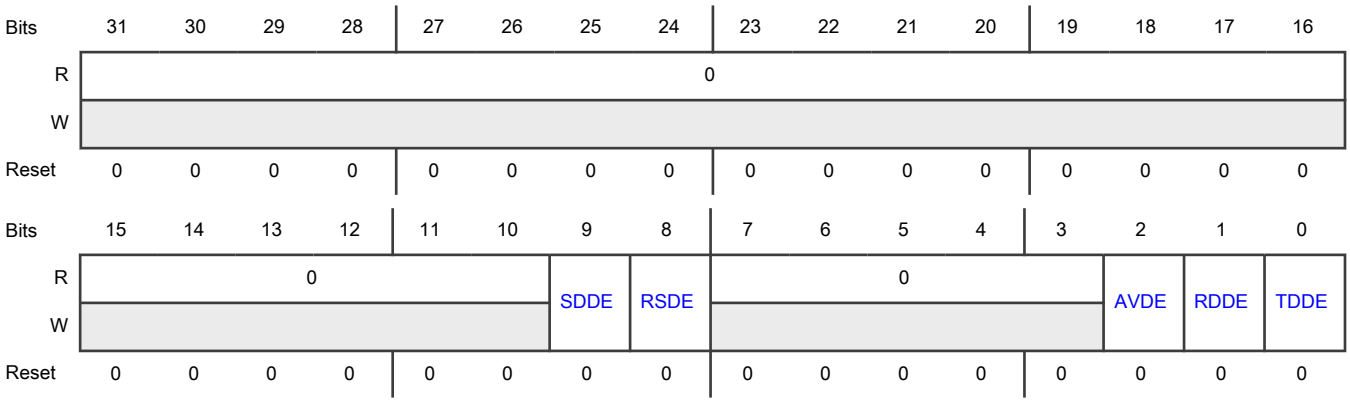
60.7.1.23 Target DMA Enable (SDER)

Contains the transmit, request, and receive enables for DMA.

Offset

Register	Offset
SDER	11Ch

Diagram



Fields

Field	Description
31-10 —	Reserved
9 SDDE	Stop Detect DMA Enable Enables DMA end-of-packet processing on stop detection. Reading Target Receive Data (SRDR) when it is empty: • Generates a DMA end-of-packet response. • Returns 0000_40FFh. • Writes 0 to SSR[SDF]. 0 - Disable 1 - Enable
8 RSDE	Repeated Start DMA Enable Enables DMA end-of-packet processing on repeated start. Reading Target Receive Data (SRDR) when it is empty: • Generates a DMA end-of-packet response. • Returns 0000_40FFh.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Writes 0 to SSR[RDF]. 0 - Disable 1 - Enable
7-3 —	Reserved
2 AVDE	Address Valid DMA Enable Enables address valid DMA request. The address valid DMA request is shared with the receive data DMA request. If both are enabled, write 1 to SCFGR1[RXCFG] to allow the DMA to read the address from Target Receive Data (SRDR). 0 - Disable 1 - Enable
1 RDDE	Receive Data DMA Enable Enables receive data for DMA. 0 - Disable DMA request 1 - Enable DMA request
0 TDDE	Transmit Data DMA Enable Enables transmit data for DMA. 0 - Disable 1 - Enable

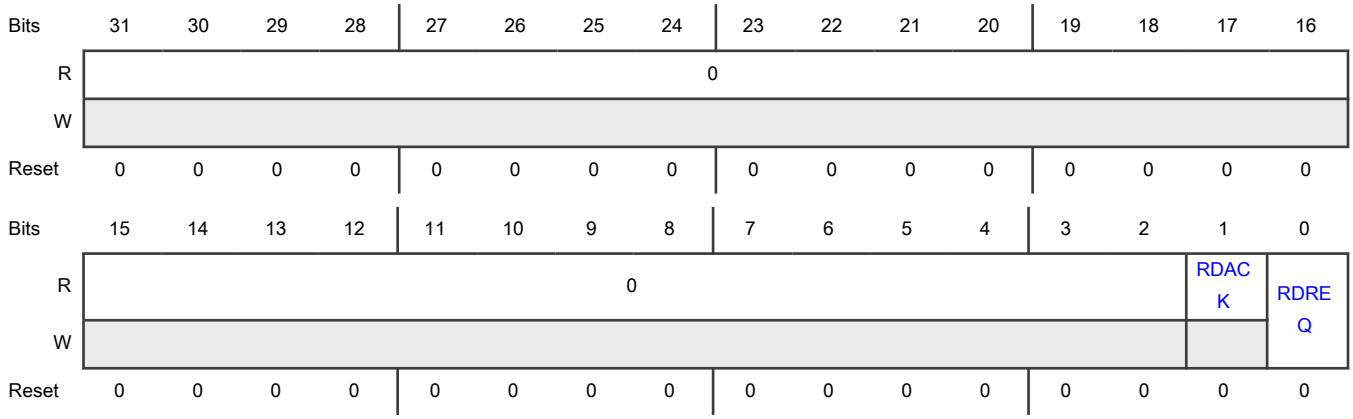
60.7.1.24 Target Configuration 0 (SCFGR0)

Configures the read request feature.

Offset

Register	Offset
SCFGR0	120h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 RDACK	Read Acknowledge Flag Indicates whether a Start then Stop sequence with one SCL pulse between them acknowledged the read request while SCFGR0[RDREQ] = 1. You can clear this flag by writing 0 to SCFGR0[RDREQ]. 0 - Read Request not acknowledged 1 - Read Request acknowledged
0 RDREQ	Read Request Enables read request. When the I2C bus is idle, writing 1 to this field causes the LPI2C target to drive SDA low, triggering a Start condition. The LPI2C target releases SDA on the next falling edge of SCL or when you write 0 to this field. To initiate a second read request (for example, following I2C bus activity that did not acknowledge the request), write 0 then 1 to this field. When the I2C bus is busy, writing to this register always writes 0 to this field. 0 - Disable 1 - Enable

60.7.1.25 Target Configuration 1 (SCFGR1)

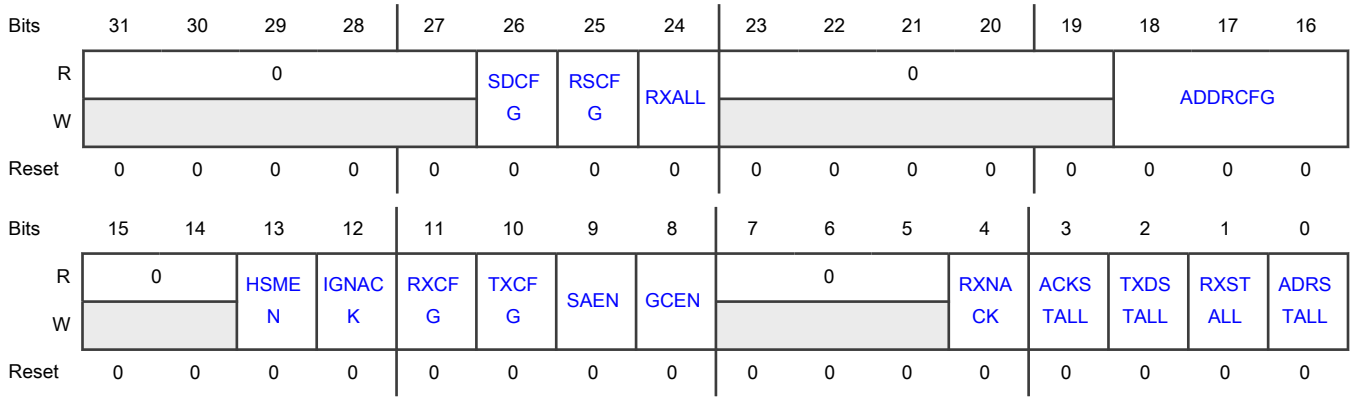
Configures various aspects of the target.

Write to this register only when the I2C target is disabled.

Offset

Register	Offset
SCFGR1	124h

Diagram



Fields

Field	Description
31-27 —	Reserved
26 SDCFG	Stop Detect Configuration Configures the conditions that set SSR[SDF]. 0 - Any Stop condition following an address match 1 - Any Stop condition
25 RSCFG	Repeated Start Configuration Configures the conditions that set SSR[RSF]. 0 - Any repeated Start condition following an address match 1 - Any repeated Start condition
24 RXALL	Receive All Enables receive-all functionality. When enabled, the LPI2C target stores all addresses on the I2C bus in Target Address Status (SASR) and all data on the I2C bus in Target Receive Data (SRDR). However, the LPI2C target does not drive SDA (for ACK or target-transmit transfer) unless there is an address match. The LPI2C target can drive SCL if clock stretching is enabled. When this field is 1, the LPI2C target only supports 7-bit addressing modes (you must configure SCFGR1[ADDRCFG] for 7-bit address match). Software can support 10-bit addresses, however. The first byte of a 10-bit address is saved to Target Address Match (SAMR). The second byte is saved as the first data byte in Target Receive Data (SRDR). Use this field to aid debugging of the I2C bus by saving all data on the bus without altering the state of the bus. When this field is 1, it is recommended to also write 1 to SCFGR1[RSCFG] and SCFGR1[SDCFG] so you can track the full state of the I2C bus. 0 - Disable 1 - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
23-19 —	Reserved
18-16 ADDRCFG	Address Configuration Configures the condition that causes an address to match. 000 - Address match 0 (7-bit) 001 - Address match 0 (10-bit) 010 - Address match 0 (7-bit) or address match 1 (7-bit) 011 - Address match 0 (10-bit) or address match 1 (10-bit) 100 - Address match 0 (7-bit) or address match 1 (10-bit) 101 - Address match 0 (10-bit) or address match 1 (7-bit) 110 - From address match 0 (7-bit) to address match 1 (7-bit) 111 - From address match 0 (10-bit) to address match 1 (10-bit)
15-14 —	Reserved
13 HSMEN	HS Mode Enable Enables detection of the HS mode controller code of target address 0000_1XX, but does not cause an address match on this code. When this field is 1 and any HS mode controller code is detected, SCR[FILTEN] and SCFGR1[ACKSTALL] are ignored until the next Stop condition is detected. 0 - Disable 1 - Enable
12 IGNACK	Ignore NACK Determines whether the target ends transfer when a NACK condition is detected. When this field is 1, the LPI2C target continues transfers after a NACK is detected. This field is required to be 1 in Ultra-Fast mode. 0 - End transfer on NACK 1 - Do not end transfer on NACK
11 RXCFG	Receive Data Configuration Configures which data is returned and which flags are cleared when reading Target Receive Data (SRDR). When this field is 0, reading SRDR returns received data and clears SSR[RDF]. When this field is 1, reading SRDR: • Returns the value of Target Address Status (SASR) and clears SSR[AVF] when SSR[AVF] is set. • Returns received data and clears SSR[RDF] when SSR[AVF] is not set. 0 - Return received data, clear SSR[RDF]

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Return SASR and clear SSR[AVF] when SSR[AVF] is set, return received data and clear SSR[RDF] when SSR[AFV] is not set
10 TXCFG	Transmit Flag Configuration Determines which conditions set SSR[TDF]. This field always becomes 1 before a NACK is detected at the end of a target-transmit transfer. This change can cause an extra word to be written to the transmit data FIFO. When this field is 0, Target Transmit Data (STDR) is automatically emptied when a target-transmit transfer is detected. SSR[TDF] is set when a target-transmit transfer is detected, and SSR[TDF] is cleared at the end of the target-transmit transfer. When this field is 1, SSR[TDF] is set when STDR is empty, and SSR[TDF] is cleared when STDR is full. This setting allows STDR to be filled before a target-transmit transfer is detected. However, it can cause STDR to be written before a NACK is detected on the last byte of a target-transmit transfer. 0 - SSR[TDF] is set only during a target-transmit transfer when STDR is empty 1 - SSR[TDF] is set whenever STDR is empty
9 SAEN	SMBus Alert Enable Enables a match on an SMBus alert. 0 - Disable 1 - Enable
8 GCEN	General Call Enable Enables a general call address. 0 - Disable 1 - Enable
7-5 —	Reserved
4 RXNACK	Receive NACK Determines whether to override the setting of STAR[TXNACK] when the LPI2C receives a matching address during an overrun. When this field is 1, the LPI2C target responds with a NACK under the following conditions: • SSR[AVF] would be set due to matching an address, but that flag is already 1 (address overrun). • SSR[RDF] would be set due to receiving data, but that flag is already 1 (receive data overrun). 0 - ACK or NACK always determined by STAR[TXNACK] 1 - NACK always generated on address overrun or receive data overrun, otherwise ACK or NACK is determined by STAR[TXNACK]
3	ACK SCL Stall

Table continues on the next page...

Table continued from the previous page...

Field	Description
ACKSTALL	Enables SCL clock stretching during target-transmit address bytes and target-receiver address and data bytes, so you can write to Target Transmit ACK (STAR) before the ACK or NACK is transmitted. Clock stretching occurs when transmitting the ninth bit, and is therefore not compatible with HS mode. If this field is 1: • You do not need to write 1 to SCFGR1[RXSTALL] or SCFGR1[ADRSTALL]. • When there is an address match on the first byte of a 10-bit address, SSR[AVF] is set, allowing you to read the received address before writing to Target Transmit ACK (STAR). 0 - Disable 1 - Enable
2 TXDSTALL	Transmit Data SCL Stall Enables SCL clock stretching when SSR[TDF] = 1 during a target-transmit transfer. Clock stretching occurs following the ninth bit, and is therefore compatible with HS mode. 0 - Disable 1 - Enable
1 RXSTALL	RX SCL Stall Enables SCL clock stretching when SSR[RDF] = 1 during a target-receive transfer. Clock stretching occurs following the ninth bit, and is therefore compatible with HS mode. 0 - Disable 1 - Enable
0 ADRSTALL	Address SCL Stall Enables SCL clock stretching when SSR[AVF] = 1. Clock stretching only occurs following the ninth bit, and is therefore compatible with HS mode. 0 - Disable 1 - Enable

60.7.1.26 Target Configuration 2 (SCFGR2)

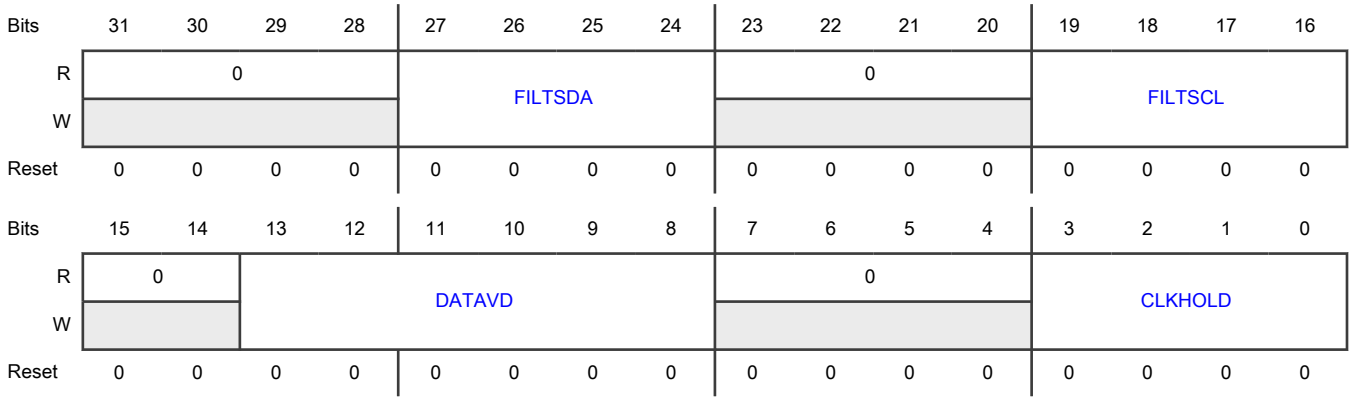
Configures data valid delay, clock hold time, and glitch filters for SDA and SCL.

Write to this register only when the I2C target is disabled.

Offset

Register	Offset
SCFGR2	128h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-24 FILTSDA	Glitch Filter SDA Configures the I2C target digital glitch filters for SDA input. Writing 0 to this field disables the glitch filter. Glitches equal to or less than the number of cycles defined by this field are filtered out and ignored. The latency through the glitch filter is equal to the number of cycles defined by this field + 3. The latency must be configured to be less than the minimum SCL low or high period. MCFGR1[PRESCALE] does not affect the glitch filter cycle count, and the glitch filter cycle count is disabled in HS mode.
23-20 —	Reserved
19-16 FILTSCL	Glitch Filter SCL Configures the I2C target digital glitch filters for SCL input. Writing 0 to this field disables the glitch filter. Glitches equal to or less than the number of cycles defined by this field are filtered out and ignored. The latency through the glitch filter is equal to the number of cycles defined by this field + 3. The latency must be configured to be less than the minimum SCL low or high period. MCFGR1[PRESCALE] does not affect the glitch filter cycle count, and the glitch filter cycle count is disabled in HS mode.
15-14 —	Reserved
13-8	Data Valid Delay

Table continues on the next page...

Table continued from the previous page...

Field	Description
DATAVD	Configures the SDA data valid delay time for the I2C target, which is equal to FILTSCL + DATAVD + 3 cycles. The data valid delay must be configured to be less than the minimum SCL low period. MCFGR1[PRESCALE] does not affect the I2C target data valid delay time, and the I2C target data valid delay time is disabled in HS mode.
7-4 —	Reserved
3-0 CLKHOLD	Clock Hold Time Configures the minimum clock hold time for the I2C target, when clock stretching is enabled. The minimum hold time is equal to the number of cycles defined by this field + 3. MCFGR1[PRESCALE] does not affect the I2C target clock hold time, and the I2C target clock hold time is disabled in HS mode.

60.7.1.27 Target Address Match (SAMR)

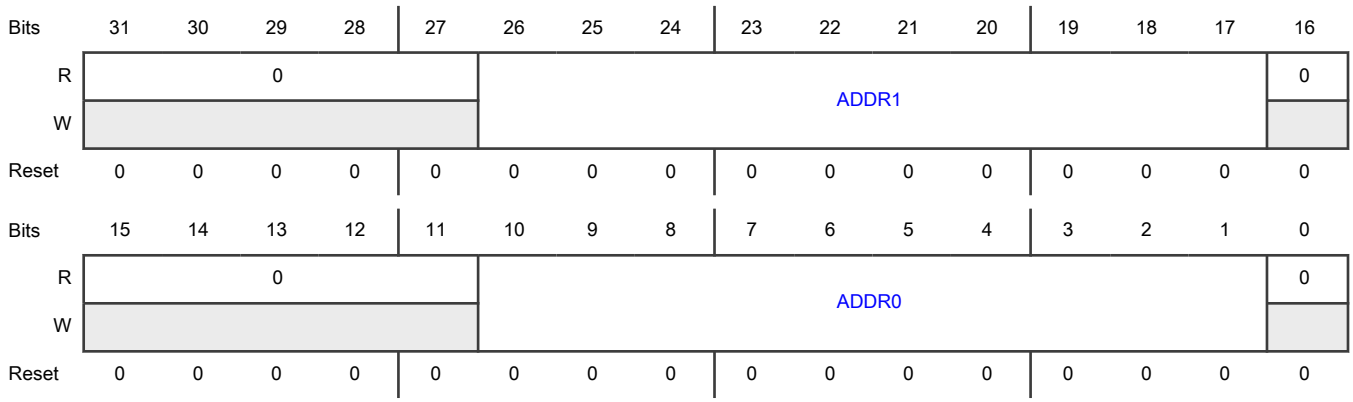
Contains address values for received target match comparison.

Write to this register only when the I2C target is disabled.

Offset

Register	Offset
SAMR	140h

Diagram



Fields

Field	Description
31-27 —	Reserved
26-17 ADDR1	Address 1 Value Contains the value of address 1, which is compared to the received address to detect the target address. In 10-bit mode, the first address byte is compared to {11110, ADDR1[26:25]} and the second address byte is compared to ADDR1[24:17]. In 7-bit mode, the address is compared to ADDR1[23:17].
16-11 —	Reserved
10-1 ADDR0	Address 0 Value Contains the value of address 0, which is compared to the received address to detect the target address. In 10-bit mode, the first address byte is compared to {11110, ADDR0[10:9]} and the second address byte is compared to ADDR0[8:1]. In 7-bit mode, the address is compared to ADDR0[7:1].
0 —	Reserved

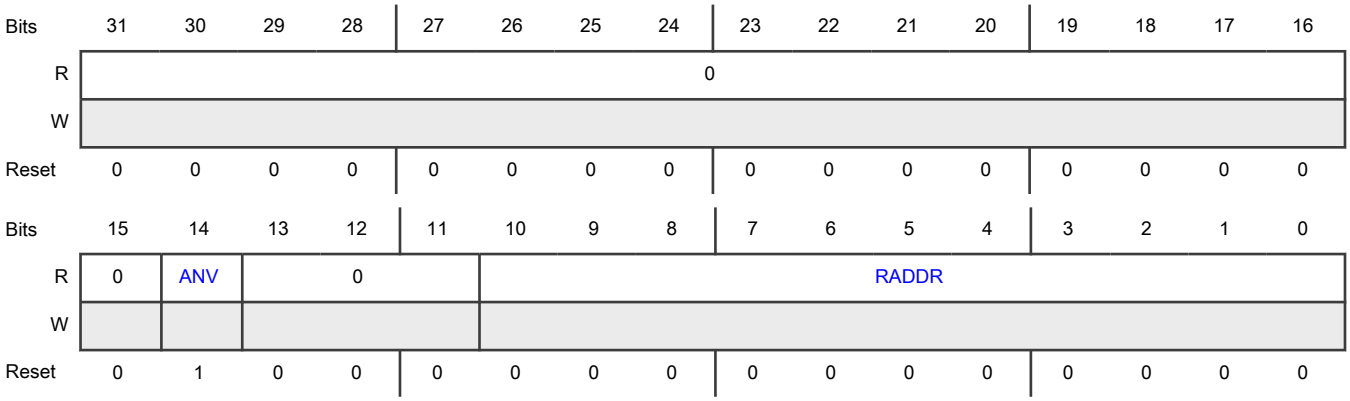
60.7.1.28 Target Address Status (SASR)

Contains the received address and its validity.

Offset

Register	Offset
SASR	150h

Diagram



Fields

Field	Description
31-15 —	Reserved
14 ANV	Address Not Valid Indicates whether SASR[RADDR] is valid. 0 - Valid 1 - Not valid
13-11 —	Reserved
10-0 RADDR	Received Address Contains the received address. Updates whenever SSR[AM0F] or SSR[AM1F] is set. Reading Target Address Status (SASR) clears SSR[AM0F] and SSR[AM1F]. In 7-bit mode, the address byte is stored in RADDR[7:0]. In 10-bit mode, the first address byte is {11110, RADDR[10:9], RADDR[0]} and the second address byte is RADDR[8:1]. The Read-or-Write bit is therefore always stored in RADDR[0]. When SCFGR1[ACKSTALL] = 1, if the first address byte matches in 10-bit mode, the first address byte is stored in RADDR[7:0] so you can read this field before writing the Transmit ACK. If the second address byte matches, this field is then updated with the full 10-bit address.

60.7.1.29 Target Transmit ACK (STAR)

Configures choice of ACK or NACK on each received word.

You can write to this register only when SCFGR1[ACKSTALL] = 1.

SCFGR1[ACKSTALL] enables clock stretching during the ACK-or-NACK bit slot. During this time, you can write to this register.

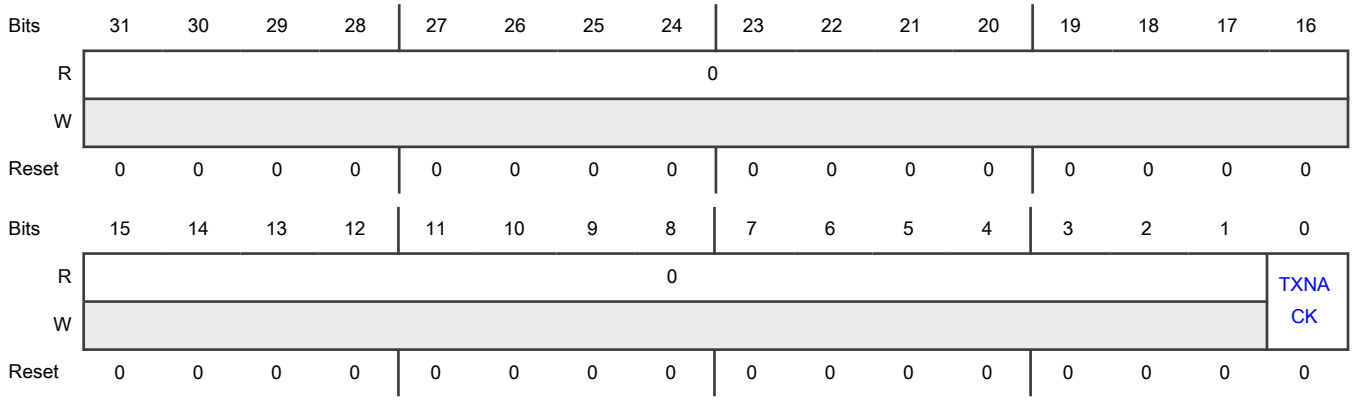
The logic ensures that the clock stretching continues for at least one bus clock cycle after this register is updated.

This clock stretching time can be extended via SCFGR2[CLKHOLD].

Offset

Register	Offset
STAR	154h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 TXNACK	Transmit NACK Selects whether transmit ACK (logic 0) or NACK (logic 1) is returned on the bus by the I2C target after receiving each word. • When SCFGR1[ACKSTALL] = 1, a transmit NACK signal must be written once for each matching address byte and each received word. SCFGR1[ACKSTALL] must be 1, because that setting stalls the data transfer until software reads the received word (and determines whether to respond with an ACK or NACK). • To configure the default (ACK or NACK), you can write to this field when LPI2C target is disabled or idle. 0 - Transmit ACK 1 - Transmit NACK

60.7.1.30 Target Transmit Data (STDR)

Contains the I2C target data to transmit.

Clock stretching (enabled or disabled) affects when the transmit data is transferred. SCFGR1[TXDSTALL] enables clock stretching during the first data bit of a target-transmit transfer.

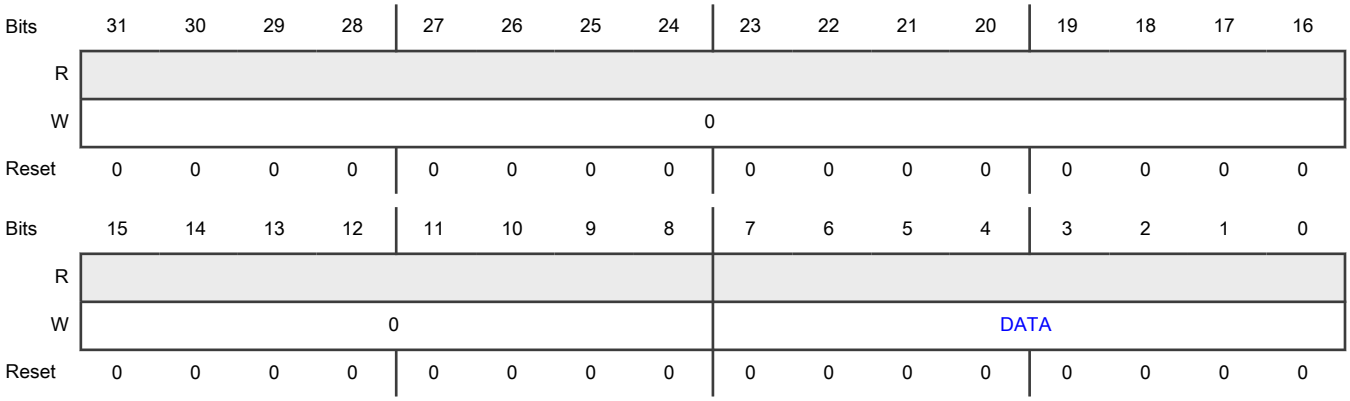
If clock stretching is enabled (SCFGR1[TXDSTALL] = 1), the transmit data transfer is stalled until this register is updated. Clock stretching is extended by at least 1 bus clock cycle after this register is updated. Clock stretching can be delayed further by using SCFGR2[CLKHOLD].

If clock stretching is disabled (SCFGR1[TXDSTALL] = 0), the transmit data must be written before the start of the target-transmit transfer, otherwise SSR[FEF] is set.

Offset

Register	Offset
STDR	160h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 DATA	Transmit Data Contains the I2C target data to transmit. Writing data to this register stores I2C target transmit data in this register.

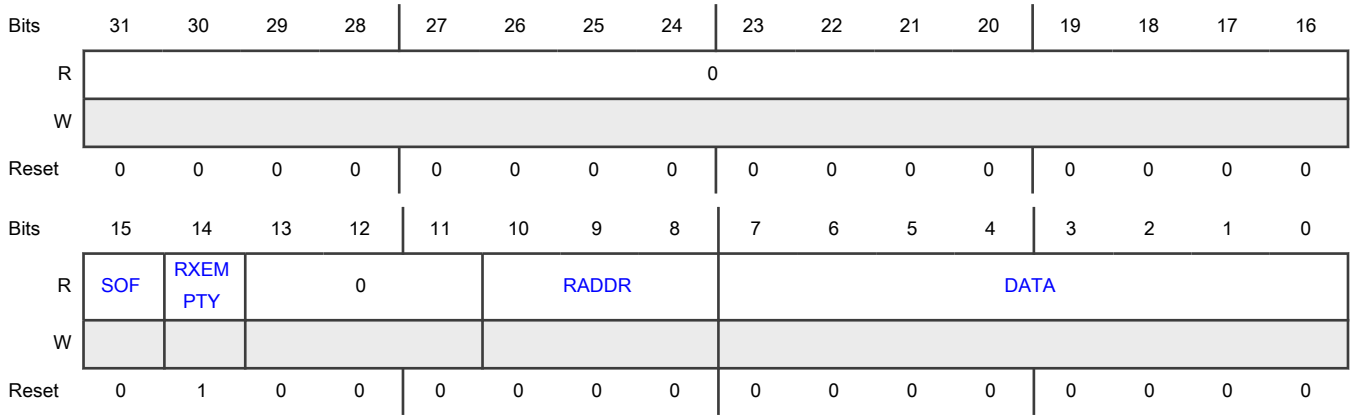
60.7.1.31 Target Receive Data (SRDR)

Contains status of target receive data transfer.

Offset

Register	Offset
SRDR	170h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 SOF	Start of Frame Indicates whether this data word is the first data word since a (repeated) Start or Stop condition. 0 - Not first 1 - First
14 RXEMPTY	Receive Empty Indicates whether this register is empty. 0 - Not empty 1 - Empty
13-11 —	Reserved
10-8 RADDR	Received Address Contains the address received by the IC2 target. When both SCFGR1[RXCFG] and SSR[AVF] are 1, bits [10:8] of SASR[RADDR] are returned. Otherwise, this field returns zero.
7-0 DATA	Received Data Contains the data received by the I2C target. When both SCFGR1[RXCFG] and SSR[AVF] are 1, bits [7:0] of SASR[RADDR] are returned.

60.7.1.32 Target Receive Data Read Only (SRDROR)

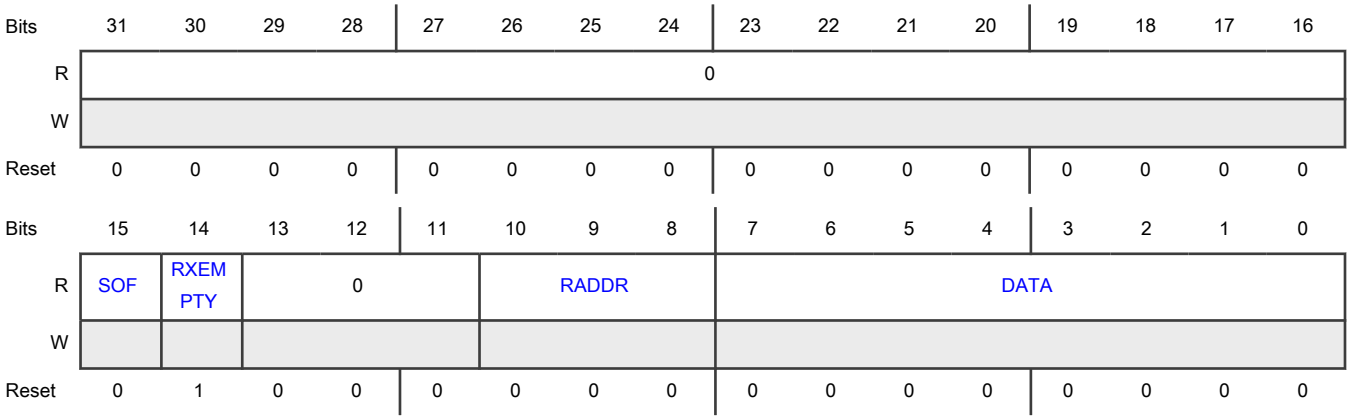
Contains the data received by the I2C target.

Reading this register returns the data received by the I2C target, but does not pull the data from the register.

Offset

Register	Offset
SRDROR	178h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 SOF	Start of Frame Indicates whether this data word is the first data word since a (repeated) Start or Stop condition. 0 - Not the first 1 - First
14 RXEMPTY	Receive Empty Indicates whether Target Receive Data (SRDR) is empty. 0 - Not empty 1 - Empty
13-11 —	Reserved
10-8 RADDR	Received Address Contains address received by the LPI2C target. When both SCFGR1[RXCFG] and SSR[AVF] are 1, bits [10:8] of SASR[RADDR] are returned. Otherwise, this field returns zero.
7-0	Receive Data

Table continues on the next page...

Table continued from the previous page...

Field	Description
DATA	Contains data received by the LPI2C target. When both SCFGR1[RXCFG] and SSR[AVF] are 1, bits [7:0] of SASR[RADDR] are returned.

60.7.1.33 Controller Transmit Command Burst (MTCBR0 - MTCBR127)

Contains transmit Command and Data fields to support burst transfers.

Is an alias of the Controller Transmit Data Register designed to support incrementing burst transfers to the transmit FIFO by a DMA controller using aligned 8-bit, 16-bit, or 32-bit writes. The size of the Controller Transmit Data Burst Register is 512-bytes.

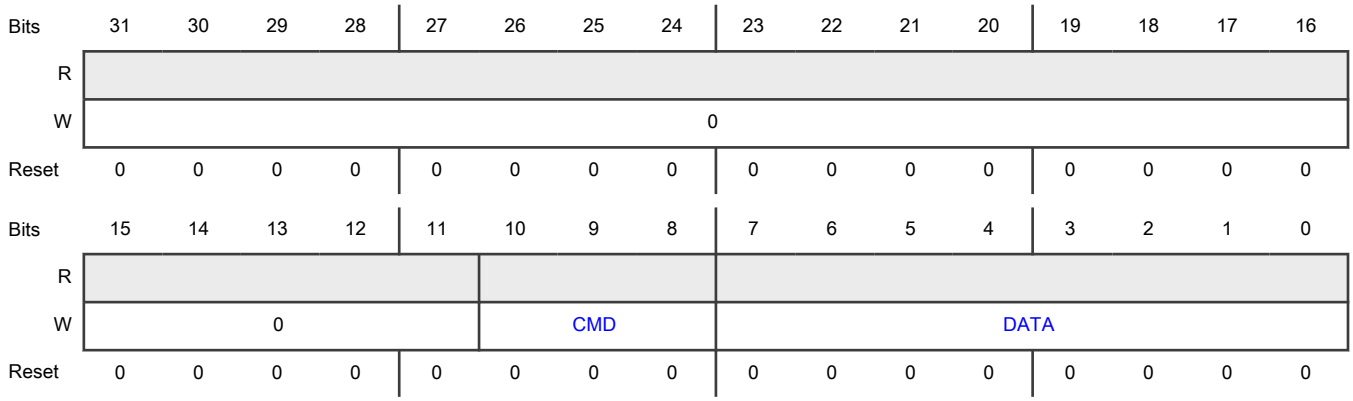
- An aligned 32-bit write in this region pushes one entry into the transmit FIFO.
- An aligned 16-bit write in this region to MTCBRn[15:0] pushes one entry into the transmit FIFO.
- An 8-bit write in this region to MTCBRn[7:0] updates DATA[7:0], but does not push the data into the transmit FIFO.
- An 8-bit write in this region to MTCBRn[15:8] pushes the data written to CMD[2:0] plus the previously written DATA[7:0] into the transmit FIFO.
- An 8-bit or 16-bit write in this region to MTXBRn[31:16] is ignored.

Offset

For n = 0 to 127:

Register	Offset
MTCBRn	200h + (n × 4h)

Diagram



Fields

Field	Description
31-11 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
10-8 CMD	Command Command is written to the Controller Transmit Data Register.
7-0 DATA	Data Data is written to the Controller Transmit Data Register.

60.7.1.34 Transmit Data Burst (MTDBR0 - MTDBR255)

Contains transmit Data fields to support burst transfers.

Is an alias of the Transmit Data Register designed to support incrementing burst transfers to the transmit FIFO by a DMA controller using 8-bit, 16-bit, or 32-bit writes. The CMD field is always zero-extended for transmit data. The size of the Transmit Data Burst Register is 1024 bytes.

- An aligned 32-bit write in this region pushes four zero-extended DATA entries into the transmit FIFO (DATA0 byte first). Note that the register access is extended by 3 wait states.
- An aligned 16-bit write in this region to either half of a 32-bit word pushes two zero-extended DATA entries into the transmit FIFO (DATA0 or DATA2 first). Note that the register access is extended by 1 wait state.
- An 8-bit write in this region pushes one zero-extended DATA entry into the transmit FIFO.

Offset

For n = 0 to 255:

Register	Offset
MTDBRn	400h + (n × 4h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DATA3								DATA2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA1								DATA0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 DATA3	Data Data is written to the MTDR[DATA] with MTDR[CMD] zero-extended.
23-16 DATA2	Data Data is written to the MTDR[DATA] with MTDR[CMD] zero-extended.
15-8 DATA1	Data Data is written to the MTDR[DATA] with MTDR[CMD] zero-extended.
7-0 DATA0	Data Data is written to the MTDR[DATA] with MTDR[CMD] zero-extended.

60.8 Glossary

HREQ	Host request
SCL	Serial clock line
SCLS	Secondary serial clock line
SDA	Serial data line
SDAS	Secondary serial data line

Chapter 61
Low Power Serial Peripheral Interface (LPSPI)

61.1 Chip-specific LPSPI information

Table 570. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing

The chip includes eight instances of LPSPI Modules. LPSPI1 and LPSPI2 are in AONMIX. LPSPI3-8 in WAKEUPMIX. SPI1 and SPI2 are only supported in Master Mode, not Slave Mode.

HREQ host request feature to allow HW triggers to initiate the start of an I2C request is not supported.

The table below summarizes the number of peripheral chip selects supported by each instance of LPSPI.

Table 571. Number of Peripheral Chip selects for each LPSI instance

MIX	Instance	PCS_NUM
AONMIX	LPSPI1	2
AONMIX	LPSPI2	2
WAKEUPMIX	LPSPI3	2
	LPSPI4	3
	LPSPI5	2
	LPSPI6	2
	LPSPI7	2
	LPSPI8	2

61.2 Overview

LPSPI provides an efficient interface (either as a controller or peripheral) to an SPI bus, which is a synchronous serial communication interface used in embedded systems. It is typically used to perform short distance communications between microcontrollers and peripheral devices, on printed circuit boards. Typical applications include interfacing with secure digital cards and LCD displays.

NOTE

The terminology in this chapter has been updated to align with NXP's inclusive language standards, as shown in the table below.

Table 572. Updated terms

Updated term	Deprecated term
Controller	Master
Peripheral	Slave

61.2.1 Block diagram

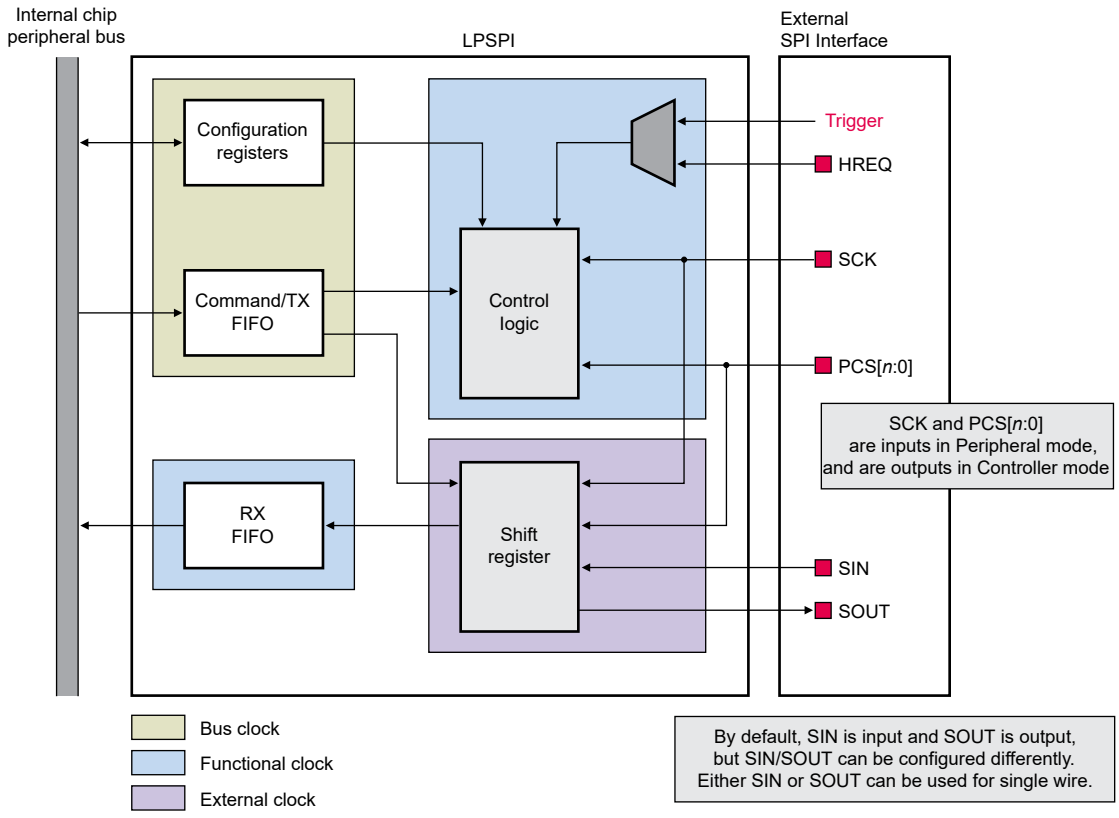


Figure 358. Block diagram

61.2.2 Features

- Minimal CPU overhead, with DMA transmit and receive requests supporting FIFO register accesses
- Operation continues in Stop mode, if configured to do so and an appropriate clock is available
- Support available for 32-bit word size
- Configurable clock polarity and phase
- Support available for 2 peripheral chip selects in Controller mode
- Support available for Peripheral mode
- 8-word transmit and command FIFO
- 8-word receive FIFO

- Flexible timing parameters in Controller mode, including SCK frequency and duty cycle, and delays between PCS and SCK edges
- Continuous transfer option to keep PCS asserted across multiple frames
- Full-duplex transfers that support 1-bit transmit and receive on each clock edge
- Half-duplex transfers that support:
  - 1-bit transmit or receive on each clock edge
- Receive data match logic that discards nonmatching data and interrupt on data match

61.3 Functional description

61.3.1 Controller mode

61.3.1.1 Transmit and command FIFO commands

The transmit and command FIFO is a combined FIFO that includes both transmit data words and command words. You store:

- Transmit data words in the transmit and command FIFO, by writing to Transmit Data (TDR).
- Command words in the transmit and command FIFO, by writing to Transmit Command (TCR).

When a command word is at the top of the transmit and command FIFO, the actions that can occur depend on whether LPSPI is busy or between frames (see TCR[CONT] and TCR[CONTC]). See Table 573 for conditions and possible corresponding actions when a command word is at the top of the transmit and command FIFO.

Table 573. Possible actions when a command word is at the top of the transmit and command FIFO

Condition	Action
LPSPI is enabled and idle.	The command word is pulled from the FIFO, and this command word controls all subsequent transfers.
LPSPI is busy and TCR[CONTC] is 0.	The SPI frame completes at the end of the existing word, ignoring TCR[FRAMESZ]. The command word is then pulled from the FIFO and that command word controls all subsequent transfers (or until the next update to the command word). Note that a command word with TCR[CONTC] = 0 always terminates the existing transfer regardless of the previous TCR[CONT] value.
LPSPI is busy; the existing TCR[CONT] value is 1 and the new TCR[CONTC] value is 1.	The command word must be updated at the frame boundary. The command word is pulled from the FIFO during the last SCK pulse of the existing frame (based on the value of FRAMESZ), and the frame continues using the new command value for the rest of the frame (or until the next update to the command word). When TCR[CONTC] = 1, only the lower 24 bits of the command word are updated. If the command word is updated at a word boundary, then the transfer halts (stops) after that word. TCR[CONTC] is ignored when not at a frame boundary, so the frame ends prematurely.

TCR[CONT] = 1 keeps PCS asserted at end of frame, allowing the transfer to continue.

TCR[CONTC] = 1 specifies that this command word must not terminate the existing frame, and the transfer can continue using the new command word.

TCR[CONTC] = 1 is restricted in the sense that the new command must load on a frame boundary, and the only way for a transfer to continue from a frame boundary is when the previous command has TCR[CONT] = 1.

You can read the current state of the existing command word from Transmit Command (TCR). It requires at least three LPSPI functional clock cycles for Transmit Command (TCR) to update after you write to it (assuming an empty FIFO), and LPSPI must be enabled (CR[MEN] = 1).

Writing to Transmit Command (TCR) does not initiate an SPI bus transfer, unless TCR[TXMSK] = 1. When TCR[TXMSK] = 1, a new command word is not loaded until the end of the existing frame (based on the value of TCR[FRAMESZ]); at the end of the transfer, TCR[TXMSK] transitions to 0.

In Controller mode, the LPSPI command word in Transmit Command (TCR) controls SPI attributes based on the selections in register fields. See Table 574 for TCR fields and associated functionality related to data transfer.

Table 574. Command word in Controller mode

Transmit Command (TCR)		Description	Can this field be modified during a data transfer?
Field	Name		
CPOL	Clock polarity	Specifies the polarity of the SCK pin. Any change of CPOL value causes a transition on the SCK pin.	N
CPHA	Clock phase	Specifies the clock phase of the transfer.	N
PRESCALE	Prescaler value	Specifies a prescaler used to divide the LPSPI functional clock, to generate the timing parameters of the SPI bus transfer. Changing PRESCALE in conjunction with PCS enables LPSPI to connect to different peripheral devices at different frequencies.	N
PCS	Peripheral chip select	Specifies which PCS pin asserts for the transfer; the polarity of PCS is static and specified by CFGR1[PCSPOL].	N
LSBF	LSB first	Specifies whether LSB (bit 0) or MSB (bit 31 for a 32-bit word) is transmitted or received first.	Y
BYSW	Byte swap	Enables byte swap on each 32-bit word when transmitting and receiving data. Byte swapping can be useful when interfacing with devices that organize data as big-endian.	Y
CONT	Continuous transfer	Configures LPSPI for a continuous transfer that keeps PCS asserted between frames (as specified by FRAMESZ). You must write a new command word to cause PCS to negate. Also, this field supports changing the command word at frame size boundaries.	Y
CONTC	Continuing command	Indicates that this is a new command word for the existing continuous transfer. When CONTC = 1, the command word must only be written to the transmit and command FIFO on a frame boundary.	Y
RXMSK	Receive data mask	Masks the receive data and does not store the masked receive data in the receive FIFO or perform receive data matching. This option is useful for half-duplex transfers or to specify which fields are compared during receive data matching.	Y
TXMSK	Transmit data mask	Masks the transmit data; masked transmit data is not pulled from the transmit FIFO, and the output data pin is 3-stated (unless otherwise configured by CFGR1[OUTCFG]). This option is useful for half-duplex transfers.	Y
FRAMESZ	Frame size	Configures the frame size in number of bits equal to (FRAMESZ + 1): • The minimum frame size is 8 bits. • If the frame size is larger than 32 bits, then the frame is divided into multiple words of 32 bits; each word is loaded from the transmit FIFO and stored in the receive FIFO separately.	Y

Table continues on the next page...

Table 574. Command word in Controller mode (continued)

Transmit Command (TCR)		Description	Can this field be modified during a data transfer?
Field	Name		
		• If the size of the frame is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO contains the remaining bits. For example, a 72-bit transfer consists of three words: the first and second words are 32 bits, and the third word is 8 bits.	

61.3.1.1.1 SPI bus transfers

LPSPI initiates an SPI bus transfer when all these conditions are true:

- Data is written to the transmit FIFO.
- LPSPI is enabled.

To perform the SPI bus transfer, LPSPI uses the attributes configured in Transmit Command (TCR) and the timing parameters defined in Clock Configuration (CCR).

The SPI bus transfer ends after the number of bits indicated by the value of FRAMESZ have been transferred (provided CONT = 0), or at the end of a word when a new transmit command word is at the top of the transmit and command FIFO. When LPSPI is disabled, the SPI bus transfers end after the transmit FIFO is empty and LPSPI is idle.

61.3.1.1.2 Circular FIFO

The transmit and command FIFO supports a circular FIFO feature. This feature enables the LPSPI controller to (periodically) repeat a short data transfer that fits within the transmit and command FIFO, without requiring additional FIFO accesses. When the circular FIFO is enabled (CFGR0[CIRFIFO] = 1), the current state of the FIFO read pointer is saved and the status flags are not updated. After the FIFO is empty and LPSPI is idle, the FIFO read pointer is restored with the saved version, so the contents of the transmit and command FIFO are not permanently pulled from the FIFO when Circular FIFO mode is enabled.

61.3.1.2 Receive FIFO and data match

The receive FIFO stores received data during SPI bus transfers. When TCR[RXMSK] = 1, the received data is discarded instead of being stored in the receive FIFO:

- Received data is written to the receive FIFO when the last bit of the word is sampled.
- If the transmit FIFO is empty during a multiple-word or continuous transfer, then the receive data is written to the receive FIFO before the transfer stalls (assuming CFGR1[NOSTALL] = 0) while waiting for new transmit data or for a command word to be written.

LPSPI provides a receive data match function that can match received data against one of the two words in DMR0 and DMR1, or against a masked data word. You can also configure the received data match function to compare only the first one or two received data words since the start of the frame:

- Received data that is already discarded because of TCR[RXMSK] cannot cause the data match flag to set, and delays the receive data match on the first received data word, until all discarded data is received.
- You can configure the receive data match function to discard all received data until a data match is detected, using CFGR0[RDMO].
- After a receive data match, to allow all subsequent data to be received, write 0 to CFGR0[RDMO], and then write 0 to SR[DMF].

61.3.1.3 Timing parameters

The timing parameters that are used for all SPI bus transfers are relative to the LPSPI functional clock divided by the selection specified in TCR[PRESCALE]. Although you can only change Clock Configuration (CCR) when LPSPI is disabled (CR[MEN] = 0), to support interfacing with different peripheral devices at different frequencies, you can change the TCR[PRESCALE] selection between SPI bus transfers by using Transmit Command (TCR).

NOTE

The minimum value shown in Table 575 is the minimum counter value, but the values of Clock Configuration (CCR) must also satisfy the data sheet specs based on the LPSPI functional clock frequency and prescaler value.

Table 575. Timing parameters

Clock Configuration (CCR) Clock Configuration 1 (CCR1)		Description	Minimum value	Maximum value
Field	Name			
SCKSET	SCK setup phase	Configures the SCK setup phase to (SCKSET + 1) cycles. The setup phase is the SCK high period when either CPHA = 0 and CPOL = 1, or CPHA = 1 and CPOL = 0. Otherwise, it is the SCK low period. The SCK period is defined as (SCKSET + SCKHLD + 2) and the duty cycle is the difference between SCKSET and SCKHLD.	0 (1 cycle)	255 (256 cycles)
SCKHLD	SCK hold phase	Configures the SCK hold phase to (SCKHLD + 1) cycles. The hold phase is the SCK low period when either CPHA = 0 and CPOL = 1, or CPHA = 1 and CPOL = 0. Otherwise, it is the SCK high period. The SCK period is defined as (SCKSET + SCKHLD + 2) and the duty cycle is the difference between SCKSET and SCKHLD.	0 (1 cycle)	255 (256 cycles)
PCSPCS	PCS-to-PCS delay	Configures the minimum delay between PCS negation and the next PCS assertion to (PCSPCS + PCSPCS + 2) cycles. When the command word is updated between transfers, there is a minimum of (PCSPCS + 1) cycles between the command word update and any change on PCS pins.	0 (2 cycles)	255 (512 cycles)
SCKSCK	SCK-to-SCK delay	Configures the delay during a continuous transfer between the last SCK edge of a frame and the first SCK edge of the continuing frame to (SCKSCK + 1) cycles. This is useful when the external peripheral requires a large delay between different words of an SPI bus transfer.	0 (1 cycle)	255 (256 cycles)
PCSSCK	PCS-to-SCK delay	Configures the minimum delay between PCS assertion and the first SCK edge to (PCSSCK + 1) cycles.	0 (1 cycle)	255 (256 cycles)
SCKPCS	SCK-to-PCS delay	Configures the minimum delay between the last SCK edge and the PCS negation to (SCKPCS + 1) cycles.	0 (1 cycle)	255 (256 cycles)

Figure 359 shows the timing settings controlled by:

- TCR[CPHA]
- TCR[CPOL]
- CCR[SCKPCS]
- CCR[PCSSCK]
- CCR1[SCKSET]
- CCR1[SCKHLD]

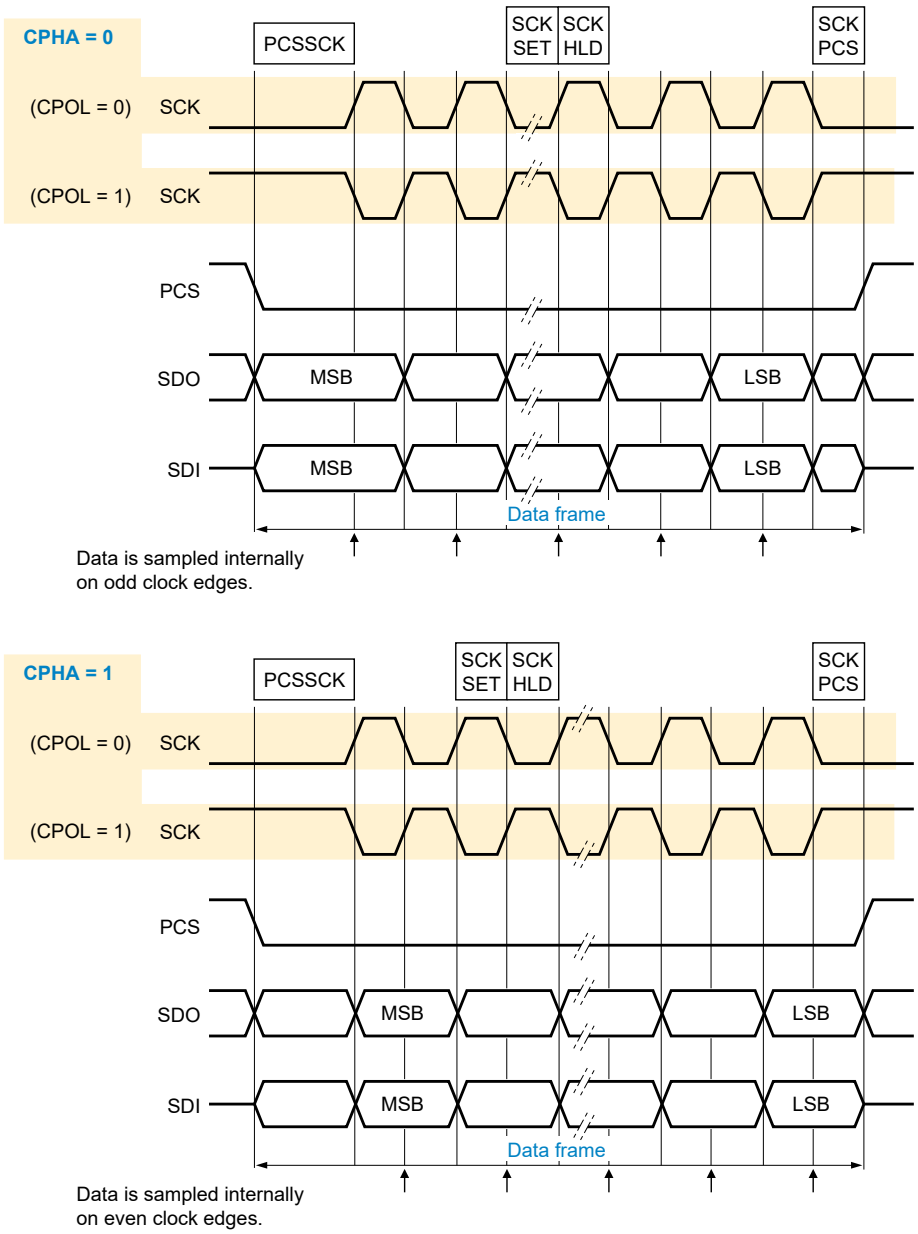


Figure 359. Clock phase (TCR[CPHA]) timing diagram example

To configure for a baud rate of 10 MHz with 50/50 duty cycle and with a functional clock frequency of 100 MHz, use the following settings:

- CCR1[SCKSET] = 0x4 (5 cycles)

- CCR1[SCKHLD] = 0x4 (5 cycles)
- CCR1[PCSPCS] = 0x8 (10 cycles)
- CCR1[SCKSCK] = 0x4 (5 cycles)
- CRR[PCSSCK] = 0x4 (5 cycles)
- CRR[SCKPCS] = 0x4 (5 cycles)
- TCR[PRESCALE] = 0x0 (divide by 1)

61.3.1.4 Pin configuration

Following are the pin configuration settings for half-duplex transfers:

- To swap directions or to support half-duplex transfers on the same pin, you can configure the SIN and SOUT pins using CFGR1[PINCFG].
- When configuring half-duplex transfers, you must configure the output data pins to 3-state when PCS is negated (CFGR1[OUTCFG] = 1).

61.3.1.5 Clock loopback

Configure the LPSPI controller to use one of the following clocks to sample the input data:

- The SCK output clock
- A delayed version of the SCK output clock

The delayed version of the SCK is chosen by the SCK pin output delay, plus the SCK pin input delay, and is selected by writing 1 to CFGR1[SAMPLE]. Enabling the loopback version of the SCK pin can improve the setup time of the input data from the peripheral.

See the chip data sheet for the specific input setup time in Controller Loopback mode.

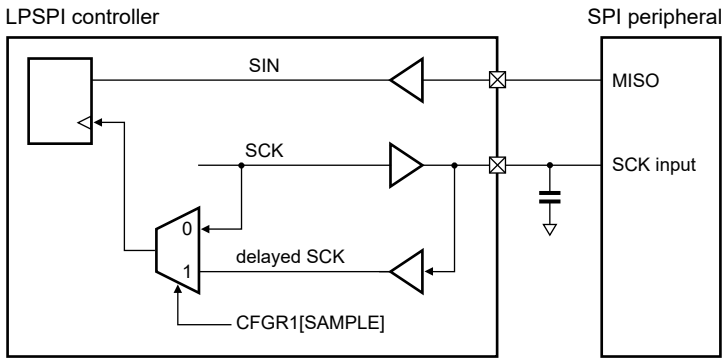


Figure 360. Clock loopback

61.3.2 Peripheral mode

LPSPI Peripheral mode:

- Uses the same shift register and logic that Controller mode uses.
- Does not use Clock Configuration (CCR).
- Requires Transmit Command (TCR) to remain static (unchanged) during SPI bus transfers.

61.3.2.1 Transmit and command FIFO commands

You must initialize Transmit Command (TCR) before enabling LPSPI in Peripheral mode, although this register is not updated until after LPSPI is enabled. After LPSPI is enabled, you must make changes to this register only when LPSPI is idle. In Peripheral mode, the LPSPI command word in this register controls SPI attributes. Before the PCS input asserts, the transmit FIFO must be filled with transmit data, or the transmit error flag sets.

Table 576. Command word in Peripheral mode

Transmit Command (TCR)		Description
Field	Name	
CPOL	Clock polarity	Specifies the polarity of the external SCK input.
CPHA	Clock phase	Specifies the clock phase of transfer.
PRESCALE	Prescaler value	Specifies the LPSPI functional clock prescaler.
PCS	Peripheral chip select	Specifies which PCS is used. The polarity of PCS is static and configured by CFGR1[PCSPOL].
LSBF	LSB first	Specifies whether LSB (bit 0) or MSB (bit 31 for a 32-bit word) is transmitted or received first.
BYSW	Byte swap	Enables byte swap on each 32-bit word when transmitting and receiving data. Byte swapping can be useful when interfacing with devices that organize data as big-endian.
CONT	Continuous transfer	When continuous transfer is selected in Peripheral mode, after the number of bits indicated by FRAMESZ are transferred, LPSPI passes through and transmits the received data until the next PCS negation. Whatever is shifted in on the receive data is shifted out as transmit data considering that there is a 32-bit shift register.
CONTC	Continuing command	When the continuing command is enabled in Peripheral mode, after the number of bits indicated by FRAMESZ are transferred, RXMSK is considered equal to 1 and TXMSK is considered equal to 0 until the next PCS negation. CONTC can be used to change the direction of a transfer after the number of bits indicated by FRAMESZ.
RXMSK	Receive data mask	Masks the receive data; LPSPI does not store masked receive data in the receive FIFO or perform receive data matching. This option is useful for half-duplex transfers or to specify which fields are compared during receive data matching.
TXMSK	Transmit data mask	Masks the transmit data so that the masked transmit data is not pulled from transmit FIFO, and the output data pin is 3-stated (unless otherwise specified in CFGR1[OUTCFG]). This option is useful for half-duplex transfers.
FRAMESZ	Frame size	Specifies the frame size in number of bits equal to (FRAMESZ + 1): • The minimum frame size is 8 bits. • If the frame size is larger than 32 bits, then the frame is divided into multiple words of 32 bits; each word is loaded from the transmit FIFO and stored in the receive FIFO separately. • If the size of the frame is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO contain the remainder bits. For example, a 72-bit transfer consists of three words: the first and second words are 32 bits, and the third word is 8 bits.

61.3.2.2 Receive FIFO and data match

The receive FIFO stores receive data during SPI bus transfers. When TCR[RXMSK] = 1, the received data is discarded instead of storing the received data in the receive FIFO.

Receive data supports a receive data match function that can match received data against one of the two words in DMR0 and DMR1 or against a masked data word. You can also configure the data match function to compare only the first one or two received data words since the start of the frame.

- Received data that is already discarded because TCR[RXMSK] = 1 cannot cause the data match to set, and delays the match on the first received data word, until all discarded data is received.
- By using CFGR0[RDMO], you can also configure the receiver match function to discard all received data until a data match is detected.
- After a receive data match, to allow all subsequent data to be received, first write 0 to CFGR0[RDMO], then clear SR[DMF].

61.3.2.3 Partial received word

When the PCS pin deasserts and the receive shift register shifts in a partial word, you can configure the receive shift register to either discard the partial word or to store it in the receive FIFO. You must specify this using CFGR1[PARTIAL].

A partial word is defined as less than TCR[FRAMESZ] bits (when TCR[FRAMESZ] is equal or less than 32 bits, or it is the last word in a multi-word frame) or less than 32 bits (when TCR[FRAMESZ] is greater than 32 bits and not the last word in a multi-word frame).

A single-bit frame is not supported. A partial received word of 1 bit is supported, but a partial received frame of 1 bit is not supported.

61.3.2.4 Clocked interface

LPSPI supports interfacing with external controllers that provide only clock and data pins (PCS is not required). This interface requires:

- Writing 1 to TCR[CPHA] (data is changed on the leading edge of SCK and captured on the following edge).
- Configuring the PCS input to be always asserted (CFGR1[PCSPOL*n*] = 1). For example, to configure PCS[0] to be always asserted, write 1 to PCSPOL[0], and do not configure PCS[0] in the pin muxing. The chip-level drives PCS to a certain value (ideally 1); you could use CFGR1[PCSPOL*n*] to invert that value.
- Writing 1 to CFGR1[AUTOPCS] to enable automatic PCS generation. When CFGR1[AUTOPCS] = 1, a minimum of four LPSPI functional clock cycles (divided by the selection specified in TCR[PRESCALE]) is required between the last SCK edge of one word and the first SCK edge of the next word.

61.3.3 Low-power modes

LPSPI remains functional during low-power modes, if CR[DOZEN] = 0 and LPSPI uses an external or internal clock source that remains enabled. LPSPI can generate an interrupt or DMA request to cause a wake-up from low-power modes.

61.3.4 Debug mode

Table 577. Debug mode

Chip mode	LPSPI operation
Debug (the core is in Debug or Halted mode)	Can continue operating in Debug mode, if CR[DBGEN] = 1

61.3.5 Clocking

Table 578. LPSPI clocks

Type of clock	Description
Functional	• Asynchronous to the bus clock. • If the LPSPI functional clock remains enabled in low-power modes, then LPSPI can perform SPI bus transfers and low-power wakeups in both Controller and Peripheral modes. • LPSPI divides the functional clock by a prescaler; the resulting frequency must be at least two times faster than the SPI external clock frequency (SCK).
External	• The LPSPI shift register is clocked directly by the SCK clock. • How the SCK clock is generated or supplied depends on the mode (Controller or Peripheral): — In Controller mode, the SCK clock is generated internally. — In Peripheral mode, the SCK clock is supplied externally.
Bus	The bus clock is only used for bus accesses to the LPSPI control and configuration registers. The bus clock frequency must be high enough to support the data bandwidth requirements of the LPSPI registers, including the FIFOs.

See the chip-specific LPSPI information for more.

61.3.6 Reset

Table 579. LPSPI resets

Type of reset	Description
Chip	Resets the LPSPI logic and registers to their default states.
Software	• Resets the LPSPI logic and registers to their default states, except for the Control register. • The LPSPI software reset is controlled using CR[RST].
FIFO	• Resets the transmit and command FIFO and the receive FIFO. • CR[RTF] and CR[RRF] are write-only. • After being reset, FIFO is empty.

61.3.7 Interrupts and DMA requests

The following table lists sources (status flags) that can generate LPSPI interrupts and LPSPI transmit and receive DMA requests.

Table 580. Interrupts and DMA requests

Status (SR)		Description	Can generate		
Status flag	Name		Interrupt?	DMA request?	Low-power wake-up?
TDF	Transmit data flag	Indicates that data can be written to transmit FIFO, as configured by the transmit FIFO watermark, FCR[TXWATER].	Y	TX	Y

Table continues on the next page...

Table 580. Interrupts and DMA requests (continued)

Status (SR)		Description	Can generate		
Status flag	Name		Interrupt?	DMA request?	Low-power wake-up?
RDF	Receive data flag	Indicates that data can be read from the receive FIFO, as configured by the receive FIFO watermark, FCR[RXWATER].	Y	RX	Y
WCF	Word complete flag	Indicates that the word is complete and the last bit of the word has been sampled.	Y	N	Y
FCF	Frame complete flag	Indicates that the frame is complete and PCS is deasserted.	Y	RX	Y
TCF	Transfer complete flag	Indicates that transfer is complete, PCS is deasserted, and the transmit and command FIFO is empty.	Y	N	Y
TEF	Transmit error flag	Indicates a transmit and command FIFO underrun. In Controller mode, when CFGR1[NOSTALL] = 0 (transfers stall when transmit FIFO is empty), TEF cannot be set.	Y	N	Y
REF	Receive error flag	Indicates a receive FIFO overflow. In Controller mode, when CFGR1[NOSTALL] = 0 (transfers stall when receive FIFO is full), REF cannot be set.	Y	N	Y
DMF	Data match flag	Indicates that the received data matches the configured data match value.	Y	N	Y
MBF	Module busy flag	Indicates that LPSPI is busy performing an SPI bus transfer.	N	N	N

61.3.7.1 End-of-packet DMA transfer

The end-of-packet functionality is designed for serial interfaces where you may not be aware of the size of the transfer in advance and the data is pushed by an external device. Examples include UART receive, I2C Peripheral mode, and SPI Peripheral mode. The end-of-packet processing is intended to ensure that data is not stranded in either the receive FIFO or the DMA receive buffer. Support for end-of-packet processing must be implemented in both the serial interfaces and the DMA controller.

The condition that signals the end of packet is different for each serial interface but the serial peripheral and DMA process it in the same way. For example, UART end of packet is signaled by an idle line condition, I2C end of packet by a stop and/or repeated start condition, and SPI end of packet by PCS negation.

If you configure the serial peripheral to signal the end-of-packet condition to the DMA and the serial interface detects an end-of-packet condition, it asserts the DMA request for the receive FIFO irrespective of the watermark configuration. For larger watermark configurations, this ensures that the last few words of the transfer are first flushed from the receive FIFO.

The DMA then reads the contents of the receive FIFO, depending on the first word in the FIFO:

- If the receive FIFO is empty, the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, but the first word in the FIFO is the start of a new packet, then data is not pulled from the receive FIFO and the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, and the first word in the FIFO is not the start of a new packet, the DMA transfers the receive data as normal.

Because the DMA may be transferring multiple words on each request, the end-of-packet condition persists until the DMA minor loop is complete and no additional data is pulled from the receive FIFO. The status flag that triggered the end-of-packet condition is cleared when the minor loop completes following end of packet being signaled to the DMA controller.

When the DMA detects the end-of-packet condition, it writes all received words up to the end of the packet into the system memory and saves the destination address for the word after the last valid data. The DMA then terminates the channel as if the major loop is complete, including final offsets and optional interrupts, channel linking and scatter/gather. The final destination address can optionally be saved in the system memory or is available in the destination address register.

After the DMA terminates the major loop, no servicing of the receive FIFO occurs until either software or hardware reconfigures the DMA (for example, channel linking or scatter/gather). This delay must be minimized to avoid receiver FIFO overrun. NXP does not recommend automatic DMA end-of-packet processing when there are only a few words transferred between end-of-packet conditions. That is because the DMA spends more time processing the end of packet than transferring the data. For example, the UART idle line length must be increased as needed to avoid an excessive number of idle conditions.

61.3.7.2 DMA support registers

To support efficient DMA transfers to the transmit and control FIFO, an alias register supports 32-bit write access to the Transmit Command (TCR) and an alias region supports incrementing 32-bit write accesses to the Transmit Data (TDR).

- Transmit Command Burst (TCBR) is a 32-bit alias register for writing to TCR.
- Transmit Data Burst (TDBR0 - TDBR127) is a 512-byte alias region that supports writing to the TDR.

The burst alias locations are contiguous. A DMA transfer can start by writing to the TCBR register to initialize the transfer and then increment into the TDBR region without changing the DMA transfer size. The alias registers can also be used by the DMA to perform burst transfers when accessing the transmit FIFO.

The transmit FIFO prevents writes that overflow the FIFO, but this prevention does not signal an error. Do not perform writes to the TDBR unless there is sufficient room in the transmit FIFO.

Receive Data Burst (RDBR0 - RDBR127) is a 512-byte alias region that supports reading the Receive Data. This can be used by the DMA to perform burst transfers when accessing the receive FIFO.

61.3.8 Peripheral triggers

The connection of the LPSPI peripheral triggers with other peripherals depends on the device that is used.

Table 581. Peripheral triggers

Type of trigger	Description	Additional information
Frame output	The frame output trigger: • Asserts at the end of each frame (when PCS deasserts). • Remains asserted for one cycle of the LPSPI functional clock divided by the configuration defined in TCR[PRESCALE].	LPSPI generates two output triggers that can be connected to other peripherals on the chip.
Word output	The word output trigger: • Asserts at the end of each received word. • Remains asserted for one cycle of the LPSPI functional clock divided by the configuration defined in TCR[PRESCALE].	

61.4 External signals

Table 582. External signals

Signal	Name	Description	I/O
SCK	Serial clock	• Input in Peripheral mode • Output in Controller mode	I/O
PCS[0]	Peripheral chip select	• Input in Peripheral mode • Output in Controller mode	I/O
PCS[1]	Peripheral chip select	: • Input in Peripheral mode • Output in Controller mode	I/O
SOUT	Serial data output	Can be configured as serial data input signal (used as data pin 0 in half-duplex parallel data transfers)	I/O
SIN	Serial data input	Can be configured as serial data output signal (used as data pin 1 in half-duplex parallel data transfers)	I/O

61.5 Initialization

This module does not require initialization.

61.6 Memory map and registers

NOTE

- Writing to a read-only register or reading a write-only register can cause bus errors.
- LPSPI does not check values programmed in registers for validity, so you must take care to write valid values only.

61.6.1 LPSPI register descriptions

61.6.1.1 LPSPI memory map

LPSPI1 base address: 4436_0000h

LPSPI2 base address: 4437_0000h

LPSPI3 base address: 4255_0000h

LPSPI4 base address: 4256_0000h

LPSPI5 base address: 426F_0000h

LPSPI6 base address: 4270_0000h

LPSPI7 base address: 4271_0000h

LPSPI8 base address: 4272_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VERID)	32	R	0200_0004
4	Parameter (PARAM)	32	R	See section
10	Control (CR)	32	RW	0000_0000
14	Status (SR)	32	RW	0000_0001
18	Interrupt Enable (IER)	32	RW	0000_0000
1C	DMA Enable (DER)	32	RW	0000_0000
20	Configuration 0 (CFGR0)	32	RW	0000_0000
24	Configuration 1 (CFGR1)	32	RW	0000_0000
30	Data Match 0 (DMR0)	32	RW	0000_0000
34	Data Match 1 (DMR1)	32	RW	0000_0000
40	Clock Configuration (CCR)	32	RW	0000_0000
44	Clock Configuration 1 (CCR1)	32	RW	0000_0000
58	FIFO Control (FCR)	32	RW	0000_0000
5C	FIFO Status (FSR)	32	R	0000_0000
60	Transmit Command (TCR)	32	RW	0000_001F
64	Transmit Data (TDR)	32	W	0000_0000
70	Receive Status (RSR)	32	R	0000_0002
74	Receive Data (RDR)	32	R	0000_0000
78	Receive Data Read Only (RDROR)	32	R	0000_0000
3FC	Transmit Command Burst (TCBR)	32	W	0000_0000
400 - 5FC	Transmit Data Burst (TDBR0 - TDBR127)	32	W	0000_0000
600 - 7FC	Receive Data Burst (RDBR0 - RDBR127)	32	R	0000_0000

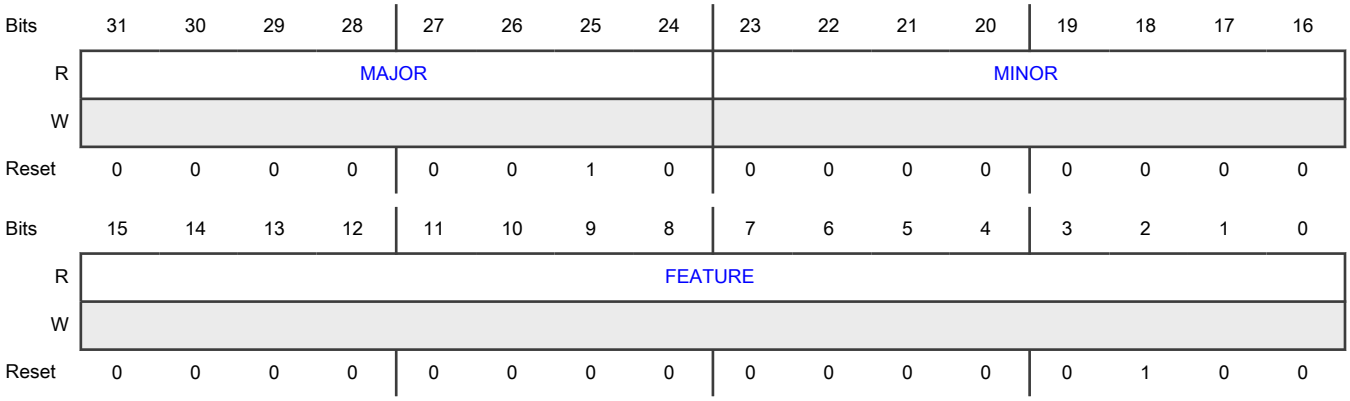
61.6.1.2 Version ID (VERID)

Contains version numbers for the module design and feature set.

Offset

Register	Offset
VERID	0h

Diagram



Fields

Field	Description
31-24 MAJOR	Major Version Number Indicates the major version number of the module specification.
23-16 MINOR	Minor Version Number Indicates the minor version number of the module specification.
15-0 FEATURE	Module Identification Number Indicates the feature set number 0000_0000_0000_0100 - Standard feature set supporting a 32-bit shift register. All other values are reserved.

61.6.1.3 Parameter (PARAM)

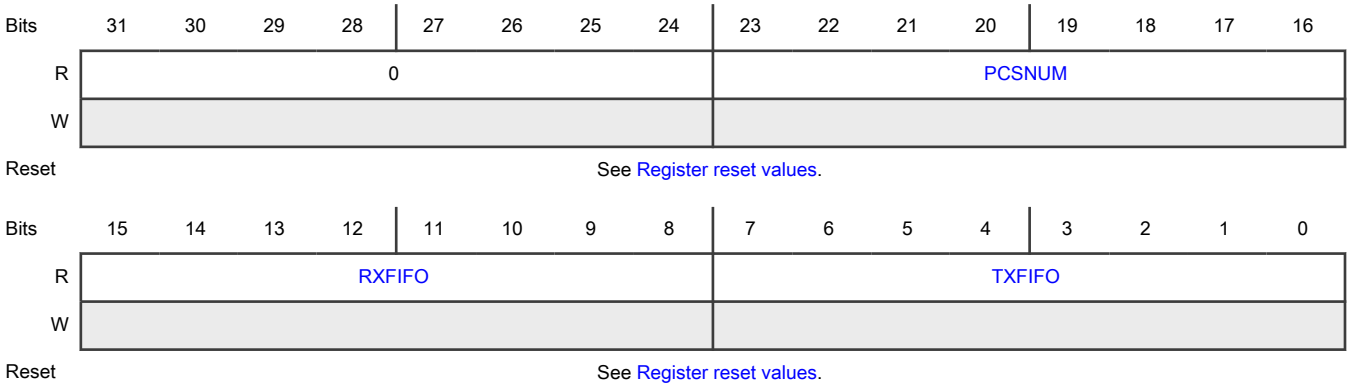
Contains:

- Number of PCS pins.
- Receive FIFO size.
- Transmit FIFO size.

Offset

Register	Offset
PARAM	4h

Diagram



Register reset values

Register	Reset value
PARAM	LPSPI1–LPSPI3: 0002_0303h LPSPI4: 0003_0303h LPSPI5–LPSPI8: 0002_0303h

Fields

Field	Description
31-24 —	Reserved
23-16 PCSNUM	PCS Number Indicates the number of PCS pins supported.
15-8 RXFIFO	Receive FIFO Size Indicates the maximum number of words in the receive FIFO. The maximum number of words is 2^{RXFIFO}.
7-0 TXFIFO	Transmit FIFO Size Indicates the maximum number of words in the transmit FIFO. The maximum number of words is 2^{TXFIFO}.

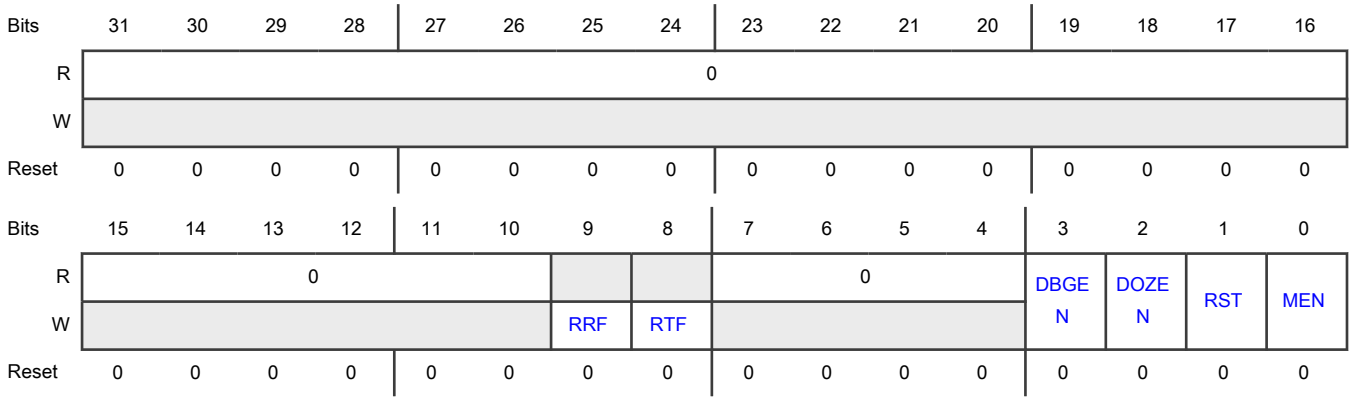
61.6.1.4 Control (CR)

Contains fields that control the module operation.

Offset

Register	Offset
CR	10h

Diagram



Fields

Field	Description
31-10 —	Reserved
9 RRF	Reset Receive FIFO Deletes all entries in the receive FIFO. This field always reads 0. 0 - No effect 1 - Reset
8 RTF	Reset Transmit FIFO Deletes all entries in the transmit FIFO. This field always reads 0. 0 - No effect 1 - Reset
7-4 —	Reserved
3 DBGEN	Debug Enable Enables LPSPI when the CPU is in Debug mode. If this field is 0, LPSPI is disabled when the CPU is halted; the PCS pin is deasserted after the transmit FIFO is empty regardless of the state of Transmit Command (TCR). You must update this field only when LPSPI is disabled (MEN = 0). 0 - Disable 1 - Enable
2 DOZEN	Doze Mode Enable Enables LPSPI when the chip is in Doze mode. You must update this field only when LPSPI is disabled (MEN = 0).

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Enable 1 - Disable
1 RST	Software Reset Resets all internal logic and registers, except Control (CR). The reset takes effect immediately and remains asserted until you write 0 to it. There is no minimum delay required before clearing the software reset by writing 0. 0 - Not reset 1 - Reset
0 MEN	Module Enable Enables the module. After writing 0, MEN remains set until LPSPI has completed the current transfer and is idle. 0 - Disable 1 - Enable

61.6.1.5 Status (SR)

Contains data flow status.

Offset

Register	Offset
SR	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							MBF	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		DMF	REF	TEF	TCF	FCF	WCF	0						RDF	TDF
W			W1C	W1C	W1C	W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31-25 —	Reserved
24 MBF	Module Busy Flag Indicates, in Controller mode, whether there is data to transmit and LPSPI is able to transmit (for example, the HREQ pin is asserted). The HREQ pin deasserts after the PCS pin deasserts and the LPSPI controller has waited for half the time specified in CCR[DBT] with no new data to transmit. Peripheral mode sets this flag when LPSPI is enabled and PCS is asserted. 0 - LPSPI is idle 1 - LPSPI is busy
23-14 —	Reserved
13 DMF	Data Match Flag Indicates whether the received data matches DMR0[MATCH0] and/or DMR1[MATCH1] (as configured by CFGR1[MATCFG]). **NOTE** This field behaves differently for register reads and writes. When reading 0 - No match 1 - Match When writing 0 - No effect 1 - Clear the flag
12 REF	Receive Error Flag Indicates a receive FIFO overflow error. When this flag is set: 1. End the transfer. 2. Empty the receive FIFO. 3. Clear this flag. 4. Restart the transfer from the beginning. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No overflow 1 - Overflow

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When writing 0 - No effect 1 - Clear the flag
11 TEF	Transmit Error Flag Indicates a transmit FIFO underrun error. When this flag is set: 1. End the transfer. 2. Clear this flag. 3. Restart the transfer from the beginning. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No underrun 1 - Underrun When writing 0 - No effect 1 - Clear the flag
10 TCF	Transfer Complete Flag Indicates, in Controller mode, whether all transfers are complete and LPSPI has returned to the Idle state and the transmit FIFO is empty. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not complete 1 - Complete When writing 0 - No effect 1 - Clear the flag
9 FCF	Frame Complete Flag Indicates whether a frame transfer is complete after PCS deasserts. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Not complete 1 - Complete When writing 0 - No effect 1 - Clear the flag
8 WCF	Word Complete Flag Indicates whether the last bit of a received word is sampled. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not complete 1 - Complete When writing 0 - No effect 1 - Clear the flag
7-2 —	Reserved
1 RDF	Receive Data Flag Indicates whether the number of words in the receive FIFO is greater than the value in FCR[RXWATER]. 0 - Receive data not ready 1 - Receive data ready
0 TDF	Transmit Data Flag Indicates whether the number of words in the transmit FIFO is equal to or less than the value in FCR[TXWATER]. 0 - Transmit data not requested 1 - Transmit data requested

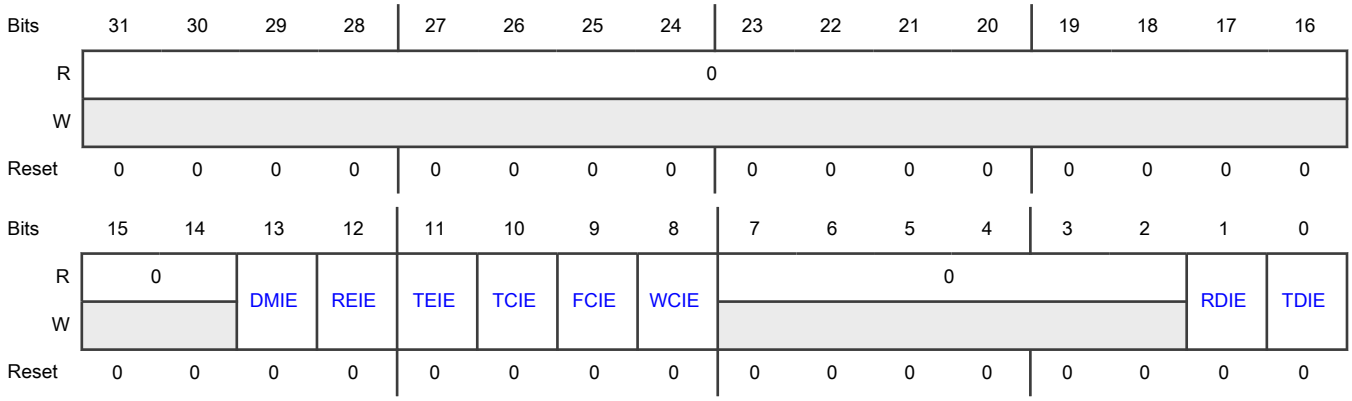
61.6.1.6 Interrupt Enable (IER)

Enables interrupts based on data flow and errors.

Offset

Register	Offset
IER	18h

Diagram



Fields

Field	Description
31-14 —	Reserved
13 DMIE	Data Match Interrupt Enable Enables the data match interrupt. 0 - Disable 1 - Enable
12 REIE	Receive Error Interrupt Enable Enables the receive error interrupt. 0 - Disable 1 - Enable
11 TEIE	Transmit Error Interrupt Enable Enables the transmit error interrupt. 0 - Disable 1 - Enable
10 TCIE	Transfer Complete Interrupt Enable Enables the transfer complete interrupt. 0 - Disable 1 - Enable
9 FCIE	Frame Complete Interrupt Enable Enables the frame complete interrupt. 0 - Disable 1 - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 WCIE	Word Complete Interrupt Enable Enables the word complete interrupt. 0 - Disable 1 - Enable
7-2 —	Reserved
1 RDIE	Receive Data Interrupt Enable Enables the receive data interrupt. 0 - Disable 1 - Enable
0 TDIE	Transmit Data Interrupt Enable Enables the transmit data interrupt. 0 - Disable 1 - Enable

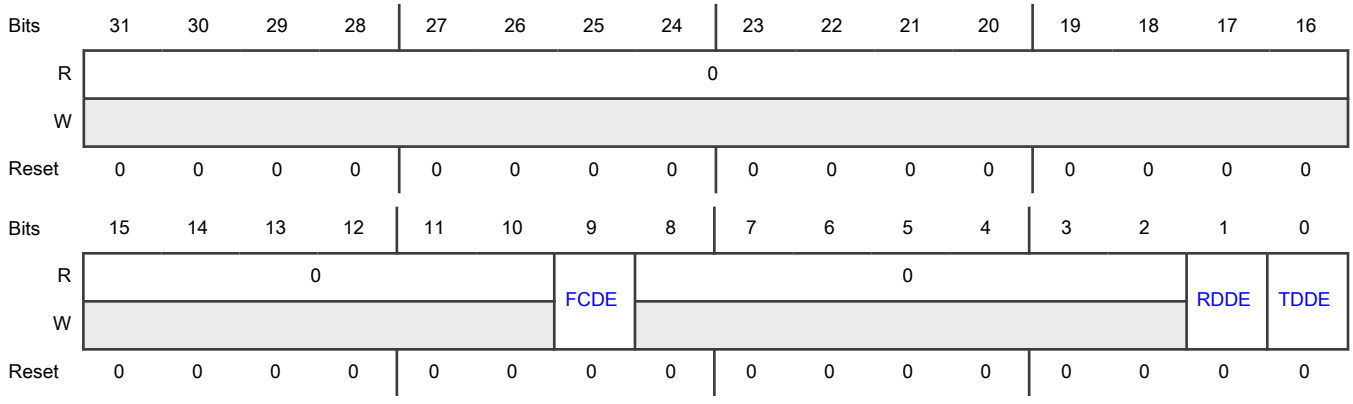
61.6.1.7 DMA Enable (DER)

Enables the DMA data flow.

Offset

Register	Offset
DER	1Ch

Diagram



Fields

Field	Description
31-10 —	Reserved
9 FCDE	Frame Complete DMA Enable Enables DMA end-of-packet processing. After the last word of a frame is read from the receive data FIFO, reading the receive data FIFO returns an end-of-packet signal with the receive data forced to FFFF_FFFFh. This continues until the DMA minor loop completes, and then SR[FCF] deasserts if the receive FIFO is empty or if LPSPI is busy (SR[MBF] = 1). 0 - Disable 1 - Enable
8-2 —	Reserved
1 RDDE	Receive Data DMA Enable Enables the receive data DMA. 0 - Disable 1 - Enable
0 TDDE	Transmit Data DMA Enable Enables the transmit data DMA. 0 - Disable 1 - Enable

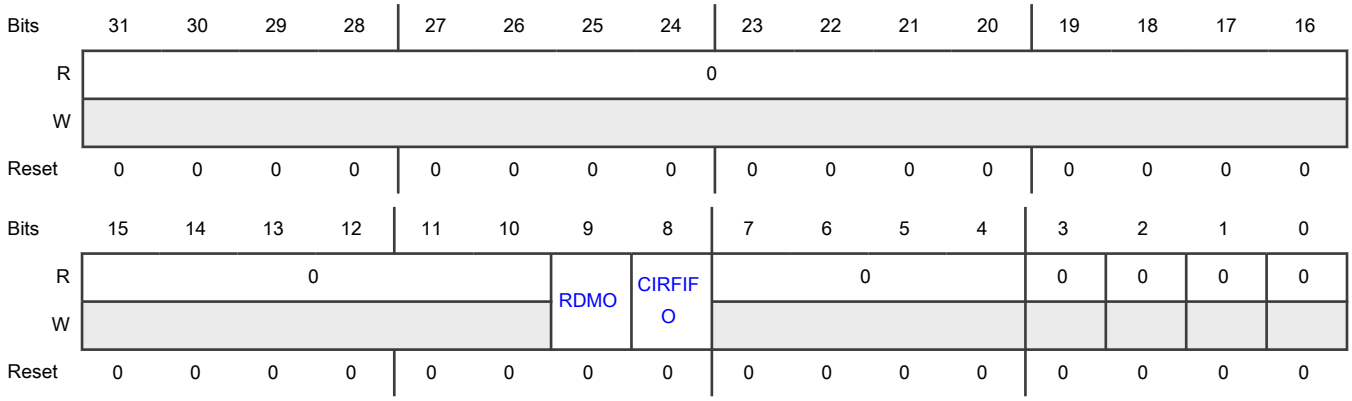
61.6.1.8 Configuration 0 (CFGR0)

Includes fields to configure LPSPI.

Offset

Register	Offset
CFGR0	20h

Diagram



Fields

Field	Description
31-10 —	Reserved
9 RDMO	Receive Data Match Only Enables receive data match. When enabled, all received data that does not cause SR[DMF] to assert is discarded: • Write 1 to this field when LPSPI is idle and SR[DMF] = 0. • After SR[DMF] = 1, this field is ignored. • To ensure that no receive data is lost when disabling RDMO, write 0 to this field before clearing SR[DMF]. See CFGR1[MATCFG] for the received data matching options. When disabled, all received data is stored in the receive FIFO. 0 - Disable 1 - Enable
8 CIRFIFO	Circular FIFO Enable Enables circular FIFO. When enabled, the transmit FIFO read pointer is saved to a temporary register. The transmit FIFO is emptied as in normal operation, but when LPSPI is idle and the transmit FIFO is empty, the read pointer value is restored from the temporary register. This restoring of the read pointer causes the contents of the transmit FIFO to be cycled through repeatedly. **NOTE** The read pointer is restored for as long as this field is 1. Writing additional words to the FIFO when this field is 1 adds them to the end of the FIFO, up to the size of the transmit FIFO. 0 - Disable 1 - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
7-4 —	Reserved
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

61.6.1.9 Configuration 1 (CFGR1)

Includes fields to configure LPSPI. You must write to this register only when LPSPI is disabled.

In addition to pin and output configurations, this register contains match configuration details; the following table shows match conditions specified in MATCFG.

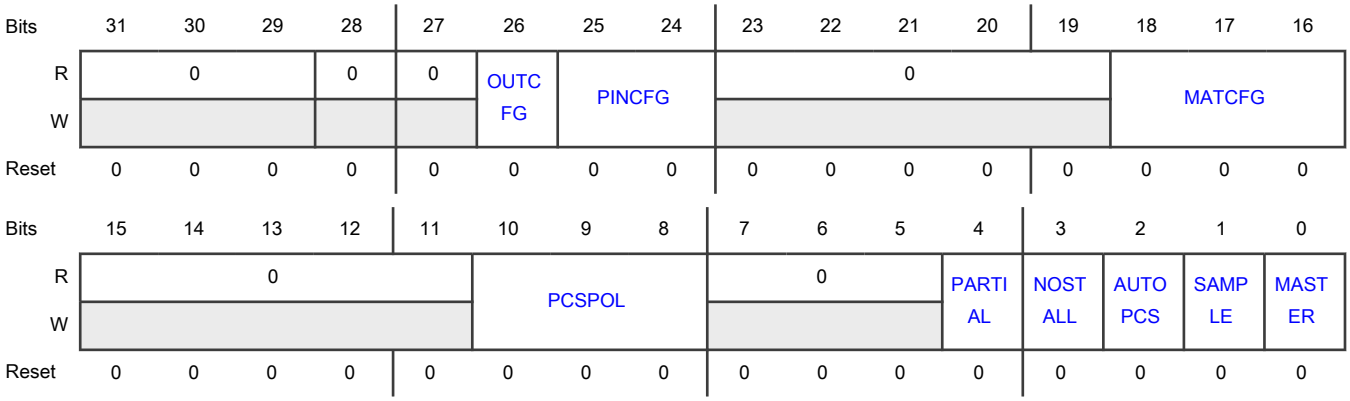
Table 583. Match conditions for CFGR1[MATCFG]

Condition	Description
Match first data word with compare word	Match if first data word equals MATCH0 logically ORed with MATCH1 first_data_word == (MATCH0 || MATCH1)
Match any data word with compare word	Match if any data word equals MATCH0 logically ORed with MATCH1 any_data_word == (MATCH0 || MATCH1)
Sequential match, first data word	Match if first data word equals MATCH0, and second data word equals MATCH1 (first_data_word == MATCH0) && (second_data_word == MATCH1)
Sequential match, any data word	Match if any data word equals MATCH0, and the next data word equals MATCH1 (any_data_word == MATCH0) && (next_data_word == MATCH1)
Match first data word (masked) with compare word (masked)	Match if first data word logically ANDed with MATCH1 equals MATCH0 logically ANDed with MATCH1 (first_data_word && MATCH1) == (MATCH0 && MATCH1)
Match any data word (masked) with compare word (masked)	Match if any data word logically ANDed with MATCH1 equals MATCH0 logically ANDed with MATCH1 (any_data_word && MATCH1) == (MATCH0 && MATCH1)

Offset

Register	Offset
CFGR1	24h

Diagram



Fields

Field	Description
31-29 —	Reserved
28 —	Reserved
27 —	Reserved
26 OUTCFG	Output Configuration Specifies whether the output data is 3-stated between accesses (when PCS is deasserted). When performing half-duplex transfers, this field must be 1. 0 - Retain last value 1 - 3-stated
25-24 PINCFG	Pin Configuration Specifies the pins used for input and output data during serial transfers. 00 - SIN is used for input data; SOUT is used for output data 01 - SIN is used for both input and output data; only half-duplex serial transfers are supported 10 - SOUT is used for both input and output data; only half-duplex serial transfers are supported 11 - SOUT is used for input data; SIN is used for output data

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>23-19
—</td><td>Reserved</td></tr>
<tr><td>18-16
MATCFG</td><td>Match Configuration
Specifies the condition that causes SR[DMF] to assert. See the match conditions listed in Table 1 for more information.
NOTE
When writing to this field, either the old value or new value must be in the disabled state (0). You cannot transition from a nonzero value to another nonzero value.
000 - Match is disabled
001 - Reserved
010 - Match first data word with compare word
011 - Match any data word with compare word
100 - Sequential match, first data word
101 - Sequential match, any data word
110 - Match first data word (masked) with compare word (masked)
111 - Match any data word (masked) with compare word (masked)</td></tr>
<tr><td>15-11
—</td><td>Reserved</td></tr>
<tr><td>10-8
PCSPOL</td><td>Peripheral Chip Select Polarity
Specifies the polarity of each PCS pin. Bit n in this field (the least-significant bit is bit 0) corresponds to PCS[n].
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>LPSPI1</td><td>CFGR1[9–8]</td><td>CFGR1[10]</td></tr><tr><td>LPSPI2</td><td>CFGR1[9–8]</td><td>CFGR1[10]</td></tr><tr><td>LPSPI3</td><td>CFGR1[9–8]</td><td>CFGR1[10]</td></tr><tr><td>LPSPI4</td><td>CFGR1</td><td>—</td></tr><tr><td>LPSPI5</td><td>CFGR1[9–8]</td><td>CFGR1[10]</td></tr></table></td></tr>
</table>

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>LPSPI6</td><td>CFGR1[9–8]</td><td>CFGR1[10]</td></tr>
<tr><td>LPSPI7</td><td>CFGR1[9–8]</td><td>CFGR1[10]</td></tr>
<tr><td>LPSPI8</td><td>CFGR1[9–8]</td><td>CFGR1[10]</td></tr>
</table>
NOTE
The descriptions of the field settings vary by module instance.
<table>
<tr><th>Instance</th><th>Field value and description</th></tr>
<tr><td>LPSPI1
LPSPI2
LPSPI3
LPSPI5
LPSPI6
LPSPI7
LPSPI8</td><td>00 - Active low
01 - Active high</td></tr>
<tr><td>LPSPI4</td><td>000 - Active low
001 - Active high</td></tr>
</table>
</td></tr>
<tr><td>7-5
—</td><td>Reserved</td></tr>
<tr><td>4
PARTIAL</td><td>Partial Enable
Specifies whether LPSPI, when in Peripheral mode, stores a partial received word in the receive FIFO, or discards it, when PCS deasserts. See Partial received word for more information.
0 - Discard
1 - Store</td></tr>
<tr><td>3
NOSTALL</td><td>No Stall
Disables a normal operating feature that causes LPSPI, when in Controller mode, to stall transfers when the transmit FIFO is empty or when the receive FIFO is full. This feature prevents transmit FIFO underruns and receive FIFO overruns. Writing 1 to this field disables this functionality.
0 - Disable
1 - Enable</td></tr>
<tr><td>2</td><td>Automatic PCS</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Description
AUTOPCS	Enables automatic PCS generation. For correct operation in Peripheral mode, LPSPI requires the PCS signal to deassert between frames. Writing 1 to this field generates an internal PCS signal at the end of each transfer word when TCR[CPHA] = 1. When this field is 1, SCK must remain idle for at least four LPSPI functional clock cycles, divided by the prescaler (see TCR[PRESCALE]) selected between each word to ensure correct operation. This field is ignored in Controller mode. **NOTE** Set TCR[CPHA] = 1 before enabling the module by setting CR[MEN] = 1. 0 - Disable 1 - Enable
1 SAMPLE	Sample Point Specifies the SCK clock edge on which LPSPI, when in Controller mode, samples input data. Writing 1 to this field causes LPSPI to sample input data on a delayed loopback SCK clock edge, which improves the setup time when sampling data (see Clock loopback). In this configuration, the input data setup time in Controller mode is equal to the input data setup time in Peripheral mode. In Peripheral mode, this field is ignored. **NOTE** When SAMPLE = 1, both the input and output buffers must be enabled for the SCK pin. Buffers are configured at the chip level. 0 - SCK edge 1 - Delayed SCK edge
0 MASTER	Controller Mode Specifies the LPSPI operating mode, Controller or Peripheral. This field directly controls the direction of the SCK and PCS pins. 0 - Peripheral mode 1 - Controller mode

61.6.1.10 Data Match 0 (DMR0)

Specifies the match data to be used when data matching is enabled. See CFGR1[MATCFG] for the received data matching options.

NOTE

Do not change the value in this register when CFGR1[MATCFG] > 0.

Offset

Register	Offset
DMR0	30h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MATCH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MATCH0	Match 0 Value Specifies the MATCH0 value to be compared against received data.

61.6.1.11 Data Match 1 (DMR1)

Specifies the match data to be used when data matching is enabled. See CFGR1[MATCFG] for the received data matching options.

NOTE

Do not change the value in this register while CFGR1[MATCFG] > 0.

Offset

Register	Offset
DMR1	34h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MATCH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 MATCH1	Match 1 Value Specifies the MATCH1 value to be compared against received data.

61.6.1.12 Clock Configuration (CCR)

Contains clock configuration fields that are used only in Controller mode; you can only change them when LPSPI is disabled (CR[MEN] = 0).

Warning

Writing a 32-bit value to this register overwrites Clock Configuration 1 (CCR1); DBT and SCKDIV always read 0.

To avoid overwriting CCR1, do one of the following:

- Write to all four fields in Clock Configuration (CCR) simultaneously and only once in a 32-bit data.
- Modify the values of CCR[SCKPCS] and/or CCR[PCSSCK]; write only these two upper bytes in a 16-bit data or one of them in an 8-bit data.
- Modify CCR1[PCSPCS] and CCR1[SCKSCK] only or CCR1[SCKSET] and CCR1[SCKHLD] only, write respectively to CCR[DBT] or CCR[SCKDIV] in 8-bit data.

Offset

Register	Offset
CCR	40h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SCKPCS								PCSSCK							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0							
W	DBT								SCKDIV							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 SCKPCS	SCK-to-PCS Delay

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Configures SCK-to-PCS delay. In Controller mode, this field helps you configure the delay from the last SCK edge to PCS negation: • The delay is equal to (SCKPCS + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. See Figure 359 for more information.
23-16 PCSSCK	PCS-to-SCK Delay Configures PCS-to-SCK delay. In Controller mode, this field helps you configure the delay from PCS assertion to the first SCK edge: • The delay is equal to (PCSSCK + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. See Figure 359 for more information.
15-8 DBT	Delay Between Transfers Configures the delay between transfers. Writing to this field updates the contents of CCR1[PCSPCS] and CCR1[SCKSCK].
7-0 SCKDIV	SCK Divider Updates the contents of CCR1[SCKSET] and CCR1[SCKHLD]. Baud rate = function clock ÷ (2^PRESCALE × (SCKSET + SCKHLD + 2))

61.6.1.13 Clock Configuration 1 (CCR1)

Contains clock configuration fields, which are used only in Controller mode. You can change them only when LPSPI is disabled (CR[MEN] = 0).

Offset

Register	Offset
CCR1	44h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SCKSCK								PCSPCS							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SCKHLD								SCKSET							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 SCKSCK	SCK Inter-Frame Delay Configures SCK inter-frame delay in Controller mode: • This field helps you configure the delay from the last SCK pulse of a frame and the first SCK pulse of the following frame, in a continuous transfer. • The delay is equal to (SCKSCK + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. **NOTE** For backward compatibility, writing to CCR[DBT] updates this field with the value written.
23-16 PCSPCS	PCS to PCS Delay Configures PCS to PCS delay in Controller mode: • This field helps you configure the delay from the PCS negation to the next PCS assertion. • The delay is equal to (PCSPCS + PCSPCS + 2) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is two cycles. • Half of the delay (PCSPCS + 1) occurs before PCS assertion and the other half of the delay (PCSPCS + 1) occurs after PCS negation. If the command word is updated between two transfers, then the command word is updated halfway between the PCS negation of the last transfer and PCS assertion of the next transfer. • The command word specifies which PCS signal is used, the polarity and phase of the SCK signal, and the selected prescaler. **NOTE** For backward compatibility, writing to CCR[DBT] updates this field with (DBT÷2) rounded up.
15-8	SCK Hold

Table continues on the next page...

Table continued from the previous page...

Field	Description
SCKHLD	Configures the hold phase of the SCK pin in Controller mode: • The hold phase is the delay between the SCK edge that samples the receive data and the SCK edge that drives the transmit data. • It is the SCK low period when CPHA = 0 and CPOL = 1, or CPHA = 1 and CPOL = 0. It is the SCK high period when CPHA = 0, CPOL = 0 and CPHA = 1, CPOL = 1. • The SCK hold phase delay is equal to (SCKHLD + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. • The SCK period is equal to (SCKSET + SCKHLD + 2) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The SCK duty cycle is based on the difference between SCKSET and SCKHLD. You must configure both these fields to the same value for a 50/50 duty cycle. See Figure 359 for more information. **NOTE** For backward compatibility, writing to CCR[SCKDIV] updates this field with (SCKDIV ÷ 2) rounded down.
7-0 SCKSET	SCK Setup Configures the setup phase of the SCK pin in Controller mode: • The setup phase is the delay between the SCK edge that drives the transmit data and the SCK edge that samples the receive data. • It is the SCK high period when CPHA = 0 and CPOL = 1, or CPHA = 1 and CPOL = 0. It is the SCK low period when CPHA = 0 and CPOL = 0, or CPHA = 1 and CPOL = 1. • The SCK setup phase delay is equal to (SCKSET + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. • The SCK period is equal to (SCKSET + SCKHLD + 2) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The SCK duty cycle is based on the difference between SCKSET and SCKHLD. You must configure both these fields to the same value for a 50/50 duty cycle. See Figure 359 for more information. **NOTE** For backward compatibility, writing to CCR[SCKDIV] updates this field with (SCKDIV ÷ 2) rounded up.

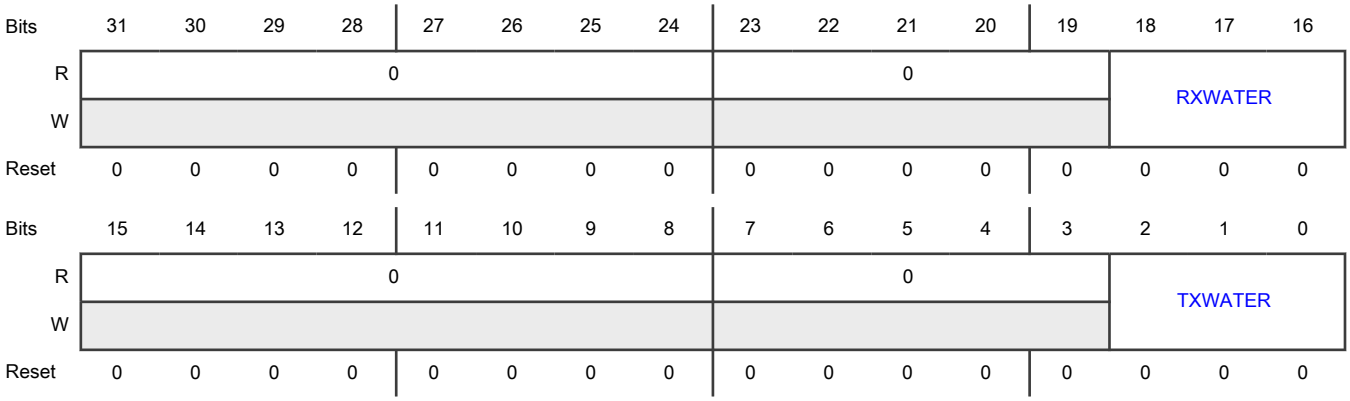
61.6.1.14 FIFO Control (FCR)

Contains the receive FIFO and transmit FIFO watermark values.

Offset

Register	Offset
FCR	58h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-19 —	Reserved
18-16 RXWATER	Receive FIFO Watermark Causes LPSPI to set SR[RDF] when the number of words in the receive FIFO is greater than RXWATER. Writing a value equal to or greater than the FIFO size truncates the written value.
15-8 —	Reserved
7-3 —	Reserved
2-0 TXWATER	Transmit FIFO Watermark Causes LPSPI to set SR[TDF] when the number of words in the transmit FIFO is equal to or less than TXWATER. Writing a value equal to or greater than the FIFO size truncates the written value.

61.6.1.15 FIFO Status (FSR)

Contains fields that indicate the number of words currently stored in the receive and transmit FIFOs.

Offset

Register	Offset
FSR	5Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0				RXCOUNT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0				TXCOUNT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23-20 —	Reserved
19-16 RXCOUNT	Receive FIFO Count Indicates the number of words currently stored in the receive FIFO.
15-8 —	Reserved
7-4 —	Reserved
3-0 TXCOUNT	Transmit FIFO Count Indicates the number of words currently stored in the transmit FIFO.

61.6.1.16 Transmit Command (TCR)

Pushes the data into the transmit FIFO, in the same order as written.

When you write to either this register or to Transmit Data (TDR), each write pushes data into the transmit FIFO. You must write to this register only using 32-bit writes, which are tagged and cause the command register to update; after that the entry reaches the top of the FIFO and LPSPI is enabled. This allows changes to the command word and the transmit data itself to be interleaved.

That is, writes to the two registers can be interleaved (write command word, then data word, then command word, and so on). Changing the command word causes all subsequent SPI bus transfers to be performed using the new command word:

- In Controller mode, writing a new command word does not initiate a new transfer, unless TXMSK is 1. Transfers are initiated by transmit data in the transmit FIFO, or by a new command word (with TXMSK = 1). Hardware writes 0 to TXMSK when PCS deasserts.
- In Controller mode, if the command word is changed before an existing frame has completed, then the existing frame terminates and the command word updates. The command word can be changed during a continuous transfer, if CONTC of the new command word is 1 and the command word is written on a frame size boundary.
- In Peripheral mode, the command word must be changed only when LPSPI is idle and there is no SPI bus transfer.

Avoid resetting the transmit FIFO after writing to this register; wait for the command register to update from the FIFO first.

Avoid register reading problems: Reading this register returns the current state of the register. Reading this register at the same time that it is loaded from the transmit FIFO can return an incorrect register value. It is recommended to:

- Read this register when the transmit FIFO is empty.
- Read this register more than once and then compare the returned values.

Offset

Register	Offset
TCR	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CPOL	CPHA	PRESCALE			Reserved	PCS		LSBF	BYSW	CONT	CONTC	RXMSK	TXMSK	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				FRAMESZ											
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Fields

Field	Description
31 CPOL	Clock Polarity Specifies the value of SCK when it is idle. You can update this field only when PCS is deasserted. See Figure 359 for more information. 0 - Inactive low 1 - Inactive high
30	Clock Phase

Table continues on the next page...

Table continued from the previous page...

Field	Description
CPHA	Indicates whether data is captured or changed on the leading edge of SCK and captured or changed on the following edge of SCK. You can update this field only when PCS is deasserted. See Figure 359 for more information. 0 - Captured 1 - Changed
29-27 PRESCALE	Prescaler Value Specifies the division of the LPSPI functional clock. For all SPI bus transfers, this value is applied to Clock Configuration (CCR). You can update this field only when PCS is deasserted. 000 - Divide by 1 001 - Divide by 2 010 - Divide by 4 011 - Divide by 8 100 - Divide by 16 101 - Divide by 32 110 - Divide by 64 111 - Divide by 128
26 —	Reserved
25-24 PCS	Peripheral Chip Select Configures the peripheral chip select used for the transfer. This field is updated only when PCS is deasserted. **NOTE** This entire field is not fully supported in every LPSPI module instance. See the chip-specific LPSPI information. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
LPSPI1	TCR[24]	TCR[25]
LPSPI2	TCR[24]	TCR[25]
LPSPI3	TCR[24]	TCR[25]

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>LPSPI4</td><td>TCR</td><td>—</td></tr>
<tr><td>LPSPI5</td><td>TCR[24]</td><td>TCR[25]</td></tr>
<tr><td>LPSPI6</td><td>TCR[24]</td><td>TCR[25]</td></tr>
<tr><td>LPSPI7</td><td>TCR[24]</td><td>TCR[25]</td></tr>
<tr><td>LPSPI8</td><td>TCR[24]</td><td>TCR[25]</td></tr>
</table>
NOTE
The descriptions of the field settings vary by module instance.
<table>
<tr><th>Instance</th><th>Field value and description</th></tr>
<tr><td>LPSPI1
LPSPI2
LPSPI3
LPSPI5
LPSPI6
LPSPI7
LPSPI8</td><td>0 - Transfer using PCS[0]
1 - Transfer using PCS[1]</td></tr>
<tr><td>LPSPI4</td><td>00 - Transfer using PCS[0]
01 - Transfer using PCS[1]
10 - Transfer using PCS[2]</td></tr>
</table>
</td></tr>
<tr><td>23
LSBF</td><td>LSB First
Indicates whether data is transferred with MSB first or LSB first.
0 - MSB first
1 - LSB first</td></tr>
<tr><td>22
BYSW</td><td>Byte Swap
Swaps the contents of [31:24] with [7:0] and [23:16] with [15:8] for each transmit data word read from the FIFO and for each received data word stored to the FIFO (or compared with match registers).
0 - Disable byte swap
1 - Enable byte swap</td></tr>
<tr><td>21</td><td>Continuous Transfer</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Description
CONT	Enables continuous transfer: • In Controller mode, this field keeps PCS asserted at the end of the frame size until a command word is received that starts a new frame. • In Peripheral mode, when this field is enabled, LPSPI only transmits the first FRAMESZ bits, after which LPSPI transmits received data (assuming a 32-bit shift register) until the next PCS negation. 0 - Disable 1 - Enable
20 CONTC	Continuing Command Enables the command word to be changed within a continuous transfer in Controller mode: • The initial command word must enable continuous transfer (CONT = 1). • The continuing command must have CONTC = 1. • The continuing command word must be loaded on a frame size boundary. For example, if the continuous transfer has a frame size of 64 bits, then a continuing command word must be loaded on a 64-bit boundary. In Peripheral mode, this field modifies the internal RXMSK and TXMSK configuration after the first FRAMESZ bits and until PCS negation: • Receive data is discarded after the first FRAMESZ bits. If CONT is also 1, this does not block the transmission of received data. • Transmit data is not masked after the first FRAMESZ bits. This allows the first FRAMESZ bits to be received and a response transmitted. 0 - Command word for start of new transfer 1 - Command word for continuing transfer
19 RXMSK	Receive Data Mask Masks receive data (receive data is not stored in the receive FIFO). 0 - Normal transfer 1 - Mask receive data
18 TXMSK	Transmit Data Mask Masks transmit data (no data is loaded from the transmit FIFO and the output pin is 3-stated). In Controller mode, TXMSK initiates a new transfer that cannot be aborted by another command word. TXMSK automatically transitions to 0 at the end of the transfer. 0 - Normal transfer 1 - Mask transmit data
17-16 —	Reserved You must write only 0 to this field.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-12 —	Reserved
11-0 FRAMESZ	Frame Size Configures the frame size in number of bits equal to (FRAMESZ + 1): • The minimum frame size is 8 bits. • If the frame size is larger than 32 bits, then the frame is divided into multiple words of 32 bits; each word is loaded from the transmit FIFO and stored in the receive FIFO separately. • If the size of the frame is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO contains the remainder bits. For example, a 72-bit transfer consists of three words: the first and second words are 32 bits, and the third word is 8 bits.

61.6.1.17 Transmit Data (TDR)

Pushes the data into the transmit FIFO, in the same order that the data is written. You can write to this register using 32-, 16-, or 8-bit writes.

When you write to this register or to Transmit Command (TCR), each write pushes data into the FIFO with zero pushed in unwritten bytes.

Offset

Register	Offset
TDR	64h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 DATA	Transmit Data Indicates transmit data. Both 8-bit and 16-bit writes of transmit data zero-extend the data written and push the data into the transmit FIFO. To zero-extend 8-bit and 16-bit writes (to 32 bits) means that the higher order (most significant) empty parts of the 8-bit and 16-bit writes are filled with zeroes.

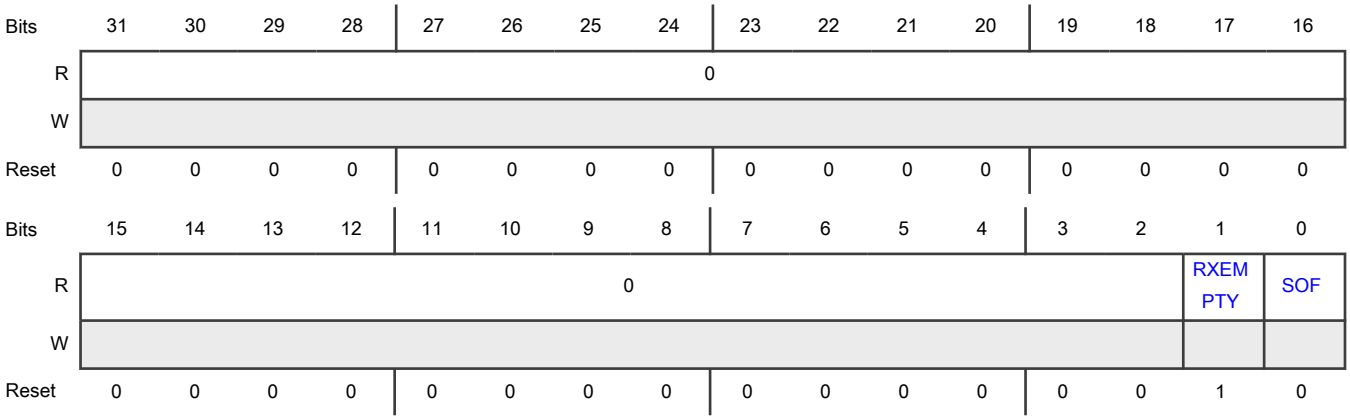
61.6.1.18 Receive Status (RSR)

Contains data flow status fields for receive FIFO.

Offset

Register	Offset
RSR	70h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 RXEMPTY	RX FIFO Empty Indicates whether the receive FIFO is empty. 0 - Not empty 1 - Empty
0 SOF	Start of Frame Indicates whether this is the first data word received after PCS assertion.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Subsequent data word or RX FIFO is empty (RXEMPTY=1). 1 - First data word

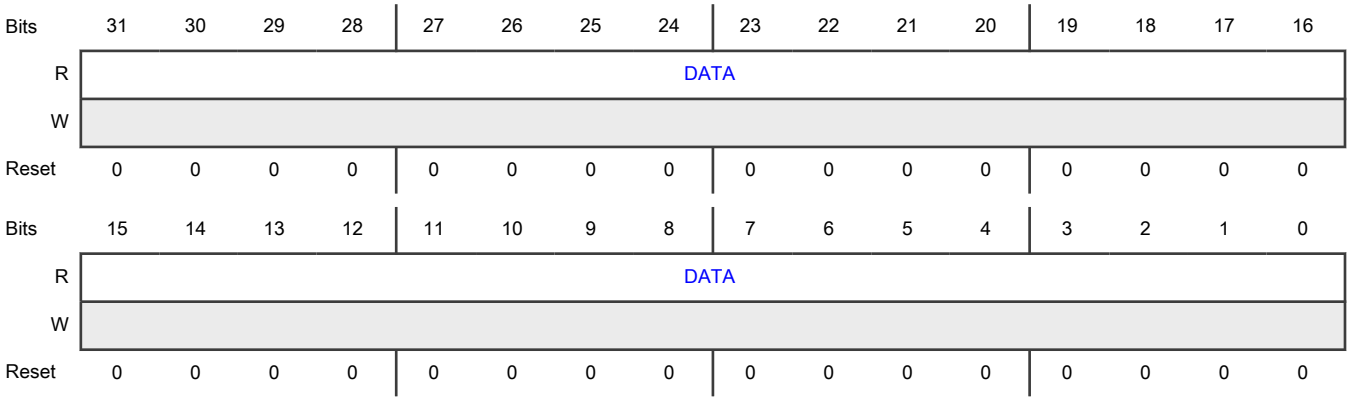
61.6.1.19 Receive Data (RDR)

Pulls the first entry from the receive FIFO.

Offset

Register	Offset
RDR	74h

Diagram



Fields

Field	Description
31-0 DATA	Receive Data

61.6.1.20 Receive Data Read Only (RDROR)

Returns the first entry in the receive FIFO but does not remove the data from the FIFO.

Offset

Register	Offset
RDROR	78h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 DATA	Receive Data

61.6.1.21 Transmit Command Burst (TCBR)

Supports burst transfers of command data to the transmit FIFO for use with the DMA controller.

See DMA support registers.

Offset

Register	Offset
TCBR	3FCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 DATA	Command Data Writes data to Transmit Command (TCR).

61.6.1.22 Transmit Data Burst (TDBR0 - TDBR127)

Supports burst transfers of data to the transmit FIFO for use with the DMA controller. The size of this register is 512 bytes.

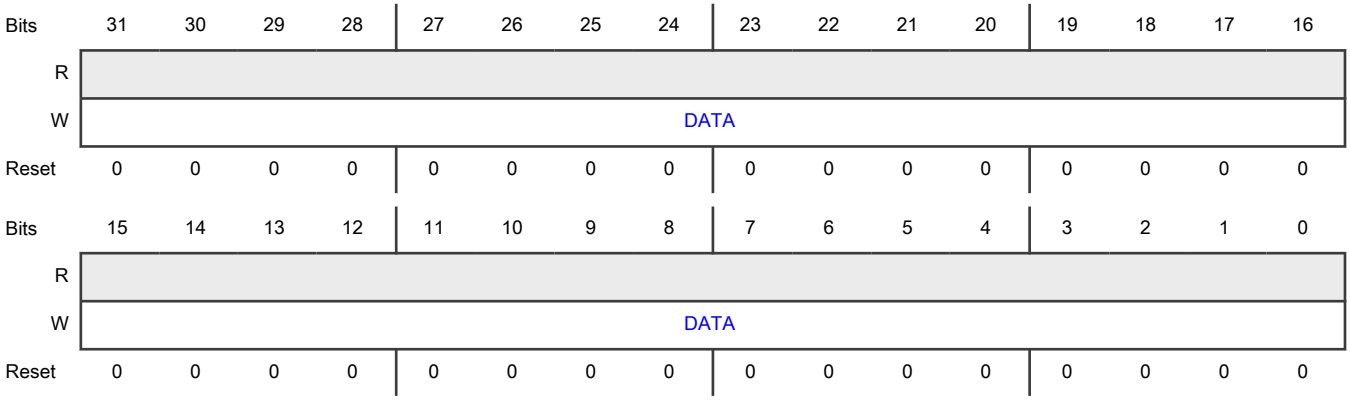
See DMA support registers.

Offset

For n = 0 to 127:

Register	Offset
TDBRn	400h + (n × 4h)

Diagram



Fields

Field	Description
31-0 DATA	Data Writes data to Transmit Data (TDR).

61.6.1.23 Receive Data Burst (RDBR0 - RDBR127)

Supports burst transfers of data from the receive FIFO. The size of this register is 512 bytes.

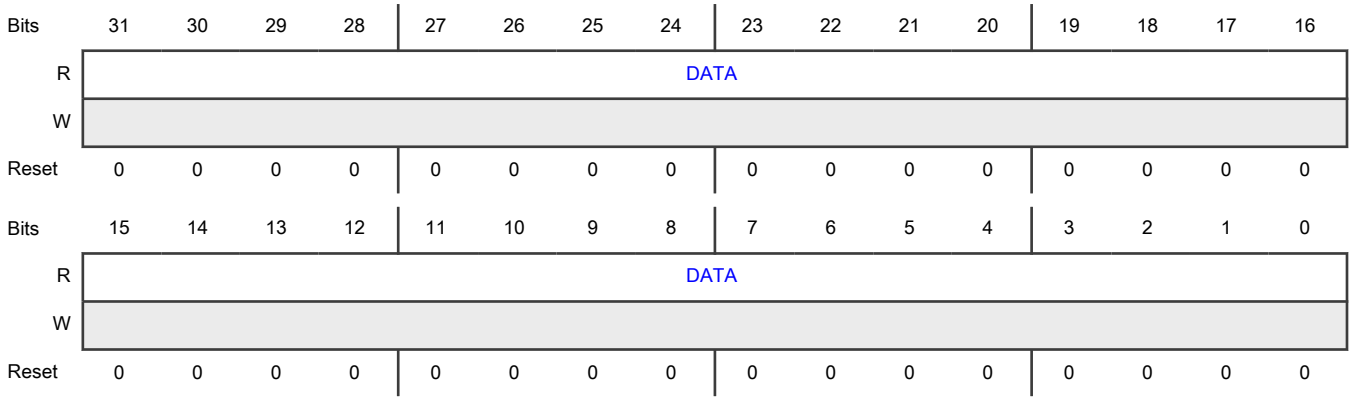
See DMA support registers.

Offset

For n = 0 to 127:

Register	Offset
RDBRn	600h + (n × 4h)

Diagram



Fields

Field	Description
31-0 DATA	Data Reads data from Receive Data (RDR).

61.7 Glossary

PCS	Peripheral chip select
SCK	Serial clock
SDI	Peripheral data in
SDO	Peripheral data out
SS	Peripheral select

Chapter 62
Low Power Universal Asynchronous Receiver/ Transmitter (LPUART)

62.1 Chip-specific LPUART information

Table 584. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing

The chip includes eight instances of LPUART Modules. LPUART1 and LPUART2 are in AONMIX. LPUART3-8 in WAKEUPMIX.

All LPUART instances support DOZE low power mode. CTRL(0x18) bit[6] is set to to 1'b1 to enable the usage of doze mode.

Entering doze mode or not, is decided by corresponding LPUART CCGRs in CCM, and the CCGRs are controlled by GPC CMCs. If the CMCs require SoC enter "WAIT" (cpu_power_mode[1:0]=2'b01) mode, then LPUART will be in "doze" mode.

LPUART loopback mode supports hardware flow control.

NOTE

LPUART DCB_B is the same as LPUART DCD_B.

NOTE

LPUART1, LPUART2, LPUART3, and LPUART4 support all signals mentioned in the External Signals section. LPUART5, LPUART6, LPUART7, and LPUART8 do not support DTR_B, DSR_B, DCD_B and RIN_B.

62.2 Overview

LPUART provides asynchronous, serial communication capabilities with external devices. It supports the non-return-to-zero (NRZ) encoding format and infrared data association (IrDA)-compatible, low-speed serial infrared (SIR) protocol. LPUART can continue operating when the processor is in Low-Power mode, if an appropriate peripheral clock is available.

62.2.1 Block diagram

Figure 361 shows the transmitter portion of LPUART.

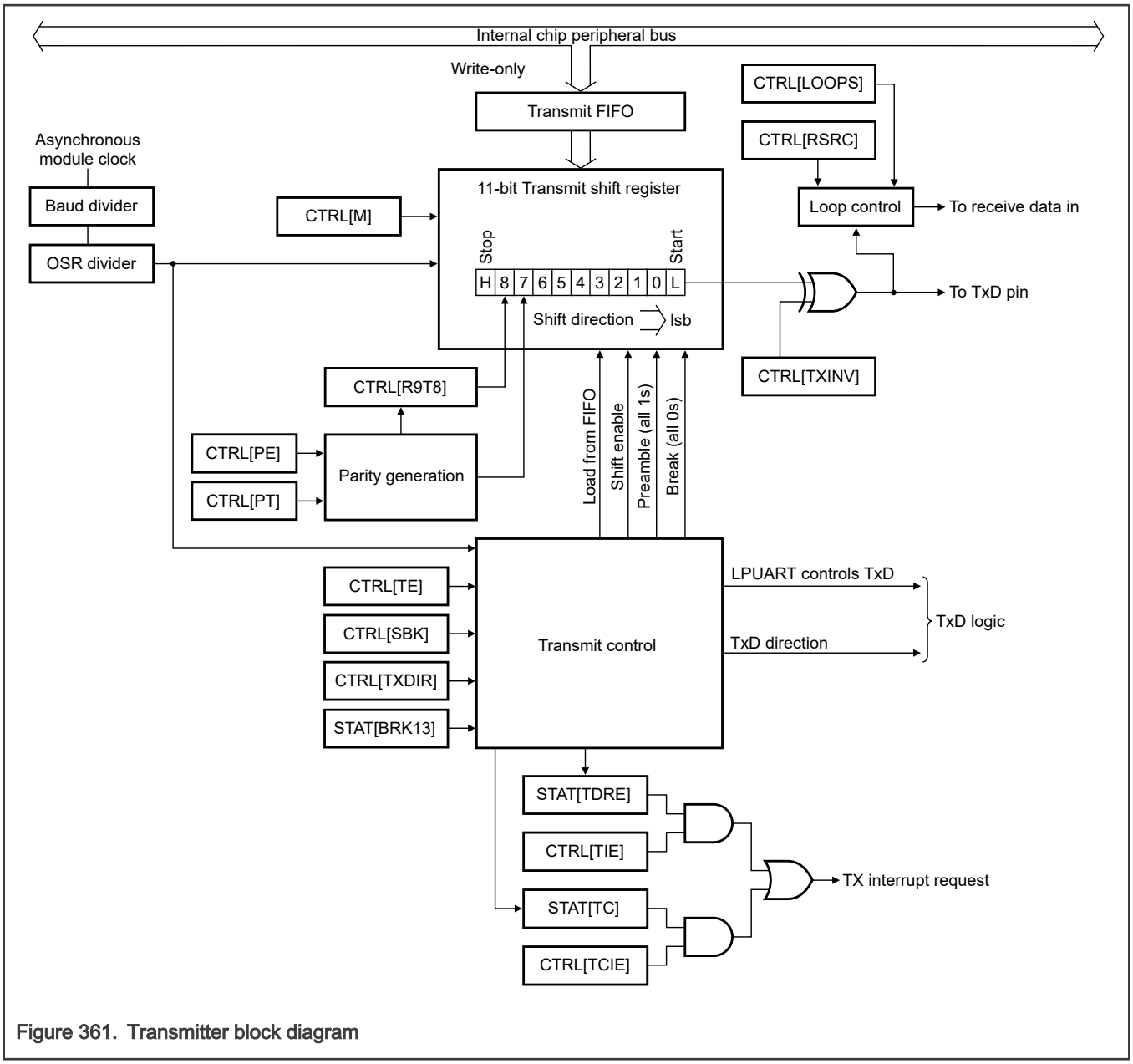


Figure 361. Transmitter block diagram

Figure 362 shows the receiver portion of LPUART.

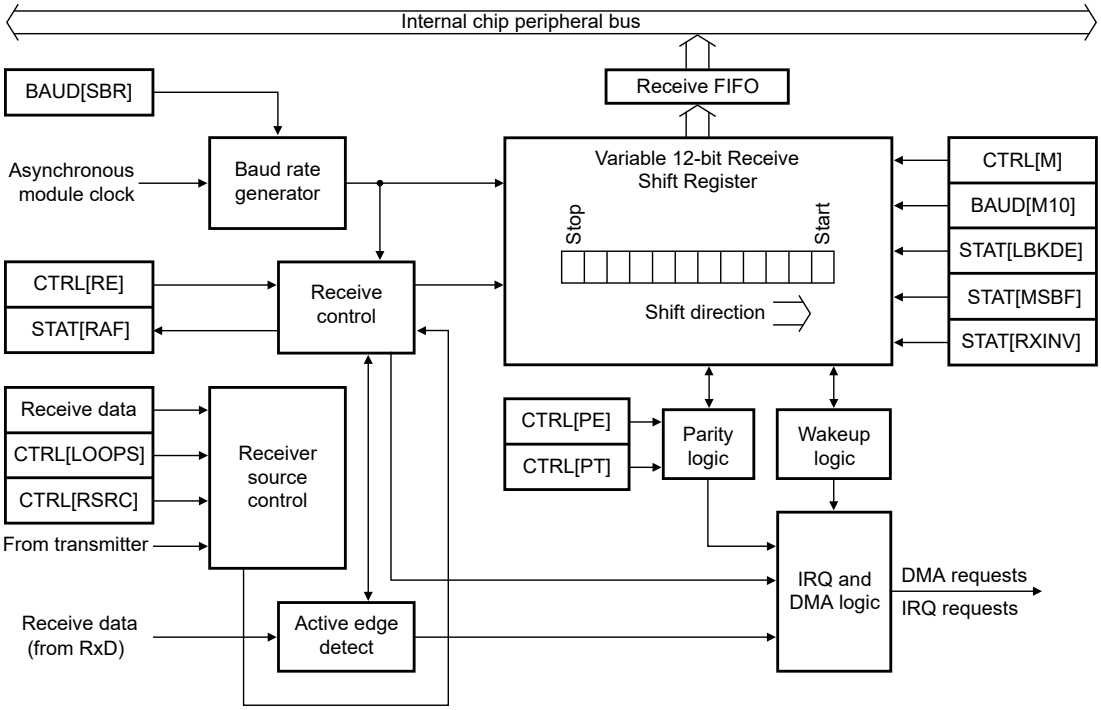


Figure 362. Receiver block diagram

62.2.2 Features

- Full-duplex, standard NRZ format
- Programmable baud rates (13-bit modulo divider) with a configurable oversampling ratio (OSR) from 4× to 32×
- Asynchronous operation of transmit and receive baud rates with respect to the bus clock:
  - Baud rate can be configured independently of the bus clock frequency.
  - Operation in Low-Power modes is supported.
- Interrupt, DMA, or polled operations:
  - Transmit data empty and transmission complete
  - Receive data full
  - Receive overrun, parity error, framing error, and noise error
  - Idle receiver detect
  - Active edge on receive pin
  - Break detect supporting LIN
  - Receive data match
- Hardware parity generation and checking
- Programmable 7-bit, 8-bit, 9-bit, or 10-bit character length
- Programmable 1-bit or 2-bit stop bits
- Support for three receiver wake-up methods:
  - Idle line wake-up

  - Address mark wake-up
  - Receive data match
- Automatic address matching to reduce ISR overhead:
  - Address mark matching
  - Idle line address matching
  - Address match start, address match end
- Optional 13-bit and 11-bit break character generation
- Configurable idle length detection supporting 1, 2, 4, 8, 16, 32, 64, or 128 idle characters
- Selectable transmitter output and receiver input polarity
- Hardware flow control support for request to send (RTS) and clear to send (CTS) signals
- Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with a programmable pulse width
- Independent FIFO structure for transmit and receive functions:
  - Separate configurable watermarks for receive and transmit requests
  - Option for receiver to assert request after a configurable number of idle characters, if receive FIFO is not empty

62.3 Functional description

LPUART supports full-duplex, asynchronous, NRZ serial communication and comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. The following sections describe all LPUART blocks.

62.3.1 Baud rate generation

A 13-bit modulus counter in the baud rate generator derives the baud rate for both the receiver and transmitter. The value, ranging from 1 to 8191, written to BAUD[SBR] determines the baud clock divisor for the asynchronous LPUART baud clock. The baud rate clock drives the receiver, while a bit clock, generated from the baud rate clock divided by the OSR, drives the transmitter. Depending on the OSR, the receiver has an acquisition rate of 4 to 32 samples per bit time. LPUART requires BAUD[SBR] and BAUD[OSR] to accurately match the baud rate requirement. There is a relationship between the required baud rate and the input clock frequency, SBR, and OSR. See the below figure for details.

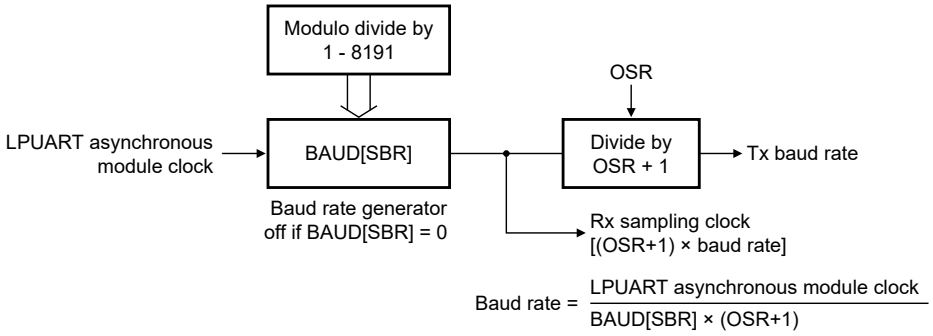


Figure 363. Baud rate generation

Baud rate generation is subject to these sources of error:

- Integer division of the asynchronous LPUART baud clock may not give the exact target frequency.
- Synchronization with the asynchronous LPUART baud clock can lead to a phase shift.

Baud rate generation is a free-running counter that continues whenever the transmitter or receiver is enabled. The transmitter bit clock continues whenever the transmitter is enabled; each transmitted character aligns to the next edge of the transmit bit clock.

In general, configuring OSR for a higher ratio and/or sampling on both edges of the clock slightly improves LPUART's tolerance to baud rate mismatch between the received data and LPUART configured baud rate. However, the three data samples in each bit (see Data sampling technique) are also closer together, which may impact noise sensitivity.

62.3.2 Baud rate tolerance

A transmitting device may operate at a baud rate below or above that of the receiver.

Accumulated bit time misalignment can cause one of the three stop bit data samples to fall outside the actual stop bit. A noise error will occur if the three samples are not all the same logical values. A framing error will occur if the receiver clock is misaligned in such a way that the majority of the three stop bit samples are a logic zero.

As the receiver samples an incoming frame, it may resynchronize the oversampling clock on any valid falling edge within the frame. Resynchronization within frames will correct a misalignment between transmitter bit times and receiver bit times.

In general, increasing the number of samples per bit will increase the baud rate tolerance and decreasing the number of samples per bit will reduce the baud rate tolerance. Note that since LPUART implements triple voting on consecutive receive data samples, increasing the number of samples per bit will move those samples closer together which would reduce the width of noise that can be filtered by the triple voting logic.

62.3.3 Calculating baud rate tolerance

Using the following definitions:

- SAM is the number of sample points per bit (valid range from 8 to 32; equal to (OSR + 1) × (BOTHEDGE + 1)).
- BIT is the number of bits in a character including start, data and stop bits (valid range from 9 to 13).

The ideal baud rate tolerance can be calculated as follows:

- Slow data rate tolerance = ((SAM ÷ 2) - 1) ÷ ((SAM × BIT) - (SAM ÷ 2) + 2)
- Fast data rate tolerance = ((SAM ÷ 2) - 2) ÷ (SAM × BIT)

As an example, if configured for 8-bit data, 1 stop bit (BIT = 10) and with OSR=0x7 and BOTHEDGE = 1 (SAM = 16):

- Slow data rate tolerance = (8 - 1) ÷ (160 - 8 + 2) = 7 ÷ 154 = 4.54%
- Fast data rate tolerance = (8 - 2) ÷ 160 = 6 ÷ 160 = 3.75%

If configured for 9-bit data with 1 stop bit (BIT = 11) with same oversampling configuration, then:

- Slow data rate tolerance = (8 - 1) ÷ (176 - 8 + 2) = 7 ÷ 170 = 4.12%
- Fast data rate tolerance = (8 - 2) ÷ 176 = 6 ÷ 176 = 3.41%

NOTE

Additional factors can contribute to a lower baud rate tolerance than the ideal. These include clock uncertainty or jitter on the LPUART functional clock source, differences in rise and fall times on the transmitter output and synchronization of the external receive pin to the local LPUART functional clock.

62.3.4 Transmitter functional description

This section describes the functioning of the LPUART transmitter, as shown in the transmitter portion of Block diagram, as well as specialized functions for sending break and idle characters.

The transmitter output (TXD) idle state defaults to logic high; the transmitter output is inverted when you write 1 to CTRL[TXINV], which becomes 0 following reset. You can enable the transmitter by writing 1 to CTRL[TE]. This queues a preamble character that is one full character frame of the Idle state. The transmitter then remains idle until data is available in the transmit FIFO and programs store data in the transmit FIFO by writing to Data (DATA).

The central element of the LPUART transmitter is the transmit shift register that is 9-bit to 13-bit long depending on the settings of CTRL[M], CTRL[M7], BAUD[M10], and BAUD[SBNS]. Going forward in this discussion, assume that CTRL[M], CTRL[M7], BAUD[M10], and BAUD[SBNS] are 0, selecting the normal 8-bit Data mode, in which the shift register holds a start bit, eight data bits, and a stop bit. When the transmit shift register is available for a new character, the value waiting in transmit FIFO is transferred to the transmit shift register, synchronized with the baud rate clock, and STAT[TDRE] becomes 1 to indicate that another character may be written to the transmit FIFO at Data (DATA).

If no new character is waiting in the transmit FIFO after a stop bit is shifted out of the TXD pin, the transmitter sets the transmit complete flag and enters an idle mode, with TXD high, waiting for more characters to transmit.

Writing 0 to CTRL[TE] does not immediately disable the transmitter. The current transmit activity in progress must first be completed (that could include a data character, idle character, or break character), although the transmitter does not start transmitting another character.

62.3.4.1 Break character length

CTRL[SBK] sends break characters, originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0, 9-bit to 12-bit times, including the start and stop bits. You can enable a longer break of 13-bit times by writing 1 to STAT[BRK13]. Normally, a program waits for STAT[TDRE] to become 1 to indicate that the last character of a message has moved to the transmit shifter. Next, the program writes 1 and then writes 0 to CTRL[SBK]. This action queues a break character to be sent as soon as the shifter is available. If CTRL[SBK] remains 1 when the queued break moves into the shifter, synchronized with the baud rate clock, an additional break character is queued. When LPUART is the receiving module, it receives a break character as 0s in all data bits and a framing error (STAT[FE] = 1) is detected.

You can also transmit a break character by writing to Data (DATA) with DATA[FRETSC] = 1 and the data bits clear. This supports transmitting the break character as part of the normal data stream and also allows DMA to transmit a break character.

When idle line wake-up is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program waits for STAT[TDRE] to become 1 to indicate that the last character of a message has moved to the transmit shifter. Next, write 0 and then write 1 to CTRL[TE]. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while CTRL[TE] becomes 0, the LPUART transmitter does not release control of the TXD pin.

You can also write to Data (DATA) to transmit an idle character, with DATA[FRETSC] and DATA[R9T9] = 1 and the values of all the other fields = 0. This supports transmitting the idle character as part of the normal data stream and also allows DMA to transmit an idle character.

As shown in the following table, STAT[BRK13], CTRL[M], CTRL[M7], BAUD[M10], and BAUD[SBNS] affect the length of the break character.

Table 585. Break character length

STAT[BRK13]	CTRL[M]	BAUD[M10]	CTRL[M7]	BAUD[SBNS]	Break character length (in bit times)
0	0	0	0	0	10
0	0	0	0	1	11
0	0	0	1	0	9
0	0	0	1	1	10
0	1	0	—	0	11
0	1	0	—	1	12
0	—	1	—	0	12
0	—	1	—	1	13
1	0	0	0	0	13

Table continues on the next page...

Table 585. Break character length (continued)

STAT[BRK13]	CTRL[M]	BAUD[M10]	CTRL[M7]	BAUD[SBNS]	Break character length (in bit times)
1	0	0	0	1	13
1	0	0	1	0	12
1	0	0	1	1	12
1	1	0	—	0	14
1	1	0	—	1	14
1	—	1	—	0	15
1	—	1	—	1	15

62.3.4.2 Hardware flow control

The transmitter supports hardware flow control by gating the transmission with the value of CTS_B. If the CTS operation is enabled, the character is transmitted when CTS_B is asserted. If CTS_B is deasserted in the middle of a transmission with characters remaining in the transmitter FIFO, the character in the transmit shift register is complete. Any characters in the FIFO wait for CTS_B to assert again, and TXD remains in the mark state (idle state) until CTS_B is reasserted. The CTS_B pin must assert for longer than one bit period to guarantee that a new transmission is started when the transmitter is idle with CTS.

If the CTS operation is disabled, the transmitter ignores the state of CTS_B.

The transmitter's CTS_B signal can be enabled even if the same LPUART receiver's RTS_B signal is disabled.

62.3.4.3 Transceiver driver enable

The transmitter can use RTS_B as an enable signal for the driver of an external transceiver. See Transceiver driver enable using RTS_B for details. If the RTS operation is enabled, when a character is placed into an empty transmit shift register, RTS_B asserts 1-bit time before the start bit is transmitted. RTS_B remains asserted for the whole time that the transmit shift register has any characters. RTS_B deasserts 1-bit time after all characters in the transmit FIFO and shift register are completely sent, including the last stop bit. In other words, when RTS_B is used as a transceiver enable, RTS_B asserts 1-bit time before the transmitter starts transmitting and negates 1-bit time after the transmitter goes idle.

Transmitting a break character also asserts RTS_B, with the same assertion and deassertion timing as having a character in the transmit shift register.

The transmitter's RTS_B signal asserts only when the transmitter is enabled. However, the transmitter's RTS_B signal is unaffected by its CTS_B signal. RTS_B remains asserted until the transfer is complete, even if the transmitter is disabled mid-way through a data transfer.

You can configure HDCR[RTSEXT] to the desired length by delaying the transmitter's RTS_B negation by up to 256-bit clock (baud rate) after the last stop bit.

62.3.4.4 Transceiver driver enable using RTS_B

RS-485 is a multiple drop communication protocol in which the LPUART transceiver's driver is three-stated unless LPUART is driving. The transmitter can use the RTS_B signal to enable the driver of a transceiver. The polarity of RTS_B can be matched to the polarity of the transceiver's driver enable signal.

The following figure shows the receiver enable signal asserted. This connection can also connect RTS_B to both DE and RE_B. The transceiver's receiver is disabled when driving. A pullup can pull RXD to a nonfloating value during this time. You can refine this option further by operating LPUART in Single-Wire mode, freeing the RXD pin for other uses.

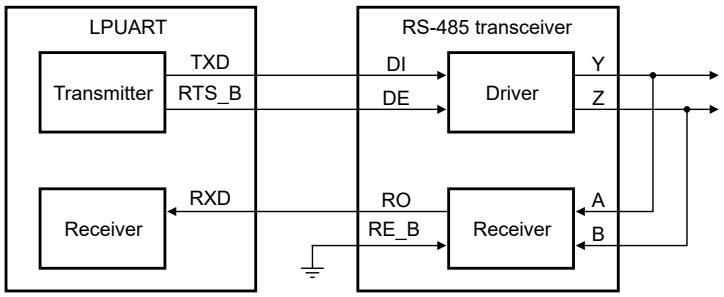


Figure 364. Transceiver driver enable using RTS_B

62.3.5 Receiver functional description

This section discusses the functioning of the LPUART receiver, as shown in the receiver portion of Block diagram. The section also discusses:

- The data sampling technique used to reconstruct receiver data.
- Different variations of the receiver wake-up function.

You can invert the receiver input by writing 1 to STAT[RXINV] and enable the receiver by writing 1 to CTRL[RE]. Character frames consist of a start bit of logic 0, along with N (7, 8, 9, 10) bits (MSB or LSB first), and one or two stop bits of logic 1. For information about 7-bit, 9-bit, or 10-bit Data mode, see Data modes. Going forward in this discussion, assume that LPUART is configured for a normal 8-bit Data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full (STAT[RDRF] = 0), the data character is transferred to the receive FIFO, resulting in STAT[RDRF] becoming 1. However, if STAT[RDRF] is already 1, indicating that the receive data buffer is already full, STAT[OR] becomes 1 and the new data is lost.

Because the LPUART receiver is separate from the receive FIFO, the receive shift register can receive the next word when the receive FIFO is full, and it is only at the end of the character that the next data is written into the receive FIFO, potentially triggering the overrun flag if the FIFO is full.

When a program detects that the receive data register is full (STAT[RDRF] = 1), it gets the data from the FIFO by reading Data (DATA). See Interrupts for details about flag clearing.

62.3.5.1 Data sampling technique

The LPUART receiver supports a configurable oversampling rate of between 4× and 32× of the baud rate clock for sampling. The receiver starts by considering logic level samples at the oversampling rate times the baud rate to search for a falling edge on the RXD serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The oversampling baud rate clock divides the bit time into 4 to 32 segments from 1 to OSR (where OSR is the configured oversampling ratio). When a falling edge is located, three more samples are taken at (OSR ÷ 2), (OSR ÷ 2) + 1, and (OSR ÷ 2) + 2 to ensure that this is a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes they are synchronized to a received character. If another falling edge is detected before the receiver is considered synchronized, the receiver restarts sampling from the first segment.

The receiver then samples each bit time, including the start and stop bits, at (OSR ÷ 2), (OSR ÷ 2) + 1, and (OSR ÷ 2) + 2, to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. If any sample in any bit time, including the start and stop bits, in a character frame fails to agree with the logic level for that bit, noise flag (STAT[NF]) becomes 1 when the received character is transferred to the receive FIFO.

When the LPUART receiver is configured to sample on both edges of the baud rate clock (that is, when BAUD[BOTHEDGE] = 1), the number of segments in each received bit is effectively doubled (from 1 to OSR× 2). The start and data bits are then sampled at OSR, OSR + 1, and OSR + 2. You must enable sampling on both edges of the clock for oversampling rates of 4× to 7×. This sampling is optional for higher oversampling rates.

The synchronization feature of LPUART synchronizes the internal oversampling counter with a detected falling edge on the receive signal, and to adjust the data sampling window. The falling edge detection needs three consecutive 1s prior to the "1->0" (one to zero) transition. After the initial falling edge detection for the start bit, the circuit continuously monitors the next falling edge, and resets the counter after another falling edge is detected. This synchronization to the start bit is termed as resynchronization.

When BAUD[RESYNCDIS] is 0, you perform this falling edge detection and resynchronization not only for the start bit but also for the rest of the character reception after the start bit.

When BAUD[RESYNCDIS] is 1, you perform the falling edge detection and resynchronization only for the start bit. The use case for disabling the resynchronization is protocols that require this (for example, LIN 2.1 prohibits resynchronization within a byte).

The following table and figure explain LPUART resynchronization.

Table 586. LPUART resynchronization settings

Resynchronization	BAUD[RESYNCDIS] = 0	BAUD[RESYNCDIS] = 1
For the starting bit falling edge	Yes	Yes
For all falling edges after the start bit	Yes	No

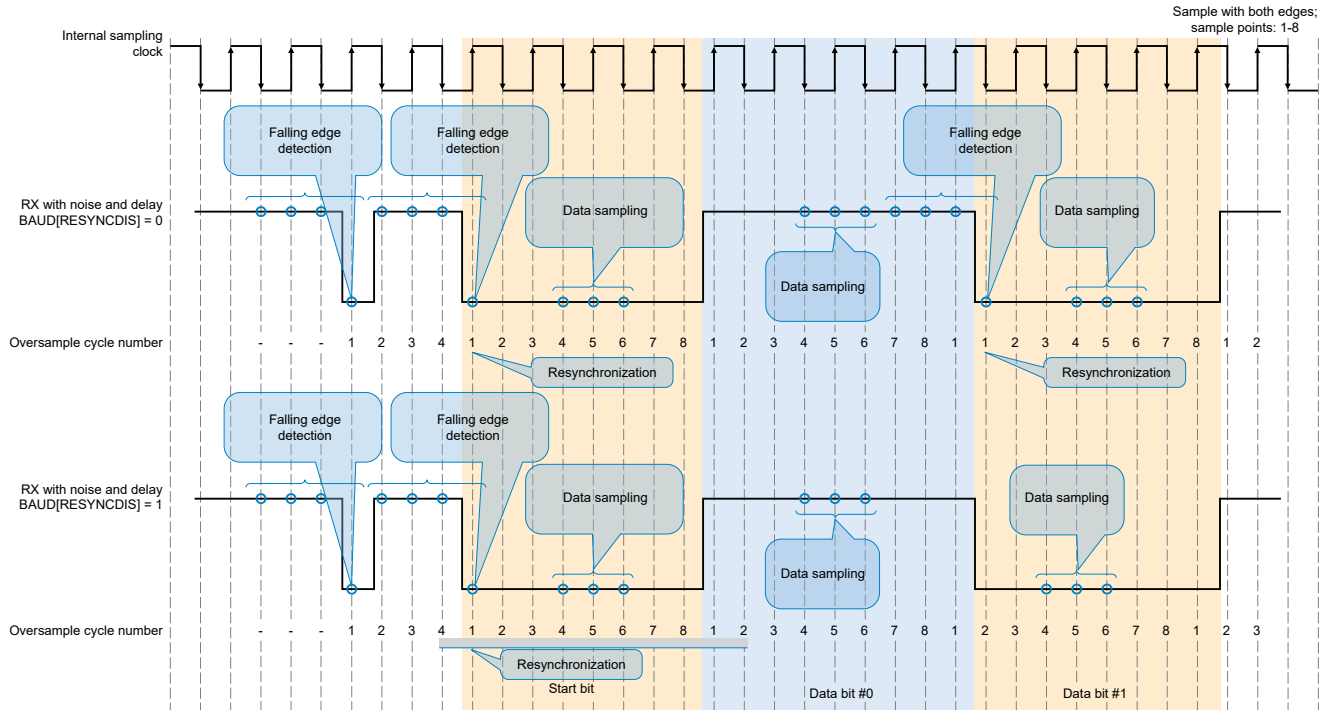


Figure 365. LPUART resynchronization diagram

62.3.5.2 Receiver wake-up operation

Receiver wake-up and receiver address matching are hardware mechanisms that allow an LPUART receiver to ignore the characters in a message intended for a different receiver.

During receiver wake-up, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write 1 to CTRL[RWU].

When CTRL[RWU] and STAT[RWUID] are 1, the status fields associated with the receiver, with the exception of STAT[IDLE], are inhibited from becoming 1, thus eliminating the software overhead for handling the unimportant message characters. At the end of a message, all receivers automatically force CTRL[RWU] to become 0. This results in all receivers waking up in time to look at the first character(s) of the next message.

During receiver address matching, the address matching is performed in hardware and the LPUART receiver ignores all characters that do not meet the address match requirements.

Table 587. Receiver wake-up options

CTRL[RWU]	BAUD[MAEN1] | BAUD[MAEN2]	BAUD[MATCFG]	CTRL[WAKE]: STAT[RWUID]	Receiver wake-up
0	0	X	X	Normal operation
1	0	00	00	Receiver wake-up on idle line; STAT[IDLE] = 0
1	0	00	01	Receiver wake-up on idle line; STAT[IDLE] = 1
1	0	00	10	Receiver wake-up on address mark
1	1	11	10	Receiver wake-up on data match
0	1	00	X0	Address mark address match; STAT[IDLE] = 0 for discarded characters
0	1	00	X1	Address mark address match; STAT[IDLE] = 1 for discarded characters
0	1	01	X0	Idle line address match
0	1	10	X0	Match on and match off; STAT[IDLE] = 0 for discarded characters
0	1	10	X1	Match on and match off; STAT[IDLE] = 1 for discarded characters

62.3.5.2.1 Idle line wake-up

When CTRL[WAKE] is 0, you can configure the receiver for an idle line wake-up. In this mode, CTRL[RWU] becomes 0 automatically when the receiver detects a full character time of the idle-line level.

When CTRL[RWU] is set the receiver ignores everything until an idle condition is detected. After the idle condition is detected the CTRL[RWU] is cleared by hardware and the receiver starts receiving data.

The STAT[RWUID] indicates that whether the STAT[IDLE] flag is set or not as a result of the idle condition detection which clears CTRL[RWU].

The STAT[RWUID] says that if the STAT[IDLE] flag is to be set as a result of the idle condition which clears CTRL[RWU].

CTRL[M], CTRL[M7], and BAUD[M10] select 7-bit to 10-bit Data mode and BAUD[SBNS] selects a 1-bit or 2-bit stop bit number that determines how many bit times of idle are needed to constitute a full character time, 9 to 13 bit times because of the start and stop bits.

When CTRL[RWU] is 1 and STAT[RWUID] is 0, the idle condition that wakes up the receiver does not lead to STAT[IDLE] becoming 1. The receiver wakes up and waits for the first data character of the next message that leads to STAT[RDRF] becoming 1 and generates an interrupt if enabled. When STAT[RWUID] is 1, any idle condition leads to STAT[IDLE] becoming 1 and generates an interrupt if enabled, regardless of whether CTRL[RWU] is 0 or 1.

These are the ways to detect an idle line:

- When CTRL[ILT] is 0, the idle bit counter starts after the start bit so that the stop bit and any logic 1s at the end of a character count to calculate the full character time of idle.
- When CTRL[ILT] is 1, the idle bit counter does not start until after the stop bit time so that the data in the last character of the previous message does not impact the idle detection.

62.3.5.2.2 Address mark wake-up

When CTRL[WAKE] is 1, you can configure the receiver for an address mark wake-up. In this mode, CTRL[RWU] becomes 0 automatically when the receiver detects a logic 1 in the most significant bit of the received character. When parity is enabled, the second most significant bit is used for address mark wake-up.

Address mark wake-up allows messages to contain idle characters, but requires one bit to be reserved for use in address frames. The logic 1 in the most significant bit (or second most significant bit when parity is enabled) of an address frame writes 0 to CTRL[RWU] and writes 1 to STAT[RDRF]. In this case, the character with the address mark bit is received even if the receiver is sleeping during most of this character time.

62.3.5.2.3 Data match wake-up

When CTRL[RWU] and CTRL[WAKE] are 1, and BAUD[MATCFG] equals 11, the receiver is configured for a data match wake-up. In this mode, CTRL[RWU] becomes 0 automatically when the receiver detects a character that matches MATCH[MA1] when BAUD[MAEN1] is 1, or that matches MATCH[MA2] when BAUD[MAEN2] is 1.

62.3.5.2.4 Address match operation

You can enable the address match operation when either BAUD[MAEN1] or BAUD[MAEN2] is 1 and BAUD[MATCFG] is 0. In this function, a character that the RXD pin receives with a logic 1 in the most significant bit (or the second most significant bit when parity is enabled) is considered an address and is compared to the associated MATCH[MA1] or MATCH[MA2]. The character is only transferred to the receive buffer, and STAT[RDRF] becomes 1 if the comparison matches. All subsequent characters received with a logic 0 in the most significant bit (or the second most significant bit when parity is enabled) are considered to be data associated with the address and are transferred to the receive FIFO. If no marked address match occurs, no transfer is made to the receive FIFO, and all the characters that follow, with logic 0 in the most significant bit (or second most significant bit when parity is enabled), are also discarded. If both BAUD[MAEN1] and BAUD[MAEN2] are 0, the receiver operates normally, and all the received data is transferred to the receive FIFO.

The address match operation functions in the same way for both MATCH[MA1] and MATCH[MA2]:

- If either BAUD[MAEN1] or BAUD[MAEN2] is 1, a marked address is compared only to the associated Match Address (MATCH) and data is transferred to the receive FIFO only on a match.
- If both BAUD[MAEN1] and BAUD[MAEN2] are 1, a marked address is compared to both MATCH[MA1] and MATCH[MA2] and data is transferred only on a match with either of these fields.

62.3.5.2.5 Idle match operation

You can enable the idle match operation when either BAUD[MAEN1] or BAUD[MAEN2] is 1 and BAUD[MATCFG] is 1. In this function, the first character that the RXD pin receives after an idle line condition is considered an address and is compared to the associated MATCH[MA1] or MATCH[MA2]. The character is transferred only to the receive buffer, and STAT[RDRF] becomes 1, if the comparison matches. All subsequent characters are considered to be data associated with the address and are transferred

to the receive FIFO until the next idle line condition is detected. If no address match occurs, no transfer is made to the receive FIFO, and all the frames that follow, until the next idle condition, are also discarded. If both BAUD[MAEN1] and BAUD[MAEN2] are 0, the receiver operates normally, and all the received data is transferred to the receive FIFO.

An idle match operation functions in the same way for both MATCH[MA1] and MATCH[MA2]:

- If either BAUD[MAEN1] or BAUD[MAEN2] is 1, the first character after an idle line is compared only to the associated Data (DATA) and data is transferred to the receive FIFO only on a match.
- If both BAUD[MAEN1] and BAUD[MAEN2] are 1, the first character after an idle line is compared to both MATCH[MA1] and MATCH[MA2] and data is transferred only on a match with either of these fields.

62.3.5.2.6 Match on, match off operation

The match on, match off operation is enabled when both BAUD[MAEN1] and BAUD[MAEN2] are 1 and BAUD[MATCFG] = 10. In this function, a character that the RXD pin receives matches MATCH[MA1] and is transferred to the receive buffer, and STAT[RDRF] becomes 1. All subsequent characters are considered to be data and are also transferred to the receive FIFO, until a character that matches MATCH[MA2] is received. The character that matches MATCH[MA2], along with all subsequent characters, is discarded; and this continues until another character that matches MATCH[MA1] is received. If both BAUD[MAEN1] and BAUD[MAEN2] are 0, the receiver operates normally, and all the received data is transferred to the receive FIFO.

NOTE

The match on, match off operation requires both BAUD[MAEN1] and BAUD[MAEN2] to be 1.

62.3.5.3 Hardware flow control

To support hardware flow control, you can program the receiver to automatically assert and deassert RTS_B:

- RTS_B remains asserted until the transfer is complete, even if the transmitter is disabled midway through a data transfer. See Transceiver driver enable using RTS_B for more information.
- If the receiver RTS functionality is enabled, the receiver automatically deasserts RTS_B if STAT[RDRF] is 1 or a start bit is detected that causes STAT[RDRF] to become 1.
- The receiver asserts RTS_B when STAT[RDRF] is 0 and has not detected a start bit that causes STAT[RDRF] to become 1. There is no impact if STAT[RDRF] is 1 already.
- Even if RTS_B is deasserted, the receiver continues to receive characters until the receive FIFO is overrun.
- If the receiver RTS functionality is disabled, the receiver's RTS_B remains deasserted.
- When RTS is driven by receiver, it will be negated when the receiver is disabled irrespective of the FIFO level.

62.3.6 Additional LPUART functions

62.3.6.1 Data modes

You can configure the LPUART transmitter and receiver to operate in 7-bit Data mode by writing 1 to CTRL[M7], 9-bit Data mode by writing 1 to CTRL[M], or 10-bit Data mode by writing 1 to BAUD[M10]. In 9-bit Data mode, there exists a ninth data bit and in 10-bit mode, there exists a tenth data bit.

When performing 8-bit writes to the transmit FIFO, the ninth and tenth bits are pushed into the FIFO from CTRL[T8] and CTRL[T9]. For coherent 8-bit writes, you must write to CTRL[T8] and CTRL[T9] before writing to Data (DATA)[7:0]. However, if the values in CTRL[T8] or CTRL[T9] do not need to change, it is not necessary to update CTRL[T8] and CTRL[T9] before every 8-bit write to Data (DATA).

When performing 16-bit or 32-bit writes to the transmit FIFO, all 10 bits are pushed into the transmit FIFO from the write data.

When performing 8-bit reads of the receive FIFO, the ninth and tenth bits are held in CTRL[R8] and CTRL[R9] but you must read them before reading Data (DATA). A 16-bit or 32-bit read of the receive FIFO returns all 10 bits in Data (DATA).

The 9-bit Data mode is typically used with parity to allow eight bits of data plus the parity in the ninth bit, or it is used with the address mark wake-up so that the ninth data bit can serve as the wake-up bit. The 10-bit Data mode is typically used with parity and address mark wake-up so that the ninth data bit can serve as the wake-up bit and the tenth bit can serve as the parity bit. In custom protocols, the ninth and/or tenth bits can also serve as software-controlled markers.

62.3.6.2 Idle length

An idle character is one where the start bit, all data bits, and stop bits are in the mark position (idle state, generally logic 1). You can configure CTRL[ILT] to start detecting an idle character from the previous start bit (any data bits and stop bits count for idle character detection) or from the previous stop bit.

You can also use CTRL[IDLECFG] to configure the number of idle characters that must be received before an idle line condition is detected. This field configures the number of idle characters that must be received before STAT[IDLE] becomes 1, STAT[RAF] becomes 0, and DATA[IDLINE] becomes 1 with the next received character.

CTRL[IDLECFG] also affects the idle line wake-up and idle match operations. When either the address match or match on/off operation is enabled, writing 1 to STAT[RWUID] causes any discarded characters to be treated as idle characters.

After the extended idle time is enabled for the receiver, you can configure an idle line condition by using REIR[IDTIME], which specifies the number of bits (baud rate) since the last stop bit that is required for an idle condition to be detected. This replaces the configuration of CTRL[ILT] and CTRL[IDLECFG].

The transmitter can also enable the extended idle time. In this case, any idle character queued through the transmit FIFO forces the transmitter to be idled for the configured number of bit clocks before the idle character is read from the FIFO and the transmitter continues.

After you enable the transmitter extended idle time, the transmitter does not automatically queue an idle character whenever it is enabled.

62.3.6.3 Loop mode

Enable Loop mode by setting CTRL[LOOPS] = 1 and CTRL[RSRC] = 0. You, sometimes, use Loop mode to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and LPUART does not use the RXD pin.

Loop mode also internally connects the RTS_B output to the CTS_B input and the DTR_B output to the DSR_B input.

62.3.6.4 Single-Wire mode

Enable Single-Wire mode by setting CTRL[LOOPS] = 1 and CTRL[RSRC] = 1. Single-Wire mode implements a half-duplex serial connection. The receiver is internally connected to the transmitter output and TXD pin (the RXD pin is not used).

In Single-Wire mode, CTRL[TXDIR] controls the direction of serial data on the TXD pin. When CTRL[TXDIR] becomes 0, the TXD pin is an input to the receiver and the transmitter is temporarily disconnected from the TXD pin so that an external device can send serial data to the receiver. When CTRL[TXDIR] = 1, the TXD pin is an output that the transmitter drives. The internal loop back connection is disabled, and as a result, the receiver is unable to receive characters that the transmitter sends out.

Half Duplex Control (HDCR) replaces the implementation of CTRL[LOOPS] and CTRL[RSRC], and you can use this register to configure various options for both Single-Wire and Half-Duplex operations, using independent RXD and TXD pins:

- HDCR[TXSTALL] replaces the CTRL[TXDIR] functionality and prevents the transmitter from becoming busy or asserting the RTS_B transmitter if STAT[RAF] is 1.
- You can select the TXD pin, as the source for the receiver, to configure HDCR[RXSEL] for a single-wire operation. If HDCR[RXSEL] is 1, you must configure the TXD pin for an open-drain operation.
- HDCR[RXMSK] masks the receiver input when the RTS_B transmitter is asserted (this applies even if you have not configured RTS_B as an output).
- HDCR[RXWRMSK] blocks storage of the receive data in the receive FIFO when the RTS_B transmitter is asserted. This setting does not affect the receiver idle functionality.
- HDCR[RTSEXT] delays the negation of the RTS_B transmitter by the configured number of bit clocks (baud rate).

62.3.6.5 Timeout counter

LPUART implements four general-purpose timeout counters; counters 0 and 1 are used to monitor the receiver and counters 2 and 3 are used to monitor the transmitter. When enabled, you can configure each counter to monitor one of the following conditions:

- Idle time in number of bits, starting to increment when first enabled and an idle condition is detected. The counter restarts whenever a character is received or transmitted.
- Idle time in number of bits, starting to increment after the next character is received or transmitted. The counter restarts whenever a character is received or transmitted.
- Idle time is more than the timeout interval, in number of bits, but less than the extended idle timeout. The counter restarts whenever a character is received or transmitted. You can use this to detect a gap between characters that is greater than a threshold (timeout) but less than the configured extended idle time. This configuration requires the extended idle feature for the transmitter or receiver to be enabled for a proper operation.
- Number of characters received or transmitted is equal to the configured timeout. The counter asserts at the start of the character that equals the timeout value.

The timeout counters restart counting whenever the counter is disabled and then enabled. These timeout counters are disabled when the corresponding STAT[TSF] field is 1. If the timeout enable is still set after STAT[TSF] becomes 0, the timeout counter restarts as if the counter had been disabled and then enabled. You can measure the idle time in bit times from the end of the last stop bit and until a start bit is validated.

62.3.7 Peripheral triggers

The connection of the LPUART peripheral triggers with other peripherals is chip-specific.

62.3.7.1 Output triggers

LPUART generates the following output triggers that can be connected to other peripherals on the chip:

- The transmit word trigger asserts at the end of each transmitted word and negates after 1-bit period.
- The transmit data trigger is identical to the TXD pin output, but without support for input trigger modulation.
- The receive word trigger asserts at the end of each received word that is written to the receive FIFO, for one oversampling clock period.
- The receive idle trigger asserts when STAT[IDLE] becomes 1, and negates when the next valid start bit is detected.

62.3.7.2 Input trigger

LPUART supports a peripheral input trigger that you can configure in one of the following ways:

- By enabling the CTS function: You can connect the input trigger (active high) instead of the CTS_B (active low) pin input. The input trigger must assert for longer than 1-bit clock period when the transmitter is idle, with data to send, to guarantee a new transmission.
- By making the input trigger modulate the transmit data output (trigger is logically ANDed with the TXD output): The input trigger is expected to be a free-running clock (carrier signal) that generates from a timer or PWM source with a frequency that is greater than the bit-clock frequency. The carrier signal must not toggle faster than the maximum supported bit time.
- By connecting the input trigger instead of the RXD pin input: The input trigger is expected to be generated from a receive data source, such as an analog comparator or external pin.

62.3.8 Infrared (IR) interface

LPUART provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses, transforming them to serial bits, which are then sent to LPUART. The IrDA physical layer specification defines a half-duplex IR communication link for exchanging data. The full standard includes data rates up to 16 Mbit/s. The LPUART IrDA support is limited to SIR mode that supports data rates only between 2.4 kbit/s and 115.2 kbit/s.

LPUART has an infrared transmit encoder and a receive decoder. The infrared decoder converts the received character from the IrDA format to the NRZ format, which the receiver uses. It also has an OSR oversampling baud rate clock counter that filters noise and indicates when a 1 is received. LPUART transmits serial bits of data, which the infrared submodule encodes, to transmit a narrow pulse for every zero bit. No pulse is transmitted for every single bit. When receiving data, an IR photo diode (external to LPUART) detects the IR pulses. The IR receive decoder transforms them to CMOS levels. The infrared receive decoder then stretches the narrow pulses to get back to a serial bit stream that LPUART receives. You can invert the polarity of transmitted pulses and expected receive pulses so that a direct connection can be made to external IrDA transceiver modules that use active-high pulses.

The IR submodule receives its clock sources from LPUART. The submodule selects one of these clocks to generate either 1 ÷ OSR, 2 ÷ OSR, 3 ÷ OSR, or 4 ÷ OSR narrow pulses during transmission.

62.3.8.1 Infrared transmit encoder

The infrared transmit encoder converts serial bits of data from the transmit shift register to the TXD signal. A narrow pulse is transmitted for a 0 bit and no pulse is transmitted for a 1 bit. The narrow pulse is sent at the start of the bit with a duration of 1 ÷ OSR, 2 ÷ OSR, 3 ÷ OSR, or 4 ÷ OSR of a bit time. A narrow low pulse is transmitted for a 0 bit when CTRL[TXINV] is 0, while a narrow high pulse is transmitted for a 0 bit when CTRL[TXINV] is 1.

62.3.8.2 Infrared receive decoder

The infrared receive block converts data from the RXD signal to the receive shift register. A narrow pulse is expected for each 0 received and no pulse is expected for each 1 received. A narrow low pulse is expected for a 0 bit when STAT[RXINV] is 0, while a narrow high pulse is expected for a 0 bit when STAT[RXINV] is 1. This receive decoder meets the edge jitter requirement as defined by the IrDA serial infrared physical layer specification.

62.3.8.3 Start-bit detection

When STAT[RXINV] is 0, the first falling edge of the received character corresponds to the start bit. The infrared decoder resets its counter. At this time, the receiver also begins its start bit detection process. After the start bit is detected, the receiver synchronizes its bit times to this start bit time. For the rest of the character reception, the infrared decoder's counter and the receiver's bit time counter count independently of each other.

62.3.8.4 Noise filtering

The decoder ignores any rising edges detected during the first half of the infrared decoder counter, and can leave any pulses less than one oversampling baud clock as undetected. This is regardless of whether the pulse is seen in the first or second half of the count.

62.3.8.5 Low-bit detection

During the second half of the decoder count, a rising edge is decoded as 0, which is sent to the receiver. The decoder counter is also reset.

62.3.8.6 High-bit detection

At OSR oversampling baud rate clocks after the previous rising edge, if a rising edge is not seen, the decoder sends a 1 to the receiver.

If the next bit is 0, which arrives late, a low bit is detected according to Low-bit detection. The value sent to the receiver is changed from 1 to 0. Then, if a noise pulse occurs outside the receiver's bit time sampling period, the delay of a 0 is not recorded as noise.

62.3.9 MODBUS protocol

62.3.9.1 MODBUS frame structure

MODBUS is an application layer messaging protocol, positioned at level 7 of the OSI model. It provides client and server communication between devices connected on different types of buses or networks.

The MODBUS protocol defines a simple protocol data unit (PDU) independent of the underlying communication layers. The mapping of the MODBUS protocol on specific buses or network can introduce some additional fields on the application data unit (ADU).

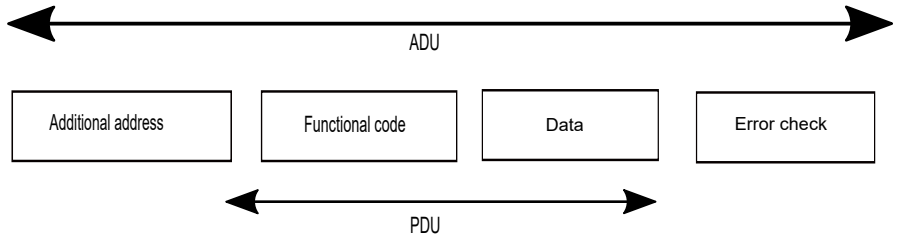


Figure 366. MODBUS frame structure

The function code field informs the server about which action to perform, as requested by the client (only the client can initiate the communication), where the data field contains additional information that the server uses to take the action defined by the function code. MODBUS PDU for serial line communication = 256 - server address (1 byte) - CRC (2 bytes) = 253 bytes.

You can set up a MODBUS controller to communicate on standard MODBUS networks using either of the two transmission modes: ASCII or RTU. RTU mode allows better character density and throughput for the same baud rate as the ASCII format allows 7 bits to represent a character where RTU keeps 8 bits for data representation in each packet.

62.3.9.2 MODBUS frame for LPUART

Though the protocol resides in the application layer, it can also be checked at the physical layer using LPUART by considering the frame structure shown in the following figure, following RTU mode of data transmission.

Start	Address	Function	Data	CRC	End
3.5 Char time	8 Bit	8 Bit	N * 8Bit	16 Bit	3.5 Char time

Figure 367. MODBUS frame for LPUART

An entire message frame of MODBUS must contain start character, address, function code, data, and end character. LPUART does not support CRC calculation for TX and RX, instead, each LPUART packet containing frame information is transmitted with its individual parity, and the same is checked in RX. Following the last transmitted character, a similar interval of at least 3.5 character times marks the end of the message and a new message can begin after this interval. The entire message frame must be transmitted as a continuous stream. If a silent interval of more than 1.5 character times occurs before the completion of the frame, the receiving device flushes the incomplete message, and assumes that the next byte is the address field of a new message. The correctness of timeout logic ensures the correctness of the MODBUS frame using LPUART.

Registers such as REIR, TEIR, TOCR, TOSR, TIMEOUT[0 - 3], TCBR, and TDBR are useful for realizing MODBUS using LPUART. The TX IDLE time is configured by using TEIR[IDTIME].

For higher baud rates, MODBUS recommends adopting fixed t1.5 and t3.5 times of 750us ms and 1.75 ms. The receiver idle line wake-up can be used by a MODBUS device to only wake-up on a matching address or broadcast address received after a valid idle time. The receiver end-of-packet DMA transfer can be used to offload the reception of a MODBUS packet onto the DMA controller.

62.3.9.3 MODBUS TX and RX programming sequence

The following programming sequences can be useful for sending and receiving MODBUS frames using LPUART.

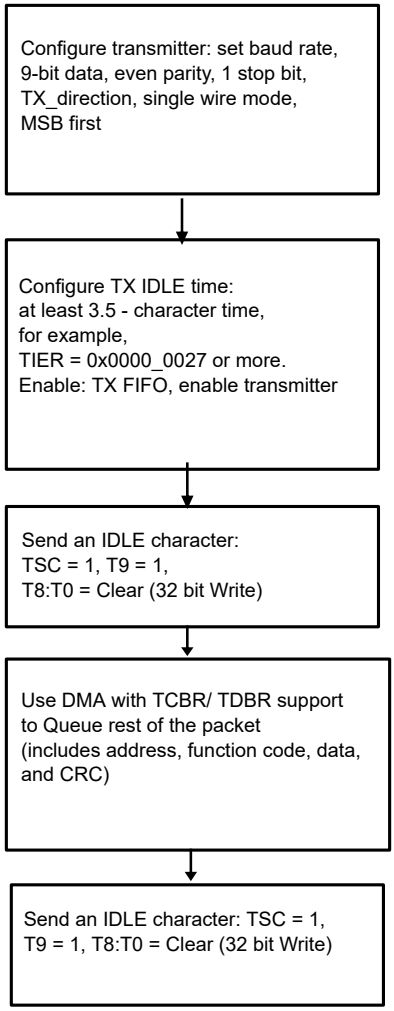


Figure 368. MODBUS TX programming

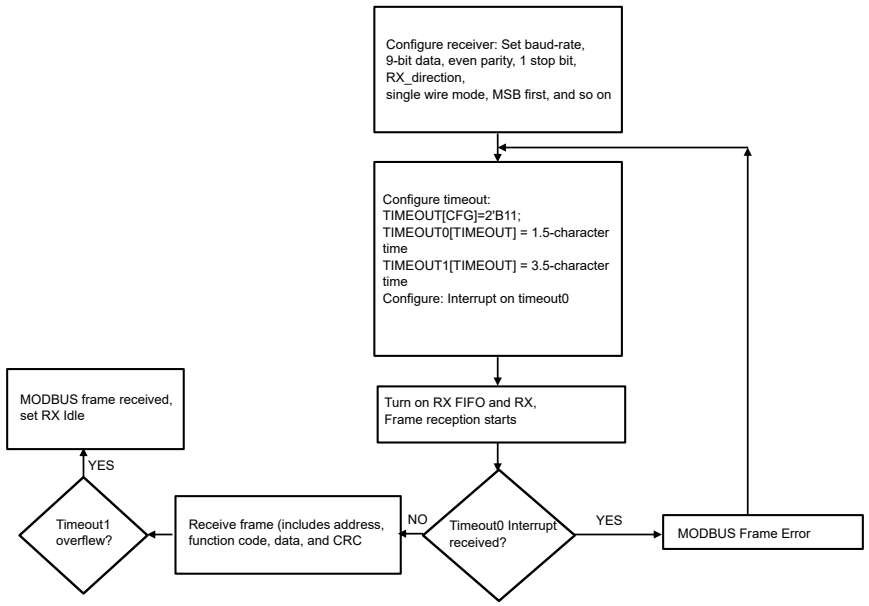


Figure 369. MODBUS RX programming

62.3.10 Modes of operation

62.3.10.1 Low-Power modes

LPUART remains functional during low-power modes, provided CTRL[DOZEEN] is 0 and the LPUART functional clock is enabled. LPUART can generate an interrupt or DMA request to cause a wake-up from low-power modes.

You can configure LPUART to be disabled in low-power modes, when CTRL[DOZEEN] is 1. In this case, the transmitter and receiver finish transmitting and receiving the current word.

If LPUART is disabled in low-power modes, it can generate a wake-up via STAT[RXEDGIF] if the receiver detects an active edge.

NOTE

See the chip-specific information for specific low-power modes available on your chip.

62.3.10.2 Debug mode

LPUART remains functional in Debug mode.

62.3.11 Clocking

Table 588. Types of clocks

Clock	Description
Functional	Is asynchronous to the bus clock and can remain enabled in Low-Power modes to support transmit and/or receive functions, including low-power wake-up.
Bus	Is only used for bus accesses to the control and configuration registers. The bus clock frequency must be sufficient to support the data bandwidth requirements of the LPUART transmit and receive registers, including the FIFOs.

62.3.12 Reset

Table 589. Types of resets

Reset	Description
Chip	Enables the logic and registers for the LPUART transmitter and receiver to reset to their default states.
Software	Resets the LPUART logic and registers to their default states, except for Global (GLOBAL). GLOBAL[RST] controls the LPUART software reset.
FIFO	Implements write-only control fields that reset the transmit FIFO (FIFO[TXFLUSH]) and receive FIFO (FIFO[RXFLUSH]). After a FIFO is reset, that FIFO becomes empty.

62.3.13 Interrupts

The LPUART transmitter has two status fields that can optionally generate hardware interrupt requests. If STAT[TDRE] is 1, it indicates that there is room in the transmit FIFO to write another transmit character to Data (DATA). If CTRL[TIE] is 1, a hardware interrupt is requested when STAT[TDRE] is 1.

STAT[TC] indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with TXD at the inactive level. This field is often used in systems with modems to determine when it is safe to turn off the modem. If CTRL[TCIE] is 1, a hardware interrupt is requested when STAT[TC] is 1. Instead of hardware interrupts, software polling may be used to monitor STAT[TDRE] and STAT[TC] if the corresponding CTRL[TIE] or CTRL[TCIE] field is 0.

When a program detects that STAT[RDRF] is 1, it gets the data from this field by reading Data (DATA). The field becomes 0 by reading Data (DATA).

STAT[IDLE] includes logic that prevents it from becoming 1 repeatedly when the RXD line remains idle for an extended period of time. STAT[IDLE] becomes 0 when you write 1 to it, and cannot become 1 again until the receiver has received at least one new character and has 1 as the value of STAT[RDRF].

If the associated error is detected in the received character that caused STAT[RDRF] to become 1, STAT[NF], STAT[FE], and STAT[PF] become 1 at the same time STAT[RDRF] becomes 1. These flags do not become 1 in overrun cases.

If STAT[RDRF] is already 1 when a new character is ready to be transferred from the receive shifter to the receive FIFO, STAT[OR] becomes 1, instead of the data along with any associated STAT[NF], STAT[FE], or STAT[PF] condition getting lost.

If the received character matches the contents of MATCH[MA1] and/or MATCH[MA2], then STAT[MA1F] and/or STAT[MA2F] become 1 at the same time that STAT[RDRF] becomes 1.

At any time, an active edge on the RXD serial data input pin causes STAT[RXEDGIF] to become 1. STAT[RXEDGIF] becomes 0 when you write 1 to it. This function depends on the receiver being enabled (the value of CTRL[RE] being 1).

MODEM Status (MSR) can generate an interrupt from a configured status field, which STAT[MSF] indicates.

Timeout Status (TOSR) can generate an interrupt from a configured status field, which STAT[TSF] indicates.

62.3.14 DMA

62.3.14.1 DMA burst support

To support efficient DMA transfers to the transmit FIFO, two alias regions are implemented to support incrementing 8-bit, 16-bit, or 32-bit write accesses to the transmit FIFO:

- Transmit Command Burst (TCBR0 - TCBR127) is a 512-byte region that supports pushing 16-bit data into the transmit FIFO.
- Transmit Data Burst (TDBR0 - TDBR255) is a 1024-byte region that supports pushing zero extended 8-bit data into the transmit FIFO.

The aforementioned regions are contiguous, so a DMA transfer can start in Transmit Command Burst (TCBR0 - TCBR127) to initialize the transfer including address mark, idle word, or break character, and then complete the transfer in Transmit Data Burst (TDBR0 - TDBR255) with the data to transmit, without changing the transfer size.

The transmit FIFO block writes overflow the FIFO, but that does not signal an error. Do not perform 32-bit writes to Transmit Data Burst (TDBR0 - TDBR255) unless there are four empty slots in the transmit FIFO, and do not perform 16-bit writes to this register unless there are two empty slots in the transmit FIFO.

62.3.14.2 End-of-packet DMA transfers

The end-of-packet functionality is designed for serial interfaces where you may not know the size of the transfer in advance and the data is being pushed by an external device. The end-of-packet processing ensures that data does not become stranded in either the receive FIFO or the DMA receive buffer. Support for end-of-packet processing must be implemented in both the serial interfaces and DMA controller.

The condition that signals the end of packet is different for each serial interface, but the serial peripheral and DMA process it in the same way. For example, an idle line condition signals the UART end of packet, the Stop and/or Repeated Start condition signals the I2C end of packet, and PCS negation signals the SPI end of packet.

When the serial peripheral is configured to signal the end-of-packet condition to the DMA and the serial interface detects an end-of-packet condition, it asserts the DMA request for the receive FIFO irrespective of the watermark configuration. For larger watermark configurations, this ensures that the last few words of the transfer are first flushed from the receive FIFO.

The DMA then reads the contents of the receive FIFO, depending on the first word in the FIFO:

- If the receive FIFO is empty, the serial interface signals an end of packet condition to the DMA controller.
- If the receive FIFO is not empty, but the first word in the FIFO is the start of a new packet, data is not pulled from the receive FIFO and the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, and the first word in the FIFO is not the start of a new packet, the DMA transfers the receive data as normal.

Because the DMA may be transferring multiple words on each request, the end-of-packet condition persists until the DMA minor loop has completed and no additional data is pulled from the receive FIFO. The field that triggered the end-of-packet condition becomes 0 when the minor loop completes following the end of packet being signaled to the DMA controller.

When the DMA detects the end-of-packet condition, it writes all received words up to the end of packet into the system memory and saves the destination address for the word after the last valid data. The DMA then terminates the channel as if the major loop completed, including final offsets and optional interrupts, channel linking, and scatter-gather. You can, optionally, save the final destination address to the system memory. The final destination address is also available in the destination address register.

Because the DMA terminates the major loop, no servicing of the receive FIFO occurs until you or the hardware reconfigures the DMA (for example, channel linking or scatter-gather). You must minimize this delay to avoid receiver FIFO overrun. The automatic DMA end-of-packet processing is not recommended when there are only a few words transferred between end-of-packet conditions. This is because the DMA spends more time processing the end of packet than transferring the data. For example, the UART idle line length must be increased as needed to avoid an excessive number of idle conditions.

62.4 External signals

Table 590. External signals

Signal	Description	I/O
TXD	Transmit data: This pin is normally an output, but is an input (tristated) in Single-Wire mode whenever the transmitter is disabled or the transmit direction is configured for receive data.	I/O

Table continues on the next page...

Table 590. External signals (continued)

Signal	Description	I/O
RXD	Receive data	I
CTS_B	Clear-to-send	I
RTS_B	Request-to-send	O
DTR_B	Data terminal ready	O
DSR_B	Data set ready	I
DCD_B	Data carrier detect	I
RIN_B	Ring indicator	I

62.5 Initialization

This module does not require initialization.

62.6 LPUART register descriptions

LPUART includes registers to control baud rate, select options, report status, and store transmit and receive data. Access to an address outside the valid memory map generates a bus error.

NOTE

Writing to a read-only (RO) register or reading a write-only (WO) register can cause bus errors. LPUART does not verify whether programmed values in the registers are correct; you must write valid values to them.

62.6.1 LPUART memory map

LPUART1 base address: 4438_0000h

LPUART2 base address: 4439_0000h

LPUART3 base address: 4257_0000h

LPUART4 base address: 4258_0000h

LPUART5 base address: 4259_0000h

LPUART6 base address: 425A_0000h

LPUART7 base address: 4269_0000h

LPUART8 base address: 426A_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VERID)	32	R	0404_0007
4	Parameter (PARAM)	32	R	0000_0404
8	Global (GLOBAL)	32	RW	0000_0000
C	Pin Configuration (PINCFG)	32	RW	0000_0000
10	Baud Rate (BAUD)	32	RW	0F00_0004

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
14	Status (STAT)	32	RW	00C0_0000
18	Control (CTRL)	32	RW	0000_0000
1C	Data (DATA)	32	RW	0000_1000
20	Match Address (MATCH)	32	RW	0000_0000
24	MODEM IrDA (MODIR)	32	RW	0000_0000
28	FIFO (FIFO)	32	RW	00C0_0033
2C	Watermark (WATER)	32	RW	0000_0000
30	Data Read-Only (DATARO)	32	R	0000_1000
40	MODEM Control (MCR)	32	RW	0000_0000
44	MODEM Status (MSR)	32	RW	0000_0000
48	Receiver Extended Idle (REIR)	32	RW	0000_0000
4C	Transmitter Extended Idle (TEIR)	32	RW	0000_0000
50	Half Duplex Control (HDCR)	32	RW	0000_0000
58	Timeout Control (TOCR)	32	RW	0000_0000
5C	Timeout Status (TOSR)	32	RW	0000_000F
60 - 6C	Timeout N (TIMEOUT0 - TIMEOUT3)	32	RW	0000_0000
200 - 3FC	Transmit Command Burst (TCBR0 - TCBR127)	32	W	0000_0000
400 - 7FC	Transmit Data Burst (TDBR0 - TDBR255)	32	W	0000_0000

62.6.2 Version ID (VERID)

Indicates the version integrated for this instance on the chip and also specifies the inclusion and exclusion of several optional features.

Offset

Register	Offset
VERID	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Fields

Field	Description
31-24 MAJOR	Major Version Number Indicates the major version number for the module specification.
23-16 MINOR	Minor Version Number Indicates the minor version number for the module specification.
15-0 FEATURE	Feature Identification Number Indicates the feature set number. 0000_0000_0000_0001 - Standard feature set 0000_0000_0000_0011 - Standard feature set with MODEM and IrDA support 0000_0000_0000_0111 - Enhanced feature set with full MODEM, IrDA, and enhanced idle detection

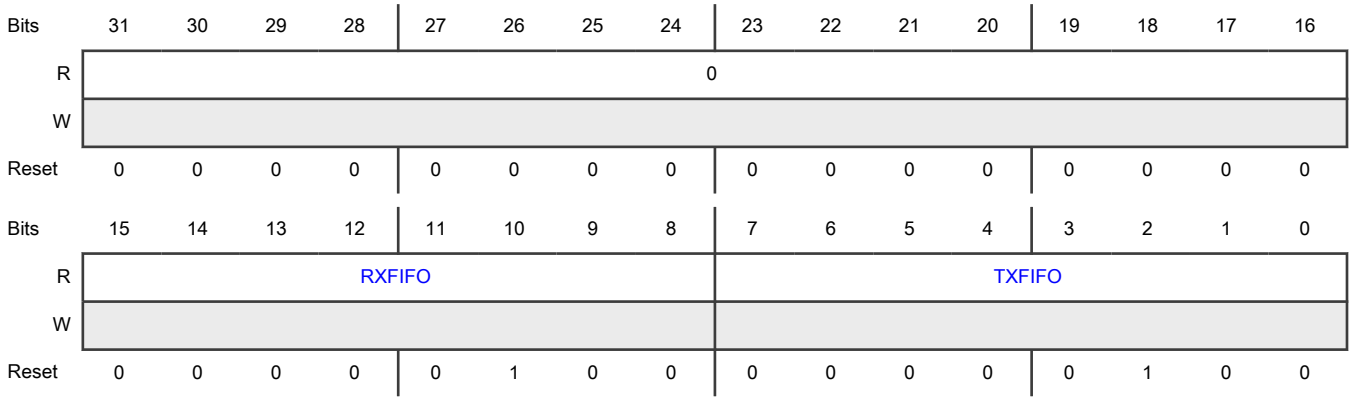
62.6.3 Parameter (PARAM)

Indicates the parameter configuration for this instance on the chip.

Offset

Register	Offset
PARAM	4h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 RXFIFO	Receive FIFO Size Indicates an index used to determine the maximum capacity of the receive FIFO in terms of data units. • 1 data unit = a character plus any additional bits. • The actual capacity is calculated as: — Capacity = 2^{RXFIFO} • Example: — If the RXFIFO field is 3, then the maximum capacity is: 2^3 = 8 data units
7-0 TXFIFO	Transmit FIFO Size Indicates an index used to determine the maximum capacity of the transmit FIFO in terms of data units. • 1 data unit = a character plus any additional bits. • The actual capacity is calculated as: — Capacity = 2^{TXFIFO} • Example: — If the TXFIFO field is 3, then the maximum capacity is: 2^3 = 8 data units

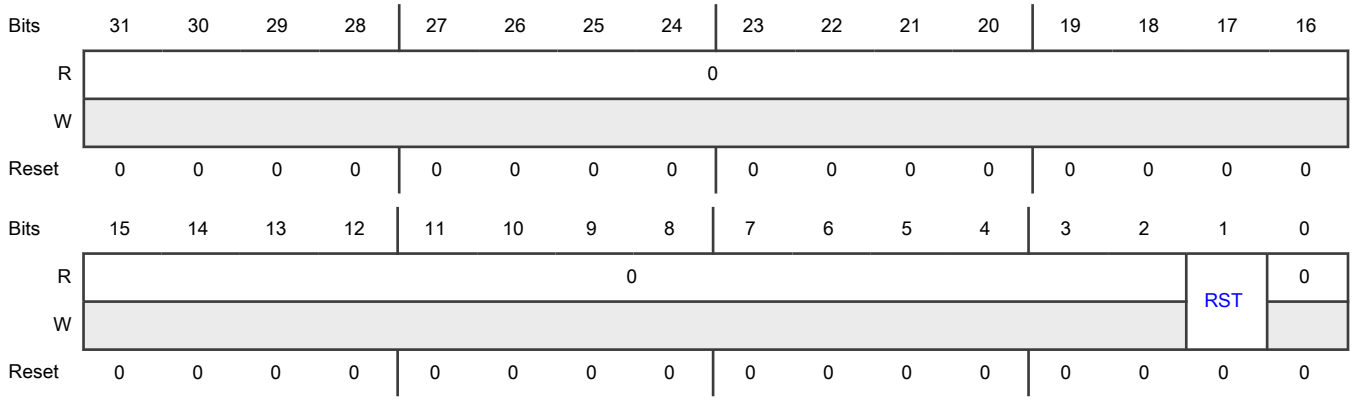
62.6.4 Global (GLOBAL)

Performs global functions.

Offset

Register	Offset
GLOBAL	8h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 RST	Software Reset Specifies whether the module is reset. This field resets all internal logic and registers, except Global (GLOBAL). The reset takes effect immediately and remains asserted until you negate it. There is no minimum delay required before clearing the software reset. 0 - Not reset 1 - Reset
0 —	Reserved

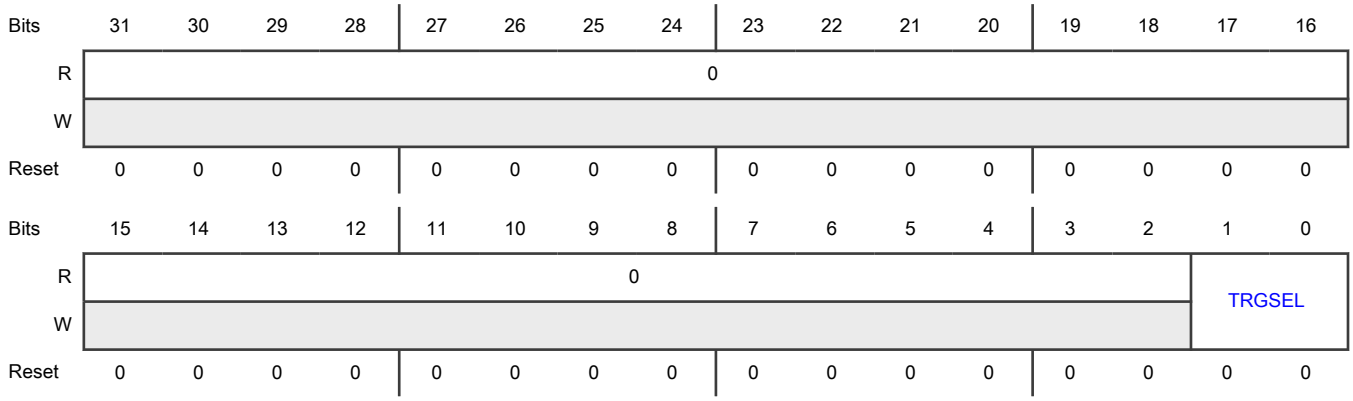
62.6.5 Pin Configuration (PINCFG)

Enables the selection of input pins.

Offset

Register	Offset
PINCFG	Ch

Diagram



Fields

Field	Description
31-2 —	Reserved
1-0 TRGSEL	Trigger Select Configures the input trigger usage. You must change the value of this field only when both the transmitter and receiver are disabled. 00 - Input trigger disabled 01 - Input trigger used instead of the RXD pin input 10 - Input trigger used instead of the CTS_B pin input 11 - Input trigger used to modulate the TXD pin output, which (after TXINV configuration) is internally ANDed with the input trigger

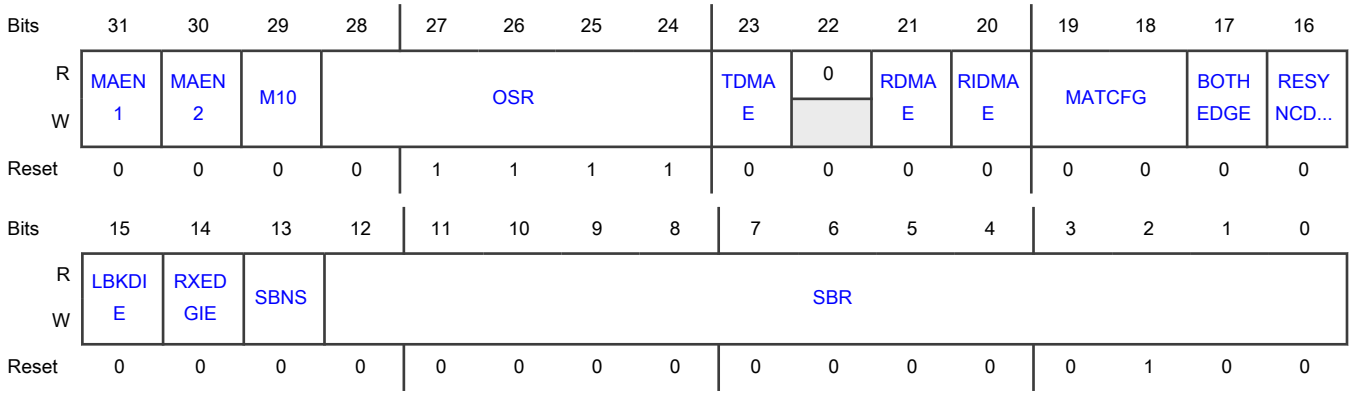
62.6.6 Baud Rate (BAUD)

Configures the baud rate.

Offset

Register	Offset
BAUD	10h

Diagram



Fields

Field	Description
31 MAEN1	Match Address Mode Enable 1 Enables automatic address matching or data matching mode for MATCH[MA1]. If this field is 0, normal operation takes place. 0 - Disable 1 - Enable
30 MAEN2	Match Address Mode Enable 2 Enables automatic address matching or data matching mode for MATCH[MA2]. If this field is 0, normal operation takes place. 0 - Disable 1 - Enable
29 M10	10-Bit Mode Select Causes the tenth bit to be a part of the serial transmission. You must change the value of this field only when both the transmitter and receiver are disabled. 0 - Receiver and transmitter use 7-bit to 9-bit data characters 1 - Receiver and transmitter use 10-bit data characters
28-24 OSR	Oversampling Ratio Configures the OSR of the receiver. You must change the value of this field only when both the transmitter and receiver are disabled. **NOTE** BAUD[OSR] results in an OSR of BAUD[OSR] + 1, for example, BAUD[OSR] = 0_0101b results in a final division by 6. 0_0000 - Results in an OSR of 16 0_0001 - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0_0010 - Reserved 0_0011 - Results in an OSR of 4 (requires BAUD[BOTHEDGE] to be 1) 0_0100 - Results in an OSR of 5 (requires BAUD[BOTHEDGE] to be 1) 0_0101 - Results in an OSR of 6 (requires BAUD[BOTHEDGE] to be 1) 0_0110 - Results in an OSR of 7 (requires BAUD[BOTHEDGE] to be 1) 0_0111 - Results in an OSR of 8 0_1000 - Results in an OSR of 9 0_1001 - Results in an OSR of 10 0_1010 - Results in an OSR of 11 0_1011 - Results in an OSR of 12 0_1100 - Results in an OSR of 13 0_1101 - Results in an OSR of 14 0_1110 - Results in an OSR of 15 0_1111 - Results in an OSR of 16 1_0000 - Results in an OSR of 17 1_0001 - Results in an OSR of 18 1_0010 - Results in an OSR of 19 1_0011 - Results in an OSR of 20 1_0100 - Results in an OSR of 21 1_0101 - Results in an OSR of 22 1_0110 - Results in an OSR of 23 1_0111 - Results in an OSR of 24 1_1000 - Results in an OSR of 25 1_1001 - Results in an OSR of 26 1_1010 - Results in an OSR of 27 1_1011 - Results in an OSR of 28 1_1100 - Results in an OSR of 29 1_1101 - Results in an OSR of 30 1_1110 - Results in an OSR of 31 1_1111 - Results in an OSR of 32
23 TDMAE	Transmitter DMA Enable Enables STAT[TDRE] to generate a DMA request. 0 - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable
22 —	Reserved
21 RDMAE	Receiver Full DMA Enable Enables STAT[RDRF] to generate a DMA request. 0 - Disable 1 - Enable
20 RIDMAE	Receiver Idle DMA Enable Enables STAT[IDLE] to generate a DMA request. If this field is 1, reading Data (DATA) when either DATA[RXEMPT] or DATA[IDLINE] is 1 generates an end-of-packet response until the completion of the DMA minor loop. During an end-of-packet response, reading Data (DATA) returns 0000_33FFh and does not pull data from the receive FIFO. STAT[IDLE] becomes 0 on completion of the minor loop, provided an end-of-packet response is generated and either the receive FIFO is empty or the receiver is active. 0 - Disable 1 - Enable
19-18 MATCFG	Match Configuration Configures the match addressing mode used. You must change the value of this field only when both the transmitter and receiver are disabled. 00 - Address match wake-up 01 - Idle match wake-up 10 - Match on and match off 11 - Enables RWU on data match and match on or off for the transmitter CTS input
17 BOTHEDGE	Both Edge Sampling Enables sampling of the received data on both edges of the baud rate clock, effectively doubling the number of times the receiver samples the input data for a given OSR. This field must be 1 for OSRs between x4 and x7 and is optional for higher OSRs. You must change the value of this field only when the receiver is disabled. If this field is 0, the receiver samples input data using the rising edge of the baud rate clock. If this field is 1, the receiver samples input data using the rising and falling edges of the baud rate clock. 0 - Rising edge 1 - Both rising and falling edges
16 RESYNCDIS	Resynchronization Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Disables resynchronization of the received data word when a data one followed by data zero transition is detected. You must change the value of this field only when the receiver is disabled. 0 - Enable 1 - Disable
15 LBKDIE	LIN Break Detect Interrupt Enable Enables STAT[LBKDIF] to generate hardware interrupt requests. If this field is 0, hardware interrupts from STAT[LBKDIF] (uses polling) are disabled. If this field is 1, hardware interrupts are requested when STAT[LBKDIF] is 1. 0 - Disable 1 - Enable
14 RXEDGIE	RX Input Active Edge Interrupt Enable Enables STAT[RXEDGIF] to generate interrupt requests. If this field is 0, hardware interrupts from STAT[RXEDGIF] are disabled. If this field is 1, hardware interrupts are requested when STAT[RXEDGIF] is 1. Changing the value of CTRL[LOOPS] or CTRL[RSRC] when this field (RXEDGIE) is 1 can cause STAT[RXEDGIF] to become 1. 0 - Disable 1 - Enable
13 SBNS	Stop Bit Number Select Determines whether data characters include one or two stop bits. You must change the value of this field only when both the transmitter and receiver are disabled. 0 - One stop bit 1 - Two stop bits
12-0 SBR	Baud Rate Modulo Divisor Sets the modulo divide rate for the baud rate generator. • If SBR is 0, baud rate generator is disabled. • If SBR is 1–8191, baud rate = baud clock ÷ ((OSR + 1) × SBR). You must update the 13-bit baud rate setting [SBR12:SBR0] only when both the transmitter and receiver are disabled (both CTRL[RE] and CTRL[TE] are 0).

62.6.7 Status (STAT)

Provides the module status.

Offset

Register	Offset
STAT	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W	W1C	W1C										W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MA1F	MA2F	0				TSF	MSF	0						AME	LBKFE
W	W1C	W1C														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 LBKDIF	LIN Break Detect Interrupt Flag Indicates whether a LIN break character is detected. This field becomes 1 when the LIN break detect circuitry is enabled and a LIN break character is detected. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not detected 1 - Detected When writing 0 - No effect 1 - Clear the flag
30 RXEDGIF	RXD Pin Active Edge Interrupt Flag Indicates whether an active edge on the receive pin has occurred. This field becomes 1 whenever the receiver is enabled and an active edge (falling if STAT[RXINV] is 0; rising if STAT[RXINV] is 1) on the RXD pin occurs. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Not occurred 1 - Occurred When writing 0 - No effect 1 - Clear the flag
29 MSBF	MSB First Specifies the first bit that is transmitted and received after the start bit. If this field is 0, LSB (bit 0) is the first bit transmitted after the start bit and the first bit received after the start bit is identified as bit 0 of the data entity. If this field is 1, MSB (identified as bit 9, bit 8, bit 7, or bit 6) is the first bit that is transmitted, after the start bit, depending on the settings of CTRL[M], CTRL[PE], and BAUD[M10]. Writing 1 to this field reverses the order of the bits that are transmitted and received on the wire. This field does not affect the polarity of the bits, the location of the parity bit, or the location of the start or stop bits. You must change the value of this field only when both the transmitter and receiver are disabled. 0 - LSB 1 - MSB
28 RXINV	Receive Data Inversion Specifies whether receive data is inverted. Writing 1 to this field reverses the polarity of the received data input before storing into FIFO. You must change the value of this field only when the receiver is disabled. **NOTE** Writing 1 to this field inverts the RXD input for all cases: data bits, start and stop bits, break, and idle. 0 - Not inverted 1 - Inverted
27 RWUID	Receive Wake Up Idle Detect Controls, for CTRL[RWU] on idle character detection, whether the idle character that wakes up the receiver writes 1 to STAT[IDLE]. For address match wake-up, this field controls whether STAT[IDLE] = 1 when the address does not match. You must change the value of this field only when the receiver is disabled. If this field is 0, during the Receive Standby state (CTRL[RWU] = 1), STAT[IDLE] does not become 1 upon detection of an idle character. During address match wake-up, STAT[IDLE] does not become 1 when an address does not match. If this field is 1, during the Receive Standby state (CTRL[RWU] = 1), STAT[IDLE] becomes 1 upon detection of an idle character. During address match wake-up, STAT[IDLE] becomes 1 when an address does not match.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - STAT[IDLE] does not become 1 1 - STAT[IDLE] becomes 1
26 BRK13	Break Character Generation Length Selects the longer transmitted break character length. The state of this field does not affect the detection of a framing error. You must change the value of this field only when the transmitter is disabled. You can send a break character by writing 1 to CTRL[SBK], or by writing the transmit FIFO when DATA[FRETSC] is 1 and DATA[R9T9] is 0. 0 - 9 to 13 bit times 1 - 12 to 15 bit times
25 LBKDE	LIN Break Detection Enable Enables LIN break detection. If this field is 0, LIN break detect is disabled, and only a normal break character can be detected. If this field is 1, LIN break detect is enabled and the LIN break character is detected at a length of 11 bit times (if CTRL[M] is 0), 12 bit times (if CTRL[M] is 1), or 13 bit times (if BAUD[M10] is 1). This field selects a longer break character detection length. When the field is 1, receive data is not stored in the receive FIFO. **NOTE** This field enables the LIN break detect circuit and disables writing receive data to FIFO. Therefore, it ignores all characters except a LIN break. 0 - Disable 1 - Enable
24 RAF	Receiver Active Flag Indicates whether the LPUART receiver is idle or active. This field becomes 1 when the receiver detects the beginning of a valid start bit, and the field becomes 0 automatically when the receiver detects an idle line. 0 - Idle, waiting for a start bit 1 - Receiver active (RXD pin input not idle)
23 TDRE	Transmit Data Register Empty Flag Indicates whether the transmit FIFO level is greater than, equal to, or less than the watermark. After the transmit FIFO is enabled, this field becomes 1 when the number of datawords in the transmit FIFO is equal to, or less than the number that WATER[TXWATER] indicates. To make the value of this field 0, write to it until the number of words in the transmit FIFO is greater than the number that WATER[TXWATER] indicates. After the transmit FIFO is disabled, this field becomes 1 to indicate that the FIFO level is less than the watermark. To make the value of this field 0 again, write to Data (DATA). This register is not affected by a character that is in the process of being transmitted; it is updated at the start of each transmitted character.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Greater than watermark 1 - Equal to or less than watermark
22 TC	Transmission Complete Flag Indicates whether the transmitter is active. This field becomes 0 when a transmission is in progress or a preamble or break character is loaded; in other words, when the transmitter is active (sending data, a preamble, or a break). The field becomes 1 when the transmit buffer is empty and no data, preamble, or break character is being transmitted; in other words, when the transmission activity is complete. When this happens, the transmit data output signal becomes idle (logic 1). This field becomes 0 after you write to Data (DATA) to transmit new data, queuing a preamble by first writing 0 and then writing 1 to CTRL[TE], queuing a break character by writing 1 to CTRL[SBK]. 0 - Transmitter active 1 - Transmitter idle
21 RDRF	Receive Data Register Full Flag Indicates whether the receive FIFO level is less than, equal to, or greater than the watermark. This field becomes 1 when the number of datawords in the receive buffer is greater than the number that WATER[RXWATER] indicates and the receive FIFO is enabled. To write 0 to this field, read Data (DATA) until the number of datawords in the receive FIFO is equal to, or less than the number that WATER[RXWATER] indicates. When the receive FIFO is disabled, this field (RDRF) becomes 1 if the receive buffer (Data (DATA)) is full. To make this field 0, read Data (DATA). A character that is in the process of being received does not cause a change in this field until the entire character is received. Even if this field is 1, the character continues to be received until an overrun condition occurs after the entire character is received. 0 - Equal to or less than watermark 1 - Greater than watermark
20 IDLE	Idle Line Flag Indicates whether an idle line is detected. This field becomes 1 when the LPUART receive line becomes idle for a number of full character time defined by CTRL[IDLECFG] field after a period of activity. There are two conditions for idle detection: 1. When CTRL[ILT] is 0, the receiver starts counting idle bit times after the start bit. If the receive character is all 1s, these bit times and the stop bit time count towards the full character time of logic high, 10 to 13 bit times, needed for the receiver to detect an idle line. 2. When CTRL[ILT] is 1, the receiver does not start counting idle bit times until after the stop bits. **NOTE** The stop bits and any logic-high periods following the previous character do not count toward the required continuous logic-high duration that the receiver uses to detect an idle line.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	To clear this field (set it to 0), write a value of 1 to it. Once the field is cleared (set to 0), you cannot write 1 to it again until either: 1. A new character is stored in the receive buffer 2. A LIN break character sets STAT[LBKDIF] to 1 This field is set to 1 only once, even if the receive line remains idle for a long time. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Idle line detected 1 - Idle line not detected When writing 0 - No effect 1 - Clear the flag
19 OR	Receiver Overrun Flag Indicates whether there is receive overrun. This field becomes 1 when you cannot prevent STAT[RDRF] from overflowing with data. The field becomes 1 immediately after the stop bit is completely received for the dataword that overflows the buffer and all the other error fields (STAT[FE], STAT[NF], and STAT[PF]) are prevented from becoming 1. The data in the shift register is lost, but the data already in the LPUART data registers is not affected. If STAT[LBKDE] is enabled and a LIN break is detected, this field becomes 1 if STAT[LBKDIF] is not 0 before the next data character is received. When this field is 1, no additional data is stored in the receive FIFO even if sufficient room exists. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No overrun 1 - Receive overrun (new LPUART data is lost) When writing 0 - No effect 1 - Clear the flag
18 NF	Noise Flag Indicates whether noise is detected in the received character of Data (DATA). The advanced sampling technique used in the receiver takes three samples in each of the received bits. If some of these samples disagree with the rest of the samples within any bit time in the frame, then noise is

Table continues on the next page...

Table continued from the previous page...

Field	Description
	detected for that character. This field becomes 1 whenever the next character to be read from Data (DATA) is received with noise detected within the character. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No noise detected 1 - Noise detected When writing 0 - No effect 1 - Clear the flag
17 FE	Framing Error Flag Indicates whether a framing error is detected. This field becomes 1 whenever the next character to be read from Data (DATA) is received with logic 0 detected where a stop bit was expected. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No framing error detected (this does not guarantee that the framing is correct) 1 - Framing error detected When writing 0 - No effect 1 - Clear the flag
16 PF	Parity Error Flag Indicates whether a parity error is detected. This field becomes 1 whenever the next character to be read from Data (DATA) is received when parity is enabled (CTRL[PE] is 1) and the parity bit in the received character does not agree with the expected parity value. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No parity error detected 1 - Parity error detected When writing 0 - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Clear the flag
15 MA1F	Match 1 Flag Indicates whether the received data is equal to MATCH[MA1]. This field becomes 1 whenever the next character to be read from Data (DATA) matches the value of MATCH[MA1]. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not equal to MA1 1 - Equal to MA1 When writing 0 - No effect 1 - Clear the flag
14 MA2F	Match 2 Flag Indicates whether the received data is equal to MATCH[MA2]. This field becomes 1 whenever the next character to be read from Data (DATA) matches the value of MATCH[MA2]. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not equal to MA2 1 - Equal to MA2 When writing 0 - No effect 1 - Clear the flag
13-10 —	Reserved
9 TSF	Timeout Status Flag Indicates whether a field in Timeout Status (TOSR) is 1 and configured to generate an interrupt. 0 - Field is 0 1 - Field is 1
8	MODEM Status Flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
MSF	Indicates whether a field in MODEM Status (MSR) is 1 and configured to generate an interrupt. 0 - Field is 0 1 - Field is 1
7-2 —	Reserved
1 AME	Address Mark Enable Configures the location of the address mark when configured for MSB first transfers. This field has no effect when configured for LSB first and you must change the value of this field only when both the transmitter and receiver are disabled. If this field is 0, address mark in character is MSB. If this field is 1, the address mark is stored in Data (DATA) at MSB (or MSB-1 when the parity bit is enabled). In other words, the address mark in character is the last bit before the stop bit (or parity bit when enabled). 0 - Disable 1 - Enable
0 LBKFE	LIN Break Flag Enable Enables the LIN break flag to assert whenever a LIN break character is detected. Unlike STAT[LBKDE], this does not impact data being stored in the receive data buffer, but does cause STAT[LBKDIF] to become 1 whenever a LIN break is detected. Because a LIN break is longer than a normal character, the LIN break triggers a write to STAT[RDRF] with the data fields as 0 and STAT[FE] as 1. The character following the LIN break has DATA[LINBRK] as 1 to indicate that the previous character was a LIN break. You must change the value of this field only when both the transmitter and receiver are disabled. If this field is 1, the LIN break character is detected at a length of 11-bit times (if CTRL[M] is 0), 12 (if CTRL[M] is 1), or 13 (if BAUD[M10] is 1). 0 - Disable 1 - Enable

62.6.8 Control (CTRL)

Controls various optional features of the LPUART system.

You must write to the fields of this register only when both the transmitter and receiver are disabled.

Offset

Register	Offset
CTRL	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	R8T9	R9T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MA1IE	MA2IE	0	0	M7	IDLECFG			LOOPS	DOZEEN	RSRC	M	WAKE	ILT	PE	PT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 R8T9	Receive Bit 8 Transmit Bit 9 Contains R8 and T9 that correspond to different functions. R8 is the ninth data bit received after you configure LPUART for 9-bit or 10-bit data formats. When reading 9-bit or 10-bit data, read R8 before reading Data (DATA). T9 is the tenth data bit transmitted after you configure LPUART for 10-bit data formats. When writing 10-bit data, write T9 before writing to Data (DATA). If T9 does not need to change from its previous value, such as when it is used to generate address mark or parity, then you need not write to it each time you write to Data (DATA). **NOTE** R8 is a read-only bit and T9 is a write-only bit; the value read is different from the value written.
30 R9T8	Receive Bit 9 Transmit Bit 8 Contains R9 and T8 that correspond to different functions. R9 is the tenth data bit received after you configure LPUART for 10-bit data formats. When reading 10-bit data, read R9 before reading Data (DATA). T8 is the ninth data bit transmitted after you configure LPUART for 9-bit or 10-bit data formats. When writing 9-bit or 10-bit data, write T8 before writing to Data (DATA). If T8 does not need to change from its previous value, such as when it is used to generate address mark or parity, then you need not write to it each time you write to Data (DATA). **NOTE** R9 is a read-only field and T8 is a write-only field; the value read is different from the value written.
29 TXDIR	TXD Pin Direction in Single-Wire Mode Determines the direction of data at the TXD pin, in Single-Wire mode, when LPUART is configured for a single-wire half-duplex operation (CTRL[LOOPS] and CTRL[RSRC] are 1). When writing 0 to this field, the transmitter finishes transmitting the current character (if any) before the receiver starts receiving data from the TXD pin.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Input 1 - Output
28 TXINV	Transmit Data Inversion Specifies whether transmit data is inverted. Writing 1 to this field reverses the polarity of the transmitted data output. This action inverts the TXD output for all cases: data bits, start and stop bits, break, and idle. 0 - Not inverted 1 - Inverted
27 ORIE	Overrun Interrupt Enable Enables STAT[OR] to generate hardware interrupt requests. When this field is 1, a hardware interrupt is requested. Use polling when OR interrupts are disabled. 0 - Disable 1 - Enable
26 NEIE	Noise Error Interrupt Enable Enables STAT[NF] to generate hardware interrupt requests. When this field is 1, a hardware interrupt is requested. Use polling when NF interrupts are disabled. 0 - Disable 1 - Enable
25 FEIE	Framing Error Interrupt Enable Enables STAT[FE] to generate hardware interrupt requests. When this field is 1, a hardware interrupt is requested. Use polling when FE interrupts are disabled. 0 - Disable 1 - Enable
24 PEIE	Parity Error Interrupt Enable Enables STAT[PF] to generate hardware interrupt requests. When this field is 1, a hardware interrupt is requested. Use polling when PF interrupts are disabled. 0 - Disable 1 - Enable
23 TIE	Transmit Interrupt Enable Enables STAT[TDRE] to generate interrupt requests if STAT[TDRE] is 1. 0 - Disable 1 - Enable
22	Transmission Complete Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
TCIE	Enables STAT[TC] to generate interrupt requests if STAT[TC] is 1. 0 - Disable 1 - Enable
21 RIE	Receiver Interrupt Enable Enables STAT[RDRF] to generate hardware interrupt requests if STAT[RDRF] is 1. 0 - Disable 1 - Enable
20 ILIE	Idle Line Interrupt Enable Enables hardware interrupts. This field enables STAT[IDLE] to generate interrupt requests. If this field is 0, hardware interrupts from STAT[IDLE] are disabled and polling is used, and if this field is 1, hardware interrupts are enabled when STAT[IDLE] is 1. 0 - Disable 1 - Enable
19 TE	Transmitter Enable Enables the LPUART transmitter. Using this field, you can also queue an idle preamble by first writing 0 and then writing 1 to this field. After this field becomes 0, the field reads 1 until the transmitter has completed the current character and the TXD pin is tristated. You can also queue a single idle character by writing to the transmit FIFO with DATA[FRETSC] and DATA[R9T9] = 1. 0 - Disable 1 - Enable
18 RE	Receiver Enable Enables the LPUART receiver. After you write 0 to this field, this field remains 1 until the receiver finishes receiving the current character (if any). 0 - Disable 1 - Enable
17 RWU	Receiver Wake-Up Control Specifies whether the LPUART receiver in standby is waiting for a wake-up condition. You can write 1 to this field to place the LPUART receiver in a Standby state. The field becomes 0 automatically when an RWU event occurs, that is, in case of an idle event when CTRL[WAKE] is 0 or an address match when CTRL[WAKE] is 1 and STAT[RWUID] is 0.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE You must write 1 to this field only when CTRL[WAKE] is 0 (wake-up on idle), if the channel is currently not idle. You can determine this by the value of STAT[RAF]. If the field is 1 to wake up an idle event and the channel is already idle, LPUART, possibly, discards the data. This is because the data must be received or a LIN break is detected after an Idle condition is detected before the IDLE flag is allowed to be reasserted. 0 - Normal receiver operation 1 - LPUART receiver in standby, waiting for a wake-up condition
16 SBK	Send Break Specifies whether queue break character(s) are to be sent. Writing 1 and then 0 to this field queues a break character in the transmit data stream. Additional break characters of 9 to 13 bits, or 12 to 15 bits if STAT[BRK13] is 1, and bit times of logic 0 are queued as long as this field is 1. Depending on the timing when this field is 1 and 0, relative to the character currently being transmitted, a second break character may be queued before you write 0 to this field. If the time taken to write 0 to this field is too long, for example, if the field does not become 0 by the end of the first break character, a second break character is sent. This is compared to queuing a break character through the transmit FIFO that guarantees only one break character is sent. You can also queue a single break character by writing to the transmit FIFO when DATA[FRETSC] is 1 and DATA[R9T9] is 0. 0 - Normal transmitter operation 1 - Queue break character(s) to be sent
15 MA1IE	Match 1 (MA1F) Interrupt Enable Enables the MA1F interrupt. 0 - Disable 1 - Enable
14 MA2IE	Match 2 (MA2F) Interrupt Enable Enables the MA2F interrupt. 0 - Disable 1 - Enable
13 —	Reserved
12 —	Reserved
11 M7	7-Bit Mode Select Specifies the data characters that the receiver and transmitter use.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You must change the value of this field only after both the transmitter and receiver are disabled. 0 - 8-bit to 10-bit 1 - 7-bit
10-8 IDLECFG	Idle Configuration Configures the number of continuous idle characters that must be received before an idle line condition is detected and STAT[IDLE] becomes 1, STAT[RAF] becomes 0, and DATA[IDLINE] becomes 1. 000 - 1 001 - 2 010 - 4 011 - 8 100 - 16 101 - 32 110 - 64 111 - 128
7 LOOPS	Loop Mode Select Selects Loop mode. After this field becomes 1, the RXD pin is disconnected from LPUART and the transmitter output is internally connected to the receiver input. The transmitter and receiver must be enabled to use the loop function. In Loop mode or Single-Wire mode, the transmitter outputs are internally connected to the receiver input (see CTRL[RSRC]). 0 - Normal operation: RXD and TXD use separate pins 1 - Loop mode or Single-Wire mode
6 DOZEEN	Doze Mode Enables LPUART in Doze mode. If this field is 1, LPUART remains active when not in Doze mode. 0 - Enable 1 - Disable
5 RSRC	Receiver Source Select Determines the source of the receiver shift register input if CTRL[LOOPS] is 1. This field has no effect unless CTRL[LOOPS] is 1. If this field is 0, internal Loopback mode is selected. LPUART does not use the RXD pin. Additionally, the CTS_B pin is not used and internally driven by the RTS_B output. If this field is 1, single-wire LPUART mode is selected where the TXD pin is connected to the transmitter output and receiver input.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Internal Loopback mode 1 - Single-wire mode
4 M	9-Bit Or 8-Bit Mode Select Specifies the data characters that the receiver and transmitter use. 0 - 8-bit 1 - 9-bit
3 WAKE	Receiver Wake-Up Method Select Determines which condition wakes up LPUART when CTRL[RWU] = 1 and BAUD[MATCFG] = 0 (this field must be 1 when BAUD[MATCFG] = 11): • Address mark in the bit preceding the stop bit (or bit preceding the parity bit when parity is enabled) of the received data character • An idle condition on the receive pin input signal If this field is 0, CTRL[RWU] is configured for idle line wake-up, and if this field is 1, CTRL[RWU] is configured with address mark wake-up. 0 - Idle 1 - Mark
2 ILT	Idle Line Type Select Determines when the receiver starts counting logic 1s as idle character bits. The count begins either after a valid start bit or the stop bit. If the count begins after the start bit, a string of logic 1s preceding the stop bit can cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. **NOTE** In case you write 1 to this field, a logic 0 is automatically shifted after a received stop bit, therefore resetting the idle count. 0 - After the start bit 1 - After the stop bit
1 PE	Parity Enable Enables hardware parity generation and checking. If parity is enabled, the bit immediately before the stop bit is treated as the parity bit. 0 - Disable 1 - Enable
0 PT	Parity Type Selects the type of parity, even or odd, if parity is enabled (CTRL[PE] = 1):

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Odd parity means that the total number of logic 1 bits in the data character, including the parity bit, is odd. • Even parity means that the total number of 1s in the data character, including the parity bit, is even. 0 - Even parity 1 - Odd parity

62.6.9 Data (DATA)

Supports 8-bit, 16-bit, or 32-bit writes, each type of write performing a separate function. An 8-bit write to DATA[7:0] pushes {CTRL[R8T9], CTRL[R9T8], DATA[7:0]} the transmit FIFO with TSC clear. A 16-bit or 32-bit write pushes the data written into the FIFO and does not update the value of CTRL[R8T9] or CTRL[R9T8].

Reads and writes of this register are also involved in the automatic flag clearing mechanisms for some of the LPUART status fields.

NOTE

Reads return the contents of the read-only receive FIFO and writes go to the write-only transmit FIFO, making this register work as a set of two separate registers.

Offset

Register	Offset
DATA	1Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NOISY	PARITYE	FRETSC	RXEMPT	IDLINE	LINBRK	R9T9	R8T8	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
15 NOISY	Noisy Data Received Indicates whether the current received dataword contained in DATA[R9:R0] is received with noise. 0 - Received without noise 1 - Received with noise
14 PARITYE	Parity Error Indicates whether the current received dataword contained in DATA[R9:R0] is received with a parity error. 0 - Received without a parity error 1 - Received with a parity error
13 FRETSC	Frame Error Transmit Special Character Specifies the way the dataword is received. For reads, this field indicates that the current received dataword contained in DATA[R9:R0] is received with a frame error. For writes, the field indicates that a break or idle character is to be transmitted instead of the contents in DATA[T9:T0]. T9 indicates a break character when it is 0 and indicates an idle character when it is 1. The contents of DATA[T8:T0] must be 0. 0 - Received without a frame error on reads or transmits a normal character on writes 1 - Received with a frame error on reads or transmits an idle or break character on writes
12 RXEMPT	Receive Buffer Empty Indicates whether the receive buffer contains valid data. This field becomes 1 when there is no data in the receive buffer. The field does not consider data in the receive shift register. 0 - Valid data 1 - Invalid data and empty
11 IDLINE	Idle Line Indicates whether the receiver line was idle before receiving the character in DATA[9:0]. It can be read as “1” on the first character when the receiver is first enabled. The difference between this field and STAT[IDLE] is that, STAT[IDLE] flag does not set on an idle line after the receiver is first enabled, it needs to receive a character before it can become set, whereas this field does not have this limitation and can be set on the first character received if an idle line is detected beforehand. 0 - Not idle 1 - Idle
10 LINBRK	LIN Break Indicates whether the receiver line detected a LIN break before receiving the character in DATA[9:0]. This field requires the value of STAT[LBKDIF] to be 1. If this field is 0, the LIN break detect circuitry is disabled.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Not detected 1 - Detected
9 R9T9	Read receive FIFO bit 9 or write transmit FIFO bit 9
8 R8T8	Read receive FIFO bit 8 or write transmit FIFO bit 8
7 R7T7	Read receive FIFO bit 7 or write transmit FIFO bit 7
6 R6T6	Read receive FIFO bit 6 or write transmit FIFO bit 6
5 R5T5	Read receive FIFO bit 5 or write transmit FIFO bit 5
4 R4T4	Read receive FIFO bit 4 or write transmit FIFO bit 4
3 R3T3	Read receive FIFO bit 3 or write transmit FIFO bit 3
2 R2T2	Read receive FIFO bit 2 or write transmit FIFO bit 2
1 R1T1	Read receive FIFO bit 1 or write transmit FIFO bit 1
0 R0T0	Read receive FIFO bit 0 or write transmit FIFO bit 0

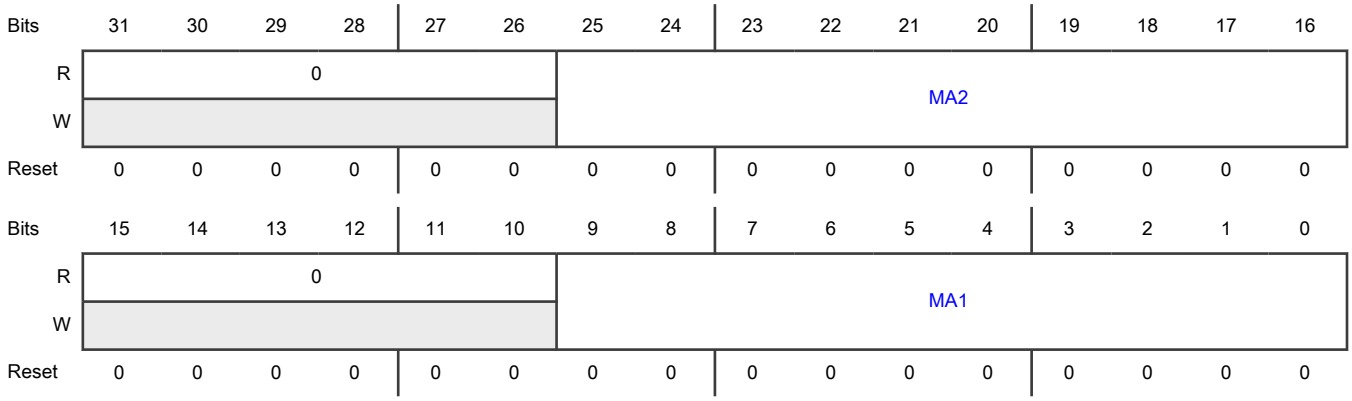
62.6.10 Match Address (MATCH)

Provides addresses for address matching during the receiver operation.

Offset

Register	Offset
MATCH	20h

Diagram



Fields

Field	Description
31-26 —	Reserved
25-16 MA2	Match Address 2 Is compared to input data addresses when the most significant bit is 1 and the associated Baud Rate (BAUD) field is 1. If a match occurs, the data that follows is transferred to Data (DATA). If a match fails, the data that follows is discarded. You must write to MATCH[MA1] and MATCH[MA2] only when the associated Baud Rate (BAUD) field is 0.
15-10 —	Reserved
9-0 MA1	Match Address 1 Is compared to input data addresses when the most significant bit is 1 and the associated Baud Rate (BAUD) field is 1. If a match occurs, the data that follows is transferred to Data (DATA). If a match fails, the data that follows is discarded. You must write to MATCH[MA1] and MATCH[MA2] fields only when the associated field in Baud Rate (BAUD) is 0.

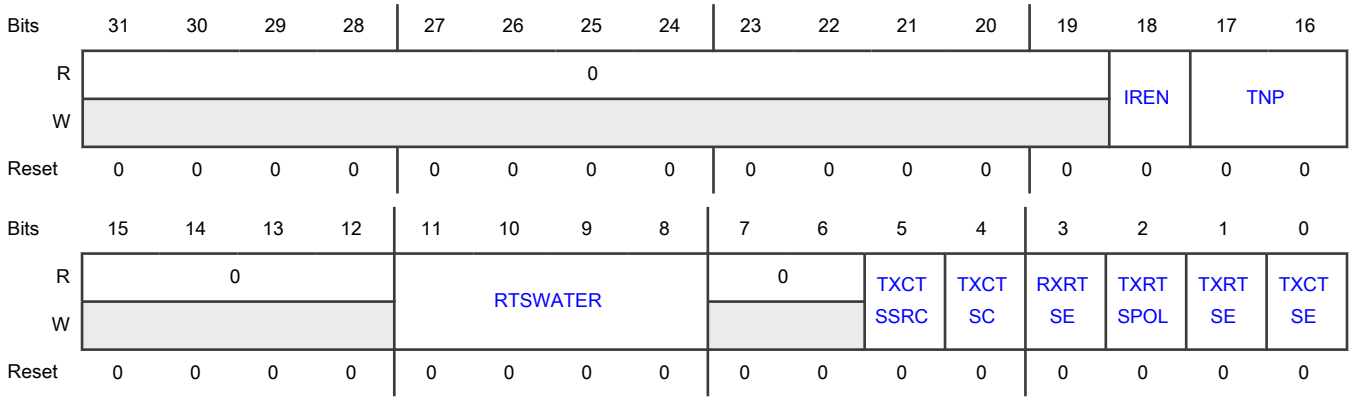
62.6.11 MODEM IrDA (MODIR)

Controls options for setting the MODEM configuration.

Offset

Register	Offset
MODIR	24h

Diagram



Fields

Field	Description
31-19 —	Reserved
18 IREN	IR Enable Enables IR modulation and demodulation. You must change the value of this field only when both the transmitter and receiver are disabled. 0 - Disable 1 - Enable
17-16 TNP	Transmitter Narrow Pulse Specifies whether LPUART transmits a 1 ÷ OSR, 2 ÷ OSR, 3 ÷ OSR, or 4 ÷ OSR narrow pulse when the IR pulse is enabled. You must change the value of this field only when both the transmitter and receiver are disabled. The IR pulse width must be configured to less than half of the OSR. Common pulse widths are 3 ÷ 16, 1 ÷ 16, 1 ÷ 32, or 1 ÷ 4 of the bit length. You can configure these by selecting the appropriate OSR and pulse width. 00 - 1 ÷ OSR 01 - 2 ÷ OSR 10 - 3 ÷ OSR 11 - 4 ÷ OSR
15-12 —	Reserved
11-8 RTSWATER	Receive RTS Configuration Configures the assertion and negation of the receiver's RTS_B output. The receiver's RTS_B output negates when the number of empty words in the receive FIFO is greater or equal to the value of this field. If this field is 0, the RTS_B pin negates when the receive FIFO is full. For the

Table continues on the next page...

Table continued from the previous page...

Field	Description
	purpose of receive RTS_B generation, the number of words in the receive FIFO updates when a start bit is detected. This supports additional latency between RTS_B negation and the external transmitter ceasing transmission. If both receive RTS_B and address or data matching is enabled, RTS_B could assert at the end of a character if there exists no match. You must change the value of this field only when the receiver is disabled.
7-6 —	Reserved
5 TXCTSSRC	Transmit CTS Source Configures the source of the CTS input. 0 - The CTS_B pin 1 - An internal connection to the receiver address match result
4 TXCTSC	Transmit CTS Configuration Configures whether the CTS state or input is checked or sampled at the start of each character or only when the transmitter is idle. 0 - Sampled at the start of each character 1 - Sampled when the transmitter is idle
3 RXRTSE	Receiver RTS Enable Allows the RTS output to control the CTS input of the transmitting device to prevent receiver overrun. You must change the value of this field only when the receiver is disabled. If this field is 0, the receiver has no effect on RTS. If this field is 1, RTS is deasserted if STAT[RDRF] is 1 or a start bit is detected that causes STAT[RDRF] to become 1. RTS is asserted if STAT[RDRF] is 0 and has not detected a start bit that causes STAT[RDRF] to become 1. **NOTE** Do not write 1 to both MODIR[RXRTSE] and MODIR[TXRTSE]. 0 - Disable 1 - Enable
2 TXRTSPOL	Transmitter RTS Polarity Controls the polarity of the transmitter RTS. This field does not affect the polarity of the receiver RTS that remains negated in the active-low state unless MODIR[TXRTSE] is 1. You must change the value of this field only when the transmitter is disabled. 0 - Active low 1 - Active high
1	Transmitter RTS Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
TXRTSE	Controls the operation of RTS before and after a transmission. You must change the value of this field only when the transmitter is disabled. If this field is 0, the transmitter has no effect on RTS, and if this field is 1, a character is placed into an empty transmit shift register. RTS asserts 1-bit time before the start bit is transmitted and deasserts 1-bit time after all characters in the transmitter FIFO and shift register are completely sent, including the last stop bit. 0 - Disable 1 - Enable
0 TXCTSE	Transmitter CTS Enable Enables the operation of the transmitter. You can write 1 to this field irrespective of the states of MODIR[TXRTSE] and MODIR[RXRTSE]. If this field is 1, the transmitter checks the state of the CTS signal each time it is ready to send a character. If CTS is asserted, the character is sent. If CTS is deasserted, the TXD signal remains in the mark state and transmission is delayed until CTS is asserted. Changes in CTS, when a character is being sent, do not affect its transmission. 0 - Disable 1 - Enable

62.6.12 FIFO (FIFO)

Provides you the ability to turn on and turn off the FIFO functionality.

This register also provides you the size of the FIFO that has been implemented. You can read this register at any time and must write to it only when CTRL[RE] and CTRL[TE] are 0 and the FIFO is empty.

Offset

Register	Offset
FIFO	28h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								TXEMPT	RXEMPT	0				TXOF	RXUF
W															W1C	W1C
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	RXIDEN			TXOFE	RXUFE	TXFE	TXFIFOSIZE			RXFE	RXFIFOSIZE		
W	TXFLUSH	RXFLUSH														
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1

Fields

Field	Description
31-24 —	Reserved
23 TXEMPT	Transmit FIFO Or Buffer Empty Indicates whether the transmit buffer is empty. This field becomes 1 when there is no data in the transmit FIFO or buffer. The field does not consider data in the transmit shift register. 0 - Not empty 1 - Empty
22 RXEMPT	Receive FIFO Or Buffer Empty Indicates whether the receive buffer is empty. This field becomes 1 when there is no data in the receive FIFO or buffer. The field does not consider data in the receive shift register. 0 - Not empty 1 - Empty
21-18 —	Reserved
17 TXOF	Transmitter FIFO Overflow Flag Indicates whether more data has been written to the transmit FIFO than it can hold. If this field is 0, no transmit FIFO overflow has occurred since the last time the field was cleared, and if this field is 1, at least one transmit FIFO overflow has occurred since the last time the field was cleared. This field becomes 1 regardless of the value of FIFO[TXOFE]. However, an interrupt is issued to the host only if FIFO[TXOFE] is 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No overflow 1 - Overflow When writing 0 - No effect 1 - Clear the flag
16 RXUF	Receiver FIFO Underflow Flag Indicates whether more data has been read from the receive FIFO than was present.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If this field is 0, no receive FIFO underflow has occurred since the last time the field was cleared, and if this field is 1, at least one receive FIFO underflow has occurred since the last time the field was cleared. This field becomes 1 regardless of the value of FIFO[RXUFE]. However, an interrupt is issued to the host only if FIFO[RXUFE] is 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No underflow 1 - Underflow When writing 0 - No effect 1 - Clear the flag
15 TXFLUSH	Transmit FIFO Flush Causes all data that is stored in the transmit FIFO to be flushed. If you write 0 to this field, no flush operation occurs, and if you write 1 to this field, all data in the transmit FIFO or buffer clears out. This does not affect data in the transmit shift register. 0 - No effect 1 - All data flushed out
14 RXFLUSH	Receive FIFO Flush Causes all data that is stored in the receive FIFO to be flushed. If you write 0 to this field, no flush operation occurs, and if you write 1 to this field, all data in the receive FIFO or buffer clears out. This does not affect data in the receive shift register. 0 - No effect 1 - All data flushed out
13 —	Reserved
12-10 RXIDEN	Receiver Idle Empty Enable Enables STAT[RDRF] to become 1 when the receiver is idle for a number of idle characters and the FIFO is not empty. This feature is not supported when the receiver extended idle time is enabled. 000 - Disable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle 001 - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for one character

Table continues on the next page...

Table continued from the previous page...

Field	Description
	010 - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for two characters 011 - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for four characters 100 - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for eight characters 101 - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for 16 characters 110 - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for 32 characters 111 - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for 64 characters
9 TXOFE	Transmit FIFO Overflow Interrupt Enable Enables FIFO[TXOF] to generate an interrupt to the host. 0 - Disable 1 - Enable
8 RXUFE	Receive FIFO Underflow Interrupt Enable Enables FIFO[RXUF] to generate an interrupt to the host. 0 - Disable 1 - Enable
7 TXFE	Transmit FIFO Enable Enables the transmit FIFO. If this field is 0, the transmit buffer operates as a FIFO of depth equal to 1 dataword, regardless of the value in FIFO[TXFIFOSIZE]. Both CTRL[TE] and CTRL[RE] must be 0 before you change the value of this field. If this field is 1, the built-in FIFO structure for the transmit buffer is enabled. FIFO[TXFIFOSIZE] indicates the size of the FIFO structure. 0 - Disable 1 - Enable
6-4 TXFIFOSIZE	Transmit FIFO Buffer Depth Indicates the maximum number of transmit datawords (transmit FIFO buffer depth) that can be stored in the transmit buffer. 000 - 1 001 - 4 010 - 8 011 - 16

Table continues on the next page...

Table continued from the previous page...

Field	Description
	100 - 32 101 - 64 110 - 128 111 - 256
3 RXFE	Receive FIFO Enable Enables the receive FIFO. If this field is 0, the receive buffer operates as a FIFO of depth equal to 1 dataword, regardless of the value in FIFO[RXFIFOSIZE]. Both CTRL[RE] and CTRL[TE] must be 0 before you change the value of this field. If this field is 1, the built-in FIFO structure for the receive buffer is enabled. FIFO[RXFIFOSIZE] indicates the size of the FIFO structure. 0 - Disable 1 - Enable
2-0 RXFIFOSIZE	Receive FIFO Buffer Depth Indicates the maximum number of receive datawords (receive FIFO buffer depth) that can be stored in the receive buffer before an overrun occurs. 000 - 1 001 - 4 010 - 8 011 - 16 100 - 32 101 - 64 110 - 128 111 - 256

62.6.13 Watermark (WATER)

Provides the ability to set a programmable threshold for notification, or sets the programmable thresholds to indicate that transmit data can be written or receive data can be read.

You may read this register at any time but must write to it only when CTRL[TE] is 0.

Offset

Register	Offset
WATER	2Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				RXCOUNT				0				RXWATER			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				TXCOUNT				0				TXWATER			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28-24 RXCOUNT	Receive Counter Indicates the number of datawords in the receive FIFO or buffer. If a dataword is being received in the receive shift register, it is not included in the count. This value may be used in conjunction with FIFO[RXFIFOSIZE] to calculate the room left in the receive FIFO or buffer.
23-20 —	Reserved
19-16 RXWATER	Receive Watermark Generates an interrupt or a DMA request if the number of datawords in the receive FIFO or buffer is greater than the value of this field. For proper operation, the value of this field must be less than the size of the receive FIFO or buffer, as indicated by FIFO[RXFIFOSIZE] and FIFO[RXFE].
15-13 —	Reserved
12-8 TXCOUNT	Transmit Counter Indicates the number of datawords in the transmit FIFO or buffer. If a dataword is being transmitted to the transmit shift register, it is not included in the count. This value may be used in conjunction with the value of FIFO[TXFIFOSIZE] to calculate the room left in the transmit FIFO or buffer.
7-4 —	Reserved
3-0	Transmit Watermark

Table continues on the next page...

Table continued from the previous page...

Field	Description
TXWATER	Generates an interrupt or a DMA request when the number of datawords in the transmit FIFO or buffer is equal to or less than the value of this field. For proper operation, the value of this field must be less than the size of the transmit buffer or FIFO, as indicated by FIFO[TXFIFOSIZE] and FIFO[TXFE].

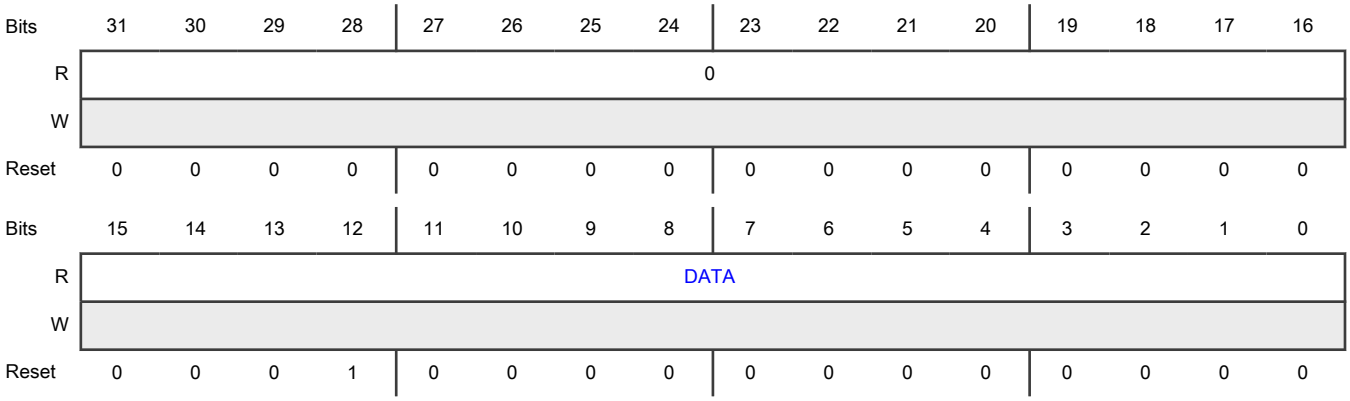
62.6.14 Data Read-Only (DATARO)

Indicates the first entry in the receive FIFO, but does not pull data from the FIFO.

Offset

Register	Offset
DATARO	30h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 DATA	Receive Data Indicates the first entry from the receive FIFO. This register has the same functionality as that of Data (DATA).

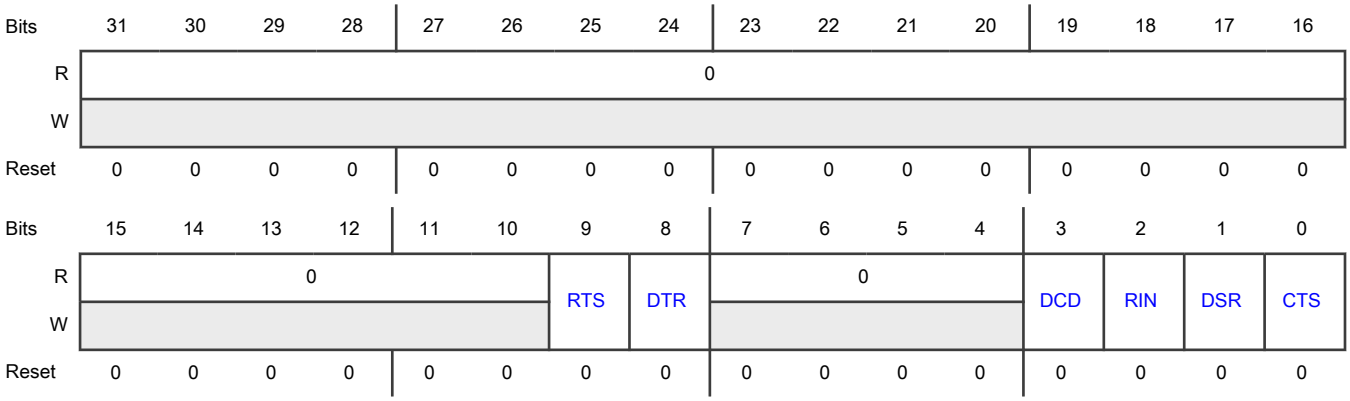
62.6.15 MODEM Control (MCR)

Controls the operation of the MODEM pins.

Offset

Register	Offset
MCR	40h

Diagram



Fields

Field	Description
31-10 —	Reserved
9 RTS	Request To Send Configures the default state of the RTS_B pin when the function is disabled. 0 - Logic one 1 - Logic zero
8 DTR	Data Terminal Ready Configures the default state of the DTR_B pin. 0 - Logic one 1 - Logic zero
7-4 —	Reserved
3 DCD	Data Carrier Detect Configures the interrupt, DCD_B, for change of state of the DCD_B pin. 0 - Disable interrupt 1 - Enable interrupt
2	Ring Indicator

Table continues on the next page...

Table continued from the previous page...

Field	Description
RIN	Configures the interrupt, RIN_B, for change of state on the RIN_B pin. 0 - Disable interrupt 1 - Enable interrupt
1 DSR	Data Set Ready Configures the interrupt, DSR_B, for change of state on the DSR_B pin. 0 - Disable interrupt 1 - Enable interrupt
0 CTS	Clear To Send Configures the interrupt, CTS_B, for change of state on the CTS_B pin. 0 - Disable interrupt 1 - Enable interrupt

62.6.16 MODEM Status (MSR)

Indicates the status of the MODEM pins.

NOTE

You must appropriately configure the PAD connecting to DCD_B, RIN_B, DSR_B, and CTS_B inputs to get appropriate reset values for the MSR register fields.

Offset

Register	Offset
MSR	44h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DCD	RIN	DSR	CTS	DDCD	DRI	DDSR	DCTS
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7 DCD	Data Carrier Detect Indicates the state of the DCD_B pin. 0 - Logic one 1 - Logic zero
6 RIN	Ring Indicator Indicates the state of the RIN_B pin. 0 - Logic one 1 - Logic zero
5 DSR	Data Set Ready Indicates the state of the DSR_B pin. 0 - Logic one 1 - Logic zero
4 CTS	Clear To Send Indicates the state of the CTS_B pin. 0 - Logic one 1 - Logic zero
3 DDCD	Delta Data Carrier Detect Indicates whether the DCD_B pin changed state since the last time this field was 0. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Did not change state 1 - Changed state When writing 0 - No effect 1 - Clear the flag
2 DRI	Delta Ring Indicator Indicates whether the RIN_B pin changed state since the last time this field was 0.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - Did not change state 1 - Changed state When writing 0 - No effect 1 - Clear the flag
1 DDSR	Delta Data Set Ready Indicates whether the DSR_B pin changed state since the last time this field was 0. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Did not change state 1 - Changed state When writing 0 - No effect 1 - Clear the flag
0 DCTS	Delta Clear To Send Indicates whether the CTS_B pin changed state since the last time this field was 0. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Did not change state 1 - Changed state When writing 0 - No effect 1 - Clear the flag

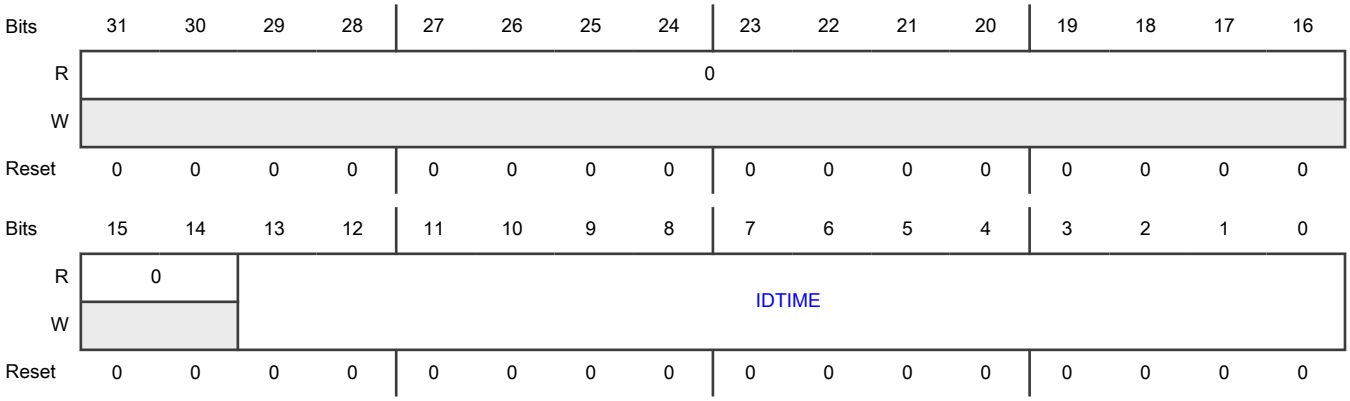
62.6.17 Receiver Extended Idle (REIR)

Configures the receiver extended idle functionality. You must not change this value when the receiver is enabled.

Offset

Register	Offset
REIR	48h

Diagram



Fields

Field	Description
31-14 —	Reserved
13-0 IDTIME	Idle Time Configures the idle length in number of bits (baud rate) since the end of the last stop bit. This affects the behavior of the idle wake-up, STAT[IDLE], DATA[IDLINE], and STAT[RAF]. The minimum supported extended idle time is equal to one idle character. The extended idle feature is disabled when this field is 0.

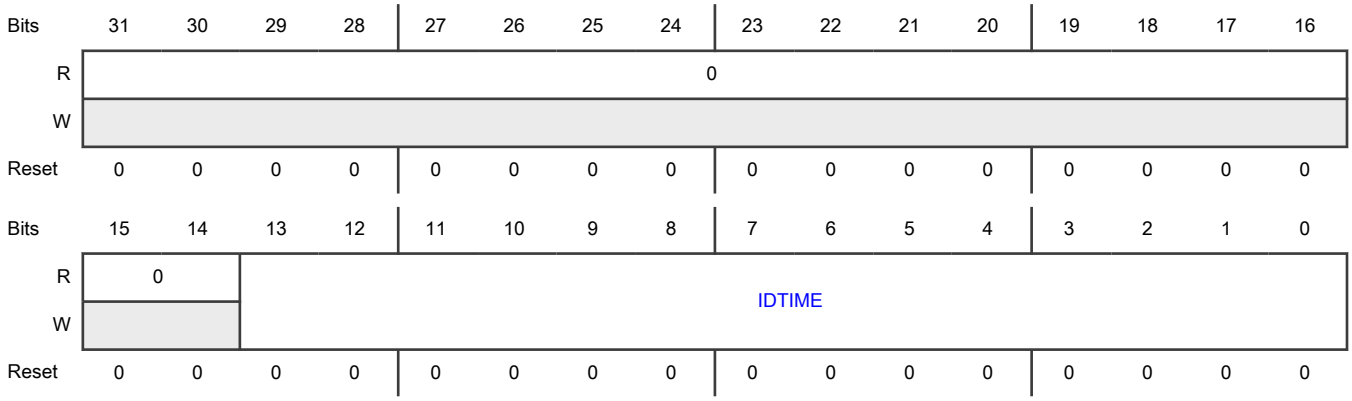
62.6.18 Transmitter Extended Idle (TEIR)

Configures the transmitter extended idle functionality. You must not change this value when the transmitter is enabled.

Offset

Register	Offset
TEIR	4Ch

Diagram



Fields

Field	Description
31-14 —	Reserved
13-0 IDTIME	Idle Time Configures the transmitter idle time in number of bits (baud rate) whenever an idle character is queued through the transmit FIFO. An idle character is not automatically queued whenever the transmitter is enabled. The minimum supported extended idle time equals one idle character. The extended idle feature is disabled when this field is 0.

62.6.19 Half Duplex Control (HDCR)

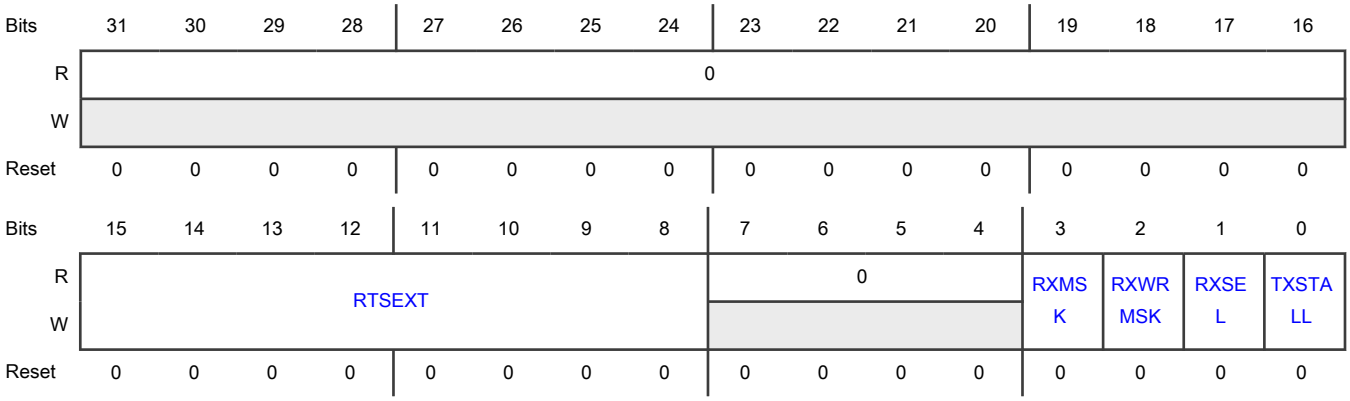
Provides control for half-duplex-related operations.

You can use this register instead of CTRL[LOOPS], CTRL[RSRC], and CTRL[TXDIR] functions, although you can use CTRL[LOOPS] to loop-back the transmitter outputs to the receiver.

Offset

Register	Offset
HDCR	50h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 RTSEXT	RTS Extended Specifies RTS extension. The transmit RTS_B remains asserted for (RTSEXT + 1) bit times after the end of the last stop bit. This applies even when the transmitter's RTS_B output is disabled and is only used internally to mask the receiver.
7-4 —	Reserved
3 RXMSK	Receive Mask Specifies whether the transmitter RTS_B masks the receive data pin. When enabled, the transmitter RTS_B masks the receive data pin. 0 - Do not mask 1 - Mask
2 RXWRMSK	Receive FIFO Write Mask Specifies whether the transmitter RTS_B masks writes to the receive FIFO. When enabled, the transmitter RTS_B masks receive FIFO writes, but the idle flag is not affected. 0 - Do not mask 1 - Mask
1 RXSEL	Receive Select Specifies the receive data pin. When enabled, the receive data is sourced from the TXD pin. 0 - RXD

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - TXD
0 TXSTALL	Transmit Stall Specifies whether the transmitter becomes busy when the receiver is active. When enabled, the transmitter does not become busy or asserts transmitter RTS_B when the receiver is active (STAT[RAF] is 1). 0 - No effect 1 - Does not become busy

62.6.20 Timeout Control (TOCR)

Configures the behavior of the timeout logic. Timeouts 0 and 1 are used to monitor the receiver and timeouts 2 and 3 are used to monitor the transmitter.

Offset

Register	Offset
TOCR	58h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				TOIE				0				TOEN			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-12 —	Reserved
11-8 TOIE	Timeout Interrupt Enable Enables the corresponding timeout flag to generate an interrupt.

Table continues on the next page...

Table continued from the previous page...

Field	Description
7-4 —	Reserved
3-0 TOEN	Timeout Enable Enables the corresponding timeout counter.

62.6.21 Timeout Status (TOSR)

Indicates the status of the timeout logic. Timeouts 0 and 1 are used to monitor the receiver and timeouts 2 and 3 are used to monitor the transmitter.

Offset

Register	Offset
TOSR	5Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				TOF				0				TOZ			
W					W1C											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Fields

Field	Description
31-12 —	Reserved
11-8 TOF	Timeout Flag Indicates whether the corresponding timeout occurred. The timeout counter is disabled when this field is 1. **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When reading 0000 - Not occurred 0001 - Occurred When writing 0000 - No effect 0001 - Clear the flag
7-4 —	Reserved
3-0 TOZ	Timeout Zero Indicates whether the corresponding timeout counter equals 0.

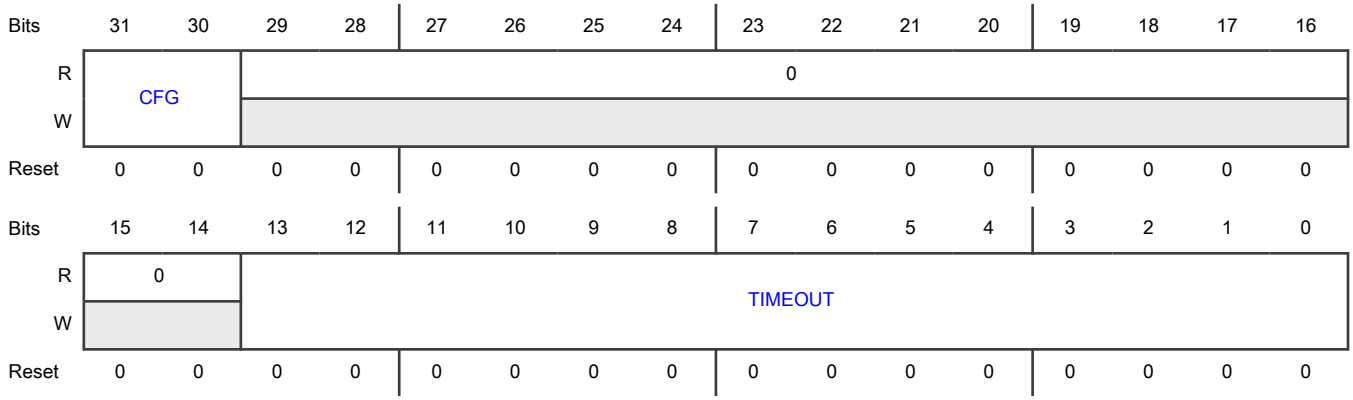
62.6.22 Timeout N (TIMEOUT0 - TIMEOUT3)

Configures the corresponding timeout counter and status field. Timeouts 0 and 1 are used to monitor the receiver and timeouts 2 and 3 are used to monitor the transmitter. You must write to this register only when the corresponding timeout is disabled or when the value of the corresponding timeout field is 1.

Offset

Register	Offset
TIMEOUT0	60h
TIMEOUT1	64h
TIMEOUT2	68h
TIMEOUT3	6Ch

Diagram



Fields

Field	Description
31-30 CFG	Idle Configuration Configures the behavior of TIMEOUT[TIMEOUT]. 00 - Becomes 1 after timeout characters are received 01 - Becomes 1 when idle for timeout bit clocks 10 - Becomes 1 when idle for timeout bit clocks following the next character 11 - Becomes 1 when idle for at least timeout bit clocks, but a new character is detected before the extended idle timeout is reached
29-14 —	Reserved
13-0 TIMEOUT	Timeout Value Configures the timeout value.

62.6.23 Transmit Command Burst (TCBR0 - TCBR127)

Acts as an alias of Data (DATA), designed to support incrementing burst transfers to the transmit FIFO by a DMA controller, using aligned 8-bit, 16-bit, or 32-bit writes. The size of this register is 512 bytes:

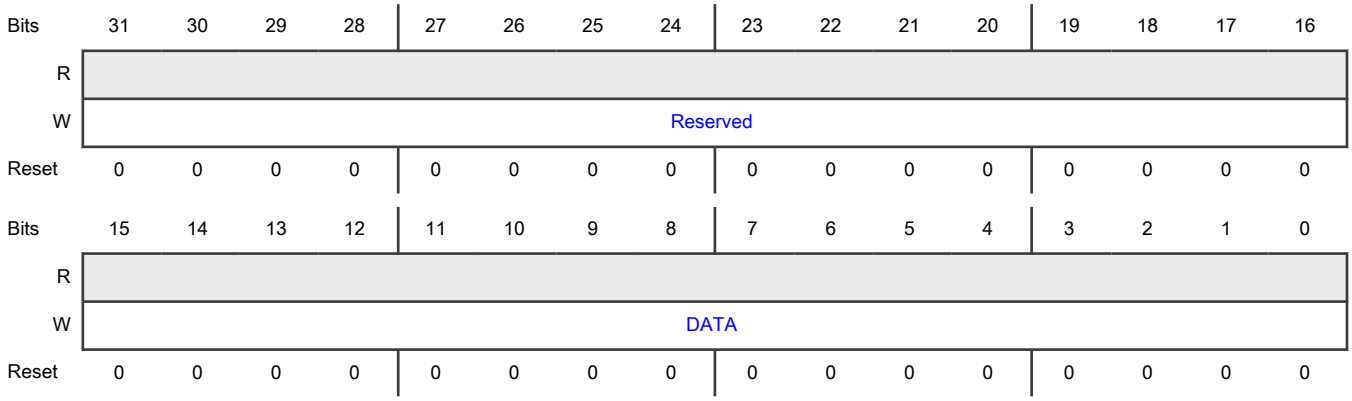
- An aligned 32-bit write in this region pushes one entry into the transmit FIFO.
- An aligned 16-bit write in this region to TCBR*x*[15:0] pushes one entry into the transmit FIFO.
- An 8-bit write in this region to TCBR*x*[7:0] updates DATA[7:0], but does not push the data into the transmit FIFO.
- An 8-bit write in this region to TCBR*x*[15:8] pushes the data written to DATA[15:8] plus the previously written DATA[7:0] into the transmit FIFO.
- An 8-bit or 16-bit write in this region to TXBR*x*[31:16] is ignored.

Offset

For a = 0 to 127:

Register	Offset
TCBRa	200h + (a × 4h)

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 DATA	Data Enables writing data to Data (DATA).

62.6.24 Transmit Data Burst (TDBR0 - TDBR255)

Acts as an alias of Data (DATA), designed to support incrementing burst transfers to the transmit FIFO by a DMA controller using 8-bit, 16-bit, or 32-bit writes. The size of this register is 1024 bytes:

- An aligned 32-bit write in this region pushes four DATA byte entries into the transmit FIFO (DATA0 byte first). The register access is extended by three wait states.
- An aligned 16-bit write in this region to either half of a 32-bit word pushes two DATA byte entries into the transmit FIFO (DATA0 or DATA2 first). The register access is extended by one wait state.
- An 8-bit write in this region pushes one DATA byte entry into the transmit FIFO.

Byte writes to Data (DATA) use the contents of CTRL[R9T8] for the ninth data bit, CTRL[R8T9] for the tenth data bit, and zero extend DATA[FRETSC].

Offset

For a = 0 to 255:

Register	Offset
TDBRa	400h + (a × 4h)

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DATA3								DATA2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA1								DATA0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 DATA3	Data3 Enables writing of data to Data (DATA).
23-16 DATA2	Data2 Enables writing of data to Data (DATA).
15-8 DATA1	Data1 Enables writing of data to Data (DATA).
7-0 DATA0	Data0 Enables writing of data to Data (DATA).

62.7 Glossary

Baud rate Number of bits per second that LPUART transmits or receives

Break character Break character is generated when the transmitter is holding the data line at the space level for at least one character time

Oversampling Number of times the receive circuitry samples the receive input per baud period (that is, per data bit)

Chapter 63
Improved Inter-Integrated Circuit (I3C)

63.1 Chip-specific I3C information

Table 591. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing

The chip includes two I3C Modules (I3C1 in AONMIX and I3C2 in WAKEUPMIX). The HCI-I3C (Host Controller Interface) is not supported.

63.2 Overview

I3C is a communications processor that improves upon the use and power of I2C, and provides an alternative to SPI for mid-speed applications.

The I3C bus protocol supports:

- In-band interrupts (IBI): these interrupts move from target to controller without extra wires, and the controller knows which target sent the interrupt
- Common command codes (CCC)
- Dynamic addressing
- Multi-controller and multi-drop
- Hot-Join (HJ)
- I2C compatibility

I3C supports all required and many optional features of the MIPI Alliance Specification for I3C, v1.0 and v1.1, except for ternary data rates (HDR-TSP and HDR-TSL). See Features, Target Capabilities (SCAPABILITIES), and Target Capabilities 2 (SCAPABILITIES2) for more information.

63.2.1 Block diagram

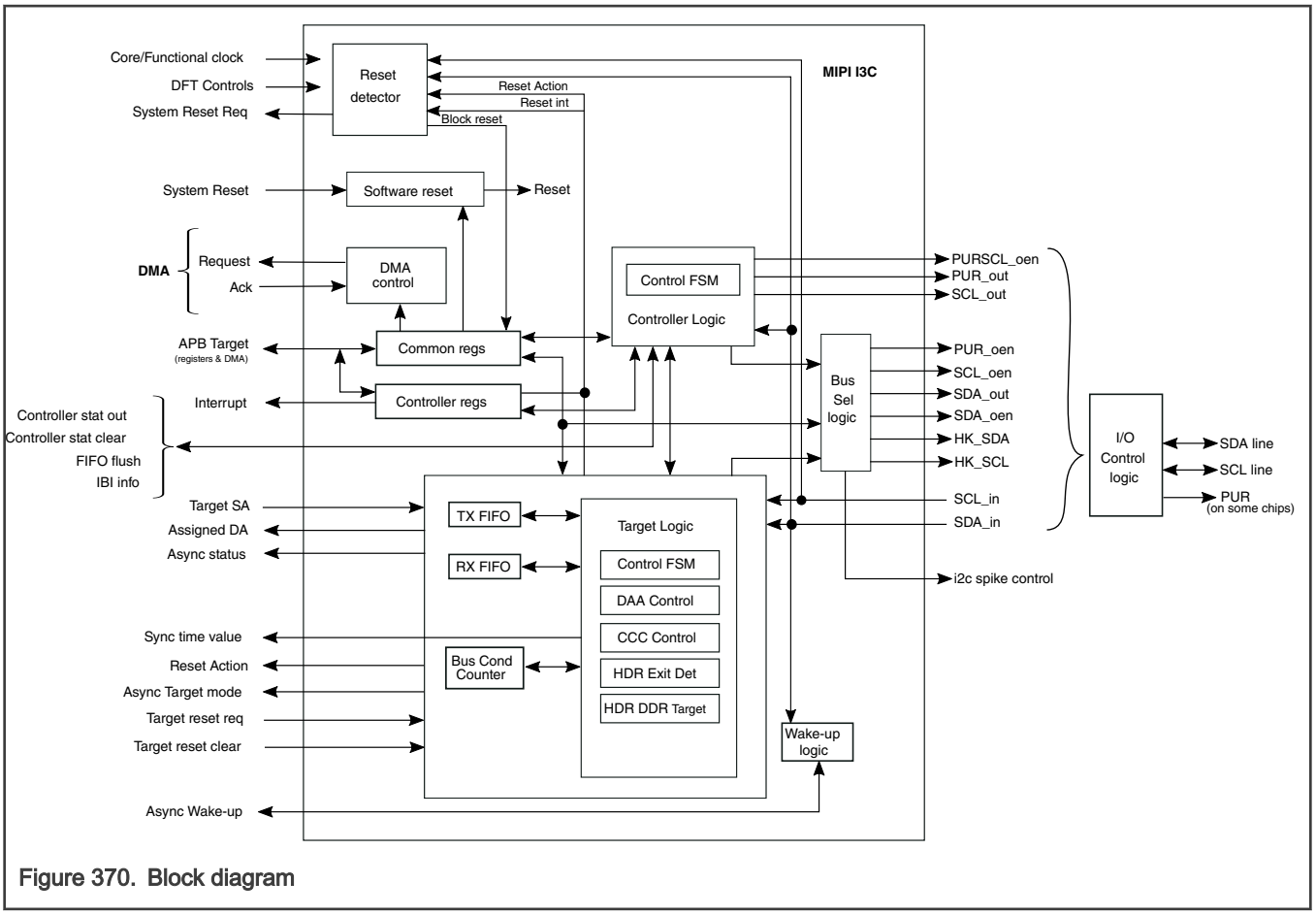


Figure 370. Block diagram

63.2.2 Features

- Two-wire multidrop bus that is capable of up to 12.5 MHz clock speeds with up to 11 devices:
  - Uses standard pads with 4 mA drive.
  - Dynamically assigns target addresses, and targets do not require static addresses (SAs). However, targets can have an I2C static address assigned at startup, so the target can operate on an I2C bus. By default, I3C supports 7-bit I2C-style addresses.
  - Supports extended I2C 10-bit addressing through Map Feature Control 1 (SMAPCTRL1).
  - Allows targets to use the inbound SCL clock as the peripheral clock (instead of the clock from the controller) so that devices can have slow or inaccurate clocks internally.
  - Allows simple targets, such as temperature sensors, to have no internal clock.
  - I3C controller supports handoff from Open-Drain to Push-Pull mode for ACK to data transfer.
  - Generally, the controller terminates the read, but for I3C, the target can also end the read.
- IBIs that allow targets to send notifications to a controller:
  - Can be equivalent to a separate GPIO, but can also be directly data-bearing.
  - Can be prioritized. When multiple targets send interrupts to a controller at the same time, the order is resolved. Dynamic addresses (DAs) establish the priority of the targets, so the controller controls the priority of the targets.

    Targets with lower-value DAs are higher-priority level IBIs.

  - Can start interrupts even when the controller is not active on the bus. A free-running clock is not required, but starting an interrupt requires a Bus Available condition.
  - Can resolve an initial event via a time-stamping option, not requiring an interrupt.
- Built-in commands for applications created in software or firmware by using the register interface maintained in a separate space. These commands do not collide with normal controller-to-target messages and:
  - Control bus behavior, modes and states, low-power state, inquiries, and more.
  - Have an extra room for new built-in commands for other groups.
- Organized forms of multicontroller modes that are secondary controllers using clean handoffs between different controllers.
- Hot-Join onto I3C bus allows devices to connect to the bus later than when the bus starts. This:
  - Enables a device or module to access the I3C bus after power up or when physically inserted onto the I3C bus.
  - Provides a clean method for notification when new devices or modules access the I3C bus.
- I3C can use both I2C and I3C buses:
  - I3C supports specific legacy I2C devices on the bus.
  - I3C target devices can operate on I2C buses.
  - In I2C controller mode, it supports only Push-Pull mode on the SCL line.
  - A reset output is available for I2C mode SW reset for application usage. Asserted when the module is an I2C target and gets a I2C software General command from a controller. See the chip-specific section, for usage if this is available.
- Higher data rate modes are available:
  - I3C has a high data rate: double data rate (HDR-DDR) mode, which is double the data rate of SDR .
  - Only the controller and the specific targets, which are capable and configured to work with HDR-DDR, support the higher data rate (the other targets can ignore it).

I3C supports most of the I3C features (see Target Capabilities (SCAPABILITIES) and Target Capabilities 2 (SCAPABILITIES2)), except for the ternary data rates (HDR-TSP and HDR-TSL), HDR-BT modes, and peer-to-peer messaging.

63.3 Functional description

63.3.1 Operating modes

This section describes all functional operation modes of the I3C module.

63.3.1.1 Target and controller roles for I3C

The I3C protocol defines these roles for devices on the I3C bus:

- Main controller: initially configures the I3C bus and serves as the first active controller
- Secondary controller: accepts the controller role from any active controller to become the new active controller (the secondary controller may then pass the controller role along: either back to the previous active controller, or on to any other controller-capable device)
- Target: responds to commands from any I3C controller
- I2C target: responds to compatible commands from any I3C controller

The I3C peripheral contains both controller and target components and can be configured to be either controller or target, or as target with secondary controller capability. However, if I3C is chosen to be a controller, then the I3C peripheral supports Target mode to facilitate handoffs to multiple controllers.

In general, any I3C controller can be a controller or a target because a controller becomes a target when giving over control to another controller. The only exceptions to this rule occur when using point-to-point communication or when using a controller that never shares controllership.

63.3.1.1.1 Controller requirements

Controller mode uses software that supports the requirements of the controller (including as a secondary controller):

- Managing the enter dynamic address assignment (ENTDAA) assigning dynamic addresses (DAs) to each target. The I3C peripheral supports this process but requires you to make choices.
- Driving the serial data (SDA) signal in both Open-Drain and Push-Pull modes. After START command, SDA is in Open-Drain mode and after Repeated START command, it is in Push-Pull mode:
  - SDA is subject to arbitration because both controller and target can drive the SDA line (arbitration is also useful for handoff from controller to target).
  - The ninth bit of SDA controller write data is an odd parity bit.
- Building a table of targets and their capabilities to control which actions and commands may be sent to the targets.
- Managing requests such as IBI and controller requests (handoff).
- Adjusting clock speed or write-to-read timing to match the limitations of the target. This adjustment can be done in hardware using dividers and uneven duty cycles, but you must decide how.
- Adjusting the maximum data length.
- Being a target after the controller role handoff to another controller-capable device.

The controller needs an accurate clock capable of running at a frequency that is the multiple of a frequency between 11 MHz and 12.5 MHz. For example:

- 24 MHz (multiple of 12 MHz)

If required to use a lower I3C SCL value, for example as small as 5 MHz (although the lower value does not support a mixed bus system):

- The controller clock can be a multiplier of that value.
- A lower SCL can also be achieved using a higher PPBAUD value.

63.3.1.1.2 I3C target acts like I2C target on I3C buses

If the target is assigned an I2C static address, the I3C target acts like an I2C device when it first powers on. If the I3C target is placed on an I2C bus with an I2C controller, then the I3C target stays in I2C mode and operates normally. The software is aware that the I3C target is in I2C mode because:

- There has not been an interrupt (see SSTATUS[DACHG]) indicating that a dynamic address was assigned.

For full I2C support of Fast mode (Fm) and Fast-mode Plus (Fm+), the pads must support a 50-ns spike filter. This filter must be turned off when the I3C 7Eh broadcast address is received (indicating an I3C controller). You can turn off the spike filters via hardware using the raw net indicating that the address was received, or via software. I3C does not need a 50-ns spike filter to operate on an I2C bus. Therefore, the spike filter is not a requirement for the I3C peripheral.

Depending on the configuration, the module supports most I2C features, except for clock stretching. Supported features include an extended 10-bit address, DeviceID, and software reset. See Target Capabilities (SCAPABILITIES) and Target Capabilities 2 (SCAPABILITIES2) for more information.

63.3.1.1.3 How a target rejoins the I3C bus

When a target tries to rejoin the I3C bus following a power-up or hard reset, the target needs a new dynamic address (DA). The target can rejoin the I3C bus in these ways:

- If the DA is lost, Hot-Join is used.

- If the DA is retained in the peripheral (for example, with state retention flip-flops), SCONFIG[OFFLINE] is used.

When SCONFIG[SLVENA] is 1, SCONFIG[OFFLINE] may become 1. This setting causes the peripheral to rejoin the bus safely. It does so by ensuring that the I3C bus is not in HDR mode, using the same approach as TE0 or TE1 exit. The peripheral waits for the HDR exit pattern from the controller, or for 60 µs of SCL and SDA lines not changing, whichever occurs first.

After using SCONFIG[OFFLINE], the I3C peripheral cannot safely use IBI until SSTATUS[STOP] becomes 1. This status ensures that the next START is safe to use for IBI.

If the application has to perform an IBI, it must wait for SSTATUS[STOP] to become 1, or for SCL and SDA lines to remain high for 200 µs. You can perform this process using the SSTATUS and SINTSET controls:

- If Target Status (SSTATUS) indicates that the bus is not busy and the peripheral interrupts on START or STOP, use a timer to measure 200 µs. If the timer finishes with no START or STOP, it is safe to use IBI.
- If a START causes an interrupt, then the timer must be turned off and the application must wait for a STOP.
- If a STOP causes an interrupt, it is safe to use IBI.

63.3.1.2 Using the I3C controller for I2C and I3C

The I3C controller operates with the following set of built-in capabilities with the application flow:

- Built-in enter dynamic address assignment (ENTDAA) mechanism to simplify the assignment of dynamic addresses to targets. This feature is used when set dynamic address from static address (SETDASA) is not available.

  NOTE

  When SETDASA CCC is available, use SETDASA CCC before using ENTDAA CCC; all targets without assigned dynamic addresses respond to ENTDAA CCC.

- Request for START + address with IBI support, including both I2C and I3C modes.
- SDR write flow via FIFO, with automatic parity in I3C and ACK or NACK detection in I2C.
- SDR read flow via the FIFO, with an automatic NACK generator for I2C, and optional use of a terminate generator for I3C.
- Request for Repeated START + address or STOP when finished with the previous request, including both I2C and I3C.
- Auto-IBI mode, which responds immediately to a target-initiated IBI and can be used when in Sleep or Deep Sleep mode.
- Special message mode for SDR and DDR to simplify DMA use.
- Automated SCL, SDA, pullup, and High-Keeper controls.
- I2C and I3C frequency and duty cycle configurations from functional clock.

  NOTE

  You must configure Controller Configuration (MCONFIG) and Controller In-band Interrupt Registry and Rules (MIBIRULES) before using the controller.

63.3.1.3 Protocol modes and states

The following modes are activated in I3C:

- I2C mode is the default mode on startup:
  - If no I2C static address is used, this mode has no effect other than to track frames looking for I3C transitional frames.
  - If an I2C static address is used, the device can interact with the I2C bus controller using the address.
- When in I2C mode, I3C Transitory Frame mode occurs whenever the I3C broadcast address is received (7Eh). In particular:
  - This mode occurs when 7Eh is broadcast followed by SETDASA from the controller. If there is a static address match, the target is assigned a dynamic address and it enters I3C SDR mode.

— This mode occurs when 7Eh is broadcast followed by ENTDAA from the controller. If the target can send a 48-bit provisioned ID (48b ID), a dynamic address is assigned, and the target enters I3C SDR mode.

— If a Hot-Join arbitrated event occurs (sending 02h when the controller generates another address such as 7Eh), there is a conceptual Hot-Join mode. ENTDAA or SETDASA sets a dynamic address to resolve the event, and the target enters I3C SDR mode.

NOTE

An I3C target can operate as a normal I2C target only when it has a static address. Otherwise, it matches nothing in I2C and waits for the aforementioned events.

- I3C SDR mode is the standard mode after I3C activates, which is defined by a dynamic address being assigned. This mode is the resting mode of I3C, and all devices are normally in SDR mode:
  — RSTDAA CCC causes an exit from SDR mode and a return to I2C mode. This transition is not normally used.

    RSTDAA can be issued to ensure that all I3C targets participate in an upcoming dynamic address assignment process, before the I3C controller starts assigning the target address.

  — SETNEWDA does not affect the SDR mode; it only changes the dynamic address of an I3C device that already had a dynamic address assigned.

  — When in SDR mode, the device ignores messages to or from its original I2C static address.
- The I3C CCC command submode of type Direct or Broadcast. Following are the options to exit the CCC:
  — Exit the Broadcast CCC submode by a Repeated START or by a STOP.

  — Exit the Direct CCC submode by a 7Eh broadcast address after a Repeated START or by a STOP.

  — I3C dynamic address assignment (DAA) CCC command enters DAA mode. This mode is a special mode for dynamic addressing. Use STOP to exit. If a target has a dynamic address, the command is ignored.
- I3C SETDASA CCC command enters Static Address Match submode, which allows matching the static address of the device (if any). The command is ignored when a target has a dynamic address, or when the target in I2C mode has no static address.

NOTE

The special point-to-point address is also matched.

- I3C HDR modes are activated by ENTHDR*n* CCC commands, where *n* represents the type of HDR (0 to 7). This mode is valid until an HDR exit pattern is detected, irrespective of whether HDR mode for a target is supported.

  Exit-pattern detection must always be on to prevent errors. Alternatively, you could set it to be activated only on ENTHDR.
- A special HDR-DDR syntax is used for transmitting CCC in the HDR-DDR protocol.
- Internal states such as IBI or no-IBI and Low-Power mode are flagged internally (see SSTATUS[EVDET] for more information). These states are exported to the application and affect the engine to restrain it from performing a prohibited operation.

63.3.1.4 Address match

For an address match to occur, the target matches the I3C broadcast address and either the I2C-style static address or I3C dynamic address (see Table 592).

The address match is inactive (just listening) for all Repeated START commands and also for the START command, unless an IBI, CR, or Hot-Join has been activated. If inactive, it simply tries to match. If not matched, it waits for a Repeated START or STOP command. If an IBI, CR, or Hot-Join has been activated, the arbitration mechanism is used for START (but never for a Repeated START).

When matching the 7Eh broadcast address, it looks for a CCC until the next Repeated START or STOP command.

Table 592. Address match conditions

For	Match condition occurs when	Notes
I3C broadcast address	The address is 111 1110b (written as 7Eh).	
I2C-style static address	• The device has a static address. • The device is not in I3C mode.	When in I3C mode, the address matches only until the ENTDAA, SETDASA, or SETAASA command has a dynamic address assigned. SETDASA matches the static address or the special one-controller-to-one-target point-to-point address.
I3C dynamic address	ENTDAA or SETDASA assigns it, or SETNEWDA modifies it.	

63.3.1.5 I3C from low power mode and wake-up

A wake-up interrupt is available to bring the I3C module out of low power mode.

This interrupt works in both Controller and Target mode of operation as described below.

63.3.1.5.1 Controller mode

In Controller mode, I3C works with 2 clocks.

- Host interface clock for accessing registers and DMA
- Functional clock (FCLK) for controller logic operation and generating SCL

The Functional clock can be switched off to keep the I3C controller in low power mode if there is no data to be sent by the host and the i3C bus is in idle state (SDA = 1).

Before switching off the clock, take the below steps.

- Ensure that you enable the Controller mode MCONFIG[MSTENA] = 01b.
- Ensure that the bus is in the idle state.
- Switch off the Functional Clock.

If any agent on the bus pulls the SDA line low and at the SoC level the ENABLE bit of the I3C SDA interrupt is set to 1, then the interrupt goes high to inform the SoC logic to switch on the FCLK again.

63.3.1.5.2 Target mode

In Target mode, I3C works with only the Host interface clock (for accessing registers and DMA)

The main target control logic works on the external SCL clock.

The Host interface clock can be switched off to keep the I3C Target in low-power mode.

Before switching off the clock, take these steps.

- Ensure that you configure the Target mode, MCONFIG[MSTENA] = 00b along with all other normal target configuration as required.
- Clear the interrupts in Target Status (SSTATUS) after handling them.
- Enable DA matched interrupt in SINTSET[MATCHED].
- The host can now stop the Host interface clock to enter low power mode.

The interrupt asserts when any address match happens, either DA or SA or group address with this target.

The host then resumes the Host interface clock to serve and clear the interrupt.

63.3.2 Operations

This section describes the operations of the module.

63.3.2.1 Reading and writing I2C messages using the normal method

1. Set up interrupts using the following fields of Controller Interrupt Set (MINTSET):
   - MCTRLDONE: Indicates when the module completes an MCTRL request.
   - COMPLETE: Indicates when data is finished sending or being received.
   - RXPEND: Used for read operations. Works with Controller Data Control (MDATACTRL) to set the FIFO trigger. You can also use this field to allow DMA to read out data.
   - TXNOTFULL: Used for write operations. Works with Controller Data Control (MDATACTRL) to set the FIFO trigger. You can also use this field to allow DMA to supply data.
   - IBIWON: Indicates that an IBI, CR, or HJ has won the arbitration on a header address.
   - Additionally, Controller Errors and Warnings (MERRWARN) indicates the causes of errors and warnings.
2. Configure the following fields in Controller Control (MCTRL) simultaneously:
   a. Write 1 to REQUEST (EmitStartAddr).
   b. Write 1 to TYPE (I2C).
   c. Configure IBIRESP to respond to IBIs in your chosen manner.
   d. Write 1 to DIR for read, or write 0 for write.
   e. Write the static address of the I2C target to ADDR.
   f. Configure RDTERM to the maximum length for read operations, to auto-terminate, or set it to STOP as the data is read out. For example, write 1 (with MCTRL[REQUEST] = 0) to stop after the next character.
3. Write or read the data.
   - For write operations, write to Controller Write Data Byte (MWDATAB) for each byte before the last byte, and then write to Controller Write Data Byte End (MWDATABE) for the last byte:
     — You can perform or start this operation before setting up interrupts.
     — If there is more data than the FIFO can hold, use the TXNOTFULL interrupt based on the trigger level. This interrupt allows the application to provide more data or to use DMA.
   - For read operations, wait for RXPEND, and then read out data via Controller Read Data Byte (MRDATAB). DMA may also be used.
4. After message transfer is complete (MSTATUS[COMPLETE] = 1), the message may end with a STOP condition, or a new message may start with a Repeated START condition:
   a. Write 2 to MCTRL[REQUEST] (EmitStop to STOP). Then wait for the MCTRLDONE status to be asserted for its completion. (When sending STOP in I2C mode, MCONFIG[ODSTOP] and MCTRL[TYPE] must be 1.)
   b. Write 1 to MCTRL[REQUEST] (EmitStartAddr to restart). IBI is not possible in this case.

NOTE

I2C Fm and I2C Fm+ modes are supported for legacy I2C devices as per the I3C standard specifications. See I2C configuration to meet timing requirements for Fm and Fm+ modes for more information.

63.3.2.2 Reading and writing I3C messages using the normal methods (SDR and HDR-DDR)

The normal method for I3C is the same as the method for I2C with a few differences:

1. Set up interrupts using the following fields of Controller Interrupt Set (MINTSET):
   - MCTRLDONE: Indicates when the I3C module completes an MCTRL request.
   - COMPLETE: Indicates when data is finished sending or being received.
   - RXPEND: Used for read operations. Works with Controller Data Control (MDATACTRL) to set the FIFO trigger. You can also use it to allow DMA to read out bytes.
   - TXNOTFULL: Used for write operations. Works with Controller Data Control (MDATACTRL) to set the FIFO trigger. You can also use it to allow DMA to supply bytes.
   - IBIWON: Indicates that an IBI, CR, or HJ has won the arbitration on a header address.
   - Additionally, Controller Errors and Warnings (MERRWARN) indicates the causes of errors and warnings.
2. Set up Controller Control (MCTRL).
   - Option 1: Configure the following fields of Controller Control (MCTRL) simultaneously in this way:
     a. Write 1 to REQUEST (EmitStartAddr).
     b. Write 0 to TYPE for I3C Single Data Rate (SDR) mode or write 2 for Double Data Rate (DDR) mode.
     c. Configure IBIRESP to respond to IBIs in your chosen manner.
     d. Write 1 to DIR for read, or write 0 for write.
     e. Write the dynamic address of the I3C target to ADDR.
     f. For read operations, you can configure RDTERM to the maximum length to terminate automatically.
     g. For write operations, prewriting the data (MWDATAB or MWDATAH) is preferred to ensure that there are no time delays waiting for the data.

        For DMA with MCTRL, use Controller Write Byte Data 1 (to Bus) (MWDATAB1).

     NOTE

     HDR-DDR mode requires writing an 8-bit command value for read or write. This value must be written into the TX FIFO via Controller Write Data Byte (MWDATAB). The END field is not used for this byte.
   - Option 2: This option is preferred when stopped (bus free condition) in SDR mode, and not in HDR-DDR mode. It allows any target to issue an IBI and avoids collisions with an IBI address. Also, it is faster (when the MSB of the dynamic address is always 0).

     Configure the following fields of Controller Control (MCTRL) in this way:
     a. Write 1 to REQUEST (EmitStartAddr). No prewritten transmit data can be in the FIFO.
     b. Write 0 to TYPE (I3C).
     c. Configure IBIRESP to respond to IBIs in your chosen manner.
     d. Write 0 to DIR.
     e. Write 7Eh to ADDR.
     f. Wait for MCTRLDONE (via interrupt, for example), then proceed as in option 1. The option 2 method has advantages for IBIs when 7Eh is sent on START (but not on Repeated START conditions).
3. Write or read the data.
   - For write operations, write to MWDATAB for each byte before the last byte, and then write to Controller Write Data Byte End (MWDATABE) for the last byte. For HDR-DDR, the byte with END must be even (second, fourth, sixth, and so on) because DDR uses byte pairs:

— You can perform or start this operation before step 1 (REQUEST = 1) of the option 1 method, but not before the option 2 method.

— If there is more data than the FIFO can hold, use the TXNOTFULL interrupt based on the trigger level. This interrupt allows the application to provide more data or to use DMA.

- For read operations, wait for RXPEND, and then read out data via Controller Read Data Byte (MRDATAB) or Controller Read Data Halfword (MRDATAH). DMA may also be used. When using DMA, read using the same registers.

4. After message transfer is complete (MSTATUS[COMPLETE] = 1), the message may be ended with STOP (or EXIT in HDR mode). Alternatively, a new message may be started with a Repeated START (or HDR-Restart in HDR mode):
   - In SDR mode, write 2 to MCTRL[REQUEST] (EmitStop to STOP). Then wait for the MCTRLDONE status to be asserted for its completion.
   - For HDR mode, write 6 to MCTRL[REQUEST] (ForceExit) to end HDR mode. Then wait for the MCTRLDONE status to be asserted for its completion. When sending the HDR exit pattern, MCONFIG[ODSTOP] must be 0.
   - Write 1 to MCTRL[REQUEST] (EmitStartAddr to start another message). IBI is not possible in this case.

63.3.2.3 Determining bus types and modes

The settings of MCTRL[TYPE] and MCTRL[REQUEST] determine bus types and modes.

Table 593. Determining bus types and modes

Value of MCTRL[TYPE]	Meaning when MCTRL[REQUEST] = 1 (EmitStartAddr)	Meaning when MCTRL[REQUEST] = 2 (EmitStop)	Meaning when MCTRL[REQUEST] = 6 (ForceExit)
0	SDR mode of I3C	SDR mode of I3C	Exit pattern
1	Standard I2C protocol	Standard I2C protocol	Reserved
2	The bus enters DDR mode (7E and then ENTHDR0), if the module is not in DDR mode already. The first byte written to the transmit FIFO must be a command and already in the FIFO. To end DDR mode, use ForceExit.	Reserved	Target reset
3	If not already in HDR-BT, the bus automatically enters BT mode (7E and then ENTHDR3). You must also configure the MHDRBTCFG register for multilane and CMD rules.	Reserved	Reserved

Table 594. Deriving the dynamic address

Value of MCTRL[TYPE]	Meaning when MCTRL[REQUEST] = 4 (ProcessDAA)
0	The dynamic address is derived from MWDATAB[VALUE]; in case of a NACK, ProcessDAA continues to the next 7E/R request.
1	Reserved
2	The dynamic address is derived from MWDATAB[VALUE]; in case of a NACK, a STOP condition is sent after the assignment.
3	Reserved

63.3.2.4 Sending a CCC to I3C targets

The normal common command code (CCC) method uses an I3C write operation with an address of 7Eh (the CCC is the first byte):

- For broadcast type, any remaining bytes are sent with the CCC. This operation ends with a STOP or a Repeated START and 7Eh.
- For direct type, only the CCC byte is sent (or by a CCC and a defining byte, if required by the CCC).

  These bytes are followed by a Repeated START and the address of the I3C target (for SETDASA, this address is its I2C static address). This sequence may be repeated with more Repeated START conditions and addresses until done. It may end in STOP or a Repeated START and 7Eh. After the Repeated START and target address, the values are read or written depending on the CCC.
- I3C provides interrupts (by hardware) for unhandled and handled CCC when received by the target. Its state is reflected in Target Status (SSTATUS).

I3C controllers require to emit a single START, 7E/W sequence with both SCL high and SCL low half periods at full open-drain timing (for example, 200 ns). This sequence allows I3C targets acting as I2C legacy devices to turn off their I2C 50-ns spike filters, if they have them.

NOTE

START, 7E/W is a notation indicating a START command, followed by the 7Eh broadcast address and then by a write command.

After that sequence, addresses following START may be sent with Open-Drain Low (for example, 200 ns) but high of Push-Pull timing (for example, 40 ns). The I3C controller must emit this START, 7E/W sequence with MCONFIG[ODHPP] = 0. ODHPP is Open-Drain High period at Push-Pull speeds, and MCONFIG[ODHPP] is usually 1.

NOTE

- MCONFIG[ODHPP] is ignored when sending a message to an I2C legacy device on an I3C bus.
- Each Repeated START is a new EmitStart request. This request can be chained by an interrupt or pushed by a message model using DMA.

63.3.2.5 IBI handling

An IBI occurs when a target sends its address after a START, and that address is numerically the lowest. That is, it is lower than the address sent by the controller and addresses sent by any other targets. When the controller sends 7Eh, the controller always loses the arbitration.

The IBI can occur unexpectedly when any new START (not a Repeated START) is sent. The IBI can also occur in response to a target pulling SDA low. This condition can occur in one of these ways:

- The controller has set the request to AutoIBI mode, so the IBI occurs automatically. The controller sends 7Eh to allow the target to win the arbitration.
- The application receives the SLVSTART (target START request) interrupt, so it sends 7Eh.

The IBI response is configured by MCTRL[IBIRESP], which can be set to:

- NACK (always reject the IBI).
- ACK (always accept the IBI):
  - You must configure Controller In-band Interrupt Registry and Rules (MIBIRULES) so that the engine knows whether IBI bytes follow.
  - If IBI bytes follow (also known as IBI mandatory byte), then the COMPLETE field does not become 1 when IBIWON becomes 1. RXPEND becomes 1 for one or more bytes in the receive FIFO. When the last byte is received, then the COMPLETE field becomes 1. The controller automatically stops IBI data after 9 bytes (including the mandatory data byte). MCTRL[RDTERM] can be used with MCTRL[REQUEST] = 0 to end the process of receiving data sooner. I3C

supports address ACK to mandatory byte transition during IBI from Open_Drain (controller acknowledges the target address) to Push Pull (target sends SDR data).

— I3C target supports up to 7 bytes (maximum limit is defined by IBIEXT1[MAX]) of extended IBI data following a mandatory data byte. Extended data to send, if any, is present in Extended IBI Data 1 (IBIEXT1) and Extended IBI Data 2 (IBIEXT2).

- Manual (allow the decision to be made by the application on a case-by-case basis):
  — The application rewrites it when stopped, pending an IBI.
  — The application can ACK or NACK the IBI based on the IBI address defined in MSTATUS.
  — This mode selects whether there is an IBI byte when accepting the IBI request from targets.

Accurate timestamping of I3C target data is supported in I3C though Async mode 0. In I3C Asynchronous Timing Control mode, a target device timestamp event occurs within that target. The target notifies the controller about the event by generating an IBI.

63.3.2.6 Assigning dynamic addresses to I3C devices

If dynamic addresses (DAs) are all assigned below 40h (7-bit values from 3Fh down to 03h, except where not allowed), then MIBIRULES[MSB0] can be 1. This setting optimizes the START timing. When dynamic addresses are assigned, you must program Controller In-band Interrupt Registry and Rules (MIBIRULES) based on which I3C target uses IBI bytes (present in SIDEXT[BCR]).

Any SETDASA assignments can be performed via the normal CCC model. These assignments cannot be performed when using the static address for the directed part.

There is a built-in mechanism to process ENTDAA mode:

1. Set up interrupts using the following fields of Controller Interrupt Set (MINTSET):
   - MCTRLDONE: Indicates when a target has sent its ID and BCR or DCR, and a new DA is needed.
   - COMPLETE: Indicates when DAA is done (NACKed by all targets).
   - RXPEND: Indicates the reading of the IDs of the targets. You can also use this field to allow DMA to read out bytes.
   - IBIWON: Indicates when an IBI has occurred (normally there are no IBI conditions when assigning dynamic addresses to the I3C device).
   - Additionally, Controller Errors and Warnings (MERRWARN) indicates the causes of errors and warnings.
2. Configure Controller Control (MCTRL):
   a. Write 4 to MCTRL[REQUEST] (ProcessDAA).
   b. Set MCTRL[IBIRESP] to IBI response, if possible. If not possible, set to the first target assignment.
3. Wait for the MCTRLDONE interrupt when reading ID using the RXPEND interrupt. If MSTATUS[STATE] = 5 (DAA mode) and MSTATUS[BETWEEN] = 1, it is waiting to write a dynamic address:
   a. Write the dynamic address into Controller Write Data Byte (MWDATAB) using bits 6:0, such as 14h for DA = 7'h14.
   b. Write 4 to MSTATUS[STATE] (ProcessDAA again). This option writes the DA, then moves to the next DA.
   c. If MSTATUS[COMPLETE] = 1 and MSTATUS[STATE] = 0, then all targets are assigned.
   d. If MSTATUS indicates NACK after writing a new DA, the DA is not accepted by the target. The next step is to write 2 to MCTRL[REQUEST] (EmitStop) and start over. It is also acceptable to write 4 to MCTRL[REQUEST] (ProcessDAA again).

63.3.2.7 Using Controller Message mode

Controller Message mode is intended for use with DMA although Message mode can also be used by the processor. In Message mode, all writes are to the same location, including control and data. Reads occur from an associated location.

NOTE

The data writes for Message mode work in the same way as the halfword access registers, Controller Write Data Halfword (MWDATAH) and Controller Read Data Halfword (MRDATAH).

To send a message via Single Data Rate (SDR), follow these steps (from DMA or processor):

1. Write the control request to Controller Write Message Control in SDR mode (MWMSG_SDR_CONTROL). This request includes the following items:
   - The address.
   - I2C or I3C.
   - Read or write.
   - The count of bytes to process.
   - How to end (on STOP or ready for Repeated START).
2. Process the data:
   - For write operations, write the rest of the data to Controller Write Message Data in SDR mode (MWMSG_SDR_DATA). Use the DMA trigger (or interrupt) to keep the transmit FIFO full, and the controller stops using the data when it has already consumed the number of bytes configured in the LEN field.
   - For read operations, a DMA trigger (or interrupt) indicates when the RX FIFO is ready to be read out via Controller Read Message in SDR mode (MRMSG_SDR).
3. Select the in-band interrupt (IBI) behavior in MCTRL[IBIRESP].
4. Use END type STOP (MCTRL[REQUEST] = 2) with a zero-length message or by using Controller Control (MCTRL). To exit Message mode with the original message, use END type STOP (MCTRL[REQUEST] = 2) with a zero-length message or by using the MCTRL register.

To send a message via Double Data Rate (DDR), follow these steps (from DMA or processor):

1. Write the control request to Controller Write Message in DDR mode: First Control Word (MWMSG_DDR_CONTROL). This request includes:
   - The count of byte pairs.
   - How to end the message (through HDR exit or to wait for HDR restart).
2. Write the second control data to MWMSG_DDR_CONTROL. This second write includes:
   - The address.
   - Read or write.
   - The 7-bit command value.
3. Process the data:
   - For write operations, write the rest of the data to MWMSG_DDR_DATA. Use the DMA trigger (or interrupt) to keep the transmit FIFO full, and the controller stops using the data when the program count MWMSG_DDR_CONTROL register reaches 0.
   - For read operations, a DMA trigger (or interrupt) indicates when the receive FIFO is ready to be read out.
4. Select in-band interrupt (IBI) behavior in MCTRL[IBIRESP].
5. Use END type exit (MCTRL[REQUEST] = 6) with a zero-length message or by using the MCTRL register to exit DDR Message mode with the original message.

63.3.2.8 Handing off controllership to another target and getting it back

To hand off controllership, the controller can perform either of the following actions:

- Wait for a controller request (CR) (that is, MSTATUS[IBIWON] = 1 interrupt), indicating that the CR is using MSTATUS[IBITYPE].
- Push the request manually.

In either case, the controller sends a GETACCMST request, which is a directed GET informing the target that it is being assigned controllership. If the target accepts this request, it returns its dynamic address in bits 7:1 and the negative parity of its dynamic address in bit 0. A STOP must be issued, and MCONFIG[MSTENA] must be set to 2 (switching to Target mode).

To gain controllership, the target sends a CR using Target Control (SCTRL):

- If MCONFIG[MSTENA] is 2, the GETACCCR CCC is ACKed.
- If MCONFIG[MSTENA] is not 2, the GETACCCR CCC is NACKed.

After controllership is granted, MSTATUS[NOWMASTER] becomes 1. The application must enable MINTSET[NOWMASTER], so the application is interrupted when a controllership transfer occurs.

63.3.2.9 Controller engine flow diagram

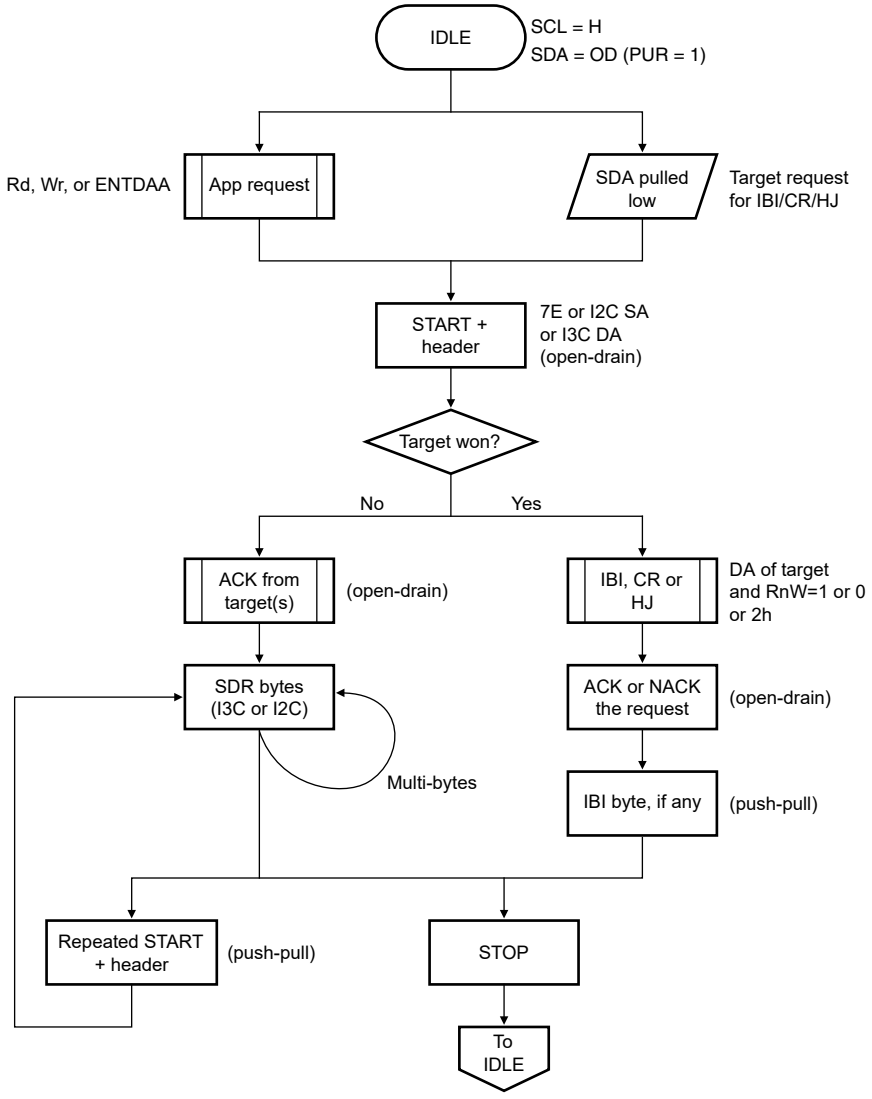


Figure 371. Controller engine flow diagram for SDR (HDR flow progresses from SDR bytes)

63.3.2.10 Target receives data from controller

When the address match is valid for the target dynamic address and the type is W (write from controller), the target ACKs the address. It does so unless some condition prevents it, such as a full buffer. On ACK, the Write state is entered. Once ACKed, it waits for each complete data byte. This process occurs in two steps:

1. Eight bits are clocked in and stored in the next buffer location when in the Write state.
2. On the ninth bit:
    - In I3C mode, the W9TH state (in I3C, the ninth data bit written by the controller is the parity of the preceding eight data bits) is used and the parity is checked. The buffer is marked as complete with or without a parity error.
    - In I2C mode, the W9TH state (in I2C, the ninth data bit written by the controller is an ACK by the target) is used and the data is ACKed and stored.

When the address match is valid for the 7Eh broadcast address, the target acknowledges it. After first data completion, it sets the in_ccc broadcast or direct flag (based on bit 7 of the command). If recognized, it parses the command, setting that bit as well. The recognized commands are for dynamic address work and modes. The CCC and DAA blocks handle the workload afterward. If not supported, then the commands are passed up to the system.

63.3.2.11 Target sends data to controller

When the address match is valid for the target dynamic address, and the type is R (read by controller), the target ACKs the address. The target does so unless there is no data waiting for the read. On ACK, the Read state is entered.

After being ACKed, it emits the data byte at a time, with the ninth bit using the R9TH state. In I2C, the ninth data bit from target to controller is an ACK by the controller. In I3C, this bit allows the target to end a read, and allows the controller to abort a read:

- In I3C mode, the T bit is emitted to allow the device to indicate whether this byte is the last. If it is not the last byte, the controller may terminate the read via the T bit.
- In I2C mode, the ninth bit allows the controller to terminate via NACK, or else it allows the read to continue via ACK. On completion of each byte read, the "done" signal is pulsed to get the next byte.

If the read is terminated by the controller unexpectedly (abort in I3C, NACK before END marked), the application is notified.

63.3.3 Clocking

The controller works on the following primary clocks:

- The system clock, which controls access to the memory-mapped registers.
- A functional clock (FCLK), which is used to generate the SCL clock rate on the I2C or I3C bus.

I3C supports adjusting clock frequency and accuracy via Target Time Control Clock (STCCLOCK).

FCLK is used to check conditions such as bus availability, bus free for IBI, or HJ:

- To determine the Bus Available condition for IBI, the target needs a clock to generate the ~1 µs timing. To support this requirement, a slow clock (CLK_SLOW) is provided to save on power.
- I3C supports a counter field, SCONFIG[BAMATCH], to calculate the Bus Available condition. BAMATCH provides the count of the slow clock . This field counts 1 µs or more to allow an IBI to drive SDA low when the controller is free. The I3C module controls the maximum width and maximum values.

I3C supports a timeout when stalled for too long in a frame. This timeout occurs when:

- The transmit FIFO or receive FIFO is not handled and the bus is stuck in the middle of a message.
- No STOP is issued and the bus is between messages.
- IBI manual is used and no decision is made.

63.3.4 Reset

- Global or system reset fed into the module: Everything is reset by this global or system reset. Release is assumed to be synchronized with the system clock. It is not synchronized with SCL or SDA. These signals must not be active when the reset is released; otherwise, I3C enters Hot-Join mode and remains inactive.
- Software reset (see Self Reset (SELFRESET) for details)

63.3.4.1 Reset detector module

The I3C module supports a target reset feature to provide the flexibility of generating different levels of reset a target device from the controller.

An application can use this detector to wake or reset the device or just a peripheral, as well as do nothing for any specific reset. The normal use is that the controller emits an RSTACT CCC message requiring not to reset when this pattern is emitted just after; broken devices are reset (because they miss the CCC) and devices in deepest sleep state (for example, an unpowered core domain) wake state (because they also miss the CCC).

The supported types of resets:

- Block reset: Resets the I3C peripheral only by software through an interrupt routine.

Reset detector diagram

The figure below shows the functional input, output, and connectivity of the Reset detector module inside the target.

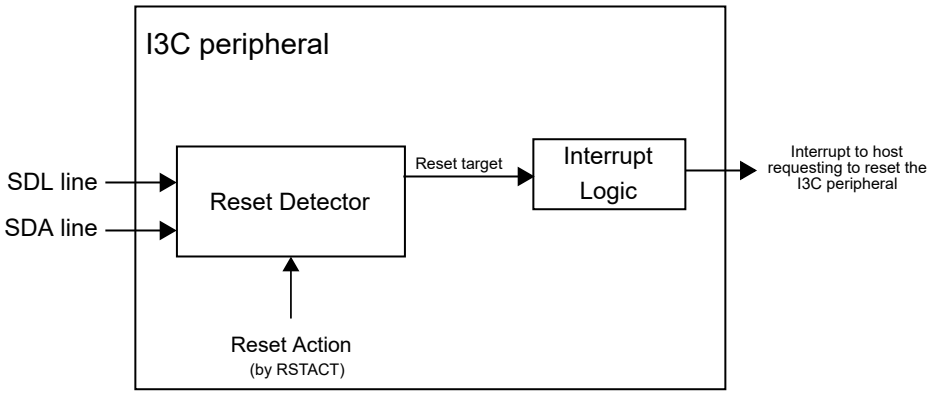


Figure 372. Target Reset Detector inside I3C module and connectivity

The active controller on the I3C bus can reset any or all the targets with a Target Reset pattern. The controller can send the Target Reset pattern with or without a preceding RSTACT CCC.

63.3.4.1.1 Behavior of block reset

The block reset output (“Reset target” in figure) is taken as an interrupt to the host. The host then takes the required action. The primary objective is to Reset the I3C peripheral but the interrupt can be used for other purposes as defined by software.

63.3.5 Interrupts

This section describes all the interrupts (IRQs) that this module generates. All status interrupts are updated in Controller Status (MSTATUS) and Target Status (SSTATUS).

Supported interrupts occur:

- For pending IBI, CR, or Hot-Join that have been sent by a target.
- When a CCC is received and is handled by the module (SSTATUS[CHANDLED]).
- When an error or warning has occurred, such as data underrun, data overrun, parity error, HDR-DDR, or CRC error.

- When a CCC is received that is unhandled by I3C. CCC Mask for Unhandled CCCs (SCCCMASK) allows you to suppress any CCC which is intended to be unhandled.

A wake-up interrupt is also available to bring the I3C module out of low power mode. See I3C from low power mode and wake-up .

63.4 External signals

Table 595. External signals

Signal	Description	Direction
SCL	Serial clock	Input or output
SDA	Serial data	Input or output
PUR	Pull up resistance. There is an internal pullup resistance on SDA, which the I3C controls. If the internal pullup is not enough, PUR can control an external pullup resistance on SDA actively. See the chip-specific I3C information on whether this external pullup is available on your chip.	Output

63.5 Initialization

63.5.1 Controller configuration

Certain configuration is performed when initializing this module. Controller Configuration (MCONFIG) controls the frequencies, duty cycle, optimizations for performance, and other parameters.

Table 596. Configuration for module initialization

Configuration parameter		Description
MSTENA	Controller enable	Determines whether the peripheral starts in Controller mode (main controller in I3C terms) or starts in Target mode (and switches to Controller mode later).
HKEEP	High-Keeper	Determines how the high-keeper (weak pullup) is implemented, depending on the device capabilities.
PPBAUD	Push-Pull baud	Sets the push-pull frequency as a divider from the FCLK (alternative clock fed to the peripheral). This frequency sets the SCL half-clock period baseline (used at least for the high time of SCL). The formula for SCL in Push-Pull mode using PPBAUD and PPLOW is: SCL frequency (in MHz) = FCLK ÷ ((2 + 2 × PPBAUD) + PPLOW) If PPLOW is 0, then SCL high = SCL low (in ns) = (1 + PPBAUD) × 1000 ÷ FCLK (in MHz) Examples: • If FCLK = 24 MHz, then PPBAUD = 0 yields 12 MHz (42.67 ns per half period) • If FCLK = 50 MHz, then PPBAUD = 1 yields 12.5 MHz (20 + 20 = 40 ns per half period)
PPLOW	Push-Pull low	Changes the duty cycle for Push-Pull. It indicates how many more FCLK cycles to use for low. The formula for SCL in Push-Pull mode using PPBAUD and PPLOW is: SCL frequency (in MHz) = FCLK ÷ ((2 + 2 × PPBAUD) + PPLOW) If PPLOW is a nonzero value, then:

Table continues on the next page...

Table 596. Configuration for module initialization (continued)

Configuration parameter		Description
		• SCL high (in ns) = (1 + PPBAUD) × 1000 ÷ FCLK (in MHz) • SCL low (in ns) = (1 + PPBAUD + PPLOW) × 1000 ÷ FCLK (in MHz) For example, when FCLK is 50 MHz, PPBAUD is 1, and PPLOW is 1, then the periods are: • (1 + 1) × 1000 ÷ 50 = 20 + 20 = 40 ns high • (1 + 1 + 1) × 1000 ÷ 50 = 20 + 20 + 20 = 60 ns low This timing is equivalent to 10 MHz SCL, but this timing maintains the 40 ns high needed; therefore, I2C devices do not see the high periods. **NOTE** The PPLOW value does not have any impact on Open-Drain mode and I2C mode SCL rate calculations.
ODBAUD	Open-Drain baud	Determines the number of PPBAUD periods to make up one I3C Open-Drain half-clock baseline. If ODHPP is 0, the ODBAUD half-clock time period = (ODBAUD + 1) × PPBAUD high period. If ODHPP is 0, the formula for SCL in Open-Drain mode is: SCL (in MHz) = FCLK ÷ (2 + 2 × PPBAUD) × (ODBAUD + 1) Target to get open-drain half-clock period is 200 ns. For example: • If PPBAUD yields 12.5 MHz (40 ns per PPBAUD period), then ODBAUD = 4 can be used to get 200 ns. See ODHPP for details on short high and long low. • If PPBAUD = 1, FCLK = 50 MHz, and ODBAUD = 4, then open-drain SCL = 50 ÷ (2 + 2 × 1) × 5 = 2.5 MHz.
ODHPP	Open-Drain High Push-Pull	Optional field that allows the I3C open drain to be long low and short high. The high period of SCL is the PPBAUD period. This period leaves enough time for the pullup resistor to pull the SDA high when SCL is low. It is also quick when SCL is high and there are no changes happening. ODBAUD low half-clock time period = (ODBAUD + 1) × PPBAUD high period. ODBAUD high half-clock time period = PPBAUD high period If ODHPP is 1, the formula for SCL in Open-Drain mode is: SCL (in MHz) = FCLK ÷ (1 + PPBAUD) × (ODBAUD + 1) + (1 + PBAUD) For example: • If PPBAUD produces 12.5 MHz (40 ns per PPBAUD period), then ODBAUD = 4 and ODHPP = 1. These settings provide a high period of 40 ns and a low period of 200 ns. • If PPBAUD = 1, FCLK = 50 MHz, and ODBAUD = 4, then open-drain SCL = 50 ÷ (1 + 1) × 5 + (1 + 1) = 4.16 MHz.
I2CBAUD	I2C Baud	Indicates the number of ODBAUD periods that are required to communicate with I2C devices (MCTRL[TYPE] = 2 or MWMSG_SDR_CONTROL[I2C] = 1).

Table continues on the next page...

Table 596. Configuration for module initialization (continued)

Configuration parameter		Description
		For example, if ODBAUD gives 200 ns, and the goal is Fm+ (Fast Mode, 1 MHz), then the sum must be 1 µs. I2CBAUD acts differently for odd and even values. For example: • If I2CBAUD = 3, it gives three ODBAUD periods low and two ODBAUD periods high. • If I2CBAUD = 4, it gives three ODBAUD periods for low and three ODBAUD periods for high. • Also, if I2CBAUD = 3, this yields 200 × 3 = 600 ns and 200 × 2 = 400 ns, with the sum 600 + 400 = 1000 ns = 1 µs.
SKEW	Skew	The normal SDA skew from SCL is handled by using the time for the SCL to reach its pad and return. This time is normally 2 ns to 5 ns (or sometimes more). If the skew is too fast, then add more delay. The skew allows specifying the number of FCLK to insert.

Additional optimizations:

- Use MIBIRULES[MSB0] to obtain faster start header times. If the controller application assigns all I3C dynamic addresses to be less than 40h (it does not have MSB set), you can write 1 to MIBIRULES[MSB0]. When the controller emits 7Eh (broadcast) and a target does not drive the first bit low, the rest of the header can be at push-pull speeds. This speed is two times faster or more, depending on optimizations.
- Auto-emit 7Eh speeds up the frame when used with MIBIRULES[MSB0]. It allows the processor to sleep when the frame starts automatically (in response to a target).

63.5.2 Interrupt service flow

Set up interrupts for Controller Status (MSTATUS) in Controller Interrupt Set (MINTSET).

Set up interrupts for Target Status (SSTATUS) in Target Interrupt Set (SINTSET).

63.6 Application information

63.6.1 I2C configuration to meet timing requirements for Fm and Fm+ modes

I2C Fm and I2C Fm+ modes are supported according to the I3C standard specifications.

Table 597 shows the configuration for Fm mode, where frequency is close to 400 kHz but timing requirement is not met (T_{SU_STA} is not met) because the actual value is 600 ns. Rest of the requirements are met per the I2C specifications in the chip data sheet.

Table 598 shows the configuration for Fm mode, where all timing requirements are met.

Table 597. Configuration for Fm mode with not all timing requirements met

FCLK	PPBAUD	ODBAUD	I2CBAUD	SCL frequency	Timing requirement status
24 MHz	0	4	11	370 kHz	T_{SU_STA} - 353 ns
48 MHz	1	4	11	370 kHz	T_{SU_STA} - 353 ns
133/134 MHz	5	4	11/12	412 kHz / 382 kHz	T_{SU_STA} - 267 ns and T_{SU_STO} - 529 ns
160 MHz	6	4	10	380 kHz	T_{SU_STA} - 312 ns

Table 598. Configuration for Fm mode with all timing requirements met

FCLK	PPBAUD	ODBAUD	I2CBAUD	SCL frequency	Timing requirement status
24 MHz	1	2	9	363 kHz	All requirements met per the I2C specifications in the chip data sheet
48 MHz	3	2	9	363 kHz	All requirements met per the I2C specifications in the chip data sheet
133/134 MHz	11	2	9/8	338 kHz / 372 kHz	All requirements met per the I2C specifications in the chip data sheet
160 MHz	13	2	9/8	345 kHz / 422 kHz	All requirements met per the I2C specifications in the chip data sheet

Table 599 shows the configuration for Fm+ mode, where the frequency is close to 1 MHz. In this case, the T_{SU_STA} timing requirement is not met because the actual value is 260 ns. Rest of the requirements are met per the I2C specifications in the chip data sheet.

Table 600 shows the configuration for Fm+ mode, where all timing requirements are met. To meet all the timing specifications, frequency may differ from 1 MHz. Use the configurations listed in Table 600.

Table 599. Configuration for Fm+ mode with not all timing requirements met

FCLK	PPBAUD	ODBAUD	I2CBAUD	SCL frequency	Timing requirement status
24 MHz	0	4	3	960 kHz	T_{SU_STA} - 228 ns
48 MHz	1	4	3	960 kHz	T_{SU_STA} - 228 ns
133/134 MHz	4	4	3	1.087 MHz	T_{SU_STA} - 156 ns
160 MHz	6	4	3	912 kHz	T_{SU_STA} - 181 ns

Table 600. Configuration for Fm+ mode with all timing requirements met

FCLK	PPBAUD	ODBAUD	I2CBAUD	SCL frequency	Timing requirement status
24 MHz	0	3	6	749 kHz	All requirements met per the I2C specifications in the chip data sheet
48 MHz	1	3	6	749 kHz	All requirements met per the I2C specifications in the chip data sheet
133/134 MHz	4	3	6	839 kHz	All requirements met per the I2C specifications in the chip data sheet
160 MHz	6	3	6	713 kHz	All requirements met per the I2C specifications in the chip data sheet

NOTE

The controller module provides an option to slow down the clock for standard speed mode to support legacy I2C standard targets as well (although this option may not necessarily meet the exact timing requirement of the standard mode according to the standard I2C timing specification).

For the timing requirements under consideration, see the "I3C timing requirements when communicating with I2C legacy devices" table in the MIPI I3C v1.1.1 specification available on www.mipi.org.

63.7 I3C register descriptions

Writing to a read-only (RO) register, except to Controller Interrupt Mask (MINTMASKED), causes bus errors. This module does not verify whether programmed values in the registers are correct. You must ensure that valid programmed values are written.

However, the peripheral ignores writes to RO registers and returns 0 for reads from WO or nonexistent registers. It requires all 32-bit registers to be read and written as 32-bit and aligned to 32 bits. The peripheral uses the advanced peripheral bus (APB: AMBA 3 APB Protocol Specification v1.0); therefore, the peripheral does not know whether partial read or write operations (less than 32 bits) are used.

63.7.1 I3C memory map

I3C1 base address: 4433_0000h

I3C2 base address: 4252_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Controller Configuration (MCONFIG)	32	RW	0000_0000
4	Target Configuration (SCONFIG)	32	RW	0017_0000
8	Target Status (SSTATUS)	32	RW	0000_1400
C	Target Control (SCTRL)	32	RW	0000_0000
10	Target Interrupt Set (SINTSET)	32	RW	0008_0000
14	Target Interrupt Clear (SINTCLR)	32	RW	See section
18	Target Interrupt Mask (SINTMASKED)	32	R	0000_0000
1C	Target Errors and Warnings (SERRWARN)	32	RW	0000_0000
20	Target DMA Control (SDMACTRL)	32	RW	0000_0010
2C	Target Data Control (SDATACTRL)	32	RW	8000_0030
30	Target Write Data Byte (SWDATAB)	32	W	See section
34	Target Write Data Byte End (SWDATABE)	32	W	See section
38	Target Write Data Halfword (SWDATAH)	32	W	See section
3C	Target Write Data Halfword End (SWDATAHE)	32	W	See section
40	Target Read Data Byte (SRDATAB)	32	R	0000_0000
48	Controller Read Data Halfword (MRDATAH)	32	R	0000_0000
48	Target Read Data Halfword (SRDATAH)	32	R	See section
54	Target Write Data Byte (SWDATAB1)	32	W	0000_0000
5C	Target Capabilities 2 (SCAPABILITIES2)	32	R	0003_0331
60	Target Capabilities (SCAPABILITIES)	32	R	FC3F_FE70
68	Target Maximum Limits (SMAXLIMITS)	32	RW	0000_0000
6C	Target ID Part Number (SIDPARTNO)	32	RW	4D58_3933
70	Target ID Extension (SIDEXT)	32	RW	0066_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
74	Target Vendor ID (SVENDORID)	32	RW	0000_0124
78	Target Time Control Clock (STCCLOCK)	32	RW	0000_3014
7C	Target Message Map Address (SMSGMAPADDR)	32	R	0000_0000
80	Controller Extended Configuration (MCONFIG_EXT)	32	RW	0000_0000
84	Controller Control (MCTRL)	32	RW	0000_0000
88	Controller Status (MSTATUS)	32	RW	0000_1000
8C	Controller In-band Interrupt Registry and Rules (MIBIRULES)	32	RW	0000_0000
90	Controller Interrupt Set (MINTSET)	32	RW	0000_0000
94	Controller Interrupt Clear (MINTCLR)	32	RW	See section
98	Controller Interrupt Mask (MINTMASKED)	32	R	0000_0000
9C	Controller Errors and Warnings (MERRWARN)	32	RW	0000_0000
A0	Controller DMA Control (MDMACTRL)	32	RW	0000_0010
AC	Controller Data Control (MDATACTRL)	32	RW	8000_0030
B0	Controller Write Data Byte (MWDATAB)	32	W	See section
B4	Controller Write Data Byte End (MWDATABE)	32	W	See section
B8	Controller Write Data Halfword (MWDATAH)	32	W	See section
BC	Controller Write Data Halfword End (MWDATAHE)	32	W	See section
C0	Controller Read Data Byte (MRDATAB)	32	R	0000_0000
CC	Controller Write Byte Data 1 (to Bus) (MWDATAB1)	32	W	0000_0000
D0	Controller Write Message Control in SDR mode (MWMSG_SDR_CONTROL)	32	W	See section
D0	Controller Write Message Data in SDR mode (MWMSG_SDR_DATA)	32	W	0000_0000
D4	Controller Read Message in SDR mode (MRMSG_SDR)	32	R	0000_0000
D8	Controller Write Message in DDR mode: First Control Word (MWMSG_DDR_CONTROL)	32	W	See section
D8	Controller Write Message in DDR Mode Control 2 (MWMSG_DDR_CONTROL2)	32	W	See section
D8	Controller Write Message Data in DDR mode (MWMSG_DDR_DATA)	32	W	0000_0000
DC	Controller Read Message in DDR mode (MRMSG_DDR)	32	R	0000_0000
E4	Controller Dynamic Address (MDYNADDR)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
100	Timing Rules for Target Reset Recovery (SRSTACTTIME)	32	RW	0000_0000
10C	CCC Mask for Unhandled CCCs (SCCCMASK)	32	RW	0000_007F
110	Target Errors and Warnings Mask (SERRWARNMASK)	32	RW	0000_0F1F
11C	Map Feature Control 0 (SMAPCTRL0)	32	R	0000_0000
120	Map Feature Control 1 (SMAPCTRL1)	32	RW	0000_0000
140	Extended IBI Data 1 (IBIEXT1)	32	RW	0000_0070
144	Extended IBI Data 2 (IBIEXT2)	32	RW	0000_0000
FF0	Self Reset (SELFRESET)	32	RW	0000_0000

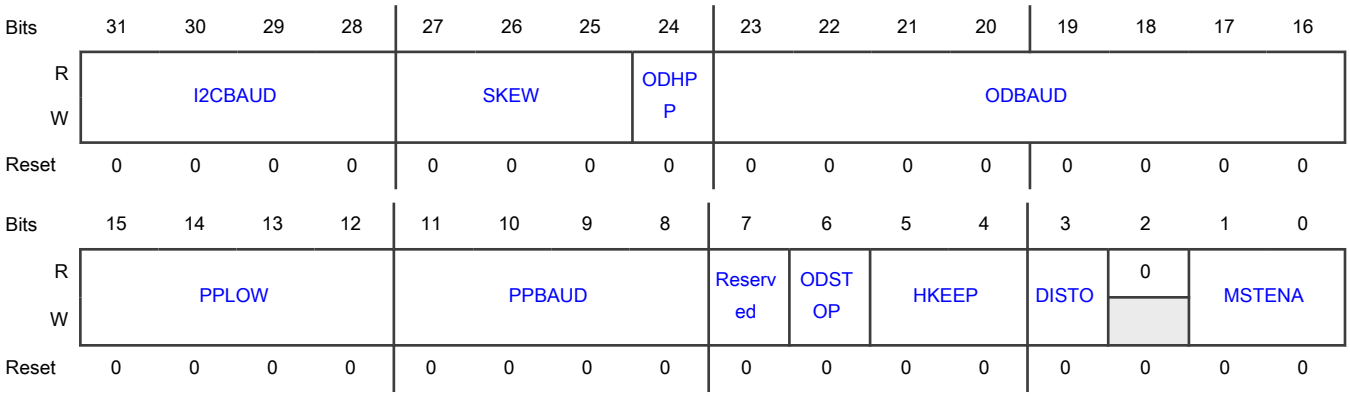
63.7.2 Controller Configuration (MCONFIG)

Controls all controller states when the controller operation is enabled. You must not change this register during an active transaction.

Offset

Register	Offset
MCONFIG	0h

Diagram



Fields

Field	Description
31-28 I2CBAUD	I2C Baud Rate Specifies the I2C low and high times in ODBAUD counts.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	See I2CBAUD in Controller configuration.
27-25 SKEW	Skew Specifies the number of FCLK counts for an SDA change after SCL for I3C push-pull. This skew is in addition to the roundtrip of the SCL line from the pad back to the module (6 ns or more): • SKEW is normally not needed, so assign SKEW = 0. • SKEW is only used if SDA is not naturally skewing from an SCL change. • I2C automatically skews SDA (but not PUR) to match I2C rules. Use the following values: • In I3C Controller mode, use SKEW = 0.
24 ODHPP	Open-drain High Push-Pull Enables ODHPP. If you write 0 to this field, open-drain SCL high half-clock period is the same as the open-drain low SCL half period. If you write 1 to this field, open-drain high SCL half-lock period is one PPBAUD count for I3C messages. This setting is faster and works for I3C devices. Any legacy I2C devices on the bus do not see the SCL high level at all (less than the spike filter period). For open-drain timing, check upon the first 7Eh broadcast allowing I3C peripherals acting as I2C legacy devices to turn off their I2C 50-ns spike filters. See Sending a CCC to I3C targets for more information. 0 - Disable 1 - Enable
23-16 ODBAUD	Open-drain Baud Rate Specifies the open-drain baud rate. ODBAUD is configured in terms of number of PPBAUD counts - 1. This field must not be 0; this setting is not the same as push-pull. See Controller configuration for more information. The open-drain baud rate is usually 200 ns (see I2CBAUD for I2C counts). When used with MCONFIG[ODHPP], this setting produces 250 ns per clock in I3C. In this case, if MCONFIG[PPBAUD] provides 12 MHz, then one PPBAUD count is half of 12 MHz, or 41.67 ns. To obtain a rate around 200 ns, use a value of 5 - 1 = 4 for ODBAUD.
15-12 PPLOW	Push-Pull Low Acts as an adder for push-pull low to create a duty cycle with a longer low period, with up to 15 more FCLK cycles low than high. PPLOW = 0 produces a 50/50 duty cycle.
11-8 PPBAUD	Push-Pull Baud Rate Specifies the push-pull baud rate. The number of FCLK counts makes each push-pull low and normally high period. PPBAUD = 0 when run at 1/2 input FCLK speed. For example, a 24 MHz FCLK produces a 12 MHz SCL because each FCLK is SCL low or SCL high.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	MCONFIG[PPLOW] has an effect on the duty cycle. For example, 24 MHz with 50/50 duty cycle is 12 MHz. However, when PPLOW adds three more low beats, the push-pull baud rate becomes 4.8 MHz (from 24 MHz or 5 beats).
7 —	Reserved
6 ODSTOP	Open-drain Stop Enables open-drain stop. If you write 0 to this field, open-drain stop is disabled. ODSTOP must be disabled when sending an HDR exit pattern. If you write 1 to this field, open-drain stop is enabled. STOP condition is emitted at open-drain speeds even for I3C messages. In legacy devices, this feature can ensure that the legacy devices see the STOP condition. 0 - Disable 1 - Enable
5-4 HKEEP	High-Keeper Indicates how High-Keeper is supported. If this field is 00b, use pull-up resistor (PUR). No separate pin_HK_SDA or pin_HK_SCL is used. Only pin_PUR_oena is used for SDA. The pin_SCL_oena pin is held high in this mode, SCL clock is push-pull, and pin_SCL_out toggles, which is the actual clock. If this field is 01b, High-Keeper controls are in place; pin_HK_SDA or pin_HK_SCL (High-Keeper) controls are used. SCL may use an HK or that signal (pin_HK_SCL) may only OR into pin_SCL_oena. The pin_HK_SDA pin is used as well for SDA high keeper, along with pin_PUR_oena. Use HKEEP = 10b for I2C modes. Use HKEEP = 00b or 01b for I3C modes. **NOTE** Your chip may not support any or all High-Keeper methods. You must check the availability of High-Keeper methods. 00 - None 01 - WIRED_IN 10 - PASSIVE_SDA (I2C mode, no clock stretches mode). SDA is open drain, and will be Hi-Z (high-impedance) for Bus Free (IDLE) and hold. The pin pin_HK_SDA is not used in this option, only a combination of SDA_oena and pin_PUR_oena are used. SCL is push-pull. The pin pin_HK_SCL is not used in this option, only pin_SCL_oena is used. Since the controller does not support multi-controller mode, there is no need to have open-drain SCL for I2C. For a single controller application, the controller's SCL output can be a push-pull driver design if there are no devices on the bus which would stretch the clock. 11 - Reserved
3 DISTO	Disable Timeout Disables the timeout that produces application errors.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If the controller is left in a state other than Stopped for more than 100 µs (because 10 kHz is the slowest allowed I3C speed), the timeout sends a MERRWARN interrupt. To prevent the MERRWARN interrupt during development or testing, write 1 to DISTO to disable the timeout. In systems that support timeouts, timeout is disabled automatically during debug. To obtain a clock configuration with 100 µs timeout, see SCONFIG[BAMATCH]. 0 - Enabled 1 - Disabled
2 —	Reserved
1-0 MSTENA	Controller Enable Indicates whether the controller is enabled and the states it can use. If this field is 0b, the controller is disabled. The module can only use Target mode. If this field is 1b, the controller is enabled. When used upon start-up, this module is the main controller by default. I3C controls the bus unless the controller is handed off. When this happens, MSTENA must become 2 so that it has the capability to become the controller again. Performing the handoff means emitting the GETACCMST CCC command. If the command is accepted, I3C emits a STOP condition and sets this field to 2 (or 0). If this field is 10b, I3C is controller-capable, but is operating as a target now. When used from the start, I3C starts as a target, but is prepared to switch to Controller mode. To switch to this mode, the target emits a controller request (CR) or receives a GETACCMST CCC command and accepts it (to switch on the STOP condition). 00 - CONTROLLER_OFF 01 - CONTROLLER_ON 10 - CONTROLLER_CAPABLE 11 - Reserved

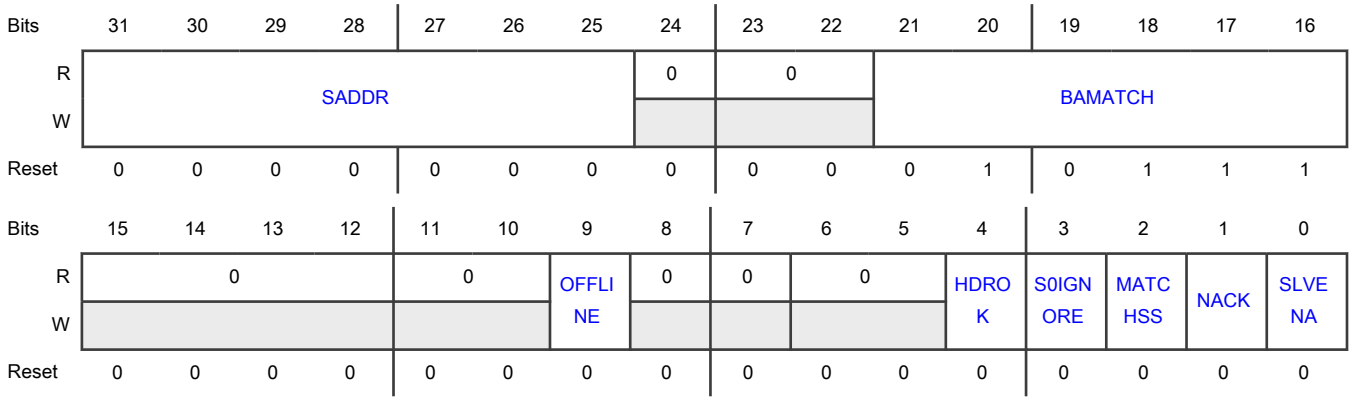
63.7.3 Target Configuration (SCONFIG)

Contains fields that must be configured before the module is activated.

Offset

Register	Offset
SCONFIG	4h

Diagram



Fields

Field	Description
31-25 SADDR	Static Address Sets the I2C 7-bit static address, which otherwise must be 0.
24 —	Reserved
23-22 —	Reserved
21-16 BAMATCH	Bus Available Match Specifies the bus available condition match value for the current slow clock. BAMATCH provides the count of the slow clock to count out 1 µs (or more). The BAMATCH should be set as (slowClk(MHz) - 1) to generate a 1 µs clock cycle. Use BAMATCH = 1 to generate 1 µs clock cycle if slow clock is 1 MHz. A value of 0 would not give a correct match indication. The default value applies to the default slow clock setting of the chip. If the I3C slow clock is updated to a new value, BAMATCH must be updated, too. When I3C operates in Controller mode, BAMATCH is used to generate a 1 µs clock cycle basic time to count 100 µs timeout. When I3C operates in Target mode, BAMATCH allows the IBI to drive SDA low when the controller is not doing so.
15-12 —	Reserved
11-10 —	Reserved
9	Offline

Table continues on the next page...

Table continued from the previous page...

Field	Description
OFFLINE	Enables wait to ensure that the bus is not in HDR mode. If this field is 1 when SCONFIG[SLVENA] is 1, then I3C waits for either 60 µs of bus quiet or an HDR exit pattern. This waiting ensures that the bus is not in HDR mode, and so can safely monitor the next activity in Single Data Rate (SDR) mode. 0 - Disable 1 - Enable
8 —	Reserved
7 —	Reserved
6-5 —	Reserved
4 HDROK	HDR OK Enables HDR OK. If you write 1 to this field, it allows HDR-DDR and/or HDR-BT messaging, if available, by writing 1 to the corresponding SIDEXT[BCR] field to indicate HDR is available, and the corresponding GETCAPS field for DDR and/or BT bit permitting use. This is a deprecated field; use the SIDEXT[BCR] field instead. 0 - Disable HDR OK 1 - Enable HDR OK
3 S0IGNORE	Ignore TE0 or TE1 Errors Ignores TE0 or TE1 errors. If you write 1 to this field, the target does not detect TE0 or TE1 errors, so it does not lock up waiting on an exit pattern. You must not use this setting when the bus does not use HDR mode. 0 - Do not ignore TE0 or TE1 errors 1 - Ignore TE0 or TE1 errors
2 MATCHSS	Match Start or Stop Enables matched START, that is Repeated START, or STOP condition. If you write 1 to this field, SINTSET[START] and SINTSET[STOP] become 1 only when SSTATUS[MATCHED] is 1. This setting allows the START and STOP fields to be used to detect the end of a message to or from this target. **NOTE** SSTATUS[START] does not assert for START of transaction if SCONFIG[MATCHSS] = 1. SSTATUS[START] only asserts for Repeated Start within that transaction, because SSTATUS[START] only asserts after SSTATUS[MATCHED] occurs, which is after START of transaction. 0 - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable
1 NACK	Not Acknowledge Controls the ACK or NACK capability. If you write 1 to this field, the target rejects all requests, except for a CCC broadcast. NACK = 1 must be used with caution because the controller may decide that the target is missing, if NACK is overused. 0 - Always disable NACK mode 1 - Always enable NACK mode (works normally)
0 SLVENA	Target Enable Enables the target. If you write 0 to this field, the target ignores the I2C or I3C bus. If you write 1 to this field, the target can operate on the I2C or I3C bus. You must not write 1 to this field before registers such as Target Configuration (SCONFIG), Target ID Part Number (SIDPARTNO), Target ID Extension (SIDEXT), and others are configured because these registers affect the data to and from the controller. Target enable is configured just once before the bus comes up. If target enable is used at other times, SCAPABILITIES[IBI_MR_HJ] must be 1 before writing 1 to SLVENA so that the device does not see a START or STOP condition incorrectly. 0 - Disable 1 - Enable

63.7.4 Target Status (SSTATUS)

Indicates the sticky status for interrupts, states, and modes related to the I3C bus. Not all fields of the register are used if the module only acts as a target. The fields are divided into current activity, interrupt maskable actions, and then states and modes on the bus.

Offset

Register	Offset
SSTATUS	8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TIMECTRL		ACTSTATE		HJDIS	0	MRDIS	IBIDIS	0		EVDET		SLVRST	EVENT	CHANDLED	HDRMATCH
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERRWARN	CCC	DACHG	TXNOTFU...	RX_PEND	STOP	MATCHED	START	0	STHDR	STDAA	STREQWR	STREQRD	STCCCH	STMSG	STNOTST...
W		W1C	W1C			W1C	W1C	W1C								
Reset	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-30 TIMECTRL	Time Control Indicates whether time control is enabled, and in which mode. 00 - NO_TIME_CONTROL (no time control is enabled) 01 - SYNC_MODE (Synchronous mode is enabled) 10 - ASYNC_MODE (Asynchronous standard mode (0 or 1) is enabled) 11 - BOTHSYNCASYNC (both Synchronous and Asynchronous modes are enabled)
29-28 ACTSTATE	Activity State from Common Command Codes (CCC) Indicates the activity state from CCC. 00 - NO_LATENCY (normal bus operations) 01 - LATENCY_1MS (1 ms of latency) 10 - LATENCY_100MS (100 ms of latency) 11 - LATENCY_10S (10 seconds of latency)
27 HJDIS	Hot-Join Disabled Indicates whether hot-join is disabled. When hot-join is disabled, CTRL requests are not responded to. 0 - Enabled 1 - Disabled
26 —	Reserved
25 MRDIS	Controller Requests Disable Indicates whether controller requests are disabled. When controller requests are disabled, CTRL requests are not responded to. 0 - Enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Disabled
24 IBIDIS	In-Band Interrupts Disable Indicates whether in-band interrupts are disabled. When in-band interrupts are disabled, CTRL requests are not responded to. 0 - Enabled 1 - Disabled
23-22 —	Reserved
21-20 EVDET	Event Details Indicates current details of the last (EVENT = 1) or pending event. 00 - NONE (no event or no pending event) 01 - NO_REQUEST (request is not sent yet; either there is no START condition yet, or is waiting for Bus-Available or Bus-Idle (HJ)) 10 - NACKed (not acknowledged, request sent and rejected); I3C tries again 11 - ACKed (acknowledged; request sent and accepted), so done (unless the time control data is still being sent)
19 SLVRST	Target Reset Flag Helps you perform follow-up tasks such as reinitialization (target reset to peripheral (not whole chip)).
18 EVENT	Event Flag Indicates, for a target, whether a pending in-band interrupt (IBI), controller request (CR), or hot-join (HJ) has been sent as requested. See the upper status register fields for details. This field becomes 1 only when acknowledged by a controller. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No event occurred 1 - IBI, CR, or HJ occurred When writing 0 - No effect 1 - Clear the flag
17 CHANDLED	Common Command Code Handled Flag Indicates whether an HDRMATCH CCC is being handled by I3C. This field is for notification only, but it may result in updates to this register.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - CCC handling not in progress 1 - CCC handling in progress When writing 0 - No effect 1 - Clear the flag
16 HDRMATCH	High Data Rate Command Match Flag Indicates whether the HDR command matched the I3C dynamic address of this device. The HDR command is available as the first byte, with RXPEND = 1. The MSB of the command byte indicates whether it is a read or a write command. If the HDR command is a read, and there are to-bus bytes waiting, then the command is ACKed and the data is sent back. Otherwise, the HDR command is NACKed. **NOTE** When this field is 1, you must check SSTATUS[ERRWARN] because the HPAR error may occur after signaling this HDR command. The parity occurs after the destination address and command. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Did not match 1 - Matched the I3C dynamic address When writing 0 - No effect 1 - Clear the flag
15 ERRWARN	Error Warning Indicates that an error or warning has occurred, such as data underrun, data overrun, parity error, HDR-DDR CRC error, or other error or warning condition (see Target Errors and Warnings (SERRWARN) for more information).
14 CCC	Common Command Code Flag Indicates whether a CCC has been received, and is not handled by I3C. There are two types of common command codes: • Broadcast CCC, which corresponds with RXPEND, and the first byte is the CCC (command). • Direct CCC, which may never be directed to this device. If the direct CCC is directed to this device, then TXSEND or RXPEND are triggered, and RXPEND contains the command.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - CCC not received 1 - CCC received When writing 0 - No effect 1 - Clear the flag
13 DACHG	Dynamic Address Change Flag Indicates an occurrence of dynamic address (DA) change. Actual DA can be seen in the DYNADDR register. If this field is 1, DA change is detected. The target DA has been assigned, reassigned, or reset (lost) and is now in the state of being valid or none. This field is also used when the MAP auto feature is configured, changing one or more MAP items. See DYNADDR and MAPCTRL*n* for more information. DYNAADDR for the main DA (0) indicates whether the last change was because of Auto-MAP. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No DA change detected 1 - DA change detected When writing 0 - No effect 1 - Clear the flag
12 TXNOTFULL	Transmit Buffer Not Full Indicates that the transmit buffer is not full. If DMA is enabled for transmitting, then it is also signaled to provide more data. To-bus buffer or FIFO can accept more data to be transmitted. For all but external FIFOs, this process uses SDATACTRL[TXTRIG], which defaults to "not full". 0 - Transmit buffer full 1 - Transmit buffer not full
11 RX_PEND	Received Message Pending Indicates whether a received message is pending. The field indicates when a message from the controller that is not being handled by I3C (not a CCC message) is received. I3C processes such messages internally. For all but external FIFOs, this process uses SDATACTRL[RXTRIG], which defaults to "not empty". If DMA is enabled for receiving, then DMA is signaled as well. This field automatically becomes 0 if data is read (from FIFO and non-FIFO sources).

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - No received message pending 1 - Received message pending
10 STOP	Stop Flag Indicates whether the Stopped state is detected. The STNOTSTOP state also indicates when the module is in Stop mode. A fast STOP and START combination may not trigger the Stop status. In that case, the Start status is always set. If this field is 1, it indicates that the Stop state was present on the bus since the bus was last cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No Stopped state detected 1 - Stopped state detected When writing 0 - No effect 1 - Clear the flag
9 MATCHED	Matched Flag Indicates whether an incoming header matched the I3C dynamic or I2C static address (if any) of this device since the bus was last cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Header not matched 1 - Header matched When writing 0 - No effect 1 - Clear the flag
8 START	Start Flag Indicates whether a START or Repeated START was detected after this flag was last cleared. This flag is not usually needed, but can be used for wake events. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Not detected 1 - Detected When writing 0 - No effect 1 - Clear the flag
7 —	Reserved
6 STHDR	Status High Data Rate Indicates whether the I3C bus is in HDR-DDR mode, regardless of whether HDR mode is supported by this module, and regardless of whether the message is intended for this module or some other module. 0 - I3C bus not in HDR-DDR mode 1 - I3C bus in HDR-DDR mode
5 STDAA	Status Dynamic Address Assignment Indicates whether the I3C bus is in Enter Dynamic Address Assignment (ENTDAA) mode, regardless of whether this bus target has a dynamic address. 0 - Not in ENTDAA mode 1 - In ENTDAA mode
4 STREQWR	Status Request Write Indicates whether the REQ in process is SDR write data from the controller to this bus target (or all targets), but not in ENTDAA mode. See status high data rate (STHDR) for double data rate (DDR) handling. 0 - Not an SDR write 1 - SDR write data from the controller, but not in ENTDAA mode
3 STREQRD	Status Request Read Indicates whether the REQ in process is an SDR read from this target. See status high data rate (STHDR) for double data rate (DDR) handling. 0 - Not an SDR read 1 - SDR read from this target or an IBI is being pushed out
2 STCCCH	Status Common Command Code Handler Indicates whether a CCC message is being handled automatically. 0 - No CCC message handled 1 - Handled automatically

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 STMSG	Status Message Indicates whether the bus target is busy (listening to the bus traffic or responding). If STNOSTOP = 1, STMSG is 0 when a nonmatching address is seen, until the next Repeated START or STOP condition occurs. 0 - Idle 1 - Busy
0 STNOTSTOP	Status not Stop Indicates the status of the bus. If this field is 0, I3C is in a STOP condition. If this field is 1, the bus is busy (has activity). Other fields of this register may also be set when busy. STNOTSTOP can also become 1 after a TE0 or TE1 error, when I3C is waiting for an exit pattern. 0 - In STOP condition 1 - Busy

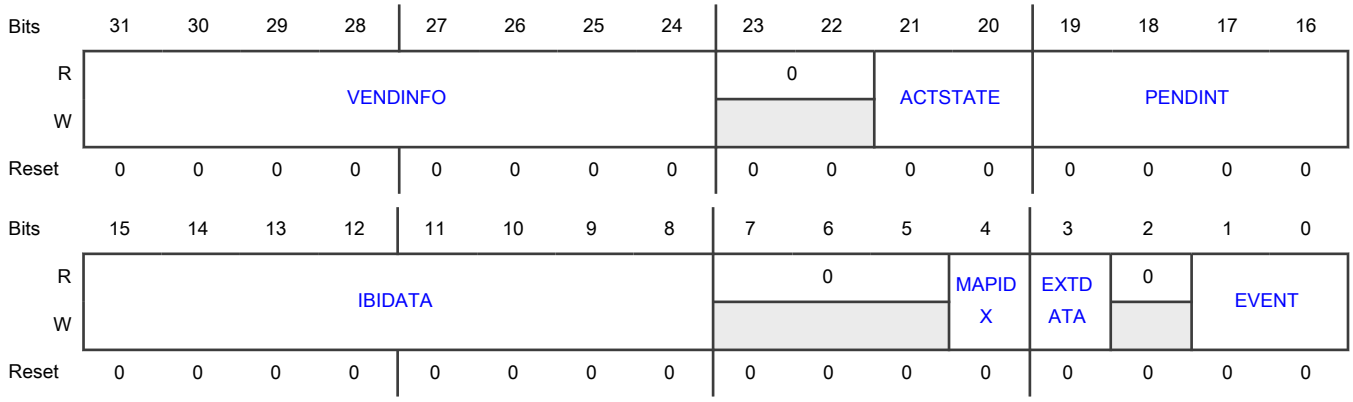
63.7.5 Target Control (SCTRL)

Contains controls for the active use of the I3C bus (for example, event generation such as interrupts to the controller). Only if I3C is configured to support various special operations for the target, this register is used to activate those operations. These events include IBI and GETSTATUS fields (except the protocol error, which is automatically set).

Offset

Register	Offset
SCTRL	Ch

Diagram



Fields

Field	Description
31-24 VENDINFO	Vendor Information Controls vendor information that the GETSTATUS CCC returns. You must set this field to the Vendor Reserved field that the GETSTATUS CCC returns. The application must maintain the vendor information because the controller reads this field. If this field is not configured, then the GETSTATUS field always returns 0.
23-22 —	Reserved
21-20 ACTSTATE	Activity State of Target Controls the activity state of the target. You must set this field to the activity state of the target that the GETSTATUS CCC command returns as activity mode. The application must maintain the activity state because the controller reads this field. If you do not configure the activity state, then the GETSTATUS command always returns 0.
19-16 PENDINT	Pending Interrupt Specifies whether an IBI interrupt is pending. You must set this field to the pending interrupt that the GETSTATUS CCC command returns. The application must maintain the pending interrupt because the controller reads this field. If PENDINT = 0: • The GETSTATUS command returns 1 if an IBI interrupt is pending. • The GETSTATUS field returns 0 if an IBI interrupt is not pending.
15-8 IBIDATA	In-Band Interrupt Data Controls the data byte accompanying the IBI, if the module is enabled for IBI. If SCTRL[IBIDATA] is enabled, then IBI is required.
7-5 —	Reserved
4 MAPIDX	Map Index Controls the index of the dynamic address of the IBI. This index is 0 for the main or base dynamic address. This field is used only if mapping is enabled. See Controller Dynamic Address (MDYNADDR) for more information.
3 EXTDATA	Extended Data Enables extended data. After IBIDATA is emitted, extended data is acquired from IBIEXT1 and IBIEXT2, if configured. If extended data is used with time control, the data follows the time information. See IBIEXT1[MAX] for more information. 0 - Disable 1 - Enable
2	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
1-0 EVENT	Event Requests an event. If this field is 0b, it indicates Normal mode, in which if this field becomes 0 from a nonzero value and event processing has not yet started, the event processing is canceled. However, event processing is not canceled if it has already started. If this field is 1b, an IBI is pushed onto the I3C bus. If there is data associated with the IBI, the data is read from SCTRL[IBIDATA]. If time control is enabled, this data includes any time-control-related bytes. Additionally, SCTRL[IBIDATA][7] becomes 1 automatically (as is required for time control). The IBI interrupt occurs after the first (mandatory) IBIDATA, if any. If this field is 10b, a controller request is started; the meaning depends on SIDEXT[BCR] configured in I3C. If this field is 11b, a hot-join (HJ) request is started, which is used when the device is powered on after the I3C bus is already powered up. It is also used when the device is connected using hot-insertion methods (the device is powered up when it is physically inserted in the powered-up I3C bus). The HJ waits for Bus Idle, and SCTRL[EVENT] = HOT_JOIN_REQUEST must be set before the target enable (SCONFIG[SLVENA]). If this field is a nonzero value, it requests an event. After the request, SSTATUS[EVENT] and SSTATUS[EVDET] show the status as it progresses. After completion, this field automatically returns to 0. After this field becomes a nonzero value, you can write only 0 it (to cancel) until the event processing is finished. 00 - NORMAL_MODE 01 - IBI 10 - CONTROLLER_REQUEST 11 - HOT_JOIN_REQUEST

63.7.6 Target Interrupt Set (SINTSET)

Sets interrupt enables for select Target Status (SSTATUS) fields. Reading this register (SINTSET) returns the status of the interrupt enable:

- To activate an interrupt enable, write 1 to its corresponding field in this register (SINTSET).
- To disable an interrupt, write 1 to its corresponding field in Target Interrupt Clear (SINTCLR). Writing 0 to the interrupt enable in this register (SINTSET) does not disable the interrupt.

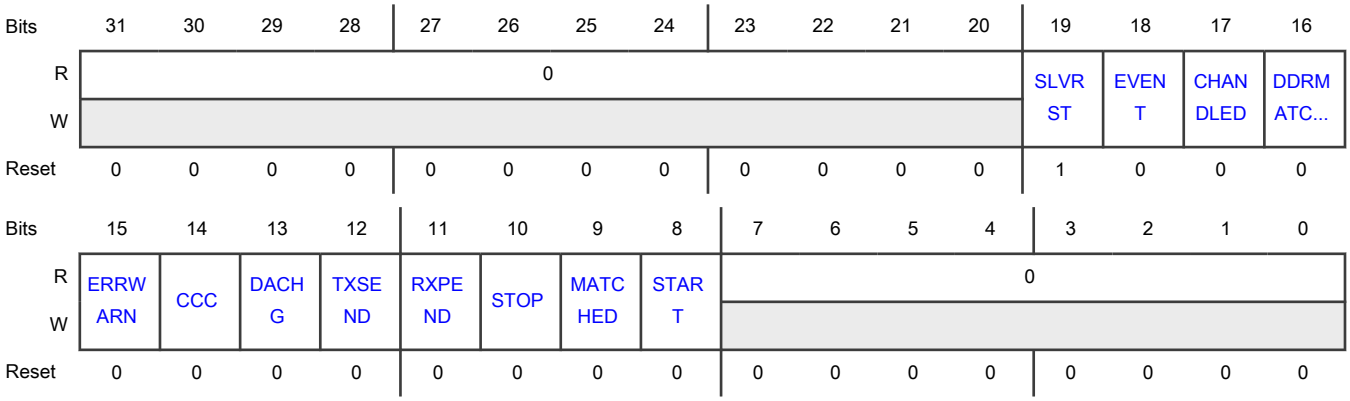
The Interrupt registers allow the masking of interrupt sources. They also allow the checking of which interrupts are activated. The normal method is to enable an interrupt, and then after the interrupt occurs, clear the interrupt either by writing to Target Status (SSTATUS) or by performing an action on the corresponding data register. The interrupt is level-held, meaning the interrupt stays set until the cause is cleared by some method. The module prevents races; if a new event occurs, that new event is not lost:

- SINTSET sets interrupt enables for Target Status (SSTATUS) fields. Reading the SINTSET register returns the status of the interrupt enables.
- SINTCLR clears interrupt enables for the SSTATUS fields.
- SINTMASKED returns the value of the SSTATUS fields ANDed with their interrupt enables.

Offset

Register	Offset
SINTSET	10h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 SLVRST	Target Reset Enables target reset interrupt. This field specifies whether target reset pattern is detected and action is set to reset peripheral. The field helps you perform follow-up tasks such as reinitialization. You must reset peripheral or verify that it is correct. This interrupt must not be unmasked. Used only if SLVRST is configured. 0 - Disable 1 - Enable
18 EVENT	Event Interrupt Enable Enables event interrupts. This field indicates, for a target, whether a pending in-band interrupt (IBI), controller request (CR), or hot-join (HJ) has been sent as requested. The field is configured to support events. See SSTATUS[EVDET] for more information. 0 - Disable 1 - Enable
17	Common Command Code (CCC) Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
CHANDLED	Enables CCC interrupts. This field specifies whether CCC is being handled by I3C. Target Status (SSTATUS) shows new results. You can use this field to track when activity states and masks on events (for example, IBIs) occur. The field is used for CCCs that are enabled. 0 - Disable 1 - Enable
16 DDRMATCHED	Double Data Rate Interrupt Enable Enables DDR match for read or write command. This field indicates when DDR matches for read or write command, and the field is used only if HDR is enabled. 0 - Disable 1 - Enable
15 ERRWARN	Error or Warning Interrupt Enable Enables error or warning interrupts. This field indicates whether an error or warning has occurred, such as data underrun, data overrun, parity error, or HDR-DDR CRC error, or any other error or warning condition. See Target Errors and Warnings (SERRWARN) for cause of error. Only available for errors related to configured features. 0 - Disable 1 - Enable
14 CCC	Common Command Code (CCC) Interrupt Enable Enables CCC interrupts. This field indicates whether a CCC has been received, and is not handled by I3C. For CCCs that the block does not handle, RXPEND also interrupts, and SSTATUS[STREQRD] indicates that it is a CCC sending a read request. 0 - Disable 1 - Enable
13 DACHG	Dynamic Address Change Interrupt Enable Enables dynamic address change interrupts: interrupt on dynamic address defined (SETDASA or ENTDAA) or lost (RSTDAA). See Controller Dynamic Address (MDYNADDR) for more information. 0 - Disable 1 - Enable
12	Transmit Interrupt Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
TXSEND	Enables transmit interrupts. This field indicates whether to-bus buffer or FIFO can accept more data to be transmitted. The corresponding interrupt occurs when the controller requests data to read from this target. This interrupt occurs on the first request (header) as well as when it is ready for more data. The corresponding interrupt occurs when the controller requests data to be read from this target. If this interrupt is for a FIFO, it triggers on the transmit emptiness trigger. If this interrupt is for DMA, then it indicates message end (DMA end or termination). 0 - Disable 1 - Enable
11 RXPEND	Receive Interrupt Enable Enables receive interrupts. This field specifies when a message from the controller that is not being handled by the module is received. For example, where data is consumed by hardware directly when it goes into the FIFO (excludes CCCs being handled automatically). If this interrupt is for a FIFO, it indicates a receive fullness trigger. If this interrupt is for DMA, then it indicates message end. 0 - Disable 1 - Enable
10 STOP	Stop Interrupt Enable Enables stop interrupts. This field detects the Stopped state present on the bus since the bus was last cleared. Use SINTSET[START] as the preferred interrupt when needed. This interrupt may not trigger for a quick STOP and START combination because it relates to the state of being stopped. 0 - Disable 1 - Enable
9 MATCHED	Match Interrupt Enable Enables match interrupts. Incoming header matched the I3C dynamic or I2C static address of this device since the bus was last cleared. If configured and if no dynamic address is set, this interrupt is also for matching a header on an I2C static address. See Controller Dynamic Address (MDYNADDR) for related information. 0 - Disable 1 - Enable
8 START	Start Interrupt Enable Enables start interrupts. A START or Repeated START (such as wakeup) is seen after this field last became 0. See SINTSET[STOP] for related information.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Disable 1 - Enable
7-0 —	Reserved

63.7.7 Target Interrupt Clear (SINTCLR)

Clears interrupt enables for select Target Status (SSTATUS) fields. To clear an interrupt enable, write 1 to the corresponding field in this register (SINTCLR). Writing 0 has no effect.

Offset

Register	Offset
SINTCLR	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												SLVRST	EVENT	CHANDLED	DDRMATC...
W													W1C	W1C	W1C	W1C
Reset	u	u	u	u	u	u	u	u	u	u	u	u	1	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERRWARN	CCC	DACHG	TXSEND	RXPEND	STOP	MATCHED	START	0							
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C								
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Description
31-20 —	Reserved
19 SLVRST	Target Reset Flag (SLVRST Interrupt Enable Clear)
18	EVENT Interrupt Enable Clear Flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
EVENT	
17 CHANDLED	CHANDLED Interrupt Enable Clear Flag
16 DDRMATCHED	DDRMATCHED Interrupt Enable Clear Flag
15 ERRWARN	ERRWARN Interrupt Enable Clear Flag
14 CCC	CCC Interrupt Enable Clear Flag
13 DACHG	DACHG Interrupt Enable Clear Flag
12 TXSEND	TXSEND Interrupt Enable Clear Flag
11 RXPEND	RXPEND Interrupt Enable Clear Flag
10 STOP	STOP Interrupt Enable Clear Flag
9 MATCHED	Matched Interrupt Enable Clear Flag
8 START	START Interrupt Enable Clear Flag
7-0 —	Reserved

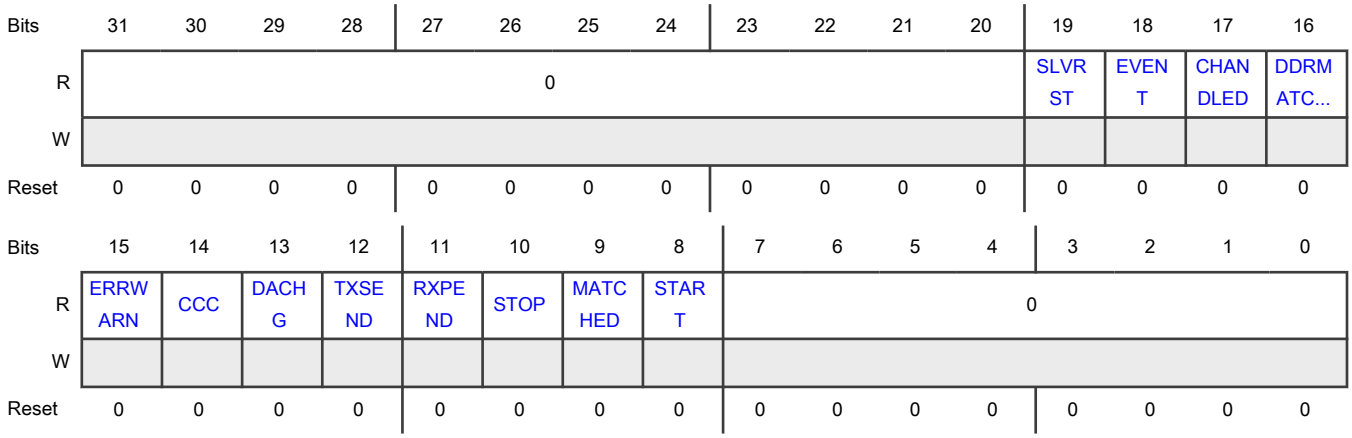
63.7.8 Target Interrupt Mask (SINTMASKED)

Returns the status of enabled interrupts (the value of Target Status (SSTATUS) ANDed with the value of Target Interrupt Set (SINTSET)).

Offset

Register	Offset
SINTMASKED	18h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 SLVRST	Target Reset Interrupt Mask
18 EVENT	EVENT Interrupt Mask
17 CHANDLED	CHANDLED Interrupt Mask
16 DDRMATCHED	DDRMATCHED Interrupt Mask
15 ERRWARN	ERRWARN Interrupt Mask
14 CCC	CCC Interrupt Mask
13 DACHG	DACHG Interrupt Mask
12 TXSEND	TXSEND Interrupt Mask
11	RXPEND Interrupt Mask

Table continues on the next page...

Table continued from the previous page...

Field	Description
RXPEND	
10 STOP	STOP Interrupt Mask
9 MATCHED	MATCHED Interrupt Mask
8 START	START Interrupt Mask
7-0 —	Reserved

63.7.9 Target Errors and Warnings (SERRWARN)

Contains errors and warnings from I3C and I2C protocols. This includes internal issues such as overrun and underrun, detected errors and conditions like parity errors, CRC errors, and read terminations by the controller. See SSTATUS[ERRWARN] and SINTSET[ERRWARN] for related information.

Offset

Register	Offset
SERRWARN	1Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0										0		0		OWRITE	OREAD
W															W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0	0	S0S1	HCRC	HPAR	SPAR	0			INVSTART	TERM	URUNNACK	URUN	ORUN
W					W1C	W1C	W1C	W1C				W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-22 —	Reserved
21 —	Reserved
20-18 —	Reserved
17 OWRITE	Over-Write Error Flag Indicates that Target Write Data Byte (SWDATAB) or Target Write Data Byte End (SWDATABE) was written to when full. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No overwrite error 1 - Overwrite error When writing 0 - No effect 1 - Clear the flag
16 OREAD	Over-Read Error Flag Indicates that Target Read Data Byte (SRDATAB) was read for more bytes than were available, by the application. This field also indicates over-read errors for Target Read Data Halfword (SRDATAH), if it is enabled. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No over-read error 1 - Over-read error When writing 0 - No effect 1 - Clear the flag
15-14 —	Reserved
13	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
12 —	Reserved
11 S0S1	TE0 or TE1 Error Flag Indicates whether a TE0 or TE1 error occurred and the target is locked and waiting for an HDR exit pattern. Writing 1 to S0S1 causes I3C to release the lock, but this method must be used with great care. S0S1 becomes 0 automatically when an exit pattern is detected, so writing 1 to S0S1 must be used under controlled circumstances to avoid problems. Before starting to operate normally after this error, I3C waits for a START (or Repeated START) or STOP state. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No TE0 or TE1 error occurred 1 - TE0 or TE1 error occurred When writing 0 - No effect 1 - Clear the flag
10 HCRC	HDR-DDR CRC Error Flag Indicates whether an HDR-DDR CRC error occurred on a message from the controller. Error reasons include an HDR restart and an exit being issued before an HDR-DDR message from the controller has finished. This error calls into question the data from the entire DDR command frame. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No HDR-DDR CRC error occurred 1 - HDR-DDR CRC error occurred When writing 0 - No effect 1 - Clear the flag
9 HPAR	HDR Parity Error Flag Indicates when an HDR parity error or framing error on a message from the controller occurs. The corresponding command or data that has the error is usually in the RX buffer, which can be read using Target Read Data Byte (SRDATAB).

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - No HDR parity error 1 - HDR parity error When writing 0 - No effect 1 - Clear the flag
8 SPAR	SDR Parity Error Flag Indicates when an SDR parity error on a message from the controller occurs. This error also sets the GETSTATUS protocol error field (which becomes 0 after a GETSTATUS read). For read operations, this field becomes 1 when a read abort (timeout) occurs because of the controller not driving clock for more than 100 µs during an SDR read. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No SDR parity error 1 - SDR parity error When writing 0 - No effect 1 - Clear the flag
7-5 —	Reserved
4 INVSTART	Invalid Start Error Flag Indicates an invalid condition with SCL falling before SDA falls, so there is no start. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No invalid start error 1 - Invalid start error When writing 0 - No effect 1 - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
3 TERM	Terminated Error Flag Indicates when the controller terminates a read from a target, if an END is not set by the target (on the same read or the previous read). **NOTE** This field behaves differently for register reads and writes. When reading 0 - No terminated error 1 - Terminated error When writing 0 - No effect 1 - Clear the flag
2 URUNNACK	Underrun and Not Acknowledged (NACKed) Error Flag Indicates when the internal to-bus buffer or FIFO is underrun in the read header and so the module NACKed the header. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No underrun; not acknowledged error 1 - Underrun; not acknowledged error When writing 0 - No effect 1 - Clear the flag
1 URUN	Underrun Error Flag Indicates when the internal to-bus buffer or FIFO is underrun during data read (the application is not providing the data fast enough). The END field or register must be used if the read was the last one. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No underrun error 1 - Underrun error When writing 0 - No effect 1 - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 ORUN	Overrun Error Flag Indicates when the internal from-bus buffer or FIFO has overrun (arrival of too many characters that you cannot process fast enough). **NOTE** This field behaves differently for register reads and writes. When reading 0 - No overrun error 1 - Overrun error When writing 0 - No effect 1 - Clear the flag

63.7.10 Target DMA Control (SDMACTRL)

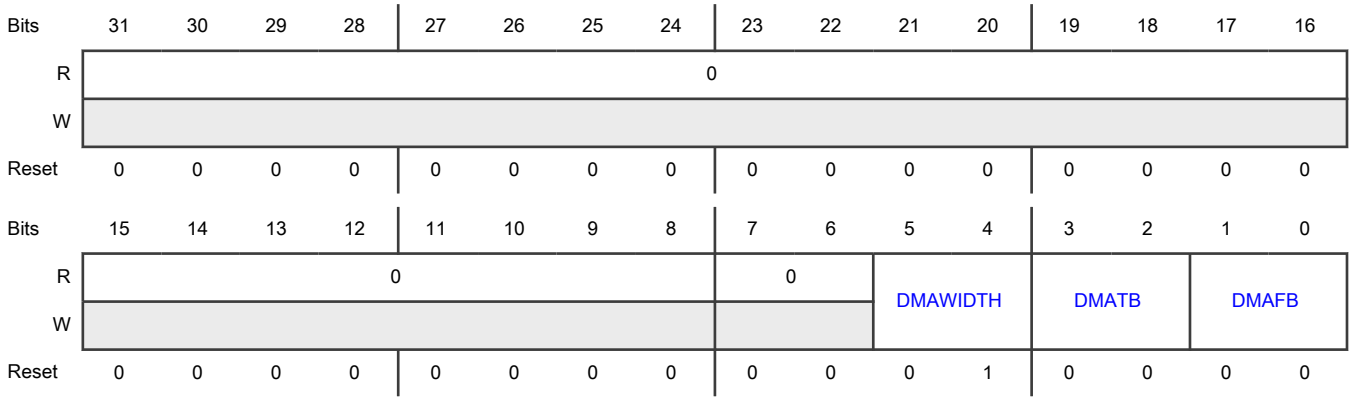
Allows DMA to be used for inbound and outbound messages. This register is limited in value for target use because the target must be reactive. These are the two common use case models:

- To avoid an overrun in from-bus collection: if SCONFIG[MATCHSS] becomes 1, then the processor enables the interrupts for START and STOP and enables the DMA to collect the data. The START or STOP interrupt only occurs after a message is directed to the target (MATCHED is 1). The DMA copied data can then be examined.
- To perform larger reads from the to-bus: I3C and I2C reads are preceded by a write that indicates what will be read (or in response to an IBI from the target). Because of this process, the DMA can be used to push through the data:
  - For I3C, the processor must manage the last value, unless the DMA moves wider words and is able to write 1 to the END field. These values are 16-bit values when in Byte mode or 32-bit values when in Halfword mode.
  - For I2C, the controller determines the last value, so the DMA may end early or run out when the controller expects more values.

Offset

Register	Offset
SDMACTRL	20h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-6 —	Reserved
5-4 DMAWIDTH	Width of DMA Operations Indicates the width of DMA operations, if configured to allow halfword data accesses. 00,01 - Byte 10 - Halfword (16 bits) (this value ensures that two bytes are available in the FIFO) 11 - Reserved
3-2 DMATB	DMA Write (To-Bus) Trigger Enables DMA writes. If this field is enabled, it starts a request DMA on a transmit trigger (see Target Data Control (SDATACTRL)). DMATB requests until full, unless the DMA is set up as a trigger. DMATB becomes 0 when MSTATUS[ERRWARN] becomes 1. If you write 1b to this field, DMA is enabled for one frame (ended by DMA or terminated). DMATB automatically becomes 0 after a STOP or repeated START. See SCONFIG[MATCHSS] for more information. If you write 10b to this field, DMA is enabled until turned off. The value must only be used with Controller Message mode. 00 - DMA not used 01 - DMA enabled for one frame 10 - DMA enabled until turned off. Recommended mode, instead of 01b, to enable DMA. This setting provides better software control. 11 - Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
1-0 DMAFB	DMA Read (From-Bus) Trigger Enables DMA reads. If this field is enabled, DMAFB requests DMA on receive trigger (see SDATACTRL). It requests until empty, unless the DMA is set up as a trigger. This field becomes 0 when SSTATUS[ERRWARN] becomes 1. If this field is 1b (DMA is enabled for one frame), it automatically becomes 0 on STOP or repeated START. See SCONFIG[MATCHSS] for more information. **NOTE** Do not enable DMA after a transaction starts. It must be enabled only when enabling the target and interrupts to avoid any kind of RX FIFO overrun. 00 - DMA not used 01 - DMA enabled for one frame 10 - DMA enabled until turned off. Recommended mode, instead of 01b, to enable DMA. This setting provides better software control. 11 - Reserved

63.7.11 Target Data Control (SDATACTRL)

Assists in data control when no FIFO is used. Also assists in data control of FIFO when the FIFO is available (regardless of size) allowing some control over the FIFO behavior. This register controls when to interrupt based on fullness or emptiness of a buffer or FIFO. It also controls behavior related to width, when the buffer or FIFO is not 1 byte wide.

Offset

Register	Offset
SDATACTRL	2Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXEMPTY	TXFULL	0	RXCOUNT					0		0		TXCOUNT			
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RXTRIG		TXTRIG			0		
W													UNLOCK		FLUSHFB	FLUSHTB
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Fields

Field	Description
31 RXEMPTY	Receive is Empty Indicates whether the receive FIFO or buffer is empty. 0 - Not empty 1 - Empty
30 TXFULL	Transmit is Full Indicates whether the transmit FIFO or buffer is full. 0 - Not full 1 - Full
29 —	Reserved
28-24 RXCOUNT	Count of Entries in Receive Contains the count of entries in the receive FIFO or buffer. Entry size for SDR and HDR-DDR is in bytes.
23-22 —	Reserved
21 —	Reserved
20-16 TXCOUNT	Count of Entries in Transmit Contains the count of entries waiting in the TXFIFO. This count is the number of entries that the application has written to the transmit FIFO that have not yet gone onto the bus. Entry size for SDR and HDR-DDR is in bytes.
15-8 —	Reserved
7-6 RXTRIG	Receive Trigger Level Indicates the trigger level for receive fullness when using a FIFO. The field affects the RXPEND interrupt. 00 - Trigger when not empty (default) 01 - Trigger when 1/4 or more full 10 - Trigger when 1/2 or more full 11 - Trigger when 3/4 or more full
5-4 TXTRIG	Transmit Trigger Level

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Indicates the trigger level for transmit emptiness when using a FIFO. The field affects the TXNOTFULL interrupt. 00 - Trigger when empty 01 - Trigger when 1/4 full or less 10 - Trigger when 1/2 full or less 11 - Default (trigger when 1 less than full or less)
3 UNLOCK	Unlock Indicates whether the RXTRIG and TXTRIG fields can be changed on a write. This field must be 1 in the same cycle while writing to TXTRIG or RXTRIG. 0 - Cannot be changed 1 - Can be changed
2 —	Reserved
1 FLUSHFB	Flush From-Bus Buffer or FIFO Not normally used. **NOTE** Before using FLUSHFB, disable the DMA (SDMACTRL[DMAFB] = 0). Then use FLUSHFB to flush the FIFO and then re-enable the DMA. 0 - No action 1 - Flush the buffer
0 FLUSHTB	Flush To-Bus Buffer or FIFO Indicates when the controller terminates a to-bus (read) message prematurely. **NOTE** Before using FLUSHTB, disable the DMA (SDMACTRL[DMATB] = 0). Then use FLUSHTB to flush the FIFO and then re-enable the DMA. 0 - No action 1 - Flush the buffer

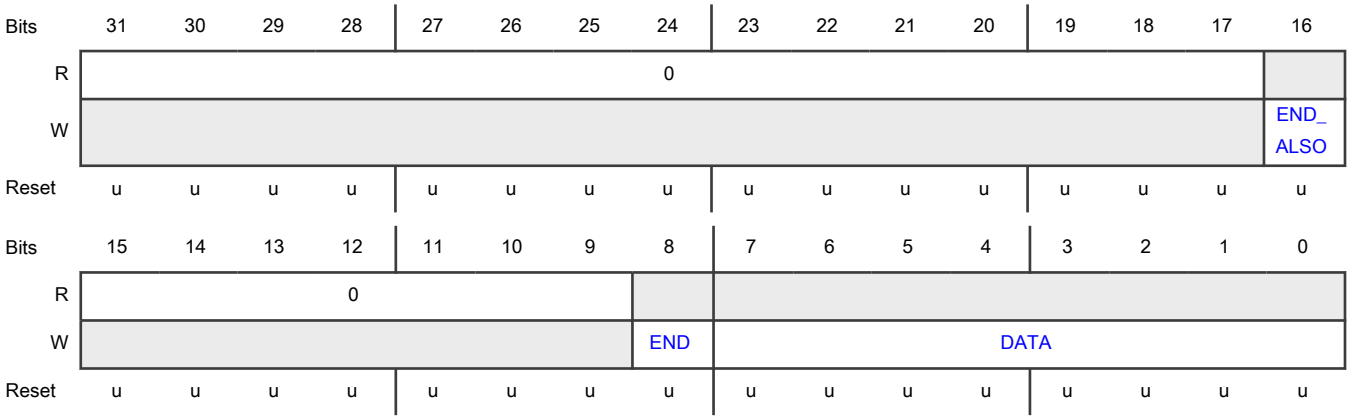
63.7.12 Target Write Data Byte (SWDATAB)

Allows writing a byte to the bus (to the controller) unless an external FIFO is used. Writing a byte requires a byte plus an end-of-data (last) marker bit. A byte must not be written unless there is room, indicated by SSTATUS[TXNOTFULL] = 1.

Offset

Register	Offset
SWDATAB	30h

Diagram



Fields

Field	Description
31-17 —	Reserved
16 END_ALSO	End Also Marks the last byte of the message. This field is required for I3C, but is optional for I2C. If this field is 0, it indicates that there are more bytes in the message. For HDR-DDR, the byte with END_ALSO must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs. 0 - Not the end 1 - End
15-9 —	Reserved
8 END	End Marks the last byte of the message. If this field is 0, it indicates that there are more bytes in the message. This field is required for I3C, but is optional for I2C. For HDR-DDR, the byte with END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs. 0 - Not the end 1 - End

Table continues on the next page...

Table continued from the previous page...

Field	Description
7-0 DATA	Data Indicates the data byte to be sent to the controller.

63.7.13 Target Write Data Byte End (SWDATABE)

Allows writing a byte to the bus (to the controller) unless an external FIFO is used.

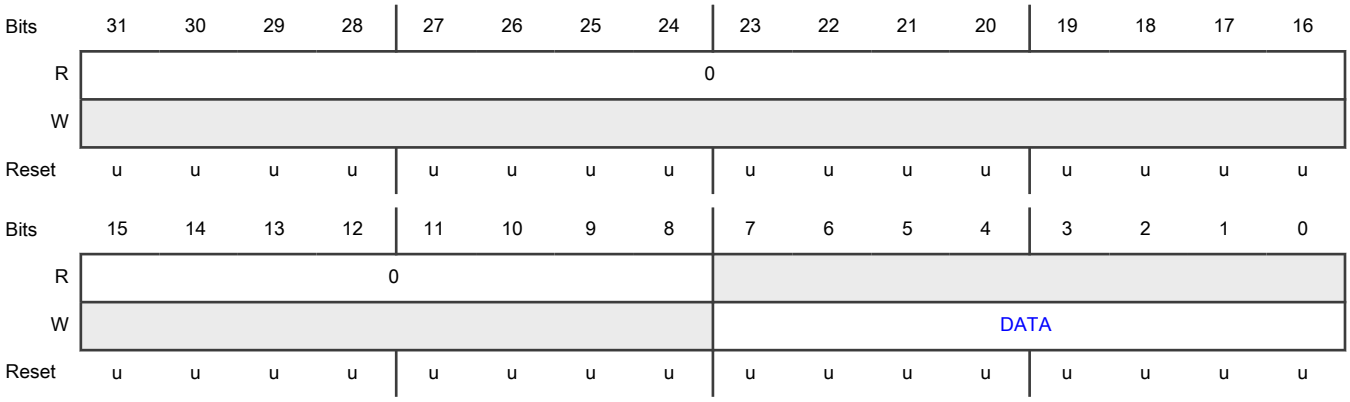
Unlike Target Write Data Byte (SWDATAB), writing a byte only requires the byte itself, and is marked as end-of-data (last byte). A byte must not be written unless there is room, indicated by SSTATUS[TXNOTFULL] = 1.

For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs.

Offset

Register	Offset
SWDATABE	34h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 DATA	Data Indicates the data byte to be sent to the controller.

63.7.14 Target Write Data Halfword (SWDATAH)

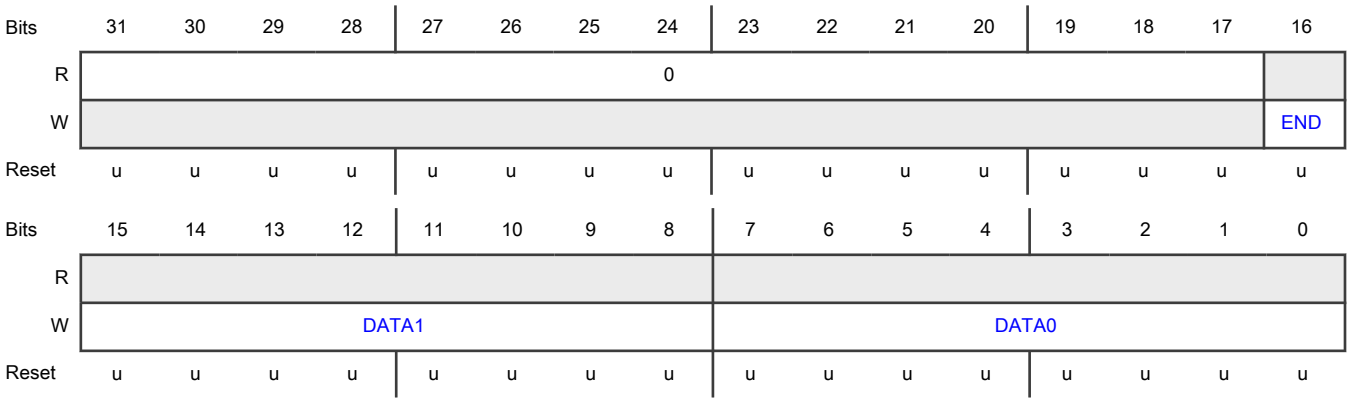
Allows writing a halfword (pair of bytes) to the bus unless an external FIFO is used. The low byte is followed by the high byte. The 16th bit marks the end; that is, the last byte of the halfword is the end.

An end-of-data (last) marker bit is allowed (or must be 0). A halfword must not be written unless there is room for both, as indicated by the use of transmit FIFO level trigger or SDATACTRL[TXCOUNT].

Offset

Register	Offset
SWDATAH	38h

Diagram



Fields

Field	Description
31-17 —	Reserved
16 END	End of Message Marks the last byte of the message. This field always marks DATA1 as the end. The field is required for I3C, but is optional for I2C. If this field is 0, it indicates that there are more bytes in the message. For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs. 0 - Not the end 1 - End
15-8 DATA1	Data 1 Indicates the second byte to be sent to the controller.
7-0 DATA0	Data 0 Indicates the first byte to be sent to the controller.

63.7.15 Target Write Data Halfword End (SWDATAHE)

Allows writing a halfword of data, which is the end (the last byte of the halfword is the end). The target writes the halfword (byte pair) in the same way it writes to Target Write Data Halfword (SWDATAH), but marks the second byte as end-of-data (last byte).

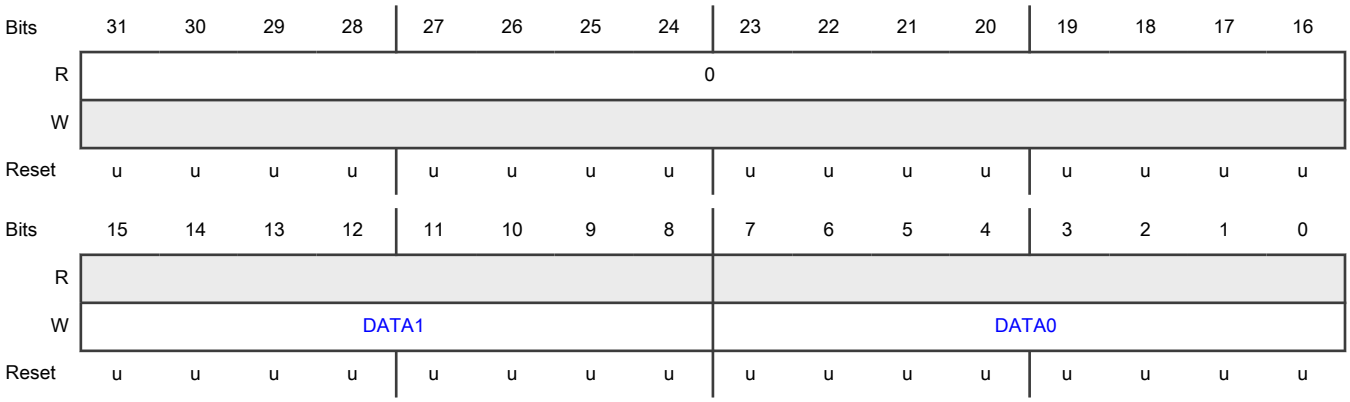
For HDR-DDR, the byte with the END must be even (second, fourth, sixth, and so on) because DDR uses byte pairs.

A halfword must not be written unless there is room for both, as indicated by the use of transmit FIFO level trigger or SDATACTRL[TXCOUNT].

Offset

Register	Offset
SWDATAHE	3Ch

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 DATA1	Data 1 Indicates the second byte to be sent to the controller.
7-0 DATA0	Data 0 Indicates the first byte to be sent to the controller.

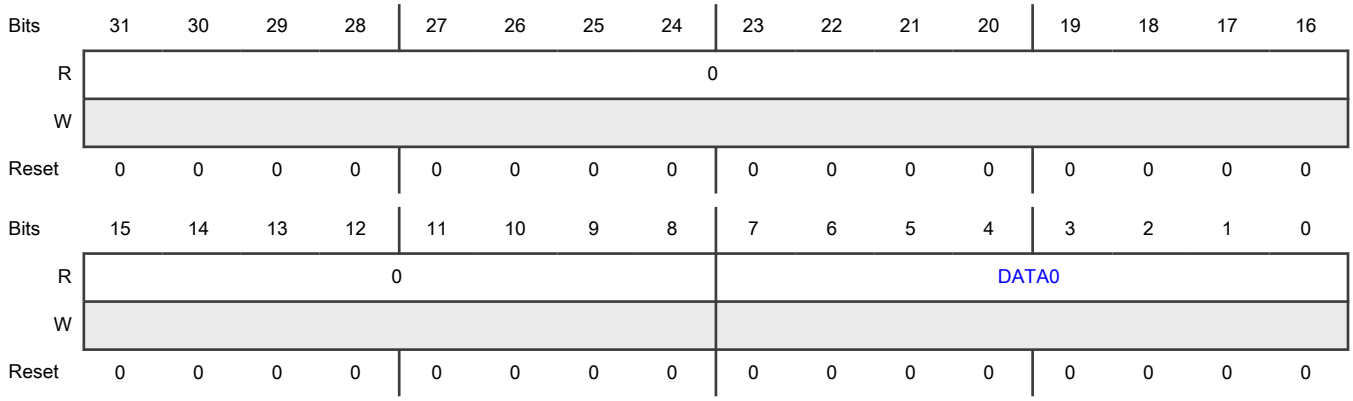
63.7.16 Target Read Data Byte (SRDATAB)

Allows reading a byte from the bus (controller). A byte must not be read unless there is data waiting, as indicated by SSTATUS[RX_PEND] = 1.

Offset

Register	Offset
SRDATAB	40h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 DATA0	Data 0 Indicates the byte read from the controller.

63.7.17 Controller Read Data Halfword (MRDATAH)

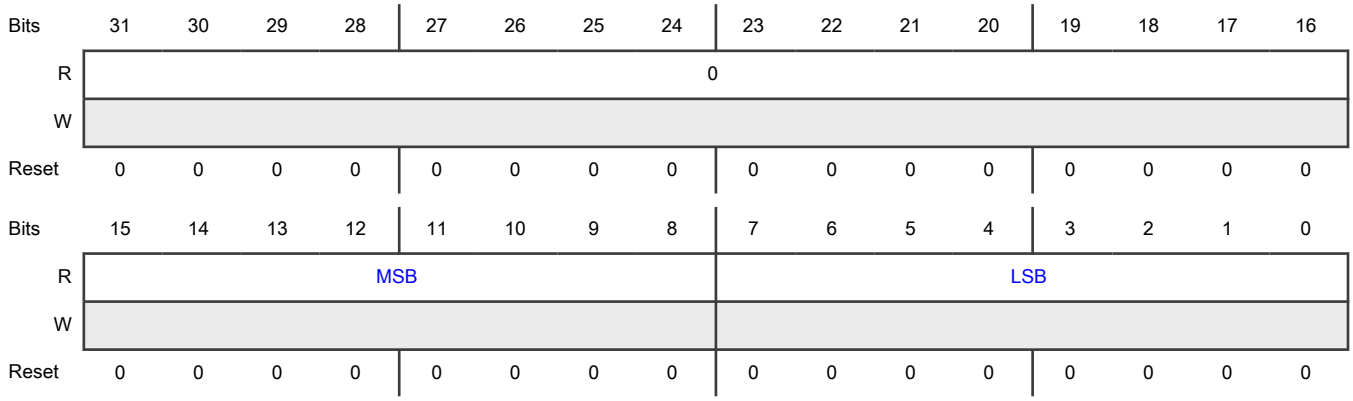
Allows reading a halfword (byte pair) written by the target after an SDR read or DAA or DDR. This register is used only when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

A halfword must not be read unless there are at least two bytes of data waiting, as indicated by the use of receive FIFO level trigger or MDATACTRL[RXCOUNT].

Offset

Register	Offset
MRDATAH	48h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 MSB	High Byte Represents the second byte read from the controller (and written by the target).
7-0 LSB	Low Byte Represents the first byte read from the controller (and written by the target).

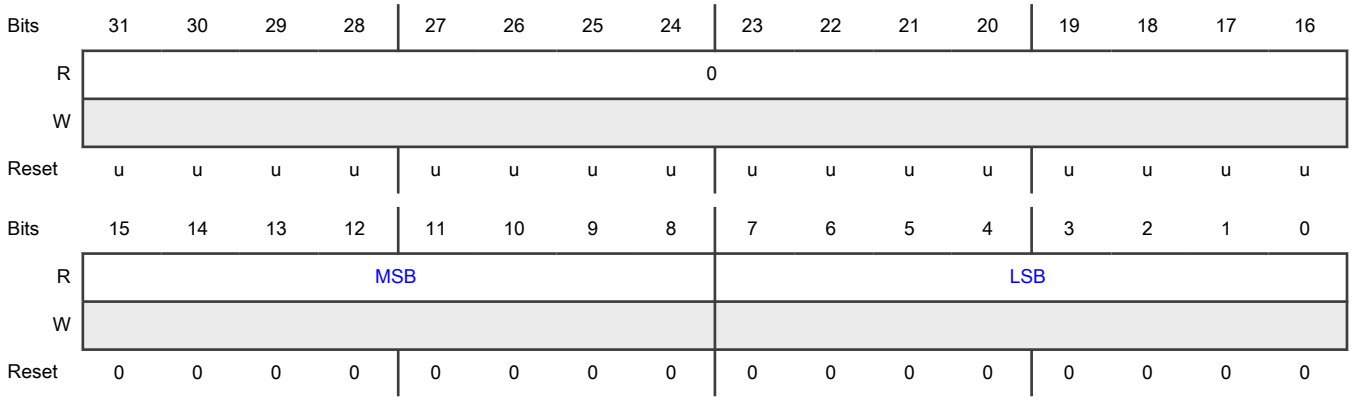
63.7.18 Target Read Data Halfword (SRDATAH)

Allows reading a halfword (byte pair) written by the target after an SDR read or DAA or DDR. This register is used only when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively. A halfword must not be read unless there are at least two bytes of data waiting, as indicated by the use of receive FIFO level trigger or SDATACTRL[RXCOUNT].

Offset

Register	Offset
SRDATAH	48h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 MSB	High Byte Indicates the second byte read from the target.
7-0 LSB	Low Byte Indicates the first byte read from the target.

63.7.19 Target Write Data Byte (SWDATAB1)

Allows writing a single byte to the bus (to the controller) in a way that only bits 7:0 are used.

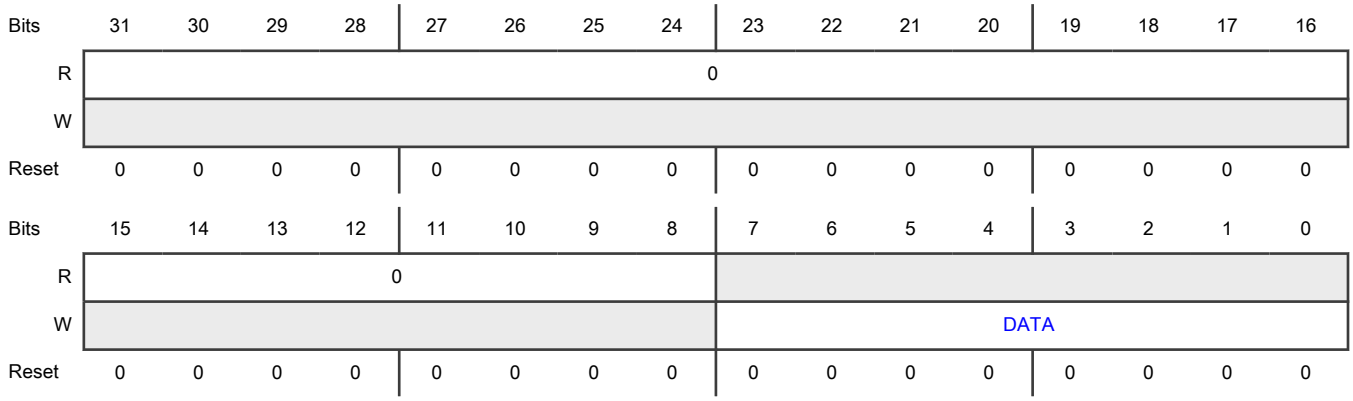
This register is intended to be used by DMAs (the upper bytes of the APB word are ignored).

A byte must not be written unless there is room, as indicated by SSTATUS[TXNOTFULL] = 1.

Offset

Register	Offset
SWDATAB1	54h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 DATA	Data Indicates the byte to be sent to the controller.

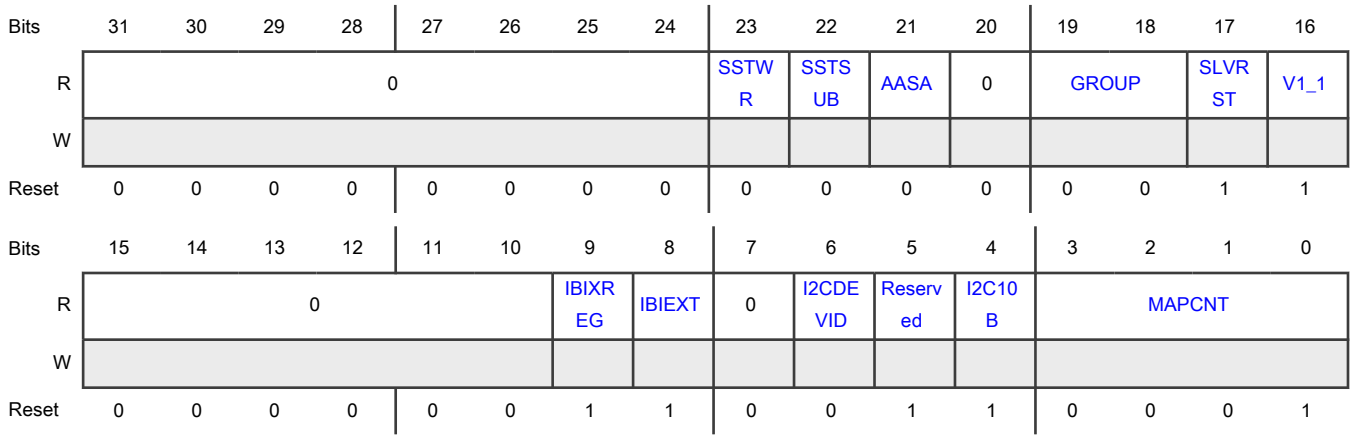
63.7.20 Target Capabilities 2 (SCAPABILITIES2)

Indicates which features are available and supported in this module, including controller and target capabilities, HDR modes, and others.

Offset

Register	Offset
SCAPABILITIES2	5Ch

Diagram



Fields

Field	Description
31-24 —	Reserved
23 SSTWR	Target-Target(s)-Tunnel Write Capable Indicates whether target-target(s)-tunnel is write capable. 0 - Not write capable 1 - Write capable
22 SSTSUB	Target-Target(s)-Tunnel Subscriber Capable Indicates whether target-target(s)-tunnel is subscriber capable. 0 - Not subscriber capable 1 - Subscriber capable
21 AASA	SETAASA Supports the set static address as dynamic address CCC (SETAASA) feature. 0 - SETAASA not supported 1 - SETAASA supported
20 —	Reserved
19-18 GROUP	Group Indicates whether v1.1 group addressing is supported. Groups use mapping, so MAPCNT must be the same or greater than GROUP. 00 - v1.1 group addressing not supported 01 - One group supported 10 - Two groups supported 11 - Three groups supported
17 SLVRST	Target Reset Indicates whether v1.1 target reset is supported. 0 - Not supported 1 - Supported
16 V1_1	Version 1.1 Indicates whether I3C v1.1 GETCAPS is supported. 0 - Not supported 1 - Supported

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-10 —	Reserved
9 IBIXREG	In-Band Interrupt Extended Register Supports extended registers for IBIs. This field indicates whether Extended IBI Data 1 (IBIEXT1) is available. Extended IBI Data 2 (IBIEXT2) is available if IBIEXT1[MAX] > 3. 0 - Not supported 1 - Supported
8 IBIEXT	In-Band Interrupt EXTDATA Supports SCTRL[EXTDATA] to allow data beyond the mandatory data byte (MDB) for IBIs. 0 - Not supported 1 - Supported
7 —	Reserved
6 I2CDEVID	I2C Device ID Indicates whether I2C device ID is supported. 0 - Not supported 1 - Supported
5 —	Reserved
4 I2C10B	I2C 10-bit Address Supports 10-bit I2C address in MAP 1. If this field is 1, MAPCNT must be 1 or greater. 0 - Not supported 1 - Supported
3-0 MAPCNT	Map Count Indicates the number of maps that are allowed. If there is no mapping, this field is 0. (DYNADDR is used to add more static and/or dynamic addresses to act as virtual targets.)

63.7.21 Target Capabilities (SCAPABILITIES)

Indicates which features are available and supported in this module, including controller and/or target capabilities, HDR modes, and others.

Offset

Register	Offset
SCAPABILITIES	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DMA	INT	FIFORX		FIFOTX		EXTFIFO		0		TIMECTRL		IBI_MR_HJ			
W																
Reset	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CCCHANDLE				SADDR		MASTER	0	HDRSUPP		IDREG				IDENA	
W																
Reset	1	1	1	1	1	1	1	0	0	1	1	1	0	0	0	0

Fields

Field	Description
31 DMA	Direct Memory Access Indicates whether DMA is supported. 0 - Not supported 1 - Supported
30 INT	Interrupts Indicates whether interrupts are supported. 0 - Not supported 1 - Supported
29-28 FIFORX	FIFO Receive Indicates the size of receive (from-bus) FIFO. **NOTE** Controller and target use the same receive FIFO, so size is the same for the controller receive FIFO as well. FIFO size of SDR and HDR-DDR is in bytes. 00 - Two or three 01 - Four 10 - Eight

Table continues on the next page...

Table continued from the previous page...

Field	Description
	11 - 16 or larger
27-26 FIFOTX	FIFO Transmit Indicates the size of transmit (to-bus) FIFO. **NOTE** Controller and target use the same transmit FIFO, so size is the same for the controller transmit FIFO as well. FIFO size of SDR and HDR-DDR is in bytes. 00 - Two 01 - Four 10 - Eight 11 - 16 or larger
25-23 EXTFIFO	External FIFO Indicates whether external FIFOs are enabled. If they are not enabled, then check FIFOTX and FIFORX for the internal FIFO. 000 - No external FIFO available 001 - Standard available or free external FIFO 010 - Request track external FIFO All other values are reserved.
22 —	Reserved
21 TIMECTRL	Time Control Specifies whether any time-control type is supported. 0 - No time control supported 1 - At least one time-control type supported
20-16 IBI_MR_HJ	In-Band Interrupts, Controller Requests, Hot-Join Events Indicates which events (IBI, CR, and HJ) are allowed. For example, if this field is 00011b, IBI (bit 0) and IBI_HAS_DATA (bit 1) functionality are both enabled. 0_0000 - Application cannot generate IBI, CR, or HJ 1_xxxx - Application can use SCONFIG[BAMATCH] for bus-available timing x_1xxx - Application can generate a Hot-Join event x_x1xx - Application can generate a controller request for a secondary controller x_xx1x - When bit 0 = 1, the IBI has data from the SCTRL register x_xxx1 - Application can generate an IBI

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-12 CCCHANDLE	Common Command Codes Handling Indicates what manages CCC between I3C and the application. 0000 - All handling features disabled 1xxx - GETSTATUS CCC returns the value of SCTRL[VENDINFO] x1xx - GETSTATUS CCC returns the values of SCTRL[PENDINT] and SCTRL[ACTSTATE] xx1x - The I3C module manages maximum read and write lengths, and max data speed xxx1 - The I3C module manages events, activities, status, HDR, and if enabled for it, ID and static-address-related items
11-10 SADDR	Static Address Indicates how the static address is managed. 00 - No static address 01 - Static address is fixed in hardware 10 - Hardware controls the static address dynamically (for example, from the pin strap) 11 - SCONFIG register supplies the static address
9 MASTER	Controller Specifies whether controller capability is supported. 0 - Not supported 1 - Supported
8 —	Reserved
7-6 HDRSUPP	High Data Rate Support Indicates which HDR modes are supported. 00 - No HDR modes supported 01 - DDR mode supported All other values are reserved.
5-2 IDREG	ID Register Indicates which ID features are available in the configurable registers. 0000 - All ID register features disabled 1xxx - A Bus Characteristics Register (BCR) is available x1xx - A Device Characteristic Register (DCR) is available xx1x - An ID Random field is available xxx1 - ID Instance is a register; used if there is no PARTNO register

Table continues on the next page...

Table continued from the previous page...

Field	Description
1-0 IDENA	ID 48b Handler Indicates what handles the 48-bit ID value. 00 - Application 01 - Hardware 10 - Hardware, but the I3C module instance handles ID 48b 11 - A part number register (PARTNO)

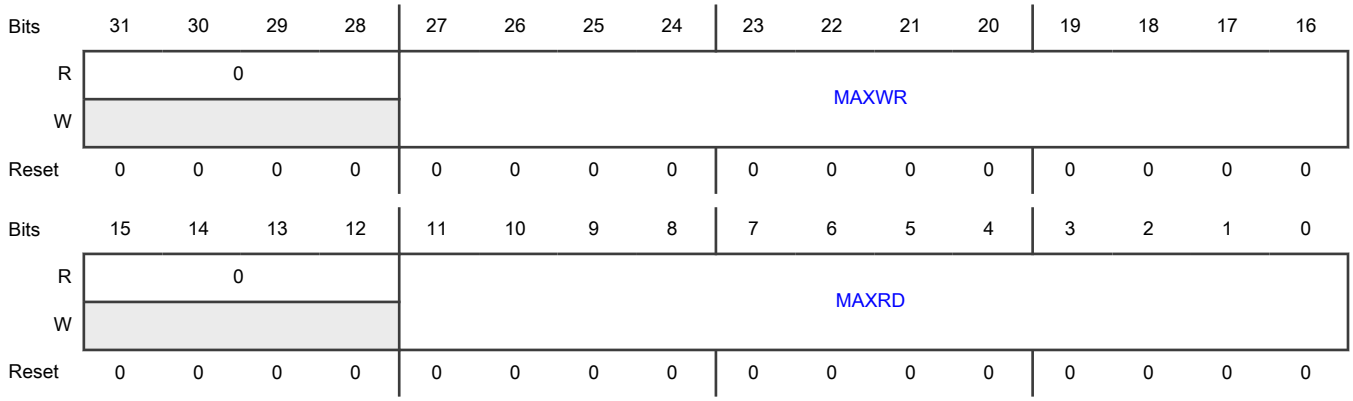
63.7.22 Target Maximum Limits (SMAXLIMITS)

Indicates the limits set by the controller (or the originally requested limits). The maximum limits are not enabled in the hardware design, including maximum read and write lengths. If the maximum read and write lengths are enabled, then the current setting (including the default request) shows up in this register (SMAXLIMITS).

Offset

Register	Offset
SMAXLIMITS	68h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 MAXWR	Maximum Write Length Indicates the maximum write length, which must be between 8 to 4095 (saturation). The application must not set the maximum write length to a higher value than the maximum write length set by the controller.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-12 —	Reserved
11-0 MAXRD	Maximum Read Length Indicates the maximum read length, which must be between 16 to 4095 (saturation). The application must not set the maximum read length to a higher value than the maximum read length set by the controller.

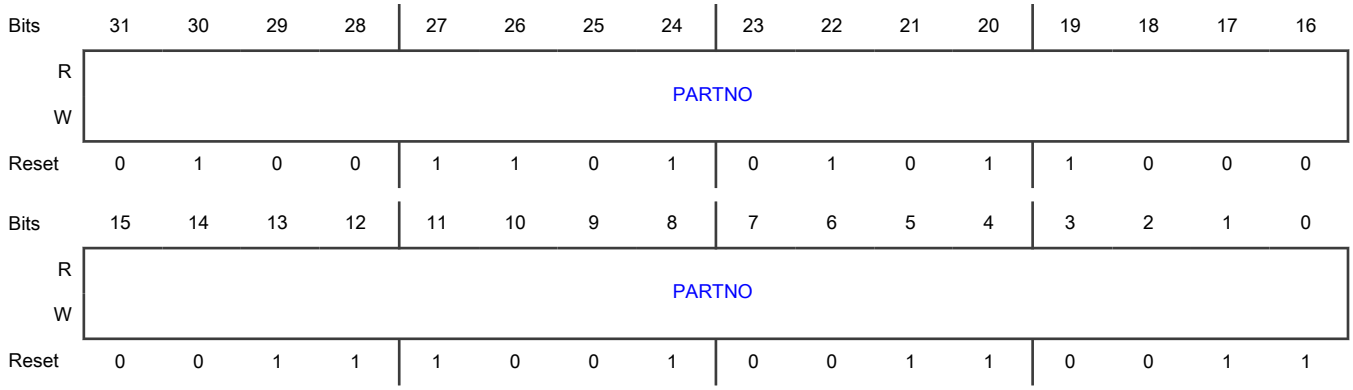
63.7.23 Target ID Part Number (SIDPARTNO)

Allows you to write the ID part number. You must write a nonzero value into the PARTNO field because 0 is not valid.

Offset

Register	Offset
SIDPARTNO	6Ch

Diagram



Fields

Field	Description
31-0 PARTNO	Part Number

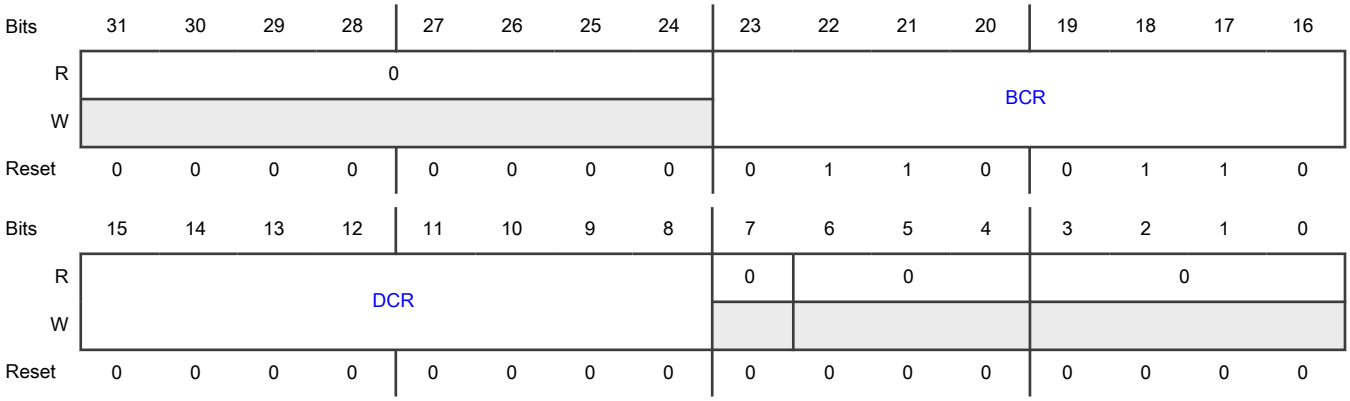
63.7.24 Target ID Extension (SIDEXT)

Allows you to write the ID extension of DCR and/or BCR.

Offset

Register	Offset
SIDEXT	70h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 BCR	Bus Characteristics Register Sets the value for BCR, if this field is configured. Otherwise, the default value is considered. This field controls features such as secondary controller and slow-speed requirements.
15-8 DCR	Device Characteristic Register Sets the value for DCR, if this field is configured. Otherwise, the default value is considered.
7 —	Reserved
6-4 —	Reserved
3-0 —	Reserved

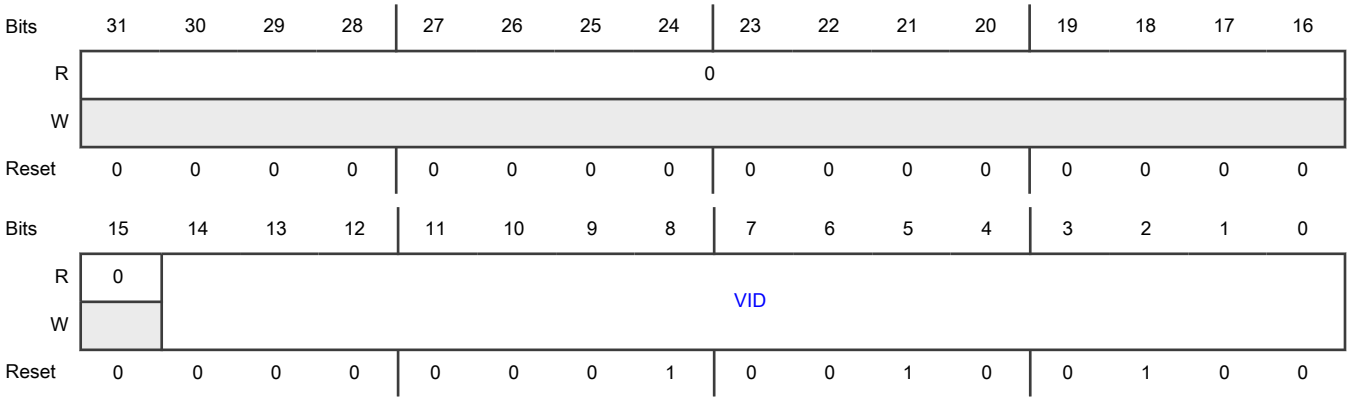
63.7.25 Target Vendor ID (SVENDORID)

Allows you to write the vendor ID. The default value is the chip vendor ID, and is set from the constant field. When using the chip vendor ID, the part number (PARTNO) does not collide with other uses. The MIPI vendor ID is available to all companies (MIPI membership is not required). To get a vendor ID, make a request at the mipi.org website.

Offset

Register	Offset
SVENDORID	74h

Diagram



Fields

Field	Description
31-15 —	Reserved
14-0 VID	Vendor ID Configures the 15-bit MIPI vendor ID. If not configured, the default value is considered.

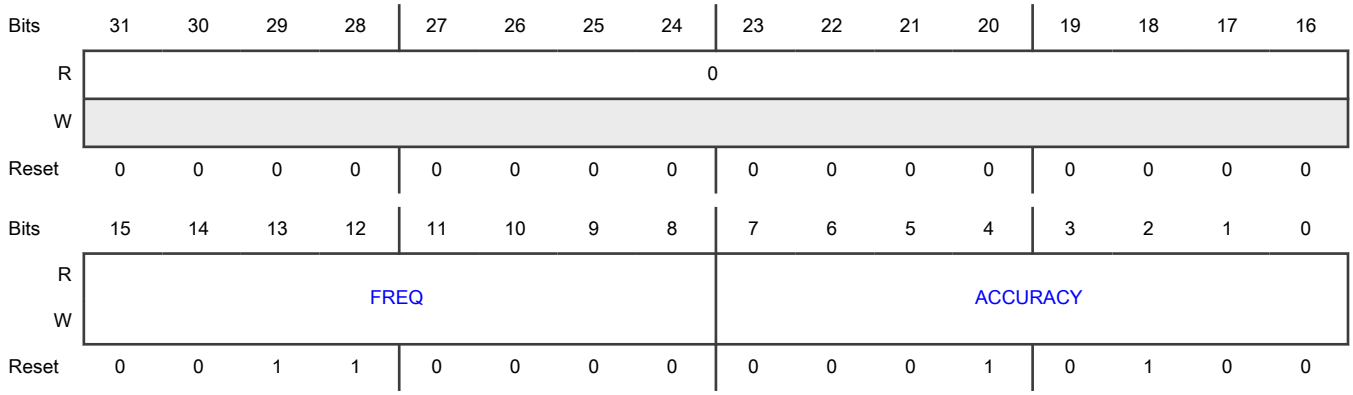
63.7.26 Target Time Control Clock (STCCLOCK)

Allows you to dynamically set the time control clock and accuracy information. The clock frequency and accuracy are constants set by the hardware. If the clock can be adjusted (that is, divided) or if the accuracy could vary with knowable information, then the clock may be set via this register, which must be updated whenever the clock source is changed.

Offset

Register	Offset
STCCLOCK	78h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 FREQ	Clock Frequency Indicates the clock frequency in 0.5 MHz steps. For example, a value of 20 in this field indicates a frequency of 10 MHz. Default set by parameters if configured.
7-0 ACCURACY	Clock Accuracy Indicates the clock accuracy in 1/10ths of %. For example, a value of 15 indicates an accuracy of 1.5%. Default set by parameters if configured.

63.7.27 Target Message Map Address (SMSGMAPADDR)

Allows you to determine which address is matched when STATUS[MATCHED] = 1. This register is used when the DYNADDR register builds a list of extra DAs or SAs to match. It holds the last three matches.

Offset

Register	Offset
SMSGMAPADDR	7Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												MAPLASTM2			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				MAPLASTM1				0			LASTSTA...	MAPLAST			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-20 —	Reserved
19-16 MAPLASTM2	Matched Previous Index 2 Indicates index 2 of the previous matched address (0 for the base address).
15-12 —	Reserved
11-8 MAPLASTM1	Matched Previous Address Index 1 Indicates index 1 of the previous matched address (0 for the base address).
7-5 —	Reserved
4 LASTSTATIC	Last Static Address Matched Indicates whether the last matched address was an I2C static address or an I3C dynamic address. 0 - I3C dynamic address 1 - I2C static address
3-0 MAPLAST	Matched Address Index Indicates the matched address index for current or last matched message (0 for the base address).

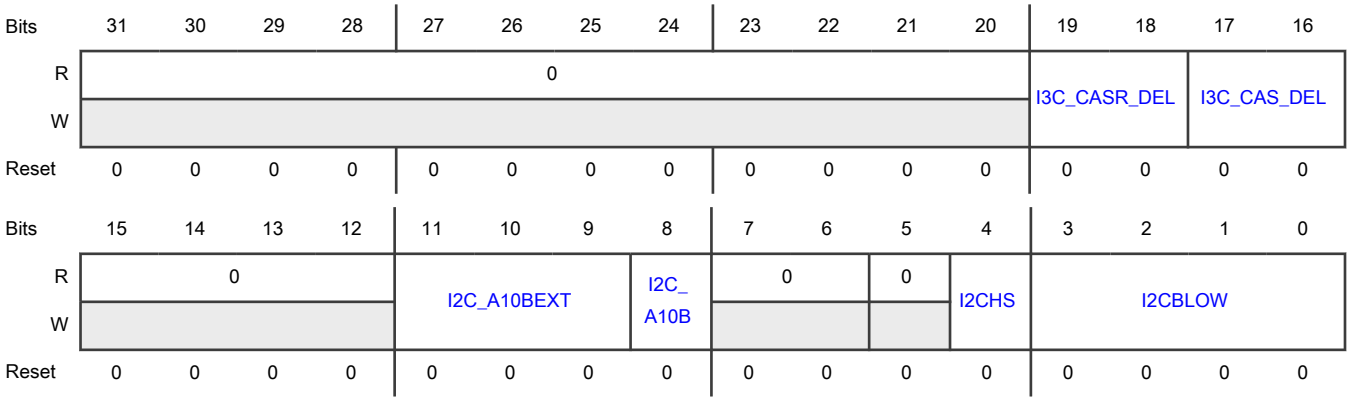
63.7.28 Controller Extended Configuration (MCONFIG_EXT)

Allows special extended configurations, including for I2C use.

Offset

Register	Offset
MCONFIG_EXT	80h

Diagram



Fields

Field	Description
31-20 —	Reserved
19-18 I3C_CASR_DEL	I3C CAS Delay After Repeated START Increases SCL clock period by a certain value. You can configure this field to delay the clock after a Repeated START to provide more time for slow I3C devices. The maximum value may be less if the internal clock is very fast. 00 - No delay 01 - Increases SCL clock period by 1/2 10 - Increases SCL clock period by 1 11 - Increases SCL clock period by 1 1/2
17-16 I3C_CAS_DEL	I3C CAS Delay After START Increases SCL clock period by a certain value. By increasing the SCL clock period, you can delay the clock to be longer after the START. The maximum value may be less if the internal clock is very fast. 00 - No delay 01 - Increases SCL clock period by 1/2 10 - Increases SCL clock period by 1 11 - Increases SCL clock period by 3/2
15-12	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
11-9 I2C_A10BEXT	I2C_A10BEXT Works with MCTRL[DA] and specifies the upper 3 bits of 10-bit address.
8 I2C_A10B	I2C_A10B Enables 10-bit addressing. When enabled for 10-bit addressing, the extra 3 bits are in bits 9 to 11. 0 - Disable 1 - Enable
7-6 —	Reserved
5 —	Reserved
4 I2CHS	I2C HS Uses the push-pull clock for I2C, and I2CBAUD and I2CBLOW must be used for 3.4MHz if this field is 1.
3-0 I2CBLOW	I2C Baud Low Overrides the I2CBAUD low period if not 0. So, I2CBAUD is used for high period, and I2CBLOW is used for low. This can allow an extended low such as for 400 kHz, as well as HS duty cycle. Note that this is 0 relative, so 1 is 1 × ODBAUD.

63.7.29 Controller Control (MCTRL)

Starts activities on the I3C or I2C bus (see the MWMSG registers). A request cannot be changed when a message is in progress; the REQUEST field becomes 0 automatically.

You must write to MCTRL fields as per use case or as mentioned in Operating modes. Bit read and write checking may not work independently.

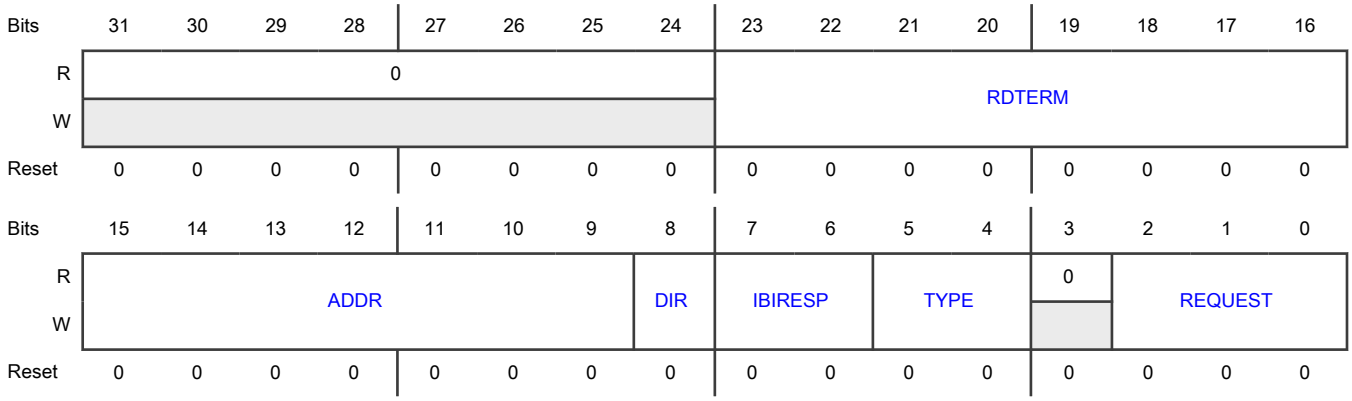
NOTE

If MCONFIG[MSTENA] is configured for I2C Controller mode (legacy I2C), only REQUEST = 1 and REQUEST = 2 are accepted. Also, fields are constrained to I2C-supported fields.

Offset

Register	Offset
MCTRL	84h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-16 RDTERM	Read Terminate Counter Determines when to terminate a read operation: • For I2C, this field controls when to NACK a read. • For I3C, this field can be used to terminate (end) a read: — If RDTERM is 0, it has no effect. — If RDTERM is 1, the read terminates after the next character. — If RDTERM is 2, the read terminates after the next two characters. The field supports up to 255 characters. In DDR mode, RDTERM terminates the read based on word counts (for DDR) instead of byte counts (for SDR). The field self-clears when MSTATUS[COMPLETE] becomes 1. **NOTE** You may write 1 to this field anytime, but a value greater than 1 must be written when starting EmitStartAddress.
15-9 ADDR	Address Indicates the type of address: I3C dynamic address or I2C static address. Some values are not allowed based on the bus that is used (I3C or I2C).
8 DIR	Direction Indicates the direction, write or read. 0 - Write 1 - Read
7-6	In-Band Interrupt Response

Table continues on the next page...

Table continued from the previous page...

Field	Description
IBIRESP	Indicates the response to use when you get an IBI from START, and when to force using an IBI ACK NACK request when completing a manual IBI. Completion of a manual IBI means that the target DA is known, and so the mandatory byte (or not) is specified by the application when acknowledging. This field is also used when a message is emitted in Message mode using the MWMSG_SDR or MWMSG_DDR registers. If an IBI with MDB (mandatory byte) is ACKed, the controller limits it to a maximum of eight more bytes after the MDB. RDTERM = 1 can be used with REQUEST = NONE to terminate data from a target sooner. If this field is 0b, it indicates ACK (acknowledge). When REQUEST = 1 (EmitStartAddr) or REQUEST = 7 (AutoIBI), Controller In-band Interrupt Registry and Rules (MIBIRULES) decides whether to ACK with mandatory byte. When REQUEST = 3 (IBIAckNack), ACK with no mandatory byte. If this field is 1b, it indicates NACK (reject) or no acknowledgement. If this field is 10b, it indicates acknowledge with mandatory byte. When REQUEST = 1 or REQUEST = 7, ignore Controller In-band Interrupt Registry and Rules (MIBIRULES). Do not use this setting unless only targets with a mandatory byte can cause an IBI. When REQUEST = 3, ACK with mandatory byte. If this field is 11b, it indicates manual. When REQUEST = 1 or REQUEST = 7, stop and wait for a decision using the IBI ACK NACK request. When REQUEST = 3, this field is reserved. **NOTE** A CR and HJ always cause IBIRESP to become 3 (manual) so the application must decide. 00 - ACK (acknowledge) 01 - NACK (reject) 10 - Acknowledge with mandatory byte 11 - Manual
5-4 TYPE	Bus Type with EmitStartAddr Works with REQUEST to determine the bus type. See Determining bus types and modes for more information. The following values are valid only if REQUEST is 1. 00 - I3C 01 - I2C 10 - DDR 11 - Reserved
3 —	Reserved
2-0 REQUEST	Request Emits the requested operation when performing in pieces, instead of performing by message. You must check Controller Status (MSTATUS) because some requests can only be made in some states. For

Table continues on the next page...

Table continued from the previous page...

Field	Description
	example, the system cannot enter SDR mode from DDR mode, and it cannot use an incorrect request in DAA mode. If this field is 0b, it indicates that no request is present (NONE). The REQUEST field returns to NONE after finishing a request. Controller Status (MSTATUS) indicates the state of the controller. See AutoIBI mode. NONE is written as 0 only in these cases: when writing 1 to MCTRL[RDTERM] (to stop a read in progress) or when configuring MCTRL[IBIRESP] for MSG use. If this field is 1b, it emits START with address and direction (EMITSTARTADDR), either from Stopped state or in the middle of a single data rate (SDR) message. If from Stopped state (Idle), then EmitStart may be prevented by an event (such as IBI, CR, HJ). In this case, an appropriate interrupt is signaled. EmitStart can be resubmitted. If this field is 10b, it emits a STOP on bus (EMITSTOP). Must be in SDR mode. In DAA mode, emitting STOP exits DAA mode. If this field is 11b, it indicates manual IBI ACK or NACK (IBIACKNACK). When MCTRL[IBIRESP] indicates a hold on an IBI to allow a manual decision, this request completes it. The field uses MCTRL[IBIRESP] to provide the information. If this field is 100b, and if currently not in DAA mode, it emits START, 7E, ENTDAA sequence, and then emits 7E/R to process the first target (PROCESSDAA). Stops just before the new dynamic address (DA) is to be emitted. The DA is written using Controller Write Data Byte (MWDATAB), then process DAA is requested again to write the new address, and then it starts the next unless marked to STOP. An MSTATUS indicating NACK means DA was not accepted (for example, parity error). If PROCESSDAA is NACKed on the 7E/R request, meaning no more targets need a DA, then a COMPLETE is signaled (along with DONE) and a STOP issued. If TYPE = 2 or 3, the DA is assigned and then it emits a STOP (instead of starting a new 7E/R request). If TYPE = 1 or 3, then the DA is taken from the ADDR field (bits 6:0). If this field is 110b, it emits an exit pattern from any state (force exit and target reset). End DDR mode (including MSGDDR), including a STOP afterward. If MCTRL[TYPE] is 2, perform a target reset action (RSTACT can prevent the reset in the target). Target reset may follow immediately after RSTACT CCC or STOP. If this field is 111b, the target is held in a Stopped state, but autoemits a START, 7E sequence when the target holds SDA low for an IBI (AUTOIBI). Actual IBI handling is defined by MCTRL[IBIRESP]. 000 - NONE 001 - EMITSTARTADDR 010 - EMITSTOP 011 - IBIACKNACK 100 - PROCESSDAA 101 - Reserved 110 - Force Exit and Target Reset 111 - AUTOIBI

63.7.30 Controller Status (MSTATUS)

Indicates the controller status, including which events cause interrupts. The peripherals share the IRQ (called parallel-to-target status).

Because a peripheral can either be in Controller or Target mode, but not both at the same time, only one (target or controller peripheral) can cause the IRQ. If there is an IRQ and the peripheral is a controller, then this register (MSTATUS) indicates the status.

If there is an IRQ and the peripheral is a target, then Target Status (SSTATUS) indicates the status.

This register self-clears if the COMPLETE field becomes 1.

If the MCONFIG register is configured only for I2C, some states and fields never remain active (for example, DAA or IBI).

Offset

Register	Offset
MSTATUS	88h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	IBIADDR							0				NOWMAST...	0		
W													W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERRWARN	0	IBIWON	TXNOTFU...	RXPEND	COMPLETE	MCTRLDO...	SLVSTART	IBITYPE		NACKED	BETWEEN	0	STATE		
W			W1C			W1C	W1C	W1C								
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30-24 IBIADDR	IBI Address Indicates the address of: • The IBI when MSTATUS[IBITYPE] = 1. • The CR, when MSTATUS[IBITYPE] = 2. • 7'h2 when HJ, when MSTATUS[IBITYPE] = 3.
23-20 —	Reserved
19 NOWMASTER	Module is now Controller Flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Indicates whether the module is now a controller. That is, it was previously a target, and controllership acceptance was requested from the previous controller and controllership was accepted. The reverse operation (controller becomes a target) does not need an interrupt because the application grants it through the GETACCMST CCC. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not a controller 1 - Controller When writing 0 - No effect 1 - Clear the flag
18-16 —	Reserved
15 ERRWARN	Error or Warning Indicates whether an error occurred, such as improper register use, overrun, or underrun of FIFO or buffer, or invalid parity or CRC in a DDR read. See Controller Errors and Warnings (MERRWARN) for more information. 0 - No error or warning 1 - Error or warning
14 —	Reserved
13 IBIWON	In-Band Interrupt (IBI) Won Flag Indicates whether an IBI, CR, or HJ has won the arbitration on a header address, regardless of whether it was NACKed or ACKed. Arbitration requires manual intervention for CR and HJ, and optionally requires it for IBI if MCTRL[IBIRESP] = 3 (manual). **NOTE** This field behaves differently for register reads and writes. When reading 0 - No IBI arbitration won 1 - IBI arbitration won When writing 0 - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Clear the flag
12 TXNOTFULL	TX Buffer or FIFO Not Full Indicates whether the buffer, FIFOm, or message register can accept another byte or halfword. FIFO uses trigger level. If DMA is enabled for transmitting, it transfers data as long as it is not full. 0 - Receive buffer or FIFO full 1 - Receive buffer or FIFO not full
11 RXPEND	RXPEND Indicates whether a message is being received from a target and bytes are in the input buffer or FIFO: • When using a FIFO, this message is at least one FIFO trigger's worth (a minimum of one byte in the FIFO). • When DMA is enabled for receiving, the DMA is signaled. RXPEND becomes 0 when the data is read. 0 - No receive message pending 1 - Receive message pending
10 COMPLETE	Complete Flag Indicates whether a message is complete: • With MWMSG_SDR or MWMSG_DDR, the COMPLETE condition occurs when MWMSG_SDR_CONTROL[LEN] or MWMSG_DDR_CONTROL[LEN] reaches 0. • When MCTRL[REQUEST] = 1 (EmitStartAddr), this COMPLETE condition occurs after the end of a write operation, or after a read operation has terminated or ended. • With an IBI (AutoIBI or EmitStartAddr), the COMPLETE condition occurs at the end of IBI data (if any). **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not complete 1 - Complete When writing 0 - No effect 1 - Clear the flag
9 MCTRLDONE	Controller Control Done Flag Indicates whether the module has completed an MCTRL request. MCTRLDONE automatically becomes 0 when writing a new control.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If MCTRL[REQUEST] = 1 (EmitStartAddr), MCTRLDONE becomes 1 when the address goes out (and is ACKed, NACKed, or ended in an IBI). If ACKed, MSTATUS[COMPLETE] becomes 1 after the write or read data is complete. If MCTRL[REQUEST] = 4 (ProcessDAA), MCTRLDONE becomes 1 when the module is ready to emit the dynamic address (DA) for the target, or when no more targets are ACKing. This condition can be determined by using the MSTATUS[BETWEEN] and MSTATUS[STATE] fields. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not done 1 - Done When writing 0 - No effect 1 - Clear the flag
8 SLVSTART	Target Start Flag Indicates whether a target is or was requesting a START by holding SDA low. Handling starts automatically when MCTRL[REQUEST] = 7 (AutoIBI). **NOTE** This field behaves differently for register reads and writes. When reading 0 - Target not requesting START 1 - Target requesting START When writing 0 - No effect 1 - Clear the flag
7-6 IBITYPE	In-Band Interrupt (IBI) Type Indicates the type of IBI of the last event that won the arbitration, whether the interrupt was ACKed, NACKed, or pending. 00 - NONE (no IBI: this status occurs when MSTATUS[IBIWON] becomes 0) 01 - IBI 10 - CR 11 - HJ
5 NACKED	Not Acknowledged

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Indicates whether the last start and address sequence was NACKed (was not ACKed by the addressed target). 0 - Not NACKed 1 - NACKed (not acknowledged)
4 BETWEEN	Between Indicates whether the controller is active between messages or dynamic address assignments (DAA). It is active when: • MSTATUS[STATE] is MSGSDR, DDR, or DAA, or NORMACT, and the state is between messages or DAAs. It is expecting a new message or DAA to start (or STOP or exit). • MSTATUS[STATE] is NORMACT. The module is waiting for the transmit FIFO to be not empty or the receive FIFO to be not full. 0 - Inactive (for other cases) 1 - Active
3 —	Reserved
2-0 STATE	State of the Controller Indicates the current controller state. If this field is 0b, it indicates that the controller is idle (the bus has stopped.) If this field is 1b, it indicates target request. The bus has stopped but a target is holding SDA low. When using auto-emit IBI (MCTRL[REQUEST] = 7), the controller does not remain in this state. If this field is 10b, it indicates entry into Single Data Rate Message mode, using MWMSG_SDR. If this field is 11b, it indicates entry into normal active SDR mode, using MCTRL, MWDATA*n*, and MRDATA*n* registers. The controller remains in this state until a STOP is issued. If this field is 100b, it indicates entry into Double Data Rate Message mode, using MWMSG_DDR or the normal method with DDR. The controller remains in the DDR state until the controller exits using EXIT (emitting the Exit pattern). If this field is 101b, it indicates entry into Dynamic Address Assignment (ENTDAA) mode. If this field is 110b, it indicates wait for an IBI ACK/NACK decision. If this field is 111b, it indicates receiving an IBI. This state is used after IBI, CR, or HJ has won an arbitration. The IBIRCV state is also used for IBI mandatory byte (if any) and any bytes that follow. 000 - IDLE (bus has stopped) 001 - SLVREQ (target request) 010 - MSGSDR 011 - NORMACT 100 - MSGDDR 101 - DAA

Table continues on the next page...

Table continued from the previous page...

Field	Description
	110 - IBIACK 111 - IBIRCV

63.7.31 Controller In-band Interrupt Registry and Rules (MIBIRULES)

Contains the rules for using IBI, and keeps a registry of the targets that use the IBI byte.

This register defines the IBI mandatory byte rules for the targets and determines which targets do (or do not) have a mandatory byte.

Concerning ADDR*n* fields: The address has 6 bits. Assuming that MIBIRULES[MSB0] = 1, the most significant bit of each address is 0. In this case, each address is 7 bits (usually written as A7 to A1) and A7 must be 0. If the application does not use that optimal convention, the "manual" method of IBI ACK handling must be used (see MCTRL[IBIRESP] for more information).

By default, the ADDR*n* values indicate the targets with a mandatory byte. If MIBIRULES[NOBYTE] = 1, the ADDR*n* values indicate targets that do not have a mandatory IBI byte.

NOTE

A7 = 0 is only needed for targets that use IBI. For legacy I2C devices, A7 can use any valid value because I2C devices cannot cause an IBI.

Offset

Register	Offset
MIBIRULES	8Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NOBYTE	MSB0	ADDR4						ADDR3						ADDR2	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR2				ADDR1				ADDR0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 NOBYTE	No IBI byte Specifies whether the ADDR*n* fields refer to targets with or without a mandatory IBI byte.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - With mandatory IBI byte 1 - Without mandatory IBI byte
30 MSB0	Most Significant Address Bit is 0 Specifies whether MSB is 0 for all I3C dynamic addresses. Assigning 1 to this field allows the START header to be optimized. 0 - MSB is not 0 1 - MSB is 0
29-24 ADDR4	ADDR4 Specifies the address of the target with or without mandatory IBI byte. If 0, then the address does not apply.
23-18 ADDR3	ADDR3 Specifies the address of the target with or without mandatory IBI byte. If 0, then the address does not apply.
17-12 ADDR2	ADDR2 Specifies the address of the target with or without mandatory IBI byte. If 0, then the address does not apply.
11-6 ADDR1	ADDR1 Specifies the address of the target with or without mandatory IBI byte. If 0, then the address does not apply.
5-0 ADDR0	ADDR0 Specifies the address of the target with or without mandatory IBI byte. If 0, then the address does not apply.

63.7.32 Controller Interrupt Set (MINTSET)

Returns the status of the interrupt enables when read. This register sets interrupt enables for select fields in Controller Status (MSTATUS):

- To activate an interrupt enable, write 1 to its corresponding field.
- To disable an interrupt enable, write 1 to the appropriate field in Controller Interrupt Clear (MINTCLR). Writing 0 to the interrupt enable in MINTSET does not disable the interrupt.

The interrupt registers allow masking of interrupt sources as well as checking which interrupts are activated in Controller Status (MSTATUS). The normal method is to enable an interrupt and then after that interrupt fires, clear the interrupt either by writing to Controller Status (MSTATUS) or via an action on the corresponding data register. The interrupt is level-held, meaning that the interrupt remains 1 until the cause is cleared by some method. The module prevents races; if a new event comes in, that new event is not lost.

These interrupts are parallel to Target Interrupt Set (SINTSET) and Target Interrupt Clear (SINTCLR), and only one interrupt set is active depending on the state (controller or target operation):

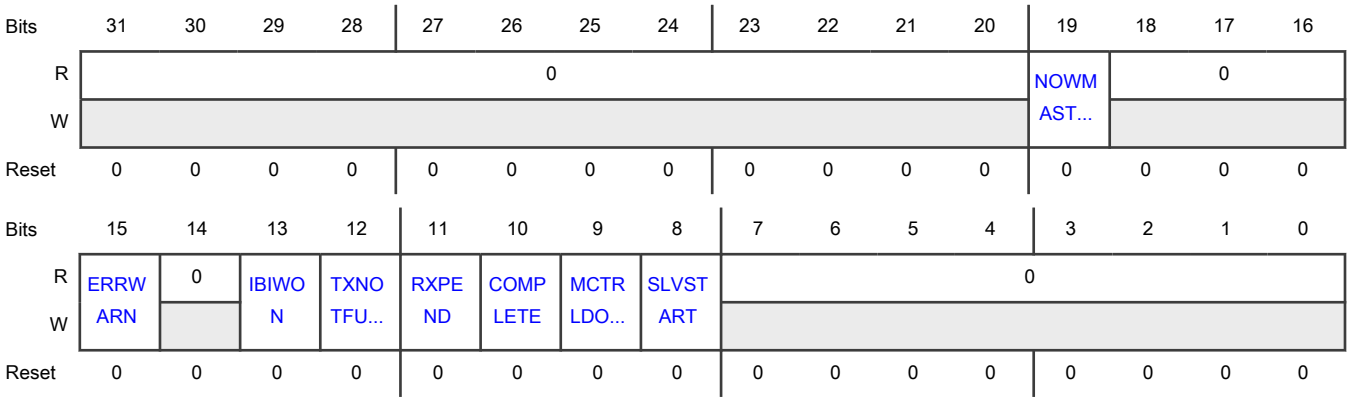
- MINTSET: Sets interrupt enables for MSTATUS fields. Reading the MINTSET register returns the status of the interrupt enables.

- MINTCLR: Clears interrupt enables for MSTATUS fields.
- MINTMASKED: Returns the value of the MSTATUS fields ANDed with their interrupt enables.

Offset

Register	Offset
MINTSET	90h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 NOWMASTER	Now Controller Interrupt Enable Enables the corresponding interrupt to indicate whether the module is now a controller (now this I3C module is a controller). That is, it was previously a target; controllership acceptance was requested from the previous controller and it was accepted. 0 - Disable 1 - Enable
18-16 —	Reserved
15 ERRWARN	Error or Warning (ERRWARN) Interrupt Enable Enables the corresponding interrupt to indicate whether an error occurred, such as improper register use, overrun or underrun of FIFO or buffer, or invalid parity or CRC in a DDR read. 0 - Disable 1 - Enable
14	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
13 IBIWON	IBI Won Interrupt Enable Enables the corresponding interrupt to indicate whether an IBI, CR, or HJ has won the arbitration on a header address, regardless of whether it was NACKed or ACKed. 0 - Disable 1 - Enable
12 TXNOTFULL	Transmit Buffer/FIFO Not Full Interrupt Enable Enables the corresponding interrupt to indicate whether the buffer, FIFOm, or message register can accept another byte or halfword. 0 - Disable 1 - Enable
11 RXPEND	Receive Pending Interrupt Enable Enables the corresponding interrupt to indicate whether a message is being received from a target and bytes are in the input buffer or FIFO.
10 COMPLETE	Completed Message Interrupt Enable Enables the corresponding interrupt to indicate whether a message is complete. 0 - Disable 1 - Enable
9 MCTRLDONE	Controller Control Done Interrupt Enable Enables the corresponding interrupt to indicate whether the module has completed an MCTRL request. 0 - Disable 1 - Enable
8 SLVSTART	Target Start Interrupt Enable Enables the corresponding interrupt to indicate whether a target is or was requesting a START by holding SDA low. 0 - Disable 1 - Enable
7-0 —	Reserved

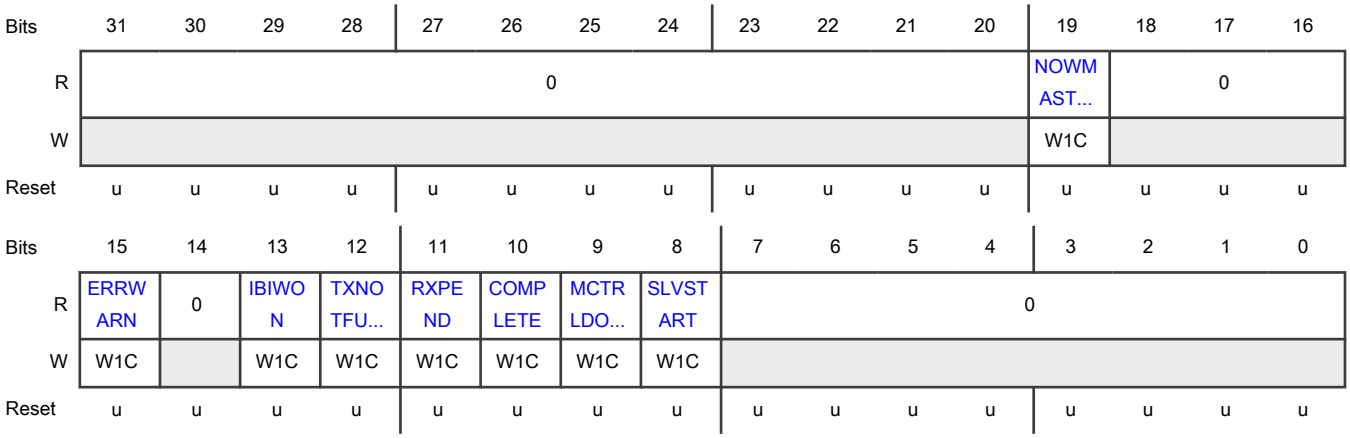
63.7.33 Controller Interrupt Clear (MINTCLR)

Clears interrupt enables for select fields in Controller Status (MSTATUS). Writing 1 clears the corresponding interrupt enable. Writing 0 has no effect.

Offset

Register	Offset
MINTCLR	94h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 NOWMASTER	NOWCONTROLLER Interrupt Enable Clear Flag Clears the corresponding NOWCONTROLLER interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No effect 1 - Interrupt enable cleared When writing 0 - No effect 1 - Clear the flag
18-16 —	Reserved
15 ERRWARN	ERRWARN Interrupt Enable Clear Flag Clears the corresponding ERRWARN interrupt enable.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - No effect 1 - Interrupt enable cleared When writing 0 - No effect 1 - Clear the flag
14 —	Reserved
13 IBIWON	IBIWON Interrupt Enable Clear Flag Clears the corresponding IBIWON interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No effect 1 - Interrupt enable cleared When writing 0 - No effect 1 - Clear the flag
12 TXNOTFULL	TXNOTFULL Interrupt Enable Clear Flag Clears the corresponding TXNOTFULL interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No effect 1 - Interrupt enable cleared When writing 0 - No effect 1 - Clear the flag
11 RXPEND	RXPEND Interrupt Enable Clear Flag Clears the corresponding RXPEND interrupt enable.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - No effect 1 - Interrupt enable cleared When writing 0 - No effect 1 - Clear the flag
10 COMPLETE	COMPLETE Interrupt Enable Clear Flag Clears the corresponding COMPLETE interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No effect 1 - Interrupt enable cleared When writing 0 - No effect 1 - Clear the flag
9 MCTRLDONE	MCTRLDONE Interrupt Enable Clear Flag Clears the corresponding MCTRLDONE interrupt enable. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No effect 1 - Interrupt enable cleared When writing 0 - No effect 1 - Clear the flag
8 SLVSTART	SLVSTART Interrupt Enable Clear Flag Clears the corresponding SLVSTART interrupt enable. **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When reading 0 - No effect 1 - Interrupt enable cleared When writing 0 - No effect 1 - Clear the flag
7-0 —	Reserved

63.7.34 Controller Interrupt Mask (MINTMASKED)

Returns the status of enabled interrupts (the value of Controller Status (MSTATUS) ANDed with the value of Controller Interrupt Set (MINTSET)).

Offset

Register	Offset
MINTMASKED	98h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												NOWMAST...	0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERRWARN	0	IBIWON	TXNOTFU...	RXPEND	COMPLETE	MCTRLDO...	SLVSTART	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-20 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
19 NOWMASTER	NOWCONTROLLER Interrupt Mask Indicates whether the NOWCONTROLLER interrupt is enabled and active. 0 - Disabled 1 - Enabled
18-16 —	Reserved
15 ERRWARN	ERRWARN Interrupt Mask Indicates whether the ERRWARN interrupt is enabled and active. 0 - Disabled 1 - Enabled
14 —	Reserved
13 IBIWON	IBIWON Interrupt Mask Indicates whether the IBIWON interrupt is enabled and active. 0 - Disabled 1 - Enabled
12 TXNOTFULL	TXNOTFULL Interrupt Mask Indicates whether the TXNOTFULL interrupt is enabled and active. 0 - Disabled 1 - Enabled
11 RXPEND	RXPEND Interrupt Mask Indicates whether the RXPEND interrupt is enabled and active.
10 COMPLETE	COMPLETE Interrupt Mask Indicates whether the COMPLETE interrupt is enabled and active. 0 - Disabled 1 - Enabled
9 MCTRLDONE	MCTRLDONE Interrupt Mask Indicates whether the MCTRLDONE interrupt is enabled and active. 0 - Disabled 1 - Enabled
8	SLVSTART Interrupt Mask

Table continues on the next page...

Table continued from the previous page...

Field	Description
SLVSTART	Indicates whether the SLVSTART interrupt is enabled and active. 0 - Disabled 1 - Enabled
7-0 —	Reserved

63.7.35 Controller Errors and Warnings (MERRWARN)

Contains errors and warnings. When any errors or warnings are not 0, then MSTATUS[ERRWARN] is 1.

Parallel-to-target ERRWARN:

- In Controller mode, use this register (MERRWARN).
- In Target mode, use Target Errors and Warnings (SERRWARN).

The error fields in both registers (MERRWARN and SERRWARN) are similar in meaning.

Offset

Register	Offset
MERRWARN	9Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0										0	TIMEOUT	INVREQ	MSGERR	OWRITE	OREAD
W												W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			0	0	HCRC	HPAR	0				TERM	WRABT	NACK	URUN	0
W						W1C	W1C					W1C	W1C	W1C	W1C	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-22 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
21 —	Reserved
20 TIMEOUT	Timeout Error Flag Indicates an error caused by the module stalling for too long in a frame. This stalling occurs when: • The transmit FIFO or receive FIFO is not handled, and the bus is stuck in the middle of a message. • No STOP is issued after data transfer (there is a gap between messages). • IBI manual is used and no decision has been made. The maximum stall period is 10 kHz or 100 µs. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
19 INVREQ	Invalid Request Error Flag Indicates an error caused by an invalid use of a request: • Not using IBI ACK NACK when stopped in manual hold for IBI acknowledgment. • Using a request other than ForceStop or ForceExit when in a message. Other requests are valid when the message is done. • Other mismatched uses (for example, IBI ACK NACK in normal states). **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
18 MSGERR	Message Error Flag Indicates an error caused by:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Trying to write to or read from the MWMSG_SDR register when in a DDR message. • Trying to write to or read from the MWMSG_DDR register when in an SDR message. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
17 OWRITE	Overwrite Error Flag Indicates an error caused by trying to write to Controller Write Data Byte (MWDATAB) when the FIFO is full. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
16 OREAD	Overread Error Flag Indicates an error caused by: • Trying to read from Controller Read Data Byte (MRDATAB) when the FIFO is empty. • Trying to read from the MRMSG_SDR register or the MRMSG_DDR register when no message has yet started. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Clear the flag
15-13 —	Reserved
12 —	Reserved
11 —	Reserved
10 HCRC	High Data Rate CRC Error Flag Indicates that a cyclic redundancy check (CRC) error occurred from a DDR read. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
9 HPAR	High Data Rate Parity Flag Indicates a parity error from a DDR read, including a bad preamble on a read. This does not stop the read because it is not safe to terminate; the read data may become misframed. Ends on a run of 1 second. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
8-5 —	Reserved
4	Terminate Error Flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
TERM	Indicates an error when this controller terminates a target read because the read exceeded the count for the message. This error is valid only when using the MWMSG_SDR or MWMSG_DDR register. If you write to the MWMSG_SDR or MWMSG_DDR register, this field automatically becomes 0. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
3 WRABT	Write Abort Error Flag Indicates an error caused by the I2C target NACKing the write data, terminating the message. For example, the controller is writing in I2C and the target NACKed the write. If you write to the MCTRL register, this field automatically becomes 0. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
2 NACK	Not Acknowledge Error Flag Indicates an error caused by the target or targets NACKing (not acknowledging) the last address. If 7Eh is the address, then it indicates all targets NACKed the last address. If you write to the MCTRL register, this field automatically becomes 0. **NOTE** In HDR mode, this error occurs when an address is not accepted (as opposed to NACKed). **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Error When writing 0 - No effect 1 - Clear the flag
1 URUN	Underrun Error Flag Indicates an underrun for HDR-BT. This error occurs when attempting to write from an empty transmit FIFO. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error 1 - Error When writing 0 - No effect 1 - Clear the flag
0 —	Reserved

63.7.36 Controller DMA Control (MDMACTRL)

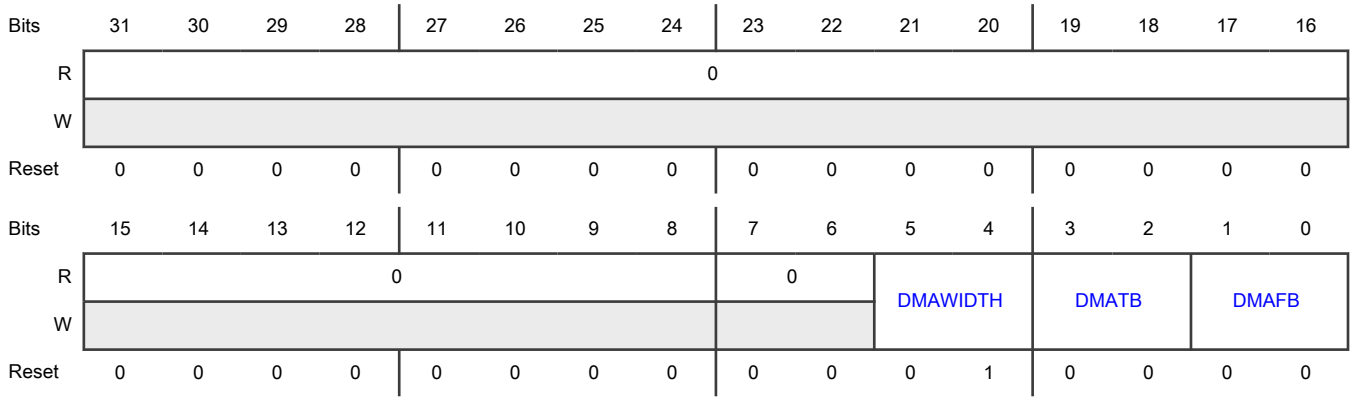
Allows DMA to be used for inbound and outbound messages. DMA is much more useful for a controller than for a target because the controller directs the bus traffic and actions. DMA can be used with a controller in these ways:

- Push or pull data for MCTRL[REQUEST] = 1 (EmitStartAddr) request written by the processor.
- Implementing message mode, completely controlled by DMA.

Offset

Register	Offset
MDMACTRL	A0h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-6 —	Reserved
5-4 DMAWIDTH	DMA Width Specifies the data width of DMA operations. The halfword (16 bits) setting ensures that two bytes are free or available in FIFO. 00,01 - Byte 10 - Halfword (16 bits) 11 - Reserved
3-2 DMATB	DMA to Bus Represents the DMA write (to-bus) trigger. When DMATB = 1 or DMATB = 2, I3C requests DMA when a transmit trigger occurs (see Controller Data Control (MDATACTRL)). I3C requests until full, unless the DMA is set up as a trigger. DMAFB becomes 0 when MSTATUS[ERRWARN] = 1. If this field is 1b, STOP or repeated START causes DMATB to become 0 (see SCONFIG[MATCHSS]). Normally, DMA enable must be used only in Controller Message mode. 00 - DMA not used 01 - Enable DMA for one frame (ended by DMA or terminated) 10 - Enable DMA until DMA is turned off. Recommended mode, instead of 01b, to enable DMA. This setting provides better software control. 11 - Reserved
1-0	DMA from Bus

Table continues on the next page...

Table continued from the previous page...

Field	Description
DMAFB	Represents the DMA read (from-bus) trigger. When DMAFB = 1 or DMAFB = 2, I3C requests DMA when a receive trigger occurs (see Controller Data Control (MDATACTRL)). I3C requests until empty unless the DMA is set up as a trigger. DMAFB becomes 0 when MSTATUS[ERRWARN] = 1. If this field is 1b, STOP or Repeated START causes DMAFB to automatically become 0 (see SCONFIG[MATCHSS]). **NOTE** Do not enable DMA after a transaction starts. It must be enabled only when enabling the controller and interrupts to avoid any kind of RX FIFO overrun. 00 - DMA not used 01 - Enable DMA for one frame 10 - Enable DMA until DMA is turned off. Recommended mode, instead of 01b, to enable DMA. This setting provides better software control. 11 - Reserved

63.7.37 Controller Data Control (MDATACTRL)

Assists in data control when there is no FIFO. This register also assists the FIFO when FIFO is available (regardless of size), and this assistance allows some control over the FIFO behavior. In particular, the register provides control over when to interrupt when a particular state of fullness or emptiness is reached. It also controls behavior related to width, when the width is not 1 byte wide.

This register acts as an alias of Target Data Control (SDATACTRL).

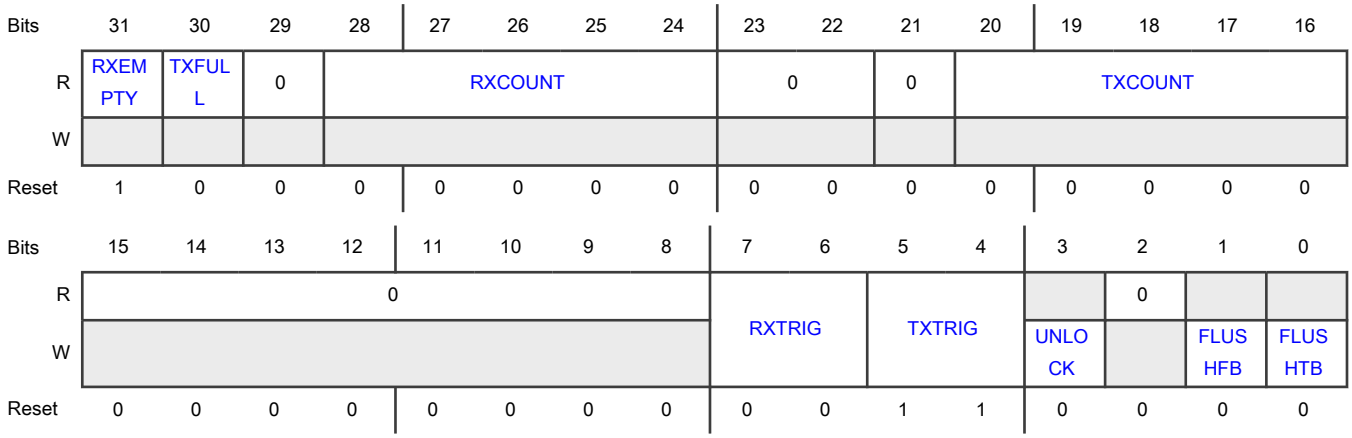
NOTE

When flushing a FIFO if DMA is in use, disable the DMA channel first. You must not use FIFO flush when a message (in that direction) is in flight.

Offset

Register	Offset
MDATACTRL	ACh

Diagram



Fields

Field	Description
31 RXEMPTY	Receive is Empty Indicates whether the receive FIFO or buffer is empty. 0 - Not empty 1 - Empty
30 TXFULL	Transmit is Full Indicates whether the transmit FIFO or buffer is full. 0 - Not full 1 - Full
29 —	Reserved
28-24 RXCOUNT	Receive Entry Count Contains the count of entries in the receive FIFO or buffer. Entry size for SDR and HDR-DDR is in bytes.
23-22 —	Reserved
21 —	Reserved
20-16 TXCOUNT	Transmit Entry Count Contains the count of entries waiting in the TXFIFO. This count is the number of entries that the application has written to the transmit FIFO that have not yet gone onto the bus. Entry size for SDR and HDR-DDR is in bytes.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-8 —	Reserved
7-6 RXTRIG	Receive Trigger Level Indicates the trigger level for receive fullness when using a FIFO. This field affects the RXPEND interrupt. 00 - Trigger when not empty (default) 01 - Trigger when 1/4 full or more 10 - Trigger when 1/2 full or more 11 - Trigger when 3/4 full or more
5-4 TXTRIG	Transmit Trigger Level Indicates the trigger level for transmit emptiness when using a FIFO. This field affects the TXNOTFULL interrupt. 00 - Trigger when empty 01 - Trigger when 1/4 full or less 10 - Trigger when 1/2 full or less 11 - Trigger when 1 less than full or less (default)
3 UNLOCK	Unlock Unlocks the functionality so that the RXTRIG and TXTRIG fields can be changed on a write. This field must be 1 (unlocked) in the same cycle when writing to TXTRIG or RXTRIG. If this field is 0 (locked), RXTRIG and TXTRIG fields cannot be changed on a write. 0 - Locked 1 - Unlocked
2 —	Reserved
1 FLUSHFB	Flush From-Bus Buffer or FIFO Flushes from-bus buffer or FIFO. You normally do not use this field. **NOTE** Before using FLUSHFB, disable the DMA (MDMACTRL[DMAFB] = 0). Then use FLUSHFB to flush the FIFO and then re-enable the DMA. 0 - No action 1 - Flush the buffer
0	Flush To-Bus Buffer or FIFO

Table continues on the next page...

Table continued from the previous page...

Field	Description
FLUSHTB	Flushes to-bus buffer or FIFO. This field is used when the controller terminates a to-bus message (read) prematurely. **NOTE** Before using FLUSHTB, disable the DMA (MDMACTRL[DMATB] = 0). Then use FLUSHTB to flush the FIFO and then re-enable the DMA. 0 - No action 1 - Flush the buffer

63.7.38 Controller Write Data Byte (MWDATAB)

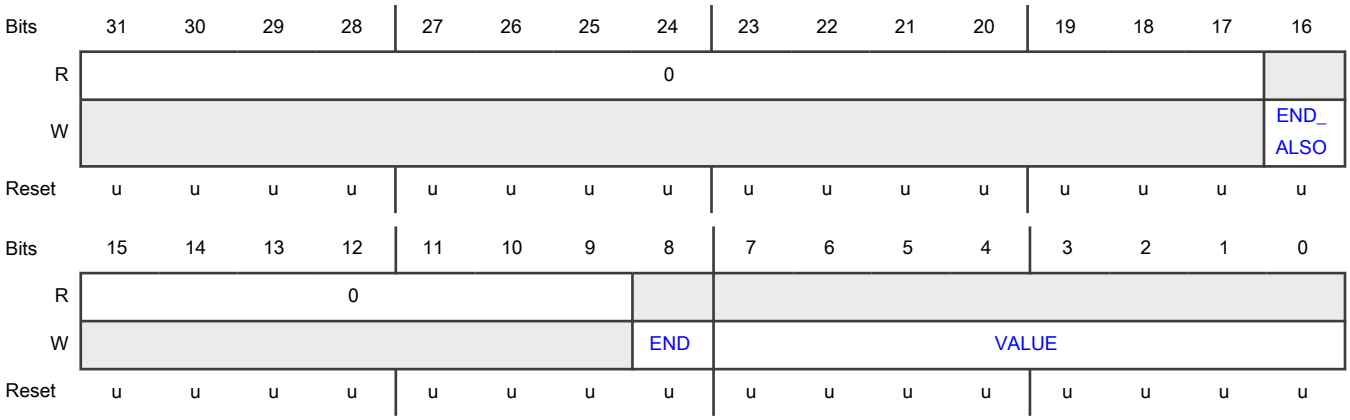
Allows writing bytes to send onto the bus. This register is used only when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

This register acts as an alias of Target Write Data Byte (SWDATAB).

Offset

Register	Offset
MWDATAB	B0h

Diagram



Fields

Field	Description
31-17 —	Reserved
16	End of Message ALSO

Table continues on the next page...

Table continued from the previous page...

Field	Description
END_ALSO	Indicates end of message, used to end an outbound message normally. Every message must indicate when it is the last message to be sent. This method can be used with the MDATABE register. This field is required for I3C and is optional for I2C. For HDR-DDR, the byte with END_ALSO must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs. If this field is 0, more bytes are assumed to be in the message. If this field is 1, the END bit marks the last byte of the message. 0 - Not the end 1 - End
15-9 —	Reserved
8 END	End of Message Indicates the end of a message. This field is required for I3C and is optional for I2C. For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs. If this field is 0, more bytes are assumed to be in the message. If this field is 1, the END bit marks the last byte of the message. 0 - Not the end 1 - End
7-0 VALUE	Data Byte Represents the byte written to the target (stored in transmit FIFO): • I3C computes the parity. • I2C manages the ACK or NACK.

63.7.39 Controller Write Data Byte End (MWDATABE)

Allows writing the last byte to send onto the bus. This register is used only when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

This register acts as an alias of Target Write Data Byte End (SWDATABE).

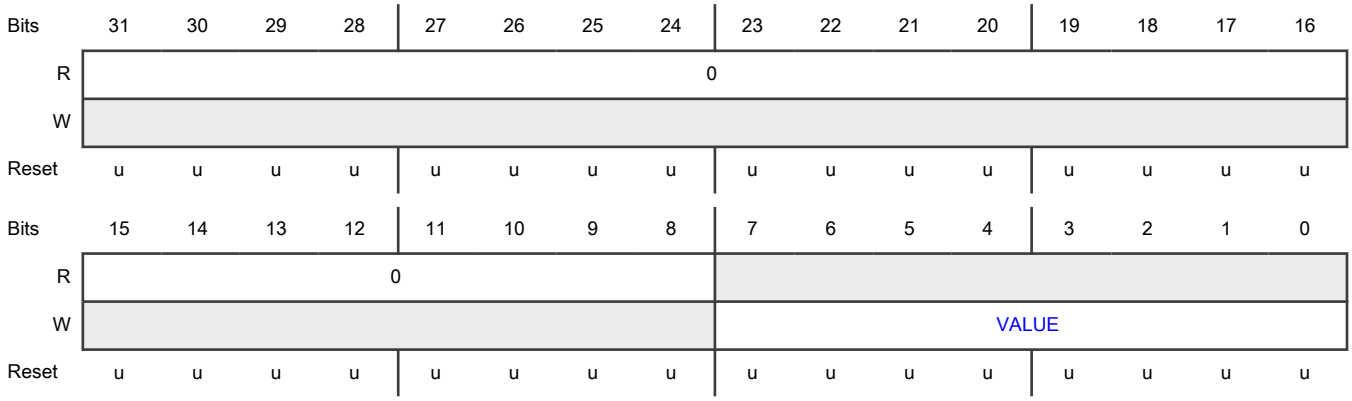
NOTE

MWDATABE can also indicate END using bit 8.

Offset

Register	Offset
MWDATABE	B4h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 VALUE	Data Represents the last byte written to the target (stored in transmit FIFO): • I3C computes the parity. • I2C manages the ACK or NACK.

63.7.40 Controller Write Data Halfword (MWDATAH)

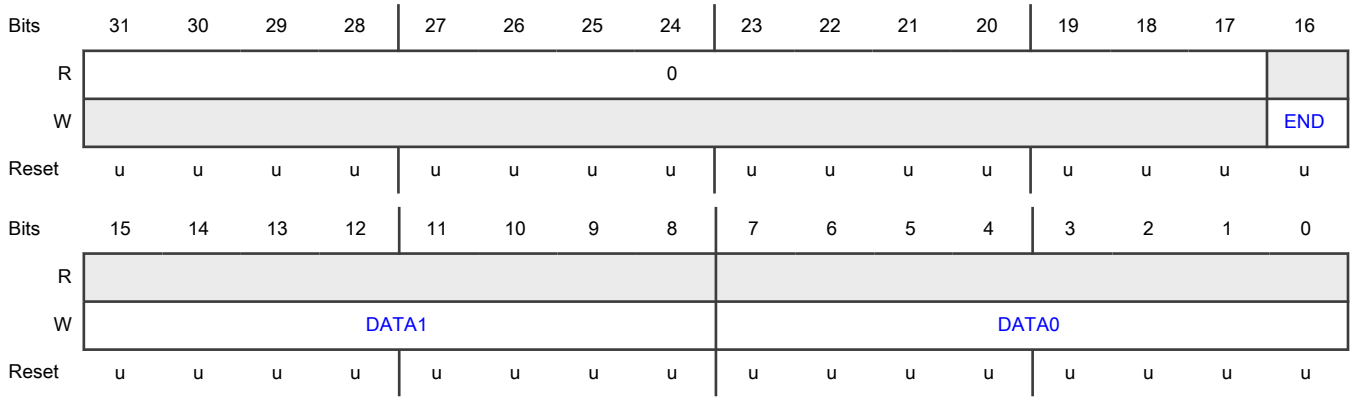
Allows writing a halfword (pair of bytes) to the bus unless an external FIFO is used, sending the low byte followed by the high byte. An end-of-data (last) marker bit is allowed (or must be 0). A halfword must not be written unless there is room for both, as indicated by the use of transmit FIFO level trigger or MDATACTRL[TXCOUNT].

This register acts as an alias of Target Write Data Halfword (SWDATAH).

Offset

Register	Offset
MWDATAH	B8h

Diagram



Fields

Field	Description
31-17 —	Reserved
16 END	End of Message Indicates the end of a message. For this register, this field always marks DATA1 as the end. This field is required for I3C and is optional for I2C. For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte pairs. If this field is 0, more bytes are assumed to be in the message. If this field is 1, the END bit marks the last byte of the message. 0 - Not the end 1 - End
15-8 DATA1	Data Byte 1 Represents the second byte that is sent to the target (written to transmit FIFO).
7-0 DATA0	Data Byte 0 Represents the first byte that is sent to the target (written to transmit FIFO).

63.7.41 Controller Write Data Halfword End (MWDATAHE)

Allows writing a halfword of data, which is the end (the last byte of the halfword is the end). The target writes the halfword (byte pair) in the same way as it does to Controller Write Data Halfword (MWDATAH), but marks the second byte as end-of-data (last byte).

For HDR-DDR, the byte with the END must be even (second, fourth, sixth, and so on) because DDR uses byte pairs.

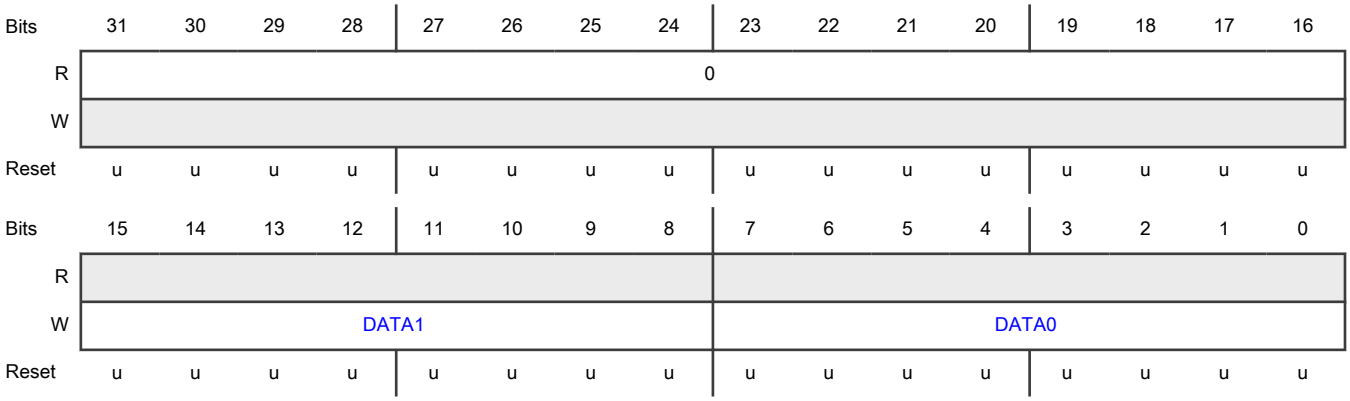
You must not write a halfword unless there is room for both, as indicated by the use of transmit FIFO level trigger or MDATACTRL[TXCOUNT].

This register acts as an alias of Target Write Data Halfword End (SWDATAHE).

Offset

Register	Offset
MWDATAHE	BCh

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 DATA1	Data Byte 1 Represents the second byte that is sent to the target (written to transmit FIFO).
7-0 DATA0	Data Byte 0 Represents the first byte that is sent to the target (written to transmit FIFO).

63.7.42 Controller Read Data Byte (MRDATAB)

Allows reading bytes written by the target after an SDR read, or DAA or DDR. This register is used only when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

This register acts as an alias of Target Read Data Byte (SRDATAB).

Offset

Register	Offset
MRDATAB	C0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								VALUE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7-0 VALUE	Value Represents the byte read from the controller (and written by the target).

63.7.43 Controller Write Byte Data 1 (to Bus) (MWDATAB1)

Allows writing bytes to send onto the bus, and is intended for DMAs that do not clear the upper bits of the word. This register does not have the END bits and is only used when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

Offset

Register	Offset
MWDATAB1	CCh

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W									VALUE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7-0 VALUE	Value Represents the byte to write out. The I3C module computes the parity for I3C, or manages the ACK or NACK for I2C.

63.7.44 Controller Write Message Control in SDR mode (MWMSG_SDR_CONTROL)

Allows setting up and writing 16-bit words in Single Data Rate (SDR) mode. The MWMSG_SDR_ register is modal and has two modes—control and data:

- For the first write to set up a new message, this register functions as the MWSMSG_SDR_CONTROL register.
- For subsequent writes, this register functions as MWMSG_SDR_DATA.
- The control information is not pipelined. You must write to control information registers for a start and do not write to them until data is sent.

When not in the middle of a message, the MWMSG_SDR_CONTROL register is used to start a new message (or emit a STOP). After starting a message, the MWMSG_SDR_DATA register is used until the length (see the LEN field) counts down, or until data with END = 1 is used.

The MWMSG_SDR_CONTROL register is written with the 16-bit control word if currently stopped or after the end of the previous message. If MSTATUS[STATE] = 2 (MSGSDR) and MSTATUS[BETWEEN] = 0, the register (at this offset address) functions as the MWMSG_SDR_DATA register. Otherwise, this register (at this offset address) functions as the MWMSG_SDR_CONTRL register, as long as MSTATUS[STATE] is not in another mode.

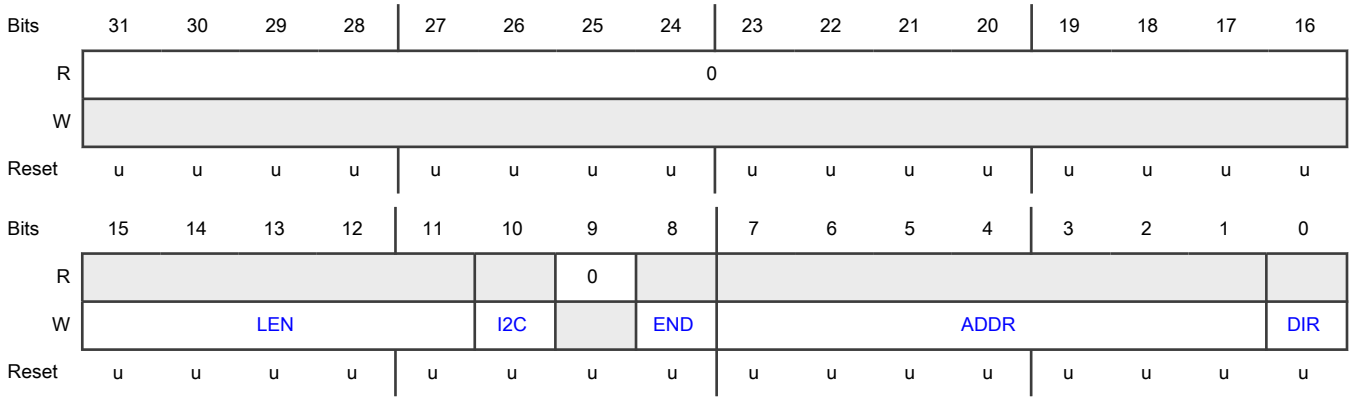
The control word contains the byte length (6-bit), address, direction, and how it ends (stop, ready for next, continuation with more length). If the command is START and an event (IBI, CR, HJ) occurs, MCTRL[IBIRESP] is used to determine action, and the corresponding interrupt occurs. In that case, the message is restarted.

The MWMSG_SDR_CONTROL and MWMSG_SDR_DATA registers are only targeted for DMA operations, but the processor can also write to the MWMSG_SDR_CONTROL register (instead of using MCTRL and MxDATAB).

Offset

Register	Offset
MWMSG_SDR_CONTROL	D0h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-11 LEN	Length Indicates the byte length of the message. If LEN = 0, then only END is used (and it does not use the address if stopped). LEN = 1 must not be used. The minimal LEN size is 2 bytes (LEN = 2).
10 I2C	I2C Specifies whether the message is I2C or I3C. 0 - I3C message 1 - I2C message
9 —	Reserved
8 END	End of SDR Message Indicates the end of SDR message. MSTATUS[COMPLETE] = 1 when done. The end can happen: • Before LEN bytes are read in I3C mode, if a target ends sooner. • Before LEN bytes are written in I2C mode, if a target NACKs. If this field is 0, the SDR message ends waiting for a new SDR message (issues a Repeated START for a new message). If this field is 1, the SDR message ends at the STOP. 0 - Not the end 1 - End
7-1 ADDR	Address Contains the address to be written.

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 DIR	Direction 0 - Write 1 - Read

63.7.45 Controller Write Message Data in SDR mode (MWMSG_SDR_DATA)

Contains the 16-bit word to be written in Single Data Rate (SDR) mode. This register functions in a way similar to MWMSG_SDR_CONTROL.

Offset

Register	Offset
MWMSG_SDR_DATA	D0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA16B															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-17 —	Reserved
16 —	Reserved
15-0 DATA16B	Data

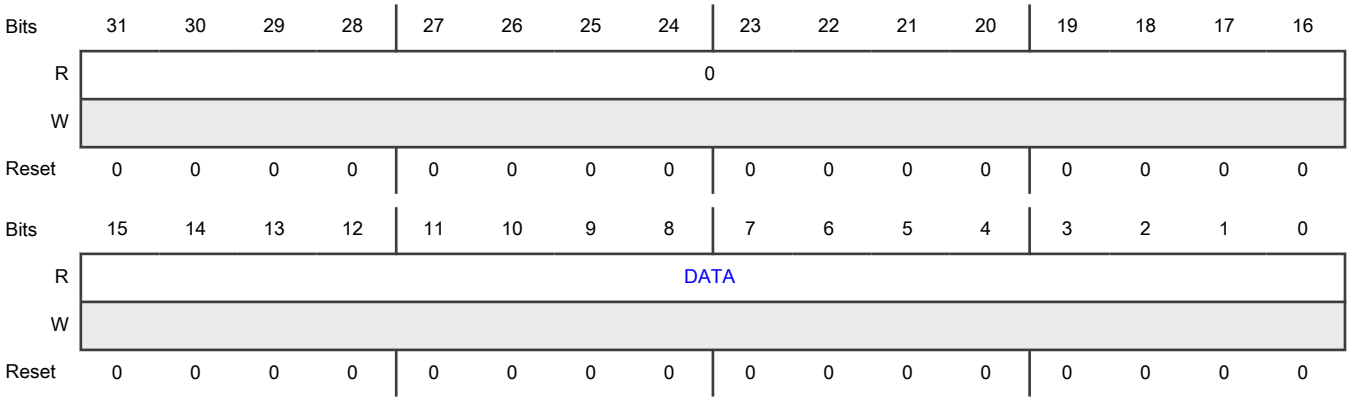
63.7.46 Controller Read Message in SDR mode (MRMSG_SDR)

Allows reading 16-bit words from a target in SDR Message mode. The MRMSG_SDR register is used to read 16-bit words from an active message started with MWMSG_SDR. These words are intended to be read by DMA.

Offset

Register	Offset
MRMSG_SDR	D4h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 DATA	Data Contains the 16-bit word read from the target: • If the length (LEN) is an odd number, the upper byte is 0. • If the target ends before LEN is finished, the module treats the read as completed. • If the target is not done before LEN is finished and END is not a continuation, the read is terminated.

63.7.47 Controller Write Message in DDR mode: First Control Word (MWMSG_DDR_CONTROL)

Allows setting up and writing 16-bit words in DDR mode. The register has three modes:

- The first write to set up a new message functions as the MWSMSG_DDR_CONTROL register.
- The second write contains the functions as shown in the MWMSG_DDR_CONTROL2 register.
- For subsequent writes, this register functions in a way similar to Controller Write Message Data in DDR mode (MWMSG_DDR_DATA).
- The control information is not pipelined. You must write to control information registers for a start and do not write to them until data is sent.

When not in the middle of a message, the MWMSG_DDR_CONTROL register is used to start a new message (or emit a STOP). After starting a message, the MWMSG_DDR_DATA register is used until the length (see LEN field) counts down or until data with END = 1 is used.

The MWMSG_DDR_CONTROL register is written with the 16-bit control word if currently stopped or after the end of the previous message. If MSTATUS[STATE] = 4 (MSGDDR) and MSTATUS[BETWEEN] = 0, then the register (at this offset address) functions as the MWMSG_DDR_DATA register. Otherwise, this register (at this offset address) functions as the MWMSG_DDR_CONTROL register, as long as MSTATUS[STATE] is not in another mode.

The main control word contains the 16-bit word length and how it ends (stop, ready for next, continuation with more length). Then the command word contains the command and address for read or write. If the command is START and an event (IBI, CR, HJ) occurs, MCTRL[IBIRESP] is used to determine action, and the corresponding interrupt occurs. In that case, the message is restarted.

The MWMSG_DDR_CONTROL, MWMSG_DDR_CONTROL2, and MWMSG_DDR_DATA registers are only targeted for DMA operations, but the processor can also write to the MWMSG_DDR_CONTROL register (instead of using MCTRL and MxDATAB).

NOTE

The module handles preamble, parity, and CRC.

Offset

Register	Offset
MWMSG_DDR_CONTROL	D8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	ADDRCMD															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Description
31-16 —	Reserved
15-0 ADDRCMD	Address Command Indicates the first data write after control write, with LEN ≠ 0. This is formatted as:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Bits 15:9: target address to read or write • Bit 8: reserved, must be 0 • Bit 7: 1 if read, 0 if write • Bits 6:0: CMD as 7-bit value, always for controller

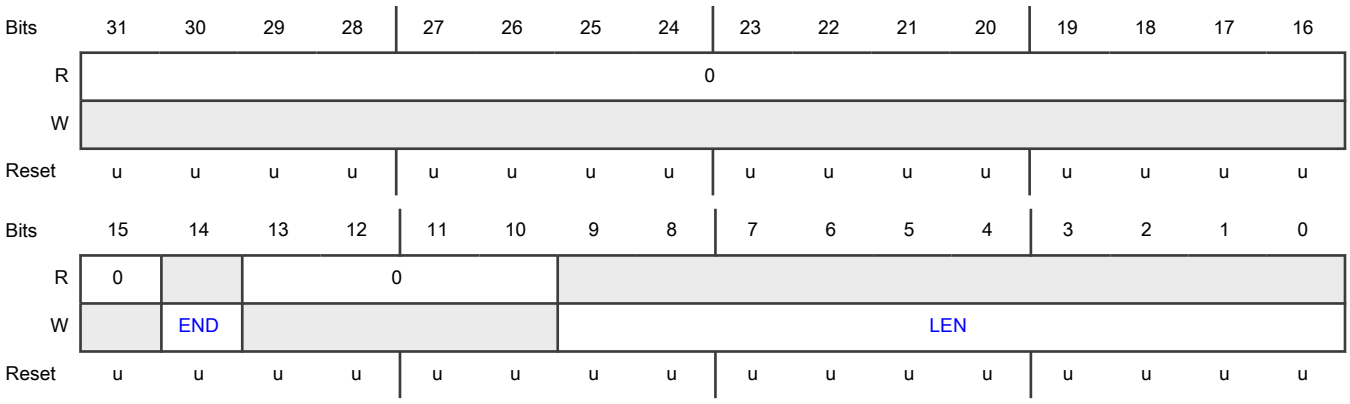
63.7.48 Controller Write Message in DDR Mode Control 2 (MWMSG_DDR_CONTROL2)

Contains the second control word instructions with the length of message and end.

Offset

Register	Offset
MWMSG_DDR_CONTROL2	D8h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 —	Reserved
14 END	End of Message Indicates the end of DDR message. MSTATUS[COMPLETE] becomes 1 when done. The end can happen before LEN bytes are read if the target ends sooner.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If this field is 0, DDR message ends waiting for a new DDR message (issues an HDR restart for the new message). If this field is 1, DDR message ends on HDR exit. 0 - Not the end 1 - End
13-10 —	Reserved
9-0 LEN	Length of Message Contains the length of the message (including the command) in halfwords, up to 2046 bytes. If LEN = 0, then only END is applied: • For reads, + 1 for the CRC. For example, to read 4 bytes (2 halfwords), use 1 + 2 + 1 for CMD + 2 halfwords + CRC. • For writes, LEN is the number of halves of data bytes + 1 (for command).

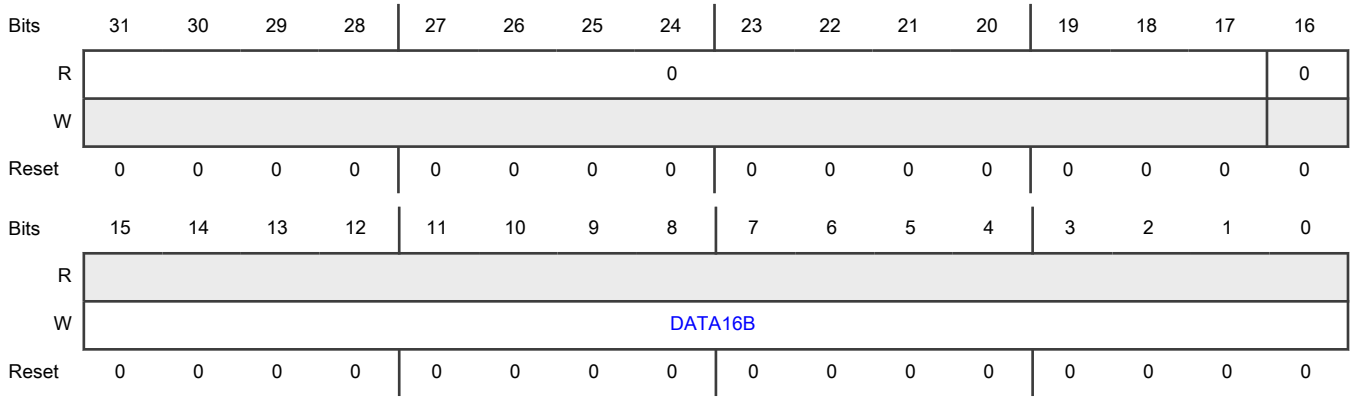
63.7.49 Controller Write Message Data in DDR mode (MWMSG_DDR_DATA)

Contains the 16-bit word to be written in DDR mode. This register functions in a way similar to Controller Write Message in DDR mode: First Control Word (MWMSG_DDR_CONTROL).

Offset

Register	Offset
MWMSG_DDR_DATA	D8h

Diagram



Fields

Field	Description
31-17 —	Reserved
16 —	Reserved
15-0 DATA16B	Data

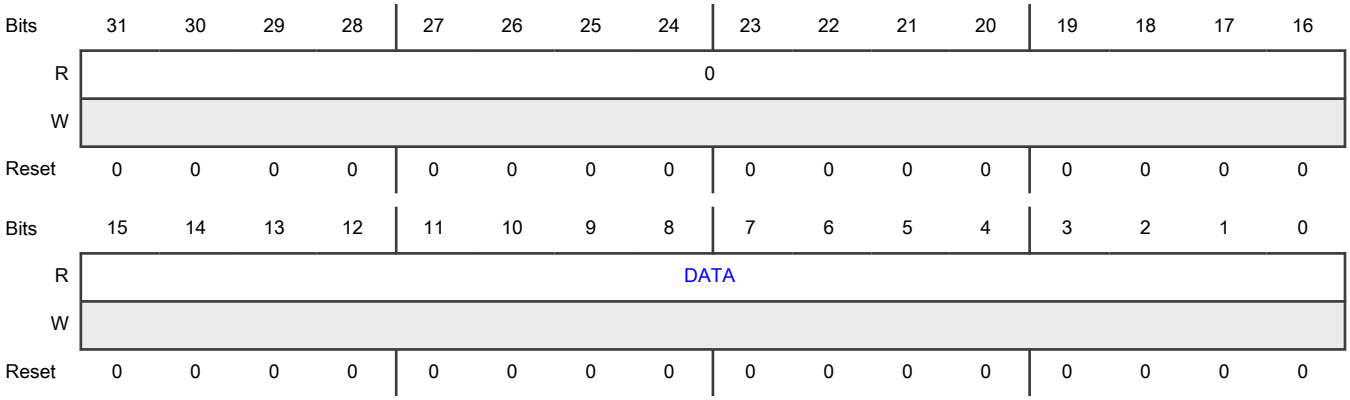
63.7.50 Controller Read Message in DDR mode (MRMSG_DDR)

Allows reading 16-bit words from a target in DDR Message mode from an active message started with MWMSG_DDR. These words are intended to be read by DMA.

Offset

Register	Offset
MRMSG_DDR	DCh

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 DATA	Data Contains the 16-bit word read from a target. The first byte is the LSB, and is in DATA[7:0]. The second byte is the MSB, and is in DATA[15:8]:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• If the target ends before the entire length of the message (MWMSG_DDR[LEN]) is read, the module considers the DATA read as completed. In I3C mode, the target can indicate the end of message (the last byte). Otherwise, the controller terminates the message if the message is more than the controller can accept. • If the target has not yet finished sending DATA before the entire length of the message (MWMSG_DDR[LEN]) is read and END is not a continuation, the DATA read is terminated.

63.7.51 Controller Dynamic Address (MDYNADDR)

Allows the I3C module to write its own dynamic address (DA) when the module changes from Controller mode to Target mode.

If this device is the main controller (the controller during bus initialization), then the device may use this mechanism to assign itself its DA. When the device hands off control to a secondary controller, it becomes a target itself. This DA must be written before switching to Target mode and must not be changed once in Target mode (it is not clock-safe to do so). It must be written with a valid address value in DADDR if DAVALID = 1.

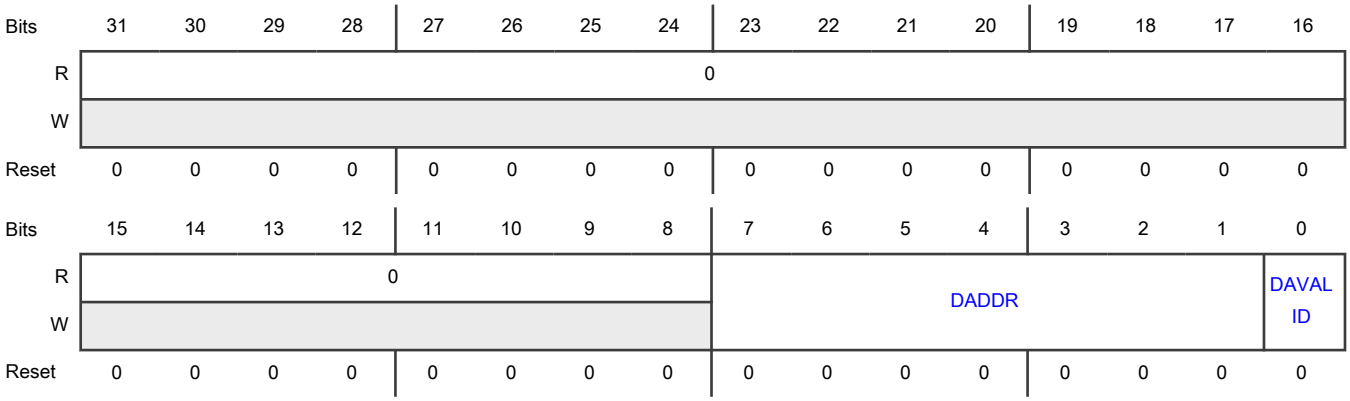
NOTE

The main controller also uses DEFSLVS CCC to define the target addresses, including itself. This mechanism is how secondary controllers know this address. If the controller is not the main controller, then this mechanism must not be used.

Offset

Register	Offset
MDYNADDR	E4h

Diagram



Fields

Field	Description
31-8	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
7-1 DADDR	Dynamic Address Contains the assigned dynamic address when DAVALID = 1.
0 DAVALID	Dynamic Address Valid 0 - No valid DA assigned 1 - Valid DA assigned

63.7.52 Timing Rules for Target Reset Recovery (SRSTACTTIME)

Indicates the timing rules for a recovery following a TargetReset.

When the controller issues a TargetReset, some part of the device may be reset, such as a peripheral, the whole device, or some subset of the device. The controller must know how long the device takes to recover and be ready for new I3C messages. This behavior may be hard-coded or configured using this register.

This register is used if the device is enabled for MMR.

Offset

Register	Offset
SRSTACTTIME	100h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SYSRSTTIM								PERRSTTIM							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
23-16 —	Reserved
15-8 SYSRSTTIM	Time to Recover from Chip Reset Specifies the time to recover, in seconds, from a full system or chip reset.
7-0 PERRSTTIM	Time to Recover from the I3C Peripheral Specifies the time to recover, in milliseconds, from the I3C peripheral reset.

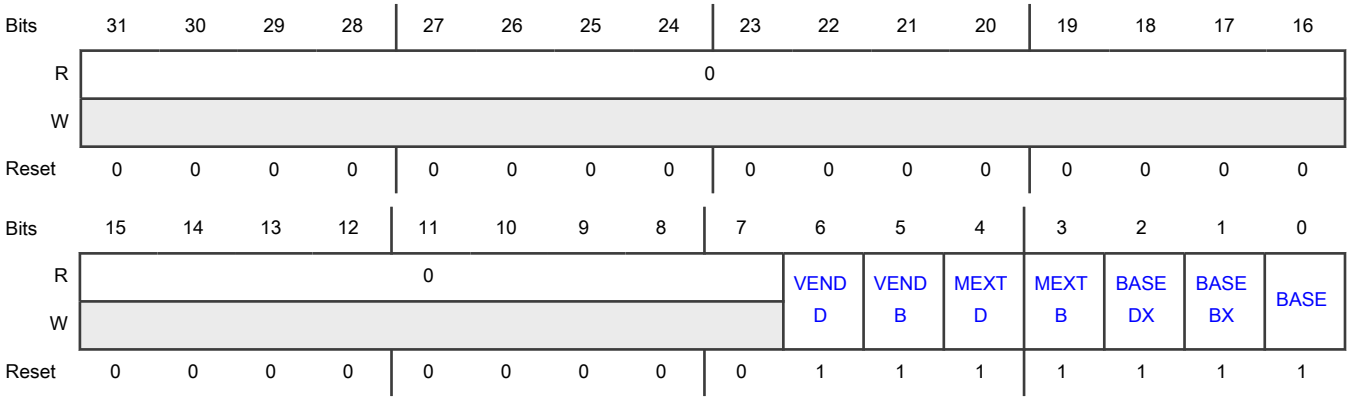
63.7.53 CCC Mask for Unhandled CCCs (SCCCMASK)

Allows masking of unhandled (not handled by hardware) CCC commands, when enabled.

Offset

Register	Offset
SCCCMASK	10Ch

Diagram



Fields

Field	Description
31-7 —	Reserved
6 VENDD	VENDD Determines whether any unhandled CCC commands in the range of E4h to FFh are passed to the application or suppressed.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Suppressed 1 - Passed to application
5 VENDB	VENDB Determines whether any unhandled CCC commands in the range of 65h to 7Fh are passed to the application or suppressed. 0 - Suppressed 1 - Passed to application
4 MEXTD	MEXTD Determines whether any unhandled CCC commands in the range of C0h to E3h are passed to the application or suppressed. 0 - Suppressed 1 - Passed to application
3 MEXTB	MEXTB Determines whether any unhandled CCC commands in the range of 49h to 64h are passed to the application or suppressed. 0 - Suppressed 1 - Passed to application
2 BASEDX	BASEDX Determines whether any unhandled CCC commands in the range of A0h to BFh are passed to the application or suppressed. 0 - Suppressed 1 - Passed to application
1 BASEBX	BASEBX Determines whether any unhandled CCC commands in the range of 20h to 48h are passed to the application or suppressed. 0 - Suppressed 1 - Passed to application
0 BASE	Base Determines whether any unhandled CCC commands in the range of 0h to 1Fh and 80h to 8Fh are passed to the application or suppressed. 0 - Suppressed 1 - Passed to application

63.7.54 Target Errors and Warnings Mask (SERRWARNMASK)

Allows masking of target errors and warnings.

Target Errors and Warnings (SERRWARN) contains I3C protocol errors and issues detected on the line. If any are 1, SSTATUS[ERRWARN] becomes 1. The corresponding interrupt field also becomes 1, if enabled. SERRWARNMASK allows masking them, so only some form the status bit.

Offset

Register	Offset
SERRWARNMASK	110h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0	0	S0S1	HCRC	HPAR	SPAR	0			INVST ART	TERM	URUN NACK	URUN	ORUN
W																
Reset	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1

Fields

Field	Description
31-14 —	Reserved
13 —	Reserved
12 —	Reserved
11 S0S1	S0S1 Mask Determines whether to allow SERRWARN[S0S1] to contribute to the status of SSTATUS[ERRWARN]. 0 - Deny 1 - Allow
10 HCRC	HCRC Mask Determines whether to allow SERRWARN[HCRC] to contribute to the status of SSTATUS[ERRWARN]. 0 - Deny 1 - Allow

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 HPAR	HPAR Mask Determines whether to allow SERRWARN[HPAR] to contribute to the status of SSTATUS[ERRWARN]. 0 - Deny 1 - Allow
8 SPAR	SPAR Mask Determines whether to allow SERRWARN[SPAR] to contribute to the status of SSTATUS[ERRWARN]. 0 - Deny 1 - Allow
7-5 —	Reserved
4 INVSTART	INVSTART Mask Determines whether to allow SERRWARN[INVSTART] to contribute to the status of SSTATUS[ERRWARN]. 0 - Deny 1 - Allow
3 TERM	TERM Mask Determines whether to allow SERRWARN[TERM] to contribute to the status of SSTATUS[ERRWARN]. 0 - Deny 1 - Allow
2 URUNNACK	URUNNACK Mask Determines whether to allow SERRWARN[URUNNACK] to contribute to the status of SSTATUS[ERRWARN]. 0 - Deny 1 - Allow
1 URUN	URUN Mask Determines whether to allow SERRWARN[URUN] to contribute to the status of SSTATUS[ERRWARN]. 0 - Deny 1 - Allow
0 ORUN	ORUN Mask Determines whether to allow SERRWARN[ORUN] to contribute to the status of SSTATUS[ERRWARN]. 0 - Deny 1 - Allow

63.7.55 Map Feature Control 0 (SMAPCTRL0)

Provides map feature control.

The SMAPCTRL*n* registers are named SMAPCTRL0, SMAPCTRL1, and so on based on the number of mapped addresses. MAPCTRL0 represents the primary DA or SA with SMAPCTRL1 onwards being the mapped addresses.

The features of the SMAPCTRL*n* registers depend on configuration. SMAPCTRL0 acts differently, as described in this chapter.

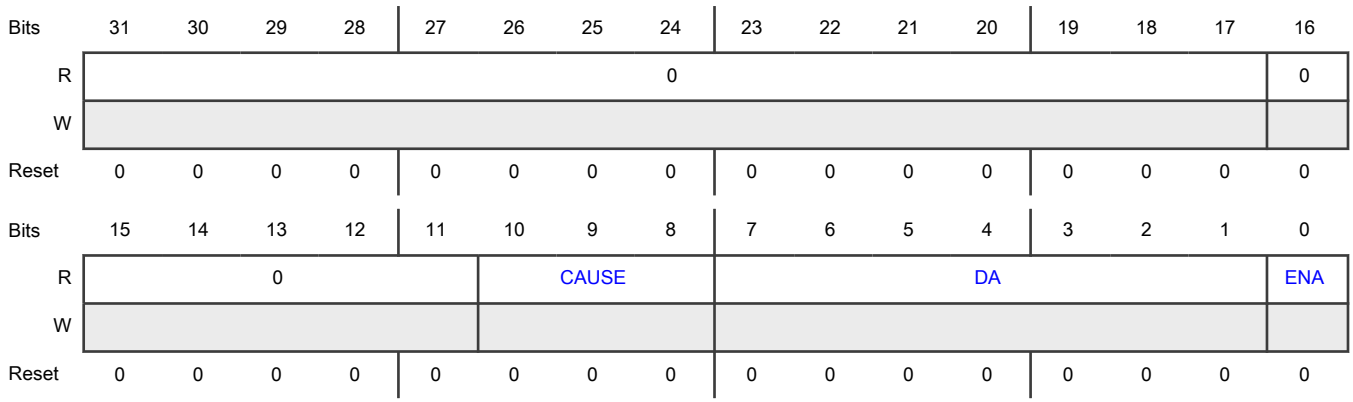
In general, this mechanism is intended to replace the DYNADDR register for all MAP-related uses.

When using the Auto-MAP and DASA or AASA, the slot is changed from SA to DA. If the controller then issues RSTDAA, the application must rewrite the static addresses and enable them because they are marked disabled.

Offset

Register	Offset
SMAPCTRL0	11Ch

Diagram



Fields

Field	Description
31-17 —	Reserved
16 —	Reserved
15-11 —	Reserved
10-8 CAUSE	Cause Indicates the cause of the most recent DA assignment, which lead to SSTATUS[DACHG] interrupt. This field has MAP enabled.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	If this field is 100b (auto MAP change happened last), the change may have changed this DA as well (for example, ENTDAA and SETAASA), but at least one MAP entry automatically changed after that. 000 - No information (this value occurs when not configured to write DA) 001 - Set using ENTDAA 010 - Set using SETDASA, SETAASA, or SETNEWDA 011 - Cleared using RSTDAA 100 - Auto MAP change happened last All other values are reserved.
7-1 DA	Dynamic Address Contains primary DA when ENA = 1. When ENA = 0, static address is used (but not described here). This field has MAP enabled.
0 ENA	Enable Primary Dynamic Address Indicates whether the MAP is enabled. 0 - Disabled 1 - Enabled

63.7.56 Map Feature Control 1 (SMAPCTRL1)

The SMAPCTRL*n* registers are named SMAPCTRL0, SMAPCTRL1, and so on based on the number of mapped addresses. SMAPCTRL0 represents the primary DA or SA, with SMAPCTRL1 onwards being mapped addresses.

The features of the SMAPCTRL*n* registers depend on configuration. SMAPCTRL0 acts somewhat differently, as described in this chapter.

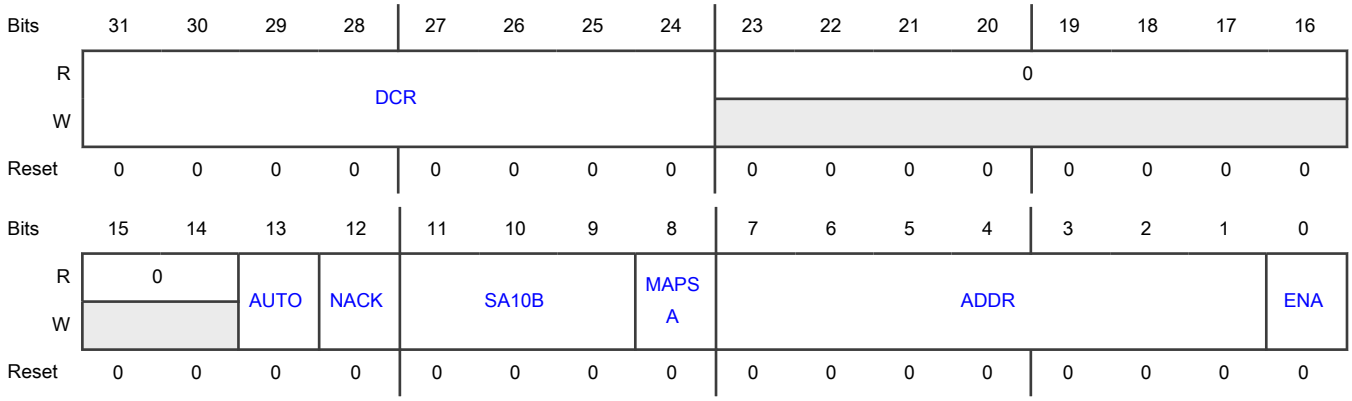
In general, this mechanism is intended to replace the DYNADDR register for all MAP-related uses.

When using the AutoMAP and DASA or AASA, the slot is changed from static address to dynamic address. If the controller then issues RSTDAA, the application must rewrite the SAs and enable them because as they are marked disabled.

Offset

Register	Offset
SMAPCTRL1	120h

Diagram



Fields

Field	Description
31-24 DCR	DCR Specifies DCR value. If DCR is to be replaced by register, this field provides the DCR value. Applicable if enabled for auto DAA using MMR. If no DCR, this field is used as extension of PID field for a total of 18 bits.
23-14 —	Reserved
13 AUTO	Auto DAA Indicates whether this slot is enabled for Auto DAA, if ENA = 0. This field is used if enabled for auto DAA using MMR. 0 - Disabled 1 - Enabled
12 NACK	Not Acknowledged Indicates whether messages to this address are always NACKed, if enabled. 0 - Do not always NACK messages 1 - Always NACK messages
11-9 SA10B	Static Address 10-Bit Extension Contains a 10-bit extension for a static address, if this field is in MAPCTRL1. In the case of a normal address, this field is 0; otherwise, it contains the 10-bit extension. This field is used if enabled for 10-bit.
8 MAPSA	MAP Static Address Indicates whether MAP entry is a static address or a dynamic address, if ENA = 1.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field has MAP enabled. 0 - I3C dynamic address 1 - Static address (I2C style)
7-1 ADDR	Address Contains static or dynamic address. If ENA = 1, contains static address or dynamic address (as indicated by the MAPSA field) with MAP enabled.
0 ENA	Enable Enables MAP entry. This entry could be a static or dynamic address with MAP enabled. 0 - Disable 1 - Enable

63.7.57 Extended IBI Data 1 (IBIEXT1)

Contains extended IBI data.

The CTRL register is used to submit IBI, CR, and Hot-Join when enabled to do so. If allowed, an IBI may have additional bytes following the mandatory data byte (MDB). Extended IBI data is allowed when SCTRL[EXTDATA] = 1. If allowed, the extra bytes are indicated using these registers.

Offset

Register	Offset
IBIEXT1	140h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EXT3								EXT2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EXT1								0	MAX			0	CNT		
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

Fields

Field	Description
31-24 EXT3	Extra Byte 3 Contains the third extra byte.
23-16 EXT2	Extra Byte 2 Contains the second extra byte.
15-8 EXT1	Extra Byte 1 Contains the first extra byte.
7 —	Reserved
6-4 MAX	Maximum Indicates the maximum number of extra bytes allowed by configuration. This field is 0 if there are no extra bytes.
3 —	Reserved
2-0 CNT	Count Contains the number of extra bytes beyond the MDB to be used. This field is 0 if there are no extra bytes.

63.7.58 Extended IBI Data 2 (IBIEXT2)

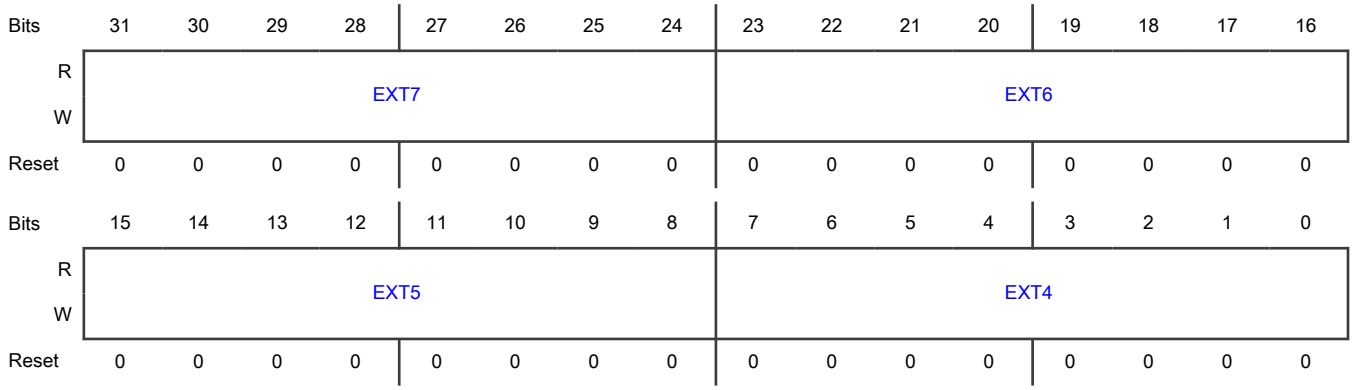
Contains extended IBI data.

The CTRL register is used to submit IBI, CR, and Hot-Join when enabled to do so. If allowed, an IBI may have additional bytes following the mandatory data byte (MDB). Extended IBI data is allowed when SCTRL[EXTDATA] = 1. If allowed, the extra bytes are indicated using these registers.

Offset

Register	Offset
IBIEXT2	144h

Diagram



Fields

Field	Description
31-24 EXT7	Extra Byte 7 Contains the seventh extra byte.
23-16 EXT6	Extra Byte 6 Contains the sixth extra byte.
15-8 EXT5	Extra Byte 5 Contains the fifth extra byte.
7-0 EXT4	Extra Byte 4 Contains the forth extra byte.

63.7.59 Self Reset (SELFRESET)

Resets the whole I3C module, if enabled. The reset is accepted if the key value is correct.

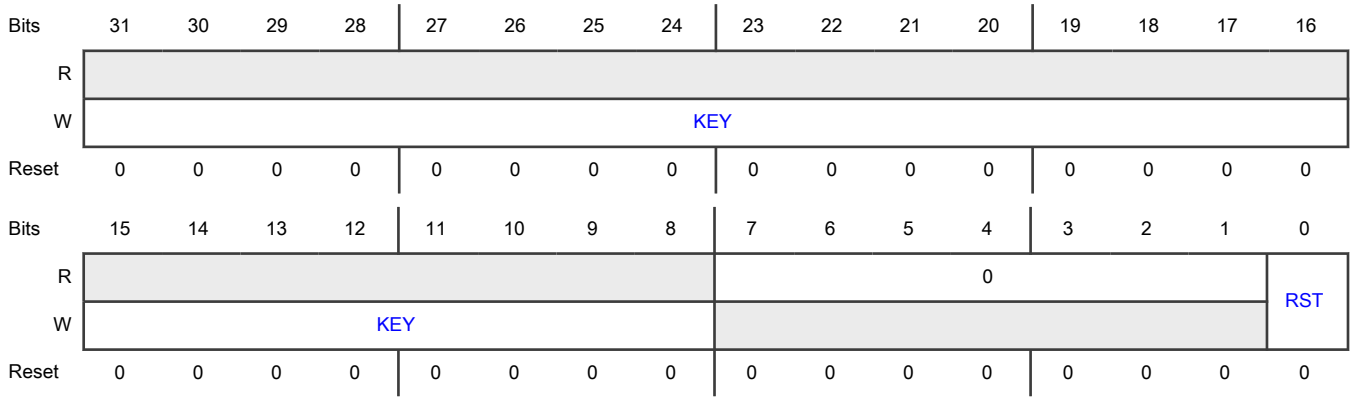
NOTE

In controller + target module, the controller is reset as well because they share resources.

Offset

Register	Offset
SELFRESET	FF0h

Diagram



Fields

Field	Description
31-8 KEY	Key Indicates information related to KEY. When requesting a reset (RST = 1), this field must be written with the key of B216E9h. Field reads back as 0.
7-1 —	Reserved
0 RST	Reset Starts a reset. Write 1 to start a reset (with a valid key). Reads back as 1 during the few bus cycles while the module is being reset. 0 - No reset 1 - Reset

63.8 Glossary

CCC	Common Command Codes
CDC	Clock-domain-crossing
CRC	Cyclic Redundancy Check
DDR	Double Data Rate
FM	Fast Mode
FM+	Fast Mode Plus
HDR	High Data Rate
HDR-BT	HDR Bulk Transfer
HDR-DDR	High Data Rate - Double Data Rate
HJ	Hot Join

IBI	In-Band interrupt
IRQ	Interrupt
CR	Controller Request
PIO	Programmable I/O
SDR	Single Data Rate

Chapter 64
Timers Overview

64.1 Overview

The following timers are supported in this chip:

- Low Power Periodic Interrupt Timer (LPIT): Two 32-bit counter timers that feature programmable count modulus, clock division features.
- Low-Power Timer (LPTMR): Two LPTMR timers, each with four timer channels with variety of controls for individual and multi-channel features.
- Timer/PWM Module (TPM): Six TPM modules, each has its own CCM clock root.
- Global System Counter (SCTR)
- Timestamp Timer Modules (TSTMR): Two satellite read-only Timestamp Timer Modules
- Watchdog timers (WDOG1-WDOG5): It is a high reliability independent timer that is available for system use

64.1.1 Low Power Periodic Interrupt Timer (LPIT)

LPIT is a low power periodic interrupt timer with four timer channels. It is a 32 bit resolution timer with programmable count modulus, clock division features, periodic interrupt generation, and ability to chain adjacent timers to achieve longer interval

64.1.2 Low-Power Timer (LPTMR)

The low-power timer (LPTMR) can be configured to operate as a time counter with optional prescaler, or as a pulse counter with optional glitch filter.

64.1.3 Global System Counter (SCTR)

The SCTR supports both a timer bus interface to the Cortex-A55 MPCore generic timers and a register interface for the Cortex-M33. Three 64kB regions are supported in the memory map: CNTControlBase, CNTReadBase and CNTCompareBase.

64.1.4 Low Power Timer/PWM (LPTPM)

TPM is a 32-bit timer with 4 channels. LPTPM1-2 in AONMIX, LPTPM3-6 in WAKEUPMIX. LPTPM1 and LPTPM3 timers are clocked from the bus clock, to ensure that LPTPM1 and LPTPM3's are synchronous with the CAN1 and CAN2. The other LPTPM instances have their own CCM clock root.

Each TPM includes 6 channels that can be configured for input capture, output compare, edge-aligned PWM mode, or center-aligned PWM mode.

64.1.5 Timestamp Timer Modules (TSTMR)

The TSTMR instances provide a 64-bit read-only snapshot of the value of the free-running incrementing counter generated by the SCTR.

64.1.7 Watchdog Timer (WDOG)

The WDOG module is an high reliability independent timer that is available for system use. It provides a safety feature to ensure that software is executing as planned and that the CPU is not stuck in an infinite loop or executing unintended code. If the WDOG module is not serviced (refreshed) within a certain period, it resets the CPU.

Main features of the WDOG module are:

- Configurable independent clock source input from bus clock

- Programmable time-out period
- Robust write sequence for counter refresh
- Windowed refresh option
- Optional timeout interrupt to allow post-processing diagnostics
- Configuration bits are write-once-after-reset to ensure watchdog configuration cannot be mistakenly altered
- Robust write sequence for unlocking write-once control/configuration bits

Chapter 65
Low-Power Timer (LPTMR)

65.1 Chip-specific LPTMR information

Table 601. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing

The chip includes two LPTMR Timer Modules (LPTMR1 in AONMIX and LPTMR2 in WAKEUPMIX). For each LPTMR, ALT1/2/3 are connected to pins, but ALT0 is connected to ADC trigger inputs. LPTMR input clocks in PSR[PCS] bits are connected as follows:

- 00 - ipg_clk_irclk: From CCM
- 01 - ipg_clk_1kHz: 32k_clk_root
- 10 - ipg_clk_32kHz: Fixed 32kHz
- 11 - ipg_clk_erclk: 32k_clk_root

CSR[TPS] Inputs are as shown in the table below. For more details please refer the Pinmux spreadsheet attached with the Reference Manual.

IO PAD	Alt0	Alt1	Alt2	Alt3	Alt4	Alt5	Alt6	Default MUX
ENET1_RD1	enet_qos.RGMII_RD1	uart3.CTS_B	xx	lptmr2.ALT1	flexio2.FLEXIO[11]	gpio4.IO[11]	xx	gpio4.IO[11]
ENET1_RD2	enet_qos.RGMII_RD2	xx	xx	lptmr2.ALT2	flexio2.FLEXIO[12]	gpio4.IO[12]	xx	gpio4.IO[12]
ENET1_RD3	enet_qos.RGMII_RD3	xx		lptmr2.ALT3	flexio2.FLEXIO[13]	gpio4.IO[13]	xx	gpio4.IO[13]
SD2_VSELECT	usdhc2.VSELECT	usdhc2.WP	lptmr2.ALT3	xx	flexio1.FLEXIO[19]	gpio3.IO[19]		gpio3.IO[19]
SD2_DATA3	usdhc2.DATA3	lptmr2.ALT1	mqs2.LEFT	xx	flexio1.FLEXIO[6]	gpio3.IO[6]	xx	gpio3.IO[6]
SD2_RESET_B	usdhc2.RESET_B	lptmr2.ALT2	xx	xx	flexio1.FLEXIO[7]	gpio3.IO[7]	xx	gpio3.IO[7]
PDM_CLK	pdm.CLK	mqs1.LEFT	xx	xx	lptmr1.ALT1	gpio1.IO[8]	can1.TX	gpio1.IO[8]

Table continues on the next page...

Table continued from the previous page...

PDM_BIT_STREAM0	pdm.BIT_STREAM[0]	mqs1.RIGHT	spi1.PCS1	tpm1.EXTCLK	lptmr1.ALT2	gpio1.IO[9]	can1.RX	gpio1.IO[9]
PDM_BIT_STREAM1	pdm.BIT_STREAM[1]	m33.NMI	spi2.PCS1	tpm2.EXTCLK	lptmr1.ALT3	gpio1.IO[10]		gpio1.IO[10]

65.2 Overview

You can configure LPTMR to operate as a time counter with an optional prescaler, or as a pulse counter with an optional glitch filter, across all power modes, including low-power modes. It is reset only on Power on Reset (POR) or Low Voltage Detect (LVD), allowing it to be used as a time-of-day counter.

65.2.1 Block diagram

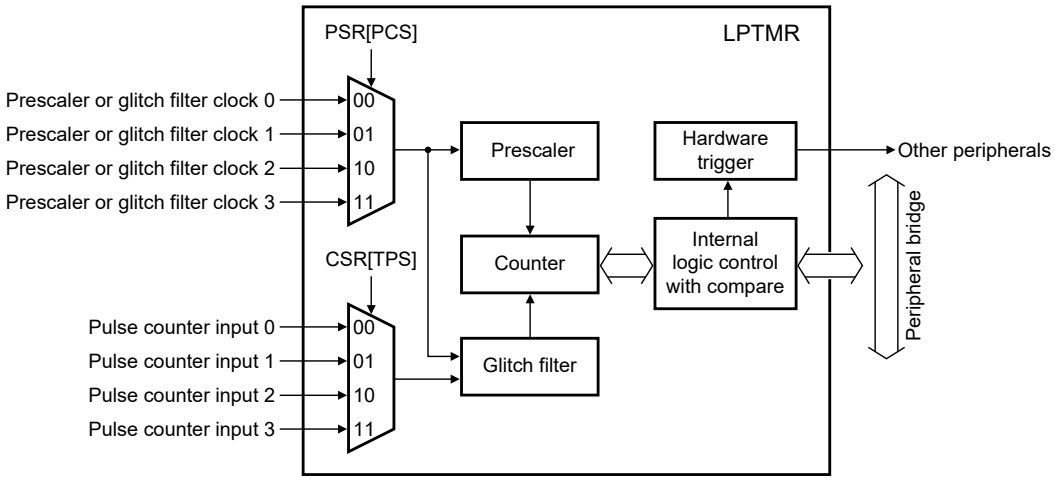


Figure 373. Block diagram

65.2.2 Features

- 32-bit time counter or pulse counter with compare:
  - Optional interrupt that can generate an asynchronous wake-up from any low-power mode
  - Hardware trigger output
  - Counter that supports a free-running mode or reset on compare
- Configurable clock source for prescaler and glitch filter
- Configurable input source for pulse counter (rising-edge or falling-edge)

65.3 Functional description

65.3.1 Low-power modes

In low-power modes, LPTMR continues to operate normally. You can configure LPTMR to exit a low-power mode by generating either an interrupt or a DMA request.

65.3.2 Clocks

The LPTMR prescaler and glitch filter can be clocked by one of the clocks that you configure by using PSR[PCS]. You must enable the clock source before you enable LPTMR.

In Pulse Counter mode, with the glitch filter bypassed, the selected input source directly clocks Counter (CNR), and no other clock source is required. To minimize power in this case, configure the glitch filter clock source for a clock that is disabled.

NOTE

- You may need to configure the clock source that you select in PSR[PCS] for it to remain enabled in low-power modes. Otherwise, LPTMR does not operate in low-power modes.
- The clock source or pulse input source selected for LPTMR must not exceed the maximum frequency of f_{LPTMR} defined in the chip data sheet.

65.3.3 Reset

LPTMR is reset only on Power on Reset (POR) or Low Voltage Detect (LVD). When configuring LPTMR registers, you must initially write to Control Status (CSR) with LPTMR disabled, before configuring Prescaler and Glitch Filter (PSR) and Compare (CMR). Then, you must write 1 to CSR[TEN] as the last step in the initialization. Doing so ensures that LPTMR is configured correctly and the LPTMR counter is reset to zero following a Power on Reset (POR) or Low Voltage Detect (LVD).

65.3.4 Prescaler and glitch filter

The LPTMR prescaler and glitch filter share the same logic, which operates as a prescaler in Time Counter mode and as a glitch filter in Pulse Counter mode.

NOTE

You must not alter the prescaler and glitch filter configuration when LPTMR is enabled.

65.3.4.1 Prescaler enabled

In Time Counter mode, when the prescaler is enabled, the output of the prescaler directly clocks Counter (CNR). When LPTMR is enabled, CNR increments every 2^1 to 2^{16} prescaler clock cycles. After LPTMR is enabled, the first increment of CNR takes an additional one or two prescaler clock cycles because of the synchronization logic.

65.3.4.2 Prescaler bypassed

In Time Counter mode, when the prescaler is bypassed, the selected prescaler clock increments Counter (CNR) on every clock cycle. When LPTMR is enabled, the first increment takes an additional one or two prescaler clock cycles because of the synchronization logic.

65.3.4.3 Glitch filter enabled

In Pulse Counter mode, when the glitch filter is enabled, the output of the glitch filter directly clocks Counter (CNR). When LPTMR is first enabled, the output of the glitch filter is asserted, that is, logic 1 for active-high and logic 0 for active-low. The following table shows the change in glitch filter output with the selected input source.

Table 602. Glitch filter output with the selected input source

If	Then
The selected input source remains deasserted for at least 2^1 to 2^{15} consecutive prescaler clock rising edges	The glitch filter output also deasserts.
The selected input source remains asserted for at least 2^1 to 2^{15} consecutive prescaler clock rising edges	The glitch filter output also asserts.

NOTE

The input is sampled only on the rising clock edge.

The value of CNR increments each time the glitch filter output asserts. In Pulse Counter mode, the maximum rate at which CNR can increment is once every 2^2 to 2^{16} glitch filter clock edges. When first enabled, the glitch filter waits for an additional one or two glitch filter clock edges because of the synchronization logic.

65.3.4.4 Glitch filter bypassed

In Pulse Counter mode, when the glitch filter is bypassed, the selected input source increments the value of Counter (CNR) every time it asserts. Before LPTMR is first enabled, the selected input source is forced to be asserted. This prevents CNR from incrementing if the selected input source is already asserted when LPTMR is first enabled.

65.3.5 Counter

The value of Counter (CNR) increments by 1 on every:

- Prescaler clock in Time Counter mode, with prescaler bypassed.
- Prescaler output in Time Counter mode, with prescaler enabled.
- Input source assertion in Pulse Counter mode, with glitch filter bypassed.
- Glitch filter output in Pulse Counter mode, with glitch filter enabled.

CNR is reset when LPTMR is disabled or if the counter register overflows. If CSR[TFC] = 0, then CNR is also reset whenever CSR[TCF] = 1.

When the core is halted in Debug mode:

- CNR continues incrementing if configured for Pulse Counter mode.
- CNR stops incrementing if configured for Time Counter mode.

You cannot initialize CNR but can read it at any time. On each read of CNR, you must first write a value to it. This synchronizes and registers the current value of CNR into a temporary register. The contents of the temporary register are returned on each read of CNR.

When reading CNR, the bus clock must be at least two times faster than the rate at which the LPTMR counter is incrementing; otherwise, incorrect data may be returned.

65.3.6 Compare

After the next Counter (CNR) increment (when its value is equal to that of Compare (CMR)), the following events occur:

- CSR[TCF] is read as 1b.
- LPTMR generates an interrupt if CSR[TIE] is 1 as well.
- LPTMR generates a hardware trigger.
- LPTMR writes 0 to CNR if CSR[TFC] is 0.

When LPTMR is enabled, you can modify the value of CMR only when CSR[TCF] is 1. When updating CMR, you must write to it and clear CSR[TCF] before the LPTMR counter increments past the new LPTMR compare value.

NOTE

When LPTMR is enabled in Time Counter mode, the first increment takes an additional one or two clock cycles because of the synchronization logic. This results in the first compare (and therefore interrupt and hardware trigger) occurring slightly later. A faster prescaler clock or larger prescaler value minimizes this impact.

65.3.7 Interrupt

LPTMR generates an interrupt whenever CSR[TIE] and CSR[TCF] are 1. CSR[TCF] is cleared by disabling LPTMR or writing a logic 1 to it.

You can modify the value of CSR[TIE] and write 1 to CSR[TCF] when LPTMR is enabled.

LPTMR generates an interrupt asynchronously to the system clock. The interrupt can be used to generate a wake-up from any low-power mode, provided LPTMR is enabled as a wake-up source.

65.3.8 Hardware trigger

The LPTMR hardware trigger asserts at the same time CSR[TCF] is set and can be used to trigger hardware events in other peripherals without your intervention. The hardware trigger is always enabled.

Table 603. Hardware trigger

When	Then
CMR[COMPARE] and CSR[TFC] are 0	The LPTMR hardware trigger asserts on the first compare and does not deassert.
CMR[COMPARE] is set to a nonzero value, or if CSR[TFC] = 1	The LPTMR hardware trigger asserts on each compare and deasserts on the following increment of Counter (CNR).

65.4 External signals

Table 604. External signals

Signal	Description		Direction
LPTMR_ALT*n*	Pulse Counter Input LPTMR can select one of the input pins to be used in Pulse Counter mode.		Input
	State meaning	Assertion—If configured for Pulse Counter mode with an active-high input, assertion causes Counter (CNR) to increment. Deassertion—If configured for Pulse Counter mode with an active-low input, deassertion causes CNR to increment.	
	Timing	Assertion or deassertion may occur at any time; input may assert asynchronously to the bus clock.	

65.5 Initialization

Perform the following procedure to initialize LPTMR:

1. Configure Control Status (CSR) for the selected mode and pin configuration, when CSR[TEN] is 0. This resets the counter and clears the flag.
2. Configure Prescaler and Glitch Filter (PSR) with the selected clock source and prescaler or glitch filter configuration.
3. Configure Compare (CMR) with the selected compare point.
4. Write 1 to CSR[TEN] to enable LPTMR.

65.6 Application information

65.6.1 Application 1: Generate an interrupt every 100 ms using 32.768 kHz clock source

1. Disable LPTMR by writing 0 to CSR[TEN].
2. Select a 32.768 kHz clock source by configuring PSR[PCS].
3. Bypass the prescaler and glitch filter by writing 1 to PSR[PBYP].
4. Assert an interrupt every 3277 cycles by configuring CMR[COMPARE] = 0CCCh.
5. Enable the LPTMR interrupt by writing 1 to CSR[TIE].
6. Enable LPTMR by writing 1 to CSR[TEN].

NOTE

This is just an example. See the chip-specific LPTMR information for the clocks supported on a given chip.

65.6.2 Application 2: Generate an interrupt once a minute using 32.768 kHz clock source

1. Disable LPTMR by writing 0 to CSR[TEN].
2. Select a 32.768 kHz clock source by configuring PSR[PCS].
3. Select the prescaler to divide the prescaler clock by 32768 to increment the counter once a second by configuring PSR[PRESCALE] = 0Eh.
4. Assert an interrupt every 60 seconds by configuring CMR[COMPARE] = 003Bh.
5. Enable the LPTMR interrupt by writing 1 to CSR[TIE].
6. Enable LPTMR by writing 1 to CSR[TEN].

NOTE

This is just an example. See the chip-specific LPTMR information for the clocks supported on a given chip.

65.7 Memory map and register definition

NOTE

The LPTMR registers are reset only on Power on Reset (POR) or Low Voltage Detect (LVD). See Reset for more information.

65.7.1 LPTMR register descriptions

65.7.1.1 LPTMR memory map

LPTMR1 base address: 4430_0000h

LPTMR2 base address: 424D_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Control Status (CSR)	32	RW	0000_0000
4	Prescaler and Glitch Filter (PSR)	32	RW	0000_0000
8	Compare (CMR)	32	RW	0000_0000
C	Counter (CNR)	32	RW	0000_0000

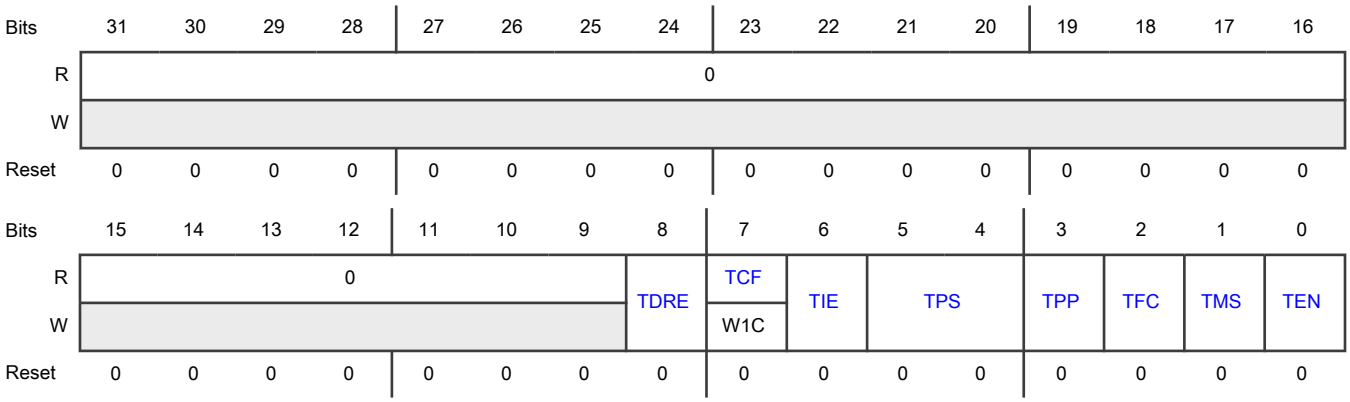
65.7.1.2 Control Status (CSR)

Controls various features of LPTMR.

Offset

Register	Offset
CSR	0h

Diagram



Fields

Field	Description
31-9 —	Reserved
8 TDRE	Timer DMA Request Enable Enables the timer DMA request. When TDRE is 1, the LPTMR DMA request is generated whenever CSR[TCF] is also set. Then, CSR[TCF] is cleared after the DMA controller completes execution. 0 - Disable 1 - Enable
7 TCF	Timer Compare Flag Compares the timer. TCF sets on the next Counter (CNR) increment when LPTMR is enabled and Counter (CNR) equals Compare (CMR). TCF is cleared when LPTMR is disabled or a logic 1 is written to it. **NOTE** You must clear this flag before enabling the timer interrupt or DMA request. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - CNR ≠ (CMR + 1) 1 - CNR = (CMR + 1) When writing 0 - No effect 1 - Clear the flag
6 TIE	Timer Interrupt Enable Enables the timer interrupt. If TIE is 1, then LPTMR generates an interrupt if CSR[TCF] is 1 as well. 0 - Disable 1 - Enable
5-4 TPS	Timer Pin Select Configures the input source to be used in Pulse Counter mode. The input connections vary by chip. For details, see the chip configuration information about connections to these inputs. You must modify this field only when LPTMR is disabled. 00 - Input 0 01 - Input 1 10 - Input 2 11 - Input 3
3 TPP	Timer Pin Polarity Configures the polarity of the input source in Pulse Counter mode. If TPP is 0, then the pulse counter input source is active-high, and Counter (CNR) increments on the rising edge. If TPP is 1, then the pulse counter input source is active-low, and CNR increments on the falling edge. You must modify this field only when LPTMR is disabled. 0 - Active-high 1 - Active-low
2 TFC	Timer Free-Running Counter Specifies when the counter resets. If TFC is 0, Counter (CNR) resets on the count cycle following Counter (CNR) becoming equal to Compare (CMR). If TFC is 1, CNR resets on overflow. In both cases, CSR[TCF] sets to 1 on the cycle after CNR and CMR match. You must modify this field only when LPTMR is disabled. 0 - Reset when TCF asserts 1 - Reset on overflow
1 TMS	Timer Mode Select Configures the mode of LPTMR. You must modify this field only when LPTMR is disabled.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Time Counter 1 - Pulse Counter
0 TEN	Timer Enable Enables the LPTMR timer. If TEN is 0, it resets the LPTMR internal logic, including CNR[COUNTER] and CSR[TCF]. If TEN is 1, LPTMR is enabled. Do not alter CSR[5:1] when writing 1 to this field. 0 - Disable 1 - Enable

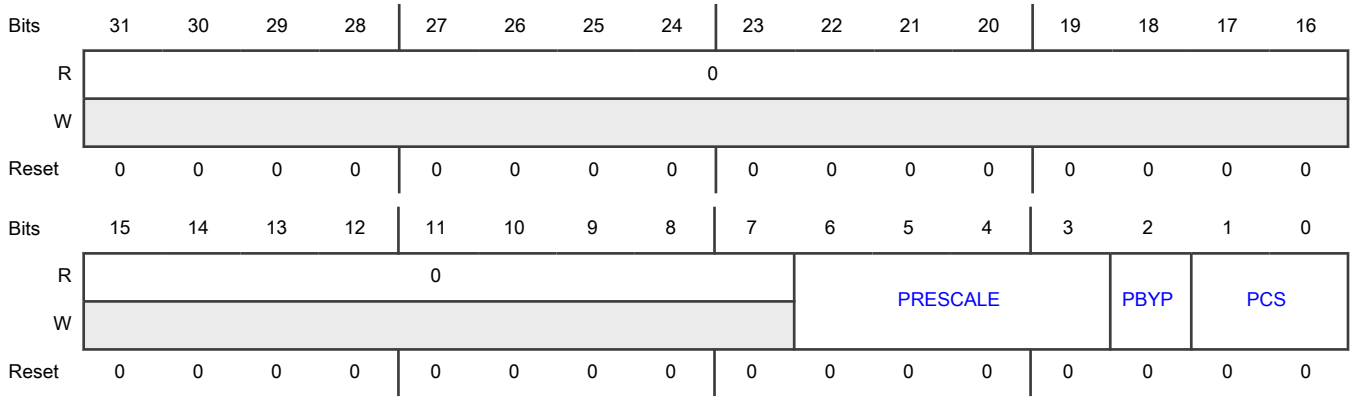
65.7.1.3 Prescaler and Glitch Filter (PSR)

Configures features related to the prescaler and glitch filter.

Offset

Register	Offset
PSR	4h

Diagram



Fields

Field	Description
31-7 —	Reserved
6-3 PRESCALE	Prescaler and Glitch Filter Value

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Configures the size of the prescaler in Time Counter mode and the width of the glitch filter in Pulse Counter mode. The width of the glitch filter can vary by one cycle because of the pulse counter input synchronization. You must modify this field only when LPTMR is disabled. 0000 - Prescaler divides the prescaler clock by 2; glitch filter does not support this configuration 0001 - Prescaler divides the prescaler clock by 4; glitch filter recognizes change on input pin after two rising clock edges 0010 - Prescaler divides the prescaler clock by 8; glitch filter recognizes change on input pin after four rising clock edges 0011 - Prescaler divides the prescaler clock by 16; glitch filter recognizes change on input pin after eight rising clock edges 0100 - Prescaler divides the prescaler clock by 32; glitch filter recognizes change on input pin after 16 rising clock edges 0101 - Prescaler divides the prescaler clock by 64; glitch filter recognizes change on input pin after 32 rising clock edges 0110 - Prescaler divides the prescaler clock by 128; glitch filter recognizes change on input pin after 64 rising clock edges 0111 - Prescaler divides the prescaler clock by 256; glitch filter recognizes change on input pin after 128 rising clock edges 1000 - Prescaler divides the prescaler clock by 512; glitch filter recognizes change on input pin after 256 rising clock edges 1001 - Prescaler divides the prescaler clock by 1024; glitch filter recognizes change on input pin after 512 rising clock edges 1010 - Prescaler divides the prescaler clock by 2048; glitch filter recognizes change on input pin after 1024 rising clock edges 1011 - Prescaler divides the prescaler clock by 4096; glitch filter recognizes change on input pin after 2048 rising clock edges 1100 - Prescaler divides the prescaler clock by 8192; glitch filter recognizes change on input pin after 4096 rising clock edges 1101 - Prescaler divides the prescaler clock by 16,384; glitch filter recognizes change on input pin after 8192 rising clock edges 1110 - Prescaler divides the prescaler clock by 32,768; glitch filter recognizes change on input pin after 16,384 rising clock edges 1111 - Prescaler divides the prescaler clock by 65,536; glitch filter recognizes change on input pin after 32,768 rising clock edges
2 PBYP	Prescaler and Glitch Filter Bypass Controls the clocking of Counter (CNR). If PBYP is 0, the output of the prescaler or glitch filter clocks CNR. If PBYP is 1, the selected prescaler clock in Time Counter mode, or else the selected input source in Pulse Counter mode, directly clocks CNR. You must modify this field only when LPTMR is disabled.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Prescaler and glitch filter enable 1 - Prescaler and glitch filter bypass
1-0 PCS	Prescaler and Glitch Filter Clock Select Selects the clock to be used by the LPTMR prescaler and glitch filter. In Time Counter mode, PCS selects the input clock to the prescaler. In Pulse Counter mode, PCS selects the input clock to the glitch filter. See the chip configuration details for information on connections to these inputs. You must modify this field only when LPTMR is disabled. 00 - Clock 0 01 - Clock 1 10 - Clock 2 11 - Clock 3

65.7.1.4 Compare (CMR)

Configures the compare values to Counter (CNR) (see Compare for more information).

Offset

Register	Offset
CMR	8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	COMPARE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMPARE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 COMPARE	Compare Value Configures the compare values to Counter (CNR). On the next CNR increment, if LPTMR is enabled and Counter (CNR) equals Compare (CMR), then: 1. LPTMR writes 1 to CSR[TCF]. 2. The hardware trigger asserts until the next time CNR increments. If CMR = 0, the hardware trigger remains asserted until LPTMR is disabled. If LPTMR is enabled, you must modify the value of CMR only if CSR[TCF] is 1.

65.7.1.5 Counter (CNR)

Specifies counter values (see Counter for more information).

Offset

Register	Offset
CNR	Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	COUNTER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COUNTER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 COUNTER	Counter Value Contains the current value of the LPTMR counter at the time you last wrote to this register.

Chapter 66
Low Power Periodic Interrupt Timer (LPIT)

66.1 Chip-specific LPIT information

Table 605. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

This chip includes two instances of LPIT timers (LPIT1 in AONMIX and LPIT2 in WAKEUPMIX).

LPIT has no IO interfaces to/from Pins.

NOTE

LPIT1 is clocked by AON bus and LPIT2 is clocked by WAKEUP bus. LPIT1 and LPIT2 have their trigger output 0 used as input by CAN1 and CAN2 respectively. Because of this, these LPIT timer instances are clocked by the bus clock - the same clock used to clock the respective CAN module (bus_aon_clk_root and bus_wakeup_clk_root respectively) in order to ensure that these signals are timed synchronously between these modules.

66.2 Overview

LPIT is a low-power periodic interrupt timer with multiple timer channels. After a timer reaches a programmed count, the respective timer channels generate pre-trigger and trigger output signals, and these outputs can be used to trigger other modules on the chip.

66.2.1 Block diagram

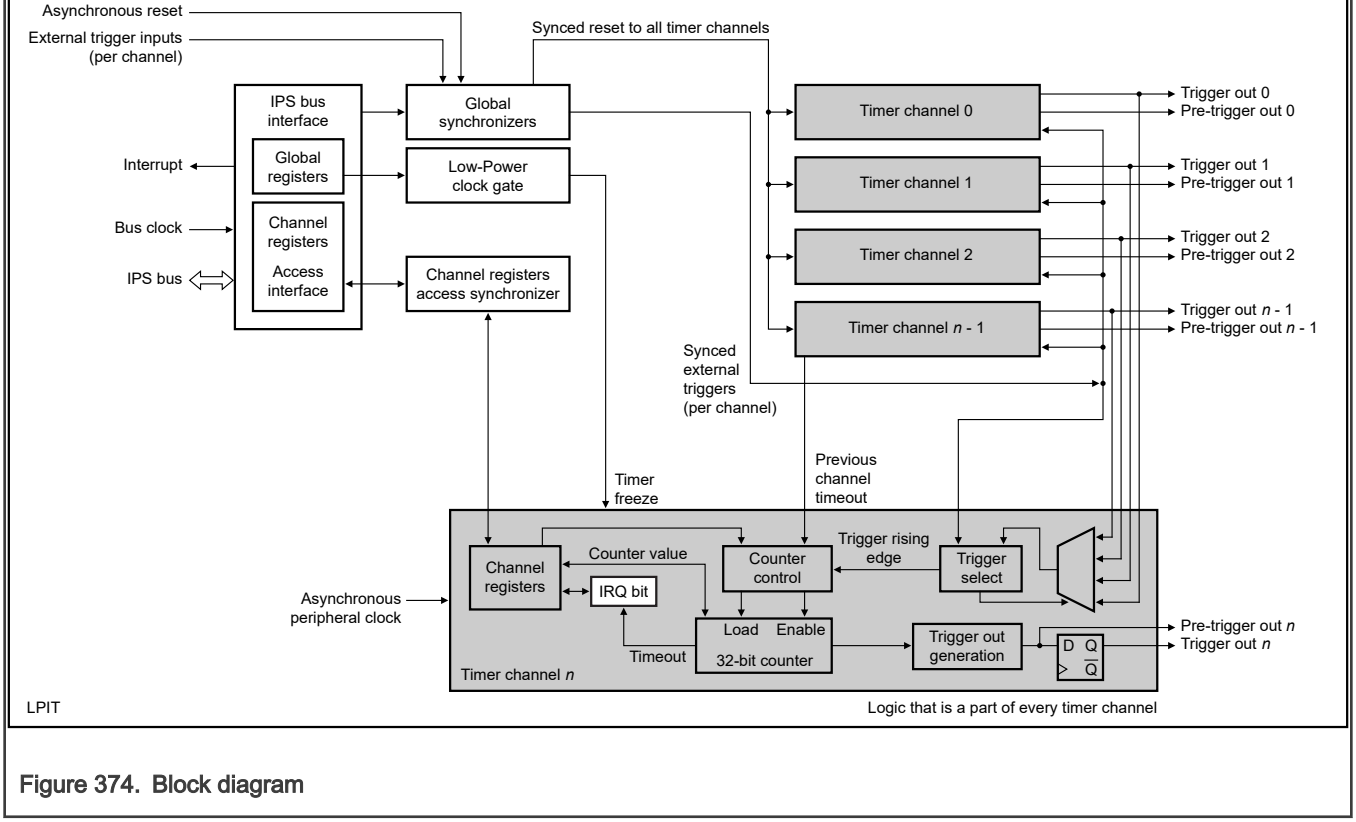


Figure 374. Block diagram

66.2.2 Features

LPIT allows you to configure timer channels in a way that they could be:

- Controlled using external triggers (triggers from outside LPIT).
- Controlled using internal triggers (triggers from other timer channels inside LPIT).
- Chained together, to form a larger width timer.
- Reloaded and counted again, or stopped after reaching the programmed count, depending on the timer modes used.

66.3 Functional description

66.3.1 Programming model

The LPIT programming model (see Figure 375) comprises:

- A global register set (common to all timer channels).
- Registers for each timer channel (that control their respective timer channels).

Access to these registers is synchronized with the asynchronous peripheral clock and then affects the timer channel registers:

- Each timer channel contains a 32-bit counter that loads the starting value and down counts after every peripheral clock's positive edge.
- After reaching a zero value (a channel timer timeout), a trigger output is generated.
- A timer enable register control field, external or internal triggers, or a previous channel timeout (when using timer chaining) control the counter enable.

- After a channel timer timeout, an interrupt flag is also set to tell the CPU about the timer timeout.

You must configure the following global registers only once:

- Module Control (MCR)
- Module Status (MSR)
- Module Interrupt Enable (MIER)
- Set Timer Enable (SETTEN)
- Clear Timer Enable (CLRTEN)

You must configure the following channel registers for each channel:

- Timer Value (TVAL0 - TVAL3)
- Timer Control (TCTRL0 - TCTRL3)

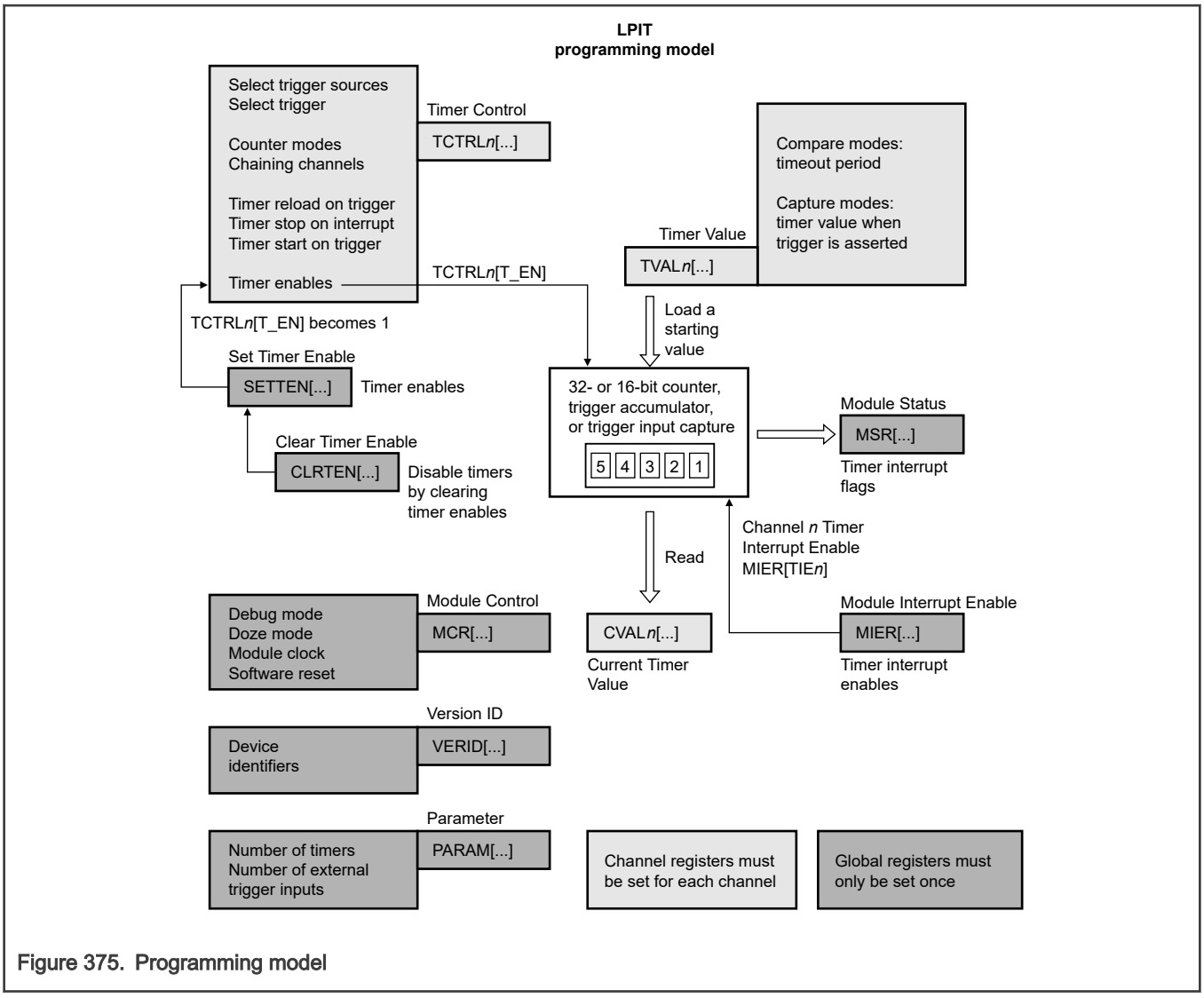


Figure 375. Programming model

66.3.2 Interfacing with other modules

The following figure shows the interface of an LPIT module with other modules on the chip:

- The CPU interface provides the clock, reset, and register read and write bus interface. It handles LPIT interrupts.

- The LPIT trigger output signals may trigger other modules on the chip.
- Similarly, other timer modules may provide trigger inputs to LPIT to control when an LPIT timer channel must start.

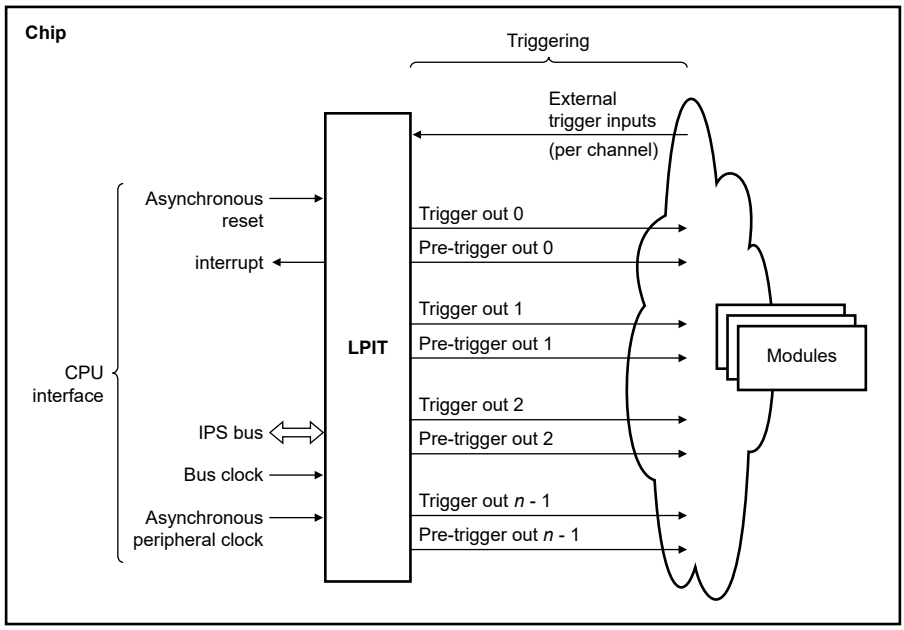


Figure 376. Interfacing with other modules

66.3.3 Chip power modes

LPIT supports the following chip power modes.

Table 606. Chip power modes

Chip mode	LPIT operation
Run	Normal operation
Stop and Wait	Can continue operating in this mode if MCR[DOZE_EN] = 1 and LPIT is using an external or internal clock source that remains operational during Stop and Wait modes.
Debug	Can continue operating in this mode if MCR[DBG_EN] = 1

66.3.4 Supported timer modes

To configure a timer mode, you must select an appropriate value using TCTRL*n*[MODE].

Table 607. Supported timer modes

Timer mode	Operation
32-bit periodic counter	The counter loads and decrements down to 0, and this sets the timer interrupt flag and asserts the output pretrigger.

Table continues on the next page...

Table 607. Supported timer modes (continued)

Timer mode	Operation
Dual 16-bit periodic counter	• The counter loads and then the lower 16 bits decrement down to 0. This asserts the output pretrigger. • The upper 16 bits then decrement down to 0. This sets the timer interrupt flag and negates the output pretrigger.
32-bit trigger accumulator	The counter loads after the first trigger rising edge and then decrements down to 0 after each trigger rising edge. This sets the timer interrupt flag and asserts the output pretrigger.
32-bit trigger input capture	• The counter loads with a value of FFFF_FFFFh and then decrements down to 0. • If a trigger rising edge is detected, it stores the inverse of the current counter value in Timer Value (TVAL0 - TVAL3). This sets the timer interrupt flag and asserts the output pretrigger.

TCTRL*n*[TSOT], TCTRL*n*[TSOI], and TCTRL*n*[TROT] control the timer operation. These fields control the timer load, reload, start, and restart of the timers.

NOTE

- The trigger output is asserted one peripheral timer clock cycle after the pre-trigger output. The trigger and pre-trigger outputs deassert at the same time.
- The pre-trigger output is asserted for two clock cycles and the trigger output is asserted for one clock cycle (except in 16-bit Periodic Counter mode, where both pre-trigger and trigger outputs are asserted for many cycles depending on the value of TVAL*n*[TMR_VAL][31:16]).
- Timer changes (that are based on external triggers) take effect after four peripheral clocks, after the actual external trigger assertion. This is because of clock synchronization, rise edge detection, and timer update.

66.3.5 Timer channel modes

You can configure each timer channel in LPIT to work in either compare modes or capture modes.

The timer channels operate on an asynchronous clock, which is independent of the register read and write access clock. Clock synchronization between clock domains ensures normal operations.

Table 608. Timer channel modes

Mode	Function
Compare	The timers decrement when enabled and generate an output pretrigger and trigger output. The trigger output is one clock cycle delayed of the pre-trigger pulse. You can configure certain control fields to control each timer channel's start, reload, and restart (see Trigger control for timers for more information). You can also configure the timer to always decrement from a programmed start value, on selected trigger inputs, or previous channel timeout (when channels are chained). By chaining timer channels, applications can achieve larger timeout durations.
Capture	You can use the timer to perform measurements by using Timer Value (TVAL0 - TVAL3), as the timer value is captured when a selected trigger input is asserted. The timer can support once-off or multiple measurements, such as frequency measurements.

66.3.6 Trigger control for timers

You can configure various LPIT register fields to control how trigger inputs and the timer operate:

- TCTRL*n*[TRG_SEL] helps you select the input trigger for the channel from all other channels' trigger outputs.
- TCTRL*n*[TRG_SRC] helps you select between the internal and external trigger inputs to the channel.

The selected trigger affects how the timer operates, using the configuration of TCTRL*n*[TSOT], TCTRL*n*[TSOI], and TCTRL*n*[TROT] (see Table 609).

Table 609. Fields that control timer operations

If	=	Then
Timer stop on interrupt (TCTRL*n*[TSOI])	1	The counter stops after MSR[TIF*n*] assertion. To reload and decrement, it requires: • A trigger (if TCTRL*n*[TSOT] = 1). • A T_EN rising edge (if TCTRL*n*[TSOT] = 0).
	0	The counter does not stop after timeout.
Timer reload on trigger (TCTRL*n*[TROT])	1	The counter is loaded after each trigger.
	0	The counter is loaded after every T_EN rising edge or timeout rising edge (timeout is not used in Capture modes).
Timer start on trigger (TCTRL*n*[TSOT])	1	The counter starts decrementing after a trigger. Subsequent triggers are ignored until the counter times out.
	0	The counter decrements immediately after the next clock edge. When the channel is chained or is in Capture mode, TCTRL*n*[TSOT] has no effect.

In different timer modes, the programmable fields listed in Table 610 affect the timer operation differently.

Table 610. Timer modes and programmable fields

Mode (TCTRL*n*[MODE])	Fields affecting timer operations
32-bit periodic counter	TCTRL*n*[TSOT], TCTRL*n*[TSOI], and TCTRL*n*[TROT] affect the timer operation as described in Table 609.
Dual 16-bit periodic counter	
32-bit trigger accumulator	• Only TCTRL*n*[TSOI] controls the timer function. • TCTRL*n*[TROT] and TCTRL*n*[TSOT] have no effect on timer operations.
32-bit input trigger capture	• Only TCTRL*n*[TSOI] and TCTRL*n*[TROT] control the timer function. • TCTRL*n*[TSOT] has no effect on timer operations.

66.3.7 Channel chaining

You can chain individual timer channels together to achieve a larger value of timeout. Chaining enables these channels to work in a "nested loop" manner, thereby leading to an effective timeout value of $TVAL_{CHn} \times (TVAL_{CHn-1} + 1)$.

To chain the channels together, write 1 to TCTRL*n*[CHAIN] for the corresponding channel. When a channel is chained, that channel's timer decrements after the previous channel's timeout pulse, regardless of the timer mode that TCTRL*n*[MODE] defines.

TCTRL*n*[TSOT] does not have any effect if you chain the channel timer (channel *n*) to the previous channel's timer (channel *n* - 1).

66.3.8 Detailed timing

The following table represents various timing diagrams. The diagrams are not "cycle-accurate," which means, they may not show some of the cycles, but they do show the timer channel behavior across several clock cycles.

Table 611. Timing diagrams

Mode (TCTRL*n*[MODE]) setting		Timing diagram	TCTRL*n*[TSOT]	TCTRL*n*[TROT]	TCTRL*n*[TSOI]	TCTRL*n*[CHAIN]
32-bit periodic counter (Compare mode)	00	Case 1: TCTRL*n*[MODE] = 0 • For a use case requiring repeated interrupts with reload • Trigger outputs have equal periods	0	0	0	0
		Case 2: TCTRL*n*[MODE] = 0 • Useful for One-Shot Trigger mode • Trigger starts again after TCTRL*n*[T_EN] becomes 1	0	0	1	0
		Case 3: TCTRL*n*[MODE] = 0 • For a use case requiring repeated interrupts with reload • Trigger outputs have unequal periods	0	1	0	0
		Case 4: TCTRL*n*[MODE] = 0 • For a one-shot timer with reload before timeout of Timer mode • Timer starts again after you write 1 to TCTRL*n*[T_EN]	0	1	1	0
		Case 5: TCTRL*n*[MODE] = 0 • Useful for generating periodic interrupts after a predefined event (input trigger) • Output triggers have equal periods	1	0	0	0
		Case 6: TCTRL*n*[MODE] = 0 • Triggers One-Shot Timer mode • Output trigger period depends on the input trigger	1	0	1	0
		Case 7: TCTRL*n*[MODE] = 0 • Repeated interrupts with Reload Timer mode • Outputs triggers have unequal periods	1	1	0	0
		Case 8: TCTRL*n*[MODE] = 0 • Case shown for a nonperiodic input trigger (might not be a use case)	1	1	1	0

Table continues on the next page...

Table 611. Timing diagrams (continued)

Mode (TCTRL*n*[MODE]) setting		Timing diagram	TCTRL*n*[TSOT]	TCTRL*n*[TROT]	TCTRL*n*[TSOI]	TCTRL*n*[CHAIN]
		• If input trigger is periodic and greater than timer timeout, it is the same as TCTRL*n*[TROT] = 0 • If input trigger is periodic and less than timer timeout, the timer never times out and always reloads on the input trigger (not a valid use case)				
16-bit dual periodic counter (Compare mode)	01	Case 1: TCTRL*n*[MODE] = 1 • The effect of TCTRL*n*[TSOT], TCTRL*n*[TROT], and TCTRL*n*[TSOI] is the same as that of TCTRL*n*[MODE] = 0 (32-bit Counter Compare mode) • Both halves of the counter are affected in the same way	0	0	0	0
32-bit trigger accumulator mode	10	Case 1: TCTRL*n*[MODE] = 10 • Useful for a continuous pulse counting mode • Trigger and timeout generated after programmed number of pulses are accumulated	X	X	0	0
		Case 2: TCTRL*n*[MODE] = 10 • Useful for One-Shot Pulse Counting mode • Trigger and timeout generated after programmed number of pulses are accumulated	X	X	1	0
32-bit trigger capture mode	11	Case 1: TCTRL*n*[MODE] = 11 • Useful for determining the duration between pulses • Proper clock selection can ensure that the timer does not rollover more than once between two pulses	X	0	0	0
		Case 2: TCTRL*n*[MODE] = 11 • Useful for determining the duration between pulses • Selecting a fast timer clock provides accurate measurements but it can also cause timer rollover between pulses	X	1	0	0

Table continues on the next page...

Table 611. Timing diagrams (continued)

Mode (TCTRL*n*[MODE]) setting		Timing diagram	TCTRL*n*[TSOT]	TCTRL*n*[TROT]	TCTRL*n*[TSOI]	TCTRL*n*[CHAIN]
		Case 3: TCTRL*n*[MODE] = 11 • One-Shot Timer Count mode • You can enable it again by writing 1 to TCTRL*n*[T_EN]	X	0	1	0
Timer chaining: effects on timing operations	XX	Timer chaining	X	X	X	1

66.3.8.1 Case 1: TCTRL*n*[MODE] = 0

The following figure represents case 1 in which TCTRL*n*[MODE] = 0 (32-bit periodic counter). LPIT works in Compare mode and is configured as follows:

- TCTRL*n*[TSOT] = 0
- TCTRL*n*[TROT] = 0
- TCTRL*n*[TSOI] = 0
- TCTRL*n*[CHAIN] = 0

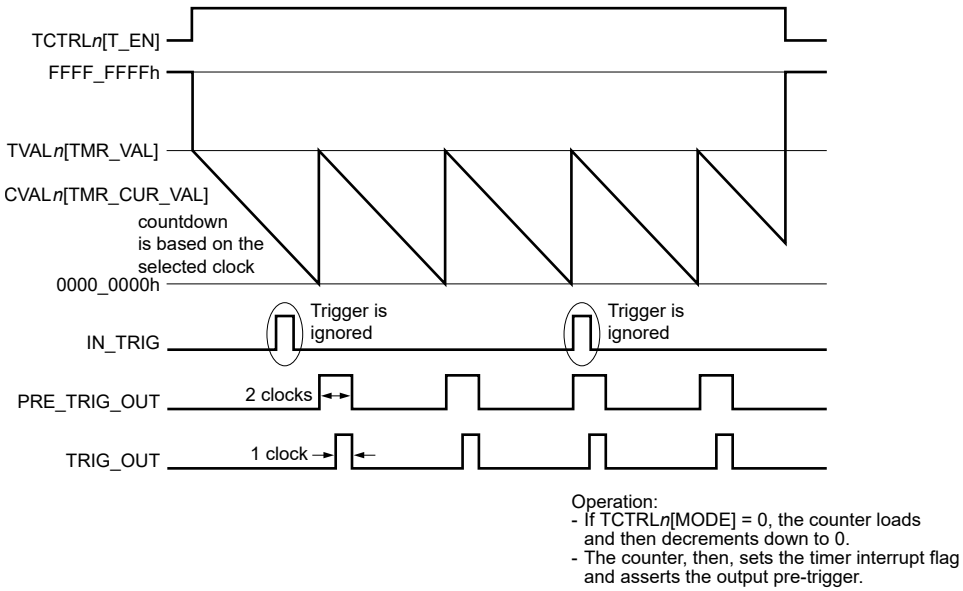


Figure 377. Case 1: TSOT = 0, TROT = 0, TSOI = 0, CHAIN = 0

66.3.8.2 Case 2: TCTRL*n*[MODE] = 0

The following figure represents case 2 in which TCTRL*n*[MODE] = 0 (32-bit periodic counter). LPIT works in Compare mode and is configured as follows:

- TCTRL*n*[TSOT] = 0
- TCTRL*n*[TROT] = 0
- TCTRL*n*[TSOI] = 1
- TCTRL*n*[CHAIN] = 0

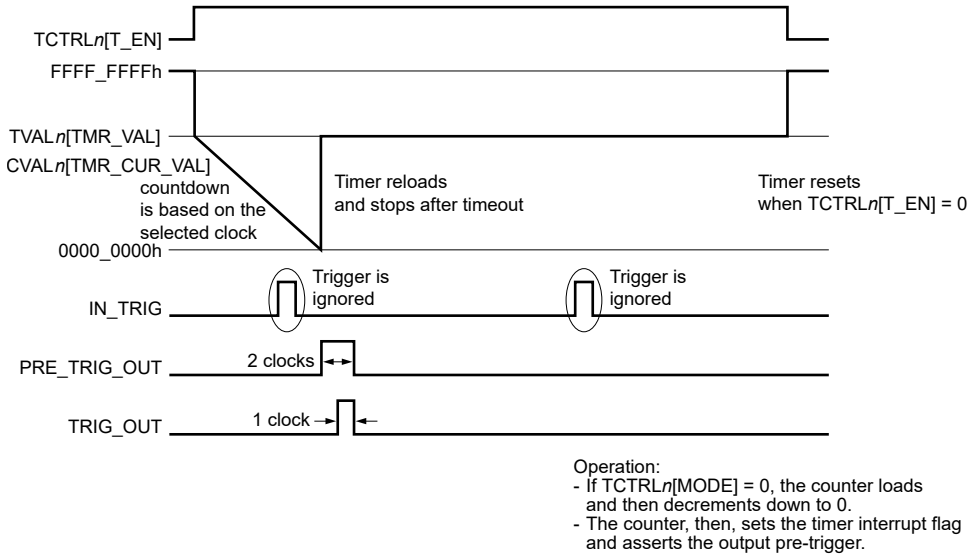


Figure 378. Case 2: TSOT = 0, TROT = 0, TSOI = 1, CHAIN = 0

66.3.8.3 Case 3: TCTRL*n*[MODE] = 0

The following figure represents case 3 in which TCTRL*n*[MODE] = 0 (32-bit periodic counter). LPIT works in Compare mode and is configured as follows:

- TCTRL*n*[TSOT] = 0
- TCTRL*n*[TROT] = 1
- TCTRL*n*[TSOI] = 0
- TCTRL*n*[CHAIN] = 0

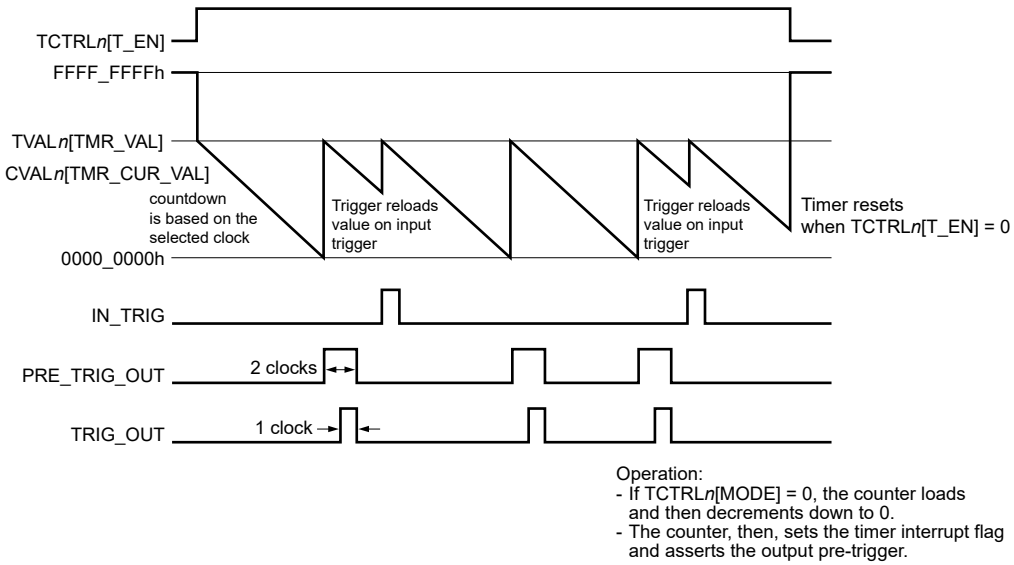


Figure 379. Case 3: TSOT = 0, TROT = 1, TSOI = 0, CHAIN = 0

66.3.8.4 Case 4: TCTRL*n*[MODE] = 0

The following figure represents case 4 in which TCTRL*n*[MODE] = 0 (32-bit periodic counter). LPIT works in Compare mode and is configured as follows:

- TCTRL*n*[TSOT] = 0
- TCTRL*n*[TROT] = 1
- TCTRL*n*[TSOI] = 1
- TCTRL*n*[CHAIN] = 0

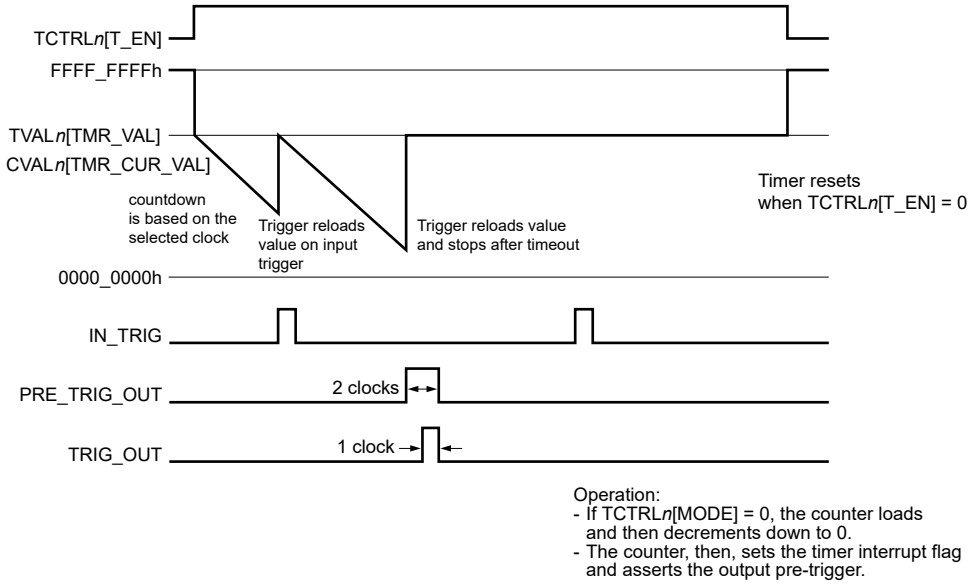


Figure 380. Case 4: TSOT = 0, TROT = 1, TSOI = 1, CHAIN = 0

66.3.8.5 Case 5: TCTRL*n*[MODE] = 0

The following figure represents case 5 in which TCTRL*n*[MODE] = 0 (32-bit periodic counter). LPIT works in Compare mode and is configured as follows:

- TCTRL*n*[TSOT] = 1
- TCTRL*n*[TROT] = 0
- TCTRL*n*[TSOI] = 0
- TCTRL*n*[CHAIN] = 0

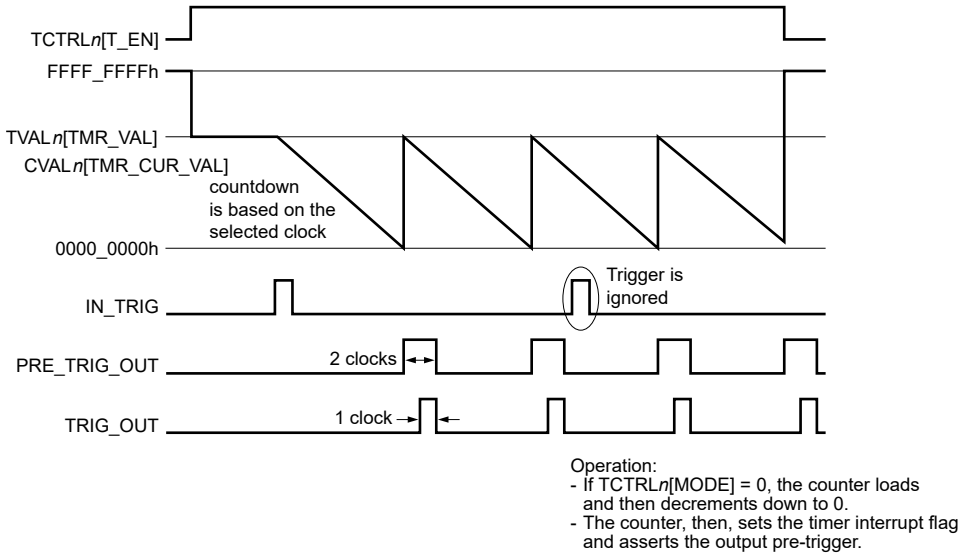


Figure 381. Case 5: TSOT = 1, TROT = 0, TSOI = 0, CHAIN = 0

66.3.8.6 Case 6: TCTRL*n*[MODE] = 0

The following figure represents case 6 in which TCTRL*n*[MODE] = 0 (32-bit periodic counter). LPIT works in Compare mode and is configured as follows:

- TCTRL*n*[TSOT] = 1
- TCTRL*n*[TROT] = 0
- TCTRL*n*[TSOI] = 1
- TCTRL*n*[CHAIN] = 0

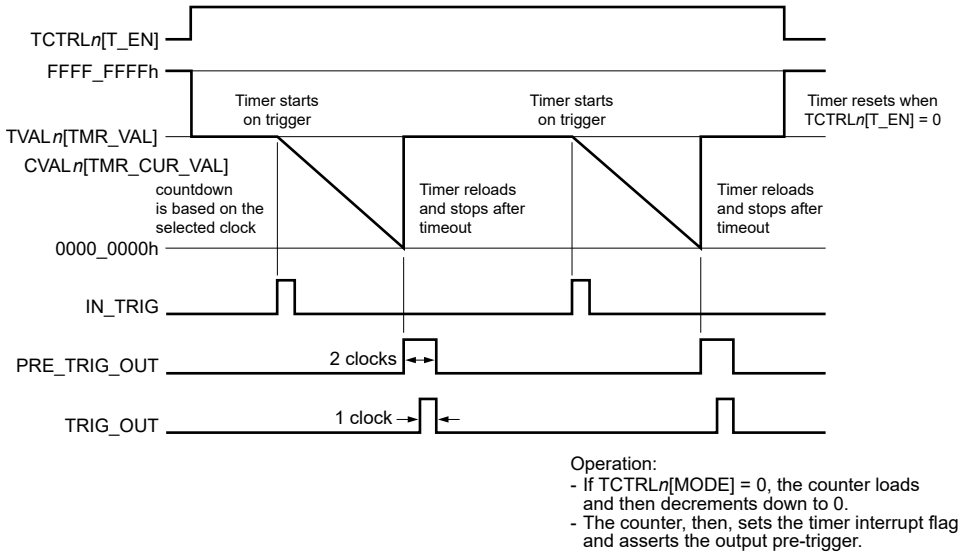


Figure 382. Case 6: TSOT = 1, TROT = 0, TSOI = 1, CHAIN = 0

66.3.8.7 Case 7: TCTRL*n*[MODE] = 0

The following figure represents case 7 in which TCTRL*n*[MODE] = 0 (32-bit periodic counter). LPIT works in Compare mode and is configured as follows:

- TCTRL*n*[TSOT] = 1
- TCTRL*n*[TROT] = 1
- TCTRL*n*[TSOI] = 0
- TCTRL*n*[CHAIN] = 0

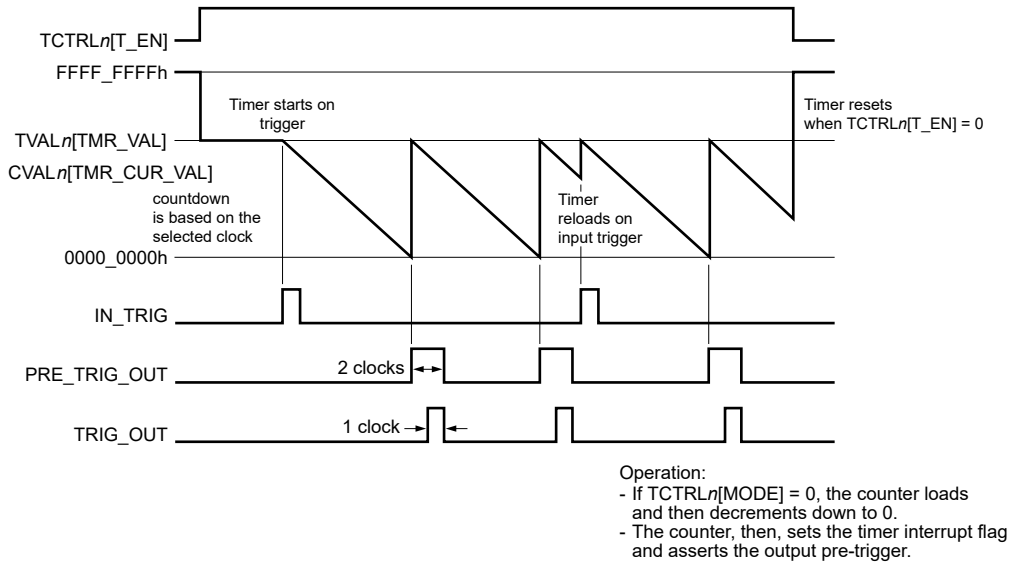


Figure 383. Case 7: TSOT = 1, TROT = 1, TSOI = 0, CHAIN = 0

66.3.8.8 Case 8: TCTRL*n*[MODE] = 0

The following figure represents case 8 in which TCTRL*n*[MODE] = 0 (32-bit periodic counter).

LPIT works in Compare mode and is configured as follows:

- TCTRL*n*[TSOT] = 1
- TCTRL*n*[TROT] = 1
- TCTRL*n*[TSOI] = 1
- TCTRL*n*[CHAIN] = 0

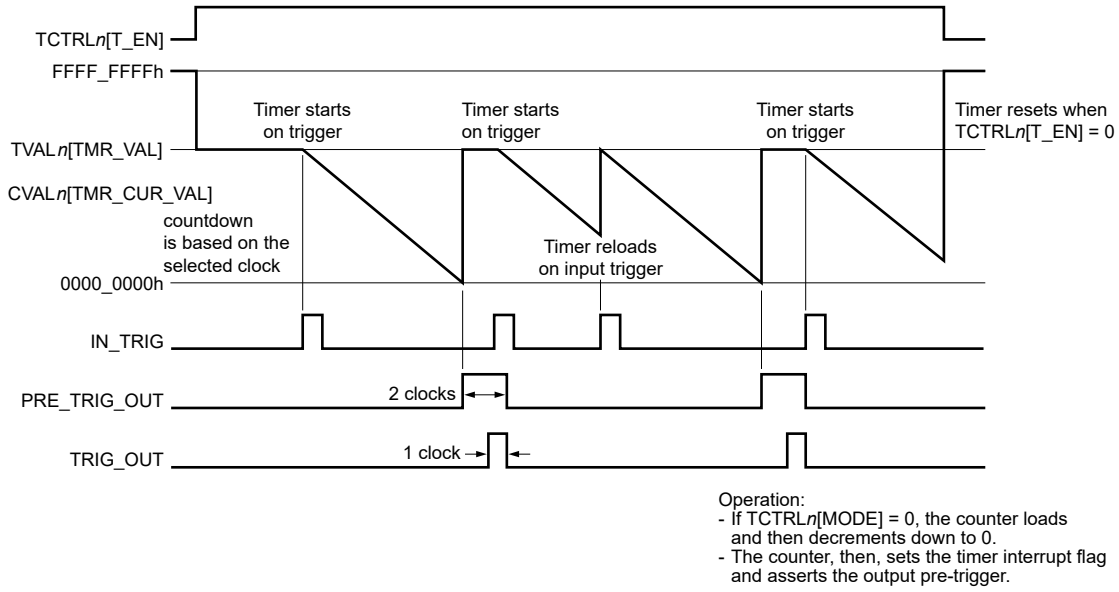


Figure 384. Case 8: TSOT = 1, TROT = 1, TSOI = 1, CHAIN = 0

66.3.8.9 Case 1: TCTRL*n*[MODE] = 1

The following figure represents case 1 in which TCTRL*n*[MODE] = 1 (16-bit dual periodic counter). LPIT works in Compare mode and is configured as follows:

- TCTRL*n*[TSOT] = 0
- TCTRL*n*[TROT] = 0
- TCTRL*n*[TSOI] = 0
- TCTRL*n*[CHAIN] = 0

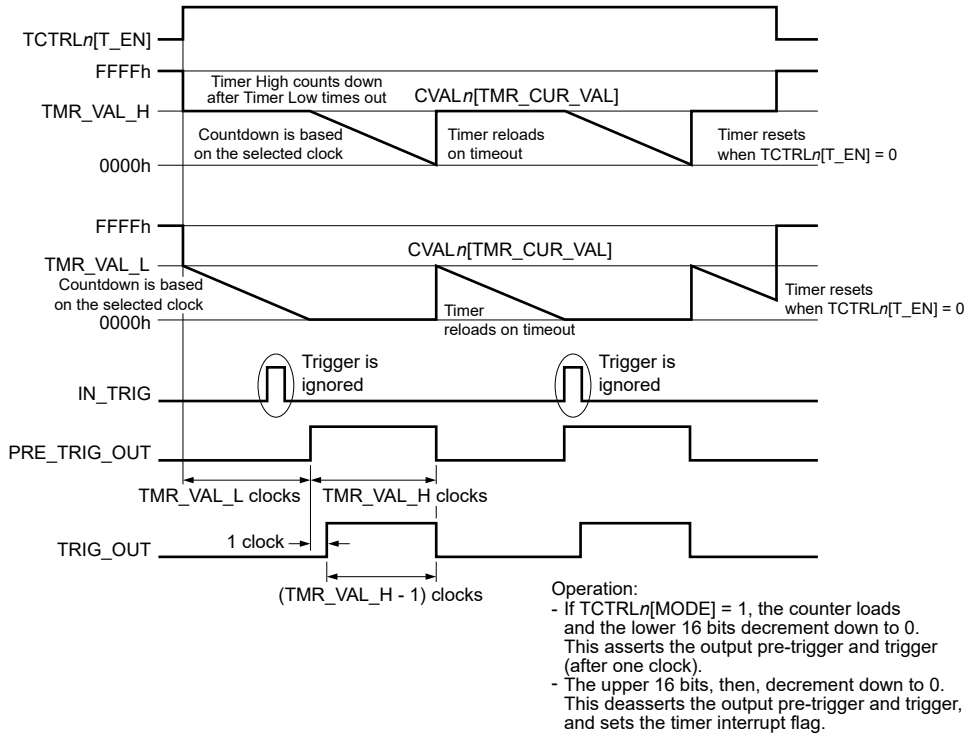


Figure 385. Case 1: TSOT = 0, TROT = 0, TSOI = 0, CHAIN = 0

If TCTRL*n*[MODE] = 1:

- The effect of timing control fields is similar to the effect of timer control fields when TCTRL*n*[MODE] = 0. See the individual timing diagrams for cases with TCTRL*n*[MODE] = 0 for more information.
- The timer interrupt (timeout) asserts when {TMR_H,TMR_L} = 0000_0000h.

See the following table for the behavior of timer control fields when TCTRL*n*[MODE] = 1.

Table 612. Timer control fields when TCTRL*n*[MODE] = 1

TCTRL*n*[TSOT]	TCTRL*n*[TROT]	TCTRL*n*[TSOI]	Function	Effect on timer
0	0	0	• For repeated interrupts with reload • Trigger outputs have equal periods	Similar to Case 1: TCTRL*n*[MODE] = 1.
0	0	1	One-shot mode	• Similar to Case 2: TCTRL*n*[MODE] = 0. • Both timers stop after first count down and then time out. • Timers do not count again until TCTRL*n*[T_EN] becomes 1.
0	1	0	• For repeated interrupts with reload • Trigger outputs have unequal periods	• Similar to Case 3: TCTRL*n*[MODE] = 0 . • Both timers reload the value of TVAL*n*[TMR_VAL] after the trigger rising edge.

Table continues on the next page...

Table 612. Timer control fields when TCTRL*n*[MODE] = 1 (continued)

TCTRL*n*[TSOT]	TCTRL*n*[TROT]	TCTRL*n*[TSOI]	Function	Effect on timer
				• Output triggers clear after reload, if asserted.
0	1	1	Reloadable one-shot mode	• Similar to Case 4: TCTRL*n*[MODE] = 0. • If a trigger occurs before timeout, then both timers reload and count down (as shown); the timers stop after timeout. • A trigger assertion after timeout reloads the value of TVAL*n*[TMR_VAL] into the timers. • The timers do not count again until TCTRL*n*[T_EN] becomes 1 again.
1	0	0	• For generating periodic interrupts after a predefined event (input trigger) • Output triggers have equal periods	• Similar to Case 5: TCTRL*n*[MODE] = 0. • After TCTRL*n*[T_EN] rises, the timers do not start until after the first trigger's rising edge. • Subsequent triggers have no effect.
1	0	1	• Triggered one-shot timer mode • Output trigger period depends on the input trigger	• Similar to Case 6: TCTRL*n*[MODE] = 0. • After TCTRL*n*[T_EN] becomes 1, the timers do not start until after the first trigger's rising edge. • The timer stops counting after a timeout assertion. • The timer does not start counting again until a new trigger's rising edge is detected.
1	1	0	• For repeated interrupts with reload timer mode • Output triggers have unequal periods	• Similar to Case 7: TCTRL*n*[MODE] = 0. • After TCTRL*n*[T_EN] becomes 1, the timers do not start until the first trigger's rising edge. • Subsequent triggers cause the timer to reload the value of TVAL*n*[TMR_VAL] into both counters. • The output triggers clear after a reload, if asserted.
1	1	1	• For a nonperiodic input trigger • If input trigger is periodic and greater than timer timeout, then it is the	• Similar to Case 8: TCTRL*n*[MODE] = 0. • After TCTRL*n*[T_EN] becomes 1, the timers do not start until the first trigger's rising edge.

Table continues on the next page...

Table 612. Timer control fields when TCTRL*n*[MODE] = 1 (continued)

TCTRL*n*[TSOT]	TCTRL*n*[TROT]	TCTRL*n*[TSOI]	Function	Effect on timer
			same as TCTRL*n*[TROT] = 0 (which is triggered one-shot timer mode; output trigger period depends on the input trigger)	• The timers stop counting after a timeout assertion. • A trigger's rising edge causes the timers to reload and then count down.

66.3.8.10 Case 1: TCTRL*n*[MODE] = 10

The following figure represents case 1 in which TCTRL*n*[MODE] = 10. LPIT works in 32-bit trigger accumulator mode and is configured as follows:

- TCTRL*n*[TSOT] = X
- TCTRL*n*[TROT] = X
- TCTRL*n*[TSOI] = 0
- TCTRL*n*[CHAIN] = 0

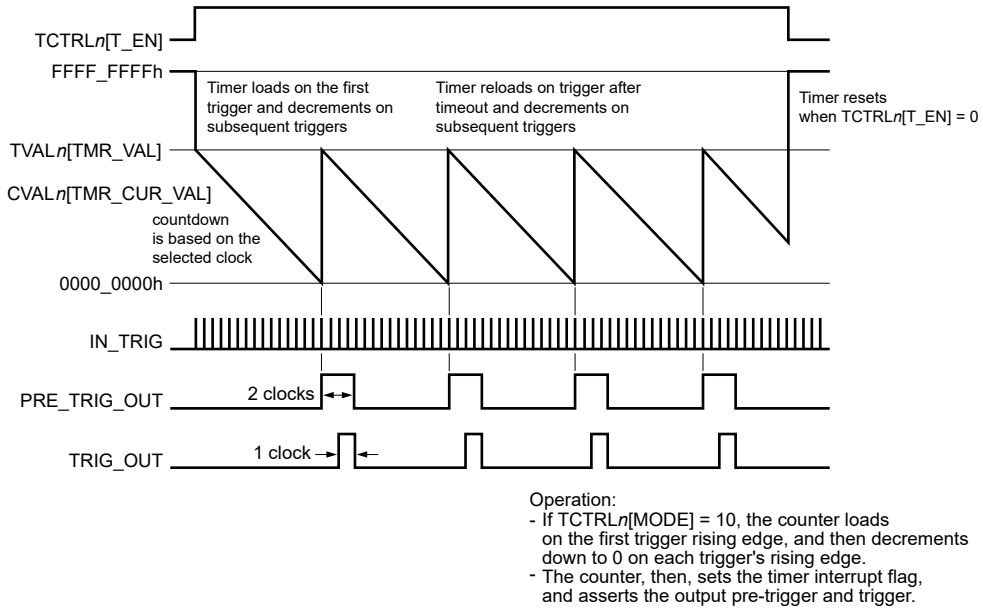


Figure 386. Case 1: TSOT = X, TROT = X, TSOI = 0, CHAIN = 0

66.3.8.11 Case 2: TCTRL*n*[MODE] = 10

The following figure represents case 2 in which TCTRL*n*[MODE] = 10. LPIT works in 32-bit trigger accumulator mode and is configured as follows:

- TCTRL*n*[TSOT] = X
- TCTRL*n*[TROT] = X
- TCTRL*n*[TSOI] = 1
- TCTRL*n*[CHAIN] = 0

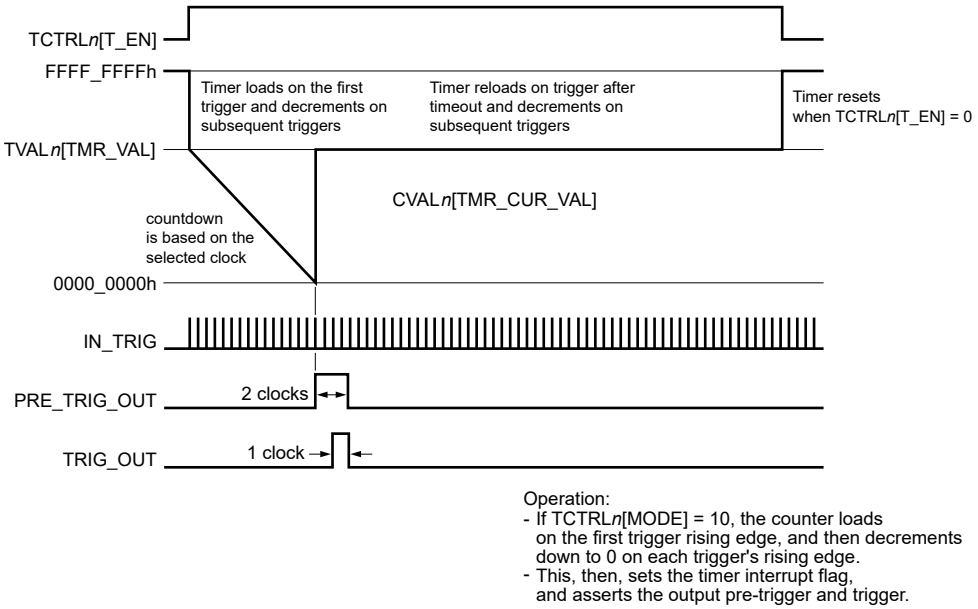


Figure 387. Case 2: TSOT = X, TROT = X, TSOI = 1, CHAIN = 0

66.3.8.12 Case 1: TCTRL*n*[MODE] = 11

The following figure represents case 1 in which TCTRL*n*[MODE] = 11. LPIT works in 32-bit trigger capture mode and is configured as follows:

- TCTRL*n*[TSOT] = X
- TCTRL*n*[TROT] = 0
- TCTRL*n*[TSOI] = 0
- TCTRL*n*[CHAIN] = 0

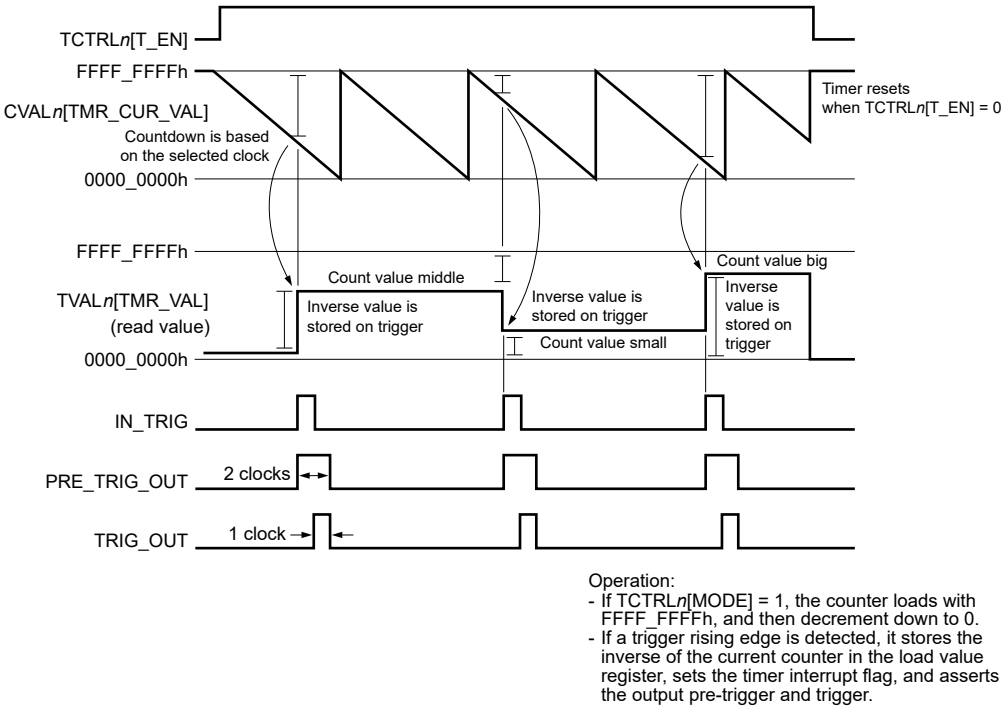


Figure 388. Case 1: TSOT = X, TROT = 0, TSOI = 0, CHAIN = 0

66.3.8.13 Case 2: TCTRL*n*[MODE] = 11

The following figure represents case 2 in which TCTRL*n*[MODE] = 11. LPIT works in 32-bit trigger capture mode and is configured as follows:

- TCTRL*n*[TSOT] = X
- TCTRL*n*[TROT] = 1
- TCTRL*n*[TSOI] = 0
- TCTRL*n*[CHAIN] = 0

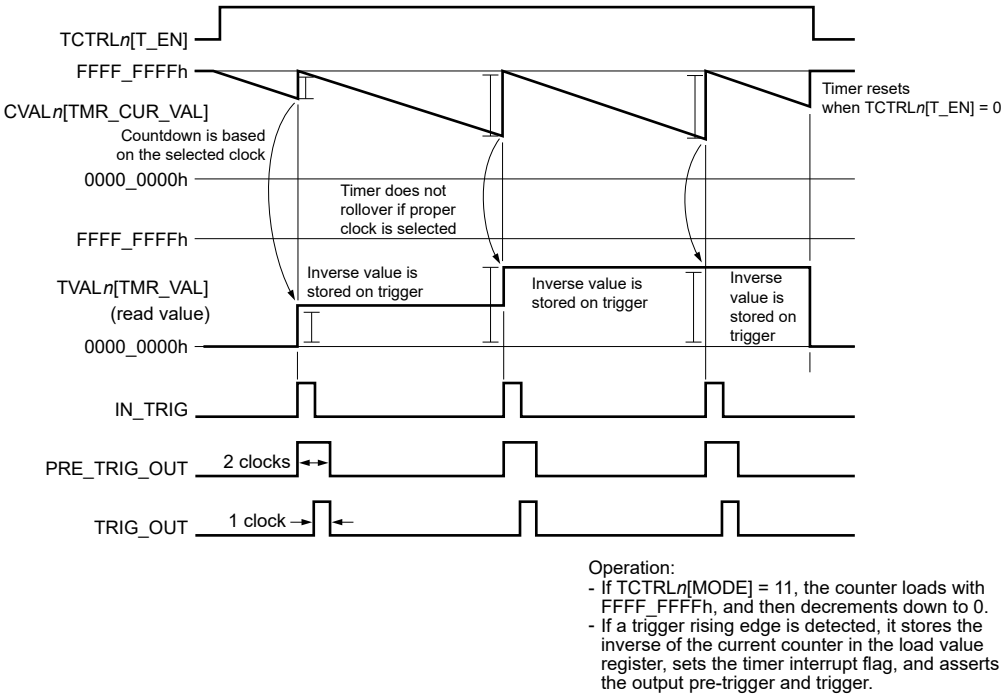


Figure 389. Case 2: TSOT = X, TROT = 1, TSOI = 0, CHAIN = 0

66.3.8.14 Case 3: TCTRL*n*[MODE] = 11

The following figure represents case 3 in which TCTRL*n*[MODE] = 11. LPIT works in 32-bit trigger capture mode and is configured as follows:

- TCTRL*n*[TSOT] = X
- TCTRL*n*[TROT] = 0
- TCTRL*n*[TSOI] = 1
- TCTRL*n*[CHAIN] = 0

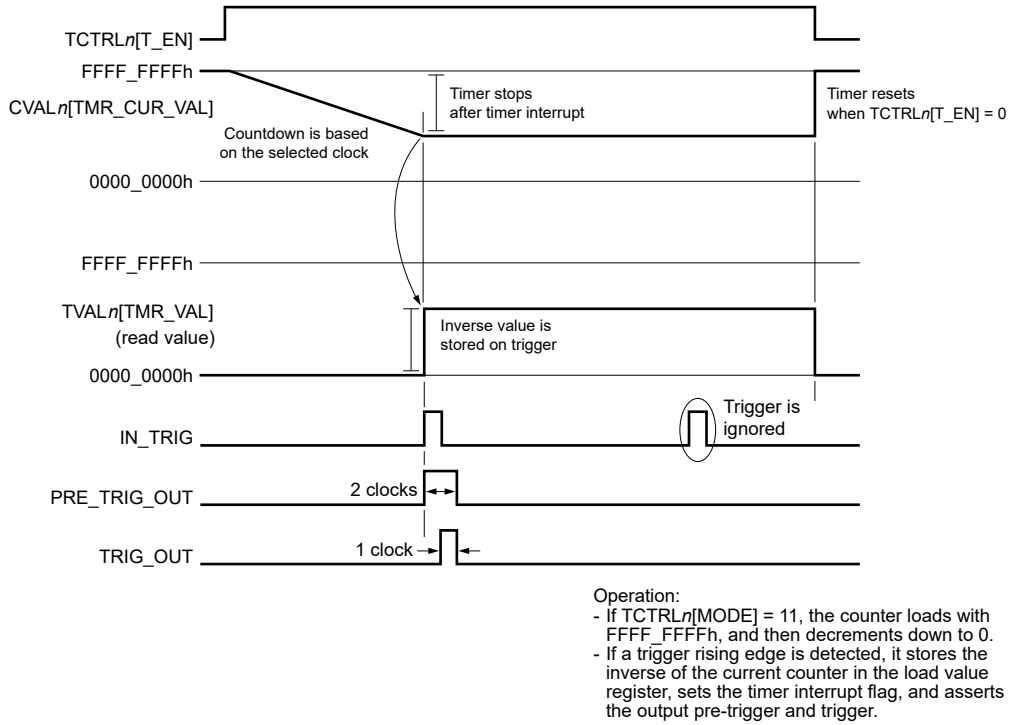


Figure 390. Case 3: TSOT = X, TROT = 0, TSOI = 1, CHAIN = 0

66.3.8.15 Case 4: TCTRL*n*[MODE] = 11

Case 4, in which TCTRL*n*[MODE] = 11, is the same as case 3, except that the timer reloads to FFFF_FFFFh and then stops. The timer does not start until TCTRL*n*[T_EN] becomes 1 again. In this case, LPIT works in 32-bit trigger capture mode and is configured as follows:

- TCTRL*n*[TSOT] = X
- TCTRL*n*[TROT] = 1
- TCTRL*n*[TSOI] = 1
- TCTRL*n*[CHAIN] = 0

66.3.9 Timer chaining

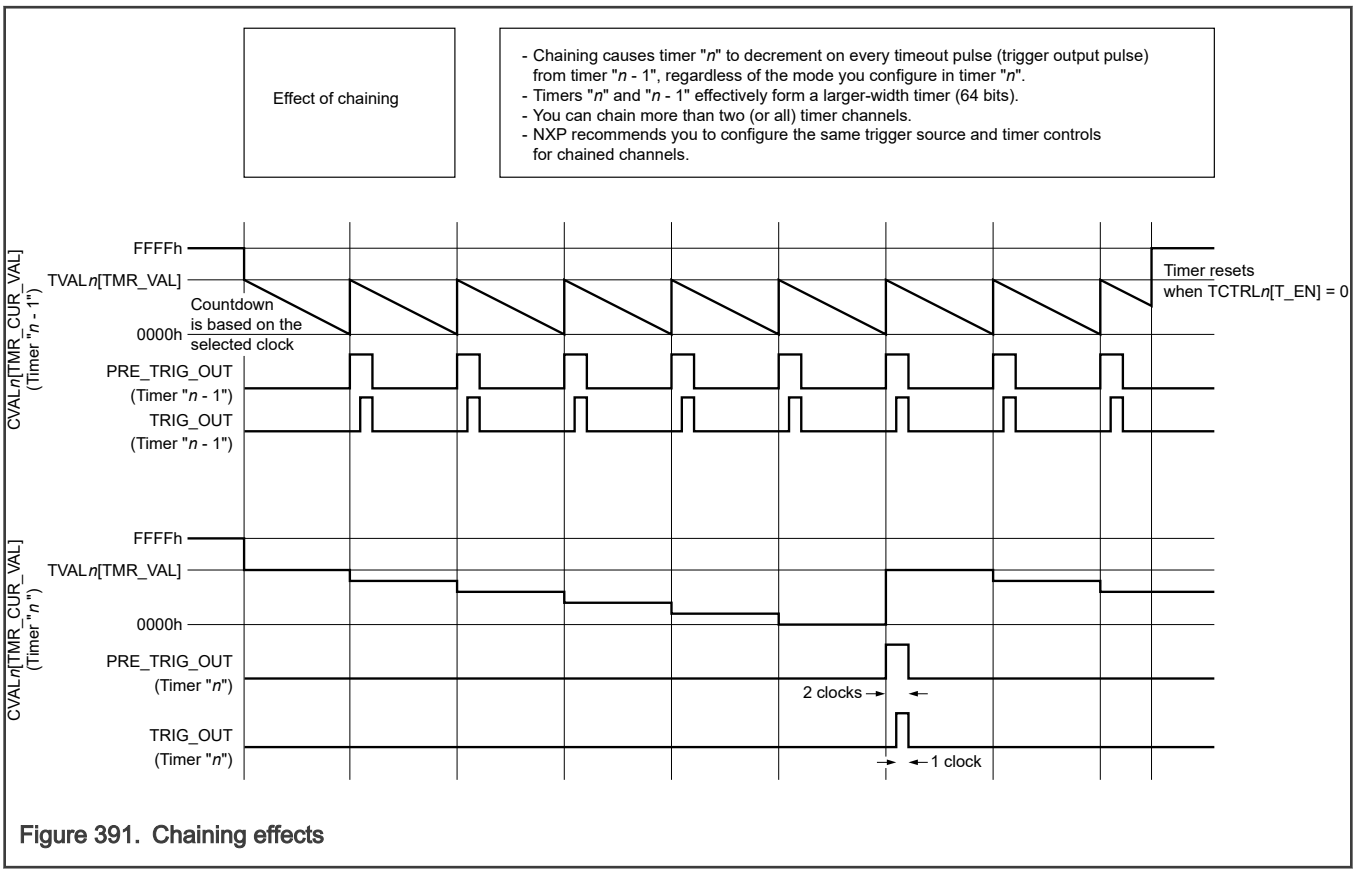


Figure 391. Chaining effects

66.4 Initialization

Table 613. Initializing LPIT

Step	Action	How or why to perform the step
1	Enable the peripheral clock	By writing 1 to MCR[M_CEN]. **NOTE** • Accessing certain registers (Module Status (MSR), Set Timer Enable (SETTEN), Clear Timer Enable (CLRTEN), Timer Value (TVAL0 - TVAL3), Current Timer Value (CVAL0 - CVAL3), and Timer Control (TCTRL0 - TCTRL3)) while MCR[M_CEN] = 0 leads to the assertion of a transfer error for that bus access. However, writing to CVAL*n* and reserved registers generates a transfer error. • There might be additional clock gating fields in the chip that gate the peripheral clock to this module. When enabling the clock to this module, you must configure those additional clock gating fields (in addition to configuring MCR[M_CEN]).
2	Wait for four peripheral clock cycles	To allow time for clock synchronization and reset deassertion.

Table continues on the next page...

Table 613. Initializing LPIT (continued)

Step	Action	How or why to perform the step
3	Configure timer control fields	For each timer channel that is to be enabled: • Timer mode of operation fields, TCTRL*n*[MODE] • Trigger source selection fields, TCTRL*n*[TRG_SEL] and TCTRL*n*[TRG_SRC] • Trigger control fields, TCTRL*n*[TROT], TCTRL*n*[TSOT], and TCTRL*n*[TSOI] **NOTE** You must not update timer control fields when the timer is disabled. You can disable a timer by using any one of the following methods: • Write 1 to the specific timer's CLRTEN[CLR_T_EN_*n*] field. • Write 0 to TCTRL*n*[T_EN] for that channel.
4	Configure the channels that are to be chained	By writing 1 to TCTRL*n*[CHAIN] in the corresponding channel's Timer Control (TCTRL0 - TCTRL3).
5	Set the timer timeout value	By programming an appropriate value in TVAL*n*[TMR_VAL] for the channels that you configure in Compare mode.
6	Configure MIER[TIE*n*]	For those channels that are required to generate interrupts after timer timeouts.
7	Configure the low-power modes of the module	By writing 1 to MCR[DBG_EN] and MCR[DOZE_EN]. This is common to all timer channels.
8	Enable the channel timers	By writing 1 to the corresponding TCTRL*n*[T_EN].

NOTE

When you enable a timer channel in Compare mode, the first decrement takes an additional one or two clock cycles because of synchronization logic. This results in the first compare (and therefore interrupt and hardware trigger) occurring slightly later. A faster counter clock minimizes this impact.

Additionally,

- For channels that you configure in Capture mode, you can read the timer value from Timer Value (TVAL0 - TVAL3) when a channel timeout occurs.
- At any time, you can read the current value of the timer for any channel by reading the corresponding channel's Current Timer Value (CVAL0 - CVAL3). Ensure that MCR[M_CEN] = 1 when doing so.
- MSR[TIF*n*] are asserted after timer timeout. To clear these timer interrupt flags, write 1 to them.

66.5 Memory map and registers

66.5.1 LPIT register descriptions

The LPIT memory map comprises 32-bit aligned registers, which you can access via 8-bit, 16-bit, or 32-bit accesses. Read and write accesses to reserved locations generate a transfer error, and the read bus shows all 0s.

NOTE

- The memory map and complete module are in big-endian (BE) format.
- LPIT does not check whether programmed values in these registers are correct—you must write the correct values.

66.5.1.1 LPIT memory map

LPIT1 base address: 442F_0000h

LPIT2 base address: 424C_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VERID)	32	R	0100_0000
4	Parameter (PARAM)	32	R	0000_0404
8	Module Control (MCR)	32	RW	0000_0000
C	Module Status (MSR)	32	RW	0000_0000
10	Module Interrupt Enable (MIER)	32	RW	0000_0000
14	Set Timer Enable (SETTEN)	32	RW	0000_0000
18	Clear Timer Enable (CLRTEN)	32	RW	0000_0000
20	Timer Value (TVAL0)	32	RW	0000_0000
24	Current Timer Value (CVAL0)	32	R	FFFF_FFFF
28	Timer Control (TCTRL0)	32	RW	0000_0000
30	Timer Value (TVAL1)	32	RW	0000_0000
34	Current Timer Value (CVAL1)	32	R	FFFF_FFFF
38	Timer Control (TCTRL1)	32	RW	0000_0000
40	Timer Value (TVAL2)	32	RW	0000_0000
44	Current Timer Value (CVAL2)	32	R	FFFF_FFFF
48	Timer Control (TCTRL2)	32	RW	0000_0000
50	Timer Value (TVAL3)	32	RW	0000_0000
54	Current Timer Value (CVAL3)	32	R	FFFF_FFFF
58	Timer Control (TCTRL3)	32	RW	0000_0000

66.5.1.2 Version ID (VERID)

Contains design version specification numbers.

Offset

Register	Offset
VERID	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-24 MAJOR	Major Version Number Indicates the major version number for the module design specification.
23-16 MINOR	Minor Version Number Indicates the minor version number for the module design specification.
15-0 FEATURE	Feature Number Indicates the feature set number.

66.5.1.3 Parameter (PARAM)

Provides parameter settings that are used when incorporating this module into the chip.

Offset

Register	Offset
PARAM	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EXT_TRIG								CHANNEL							
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

Fields

Field	Description
31-16 —	Reserved
15-8 EXT_TRIG	Number of External Trigger Inputs Specifies the number of external triggers implemented in this chip.
7-0 CHANNEL	Number of Timer Channels Specifies the number of timer channels implemented in this chip.

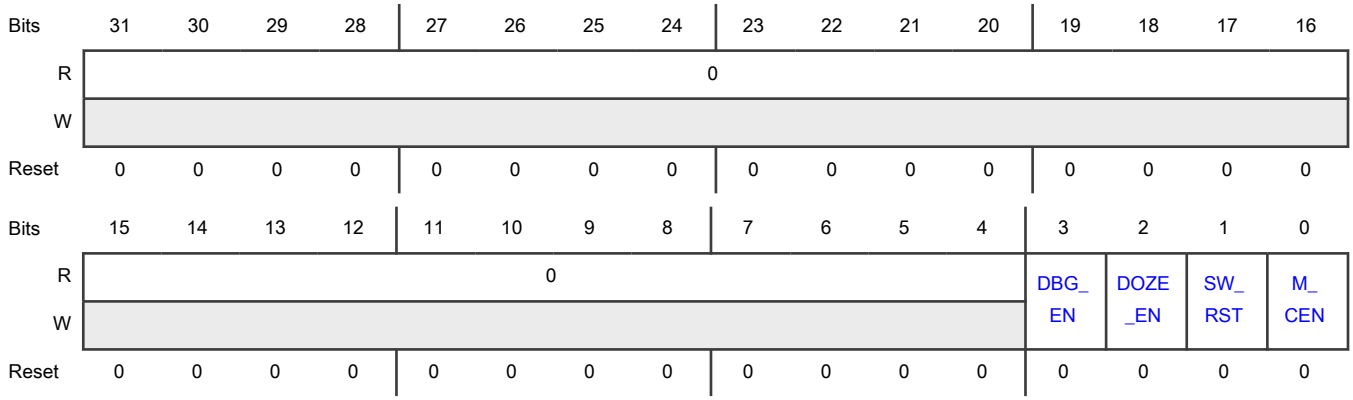
66.5.1.4 Module Control (MCR)

Contains software reset, clock enable, and mode enable fields.

Offset

Register	Offset
MCR	8h

Diagram



Fields

Field	Description
31-4 —	Reserved
3 DBG_EN	Debug Mode Enable Stops the timer channels when the chip enters Debug mode. 0 - Stops timer channels

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Allows timer channels to continue running
2 DOZE_EN	DOZE Mode Enable Stops the timer channels when the chip enters Doze mode. 0 - Stops timer channels 1 - Allows timer channels to continue running
1 SW_RST	Software Reset Resets all timer channels and registers, except Module Status (MSR). This field remains 1 until software clears it. Before clearing this field, software must wait for four peripheral clocks (for clock synchronization and reset propagation). 0 - Does not reset 1 - Resets
0 M_CEN	Module Clock Enable Enables the peripheral clock to LPIT module timers. This field must become 1 when accessing the following registers: • Module Status (MSR) • Set Timer Enable (SETTEN) • Clear Timer Enable (CLRTEN) • Timer Value (TVAL0 - TVAL3) • Current Timer Value (CVAL0 - CVAL3) • Timer Control (TCTRL0 - TCTRL3) The following considerations apply when using this field: • You must enable both the bus and peripheral clocks to allow clock synchronization and update of the aforementioned registers. Accessing these registers when MCR[M_CEN] = 0 asserts a transfer error for that bus cycle. • Writing to Current Timer Value (CVAL0 - CVAL3) and reserved registers always generates a transfer error. **NOTE** There may be additional clock gating fields available in this chip that gate the peripheral clock to LPIT. You must configure those additional clock gating fields appropriately to enable the peripheral clock to LPIT. 0 - Disable 1 - Enable

66.5.1.5 Module Status (MSR)

Contains channel timer interrupt flags.

NOTE

Unless the peripheral clock to the timers is enabled (MCR[M_CEN] = 1), reading or writing to this register generates a transfer error.

Offset

Register	Offset
MSR	Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												TIF3	TIF2	TIF1	TIF0
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved
3 TIF3	Channel 3 Timer Interrupt Flag Specifies whether the channel 3 timer has timed out. In compare modes, at the end of the timer period, this flag becomes 1. In capture modes, when the trigger asserts, this flag becomes 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not timed out 1 - Timed out When writing 0 - No effect 1 - Clear the flag
2	Channel 2 Timer Interrupt Flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
TIF2	Specifies whether the channel 2 timer has timed out. In compare modes, at the end of the timer period, this flag becomes 1. In capture modes, when the trigger asserts, this flag becomes 1. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not timed out 1 - Timed out When writing 0 - No effect 1 - Clear the flag
1 TIF1	Channel 1 Timer Interrupt Flag Specifies whether the channel 1 timer has timed out. In compare modes, this flag becomes 1 at the end of the timer period. In capture modes, this flag becomes 1 when the trigger asserts. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not timed out 1 - Timed out When writing 0 - No effect 1 - Clear the flag
0 TIF0	Channel 0 Timer Interrupt Flag Specifies whether the channel 0 timer has timed out. In compare modes, this flag becomes 1 at the end of the timer period. In capture modes, this flag becomes 1 when the trigger asserts. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Not timed out 1 - Timed out

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When writing 0 - No effect 1 - Clear the flag

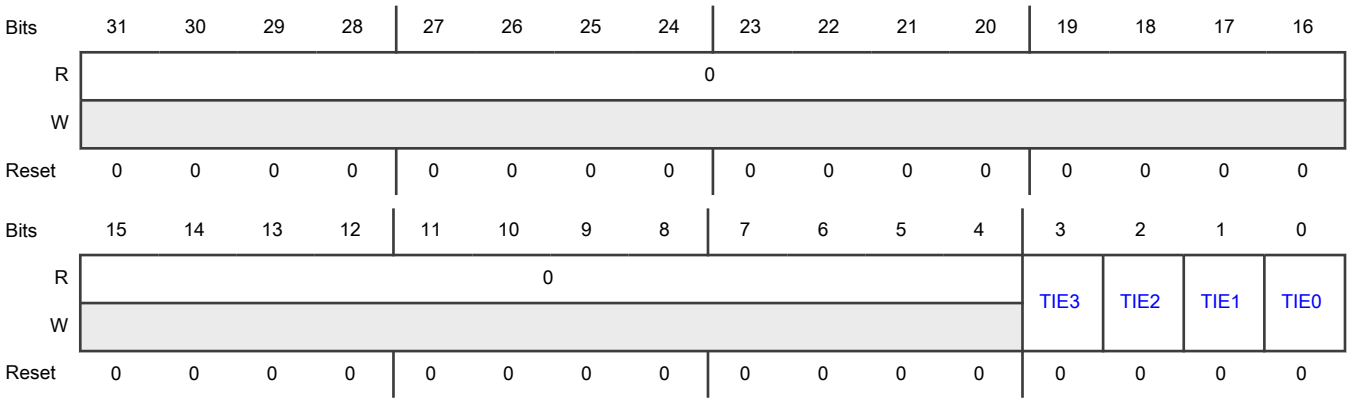
66.5.1.6 Module Interrupt Enable (MIER)

Contains channel timer interrupt enable fields.

Offset

Register	Offset
MIER	10h

Diagram



Fields

Field	Description
31-4 —	Reserved
3 TIE3	Channel 3 Timer Interrupt Enable Enables interrupt generation when: • This field = 1. • The corresponding timer interrupt flag, MSR[TIF3] = 1. 0 - Disable 1 - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 TIE2	Channel 2 Timer Interrupt Enable Enables interrupt generation when: • This field = 1. • The corresponding timer interrupt flag, MSR[TIF2] = 1. 0 - Disable 1 - Enable
1 TIE1	Channel 1 Timer Interrupt Enable Enables interrupt generation when: • This field = 1. • The corresponding timer interrupt flag, MSR[TIF1] = 1. 0 - Disable 1 - Enable
0 TIE0	Channel 0 Timer Interrupt Enable Enables interrupt generation when: • This field = 1. • The corresponding timer interrupt flag, MSR[TIF0] = 1. 0 - Disable 1 - Enable

66.5.1.7 Set Timer Enable (SETTEN)

Allows the simultaneous enabling of timer channels.

You can enable timer channels by using either of the following ways:

- Writing 1 to TCTRL*n*[T_EN] in the respective TCTRL*n* register.
- Writing 1 to the corresponding SETTEN[SET_T_EN_*n*] field.

To disable timer channels simultaneously, use Clear Timer Enable (CLRTEN). Writing 0 to the fields of this register has no effect.

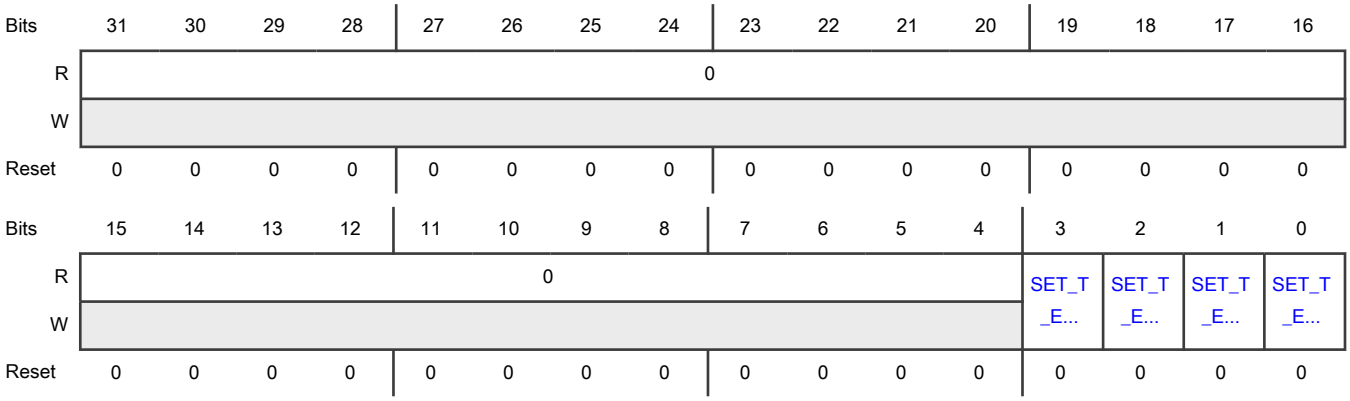
NOTE

Unless the peripheral clock to the timers is enabled (MCR[M_CEN] = 1), reading or writing to this register generates a transfer error.

Offset

Register	Offset
SETTEN	14h

Diagram



Fields

Field	Description
31-4 —	Reserved
3 SET_T_EN_3	Set Timer 3 Enable Works with TCTRL3[T_EN] and enables timer channel 3. Writing 0 to this field does not disable the counter; rather it has no effect. This field becomes 0 if any of the following conditions is true: • TCTRL3[T_EN] = 0 • You write 1 to CLRTEN[CLR_T_EN_3] 0 - No effect 1 - Enables timer channel 3
2 SET_T_EN_2	Set Timer 2 Enable Works with TCTRL2[T_EN] and enables timer channel 2. Writing 0 to this field does not disable the counter; rather it has no effect. This field becomes 0 if any of the following conditions is true: • TCTRL2[T_EN] = 0 • You write 1 to CLRTEN[CLR_T_EN_2] 0 - No Effect 1 - Enables timer channel 2
1 SET_T_EN_1	Set Timer 1 Enable Works with TCTRL1[T_EN] and enables timer channel 1. Writing 0 to this field does not disable the counter; rather it has no effect. This field becomes 0 if any of the following conditions is true:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• TCTRL1[T_EN] = 0 • You write 1 to CLRTEN[CLR_T_EN_1] 0 - No Effect 1 - Enables timer channel 1
0 SET_T_EN_0	Set Timer 0 Enable Works with TCTRL0[T_EN] and enables timer channel 0. Writing 0 to this field does not disable the counter; rather it has no effect. This field becomes 0 if any of the following conditions is true: • TCTRL0[T_EN] = 0 • You write 1 to CLRTEN[CLR_T_EN_0] 0 - No effect 1 - Enables timer channel 0

66.5.1.8 Clear Timer Enable (CLRTEN)

Allows the simultaneous disabling of timer channels.

NOTE

Unless the peripheral clock to the timers is enabled (MCR[M_CEN] = 1), reading or writing to this register generates a transfer error.

Offset

Register	Offset
CLRTEN	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												0	0	0	0
W													CLR_T_E...	CLR_T_E...	CLR_T_E...	CLR_T_E...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved
3 CLR_T_EN_3	Clear Timer 3 Enable Works with TCTRL3[T_EN] and disables timer channel 3. Writing 1 to this self-clearing field does not enable the counter. It disables timer channel 3. It also turns TCTRL3[T_EN] = 0 for timer channel 3. 0 - No action 1 - Turns TCTRL3[T_EN] = 0 for timer channel 3
2 CLR_T_EN_2	Clear Timer 2 Enable Works with TCTRL2[T_EN] and disables timer channel 2. Writing 1 to this self-clearing field does not enable the counter. It disables timer channel 2. It also turns TCTRL2[T_EN] = 0 for timer channel 2. 0 - No action 1 - Turns TCTRL2[T_EN] = 0 for timer channel 2
1 CLR_T_EN_1	Clear Timer 1 Enable Works with TCTRL1[T_EN] and disables timer channel 1. Writing 1 to this self-clearing field does not enable the counter. It disables timer channel 1. It also turns TCTRL1[T_EN] = 0 for timer channel 1. 0 - No action 1 - Turns TCTRL1[T_EN] = 0 for timer channel 1
0 CLR_T_EN_0	Clear Timer 0 Enable Works with TCTRL0[T_EN] and disables timer channel 0. Writing 1 to this self-clearing field does not enable the counter. It disables timer channel 0. It also turns TCTRL0[T_EN] = 0 for timer channel 0. 0 - No action 1 - Turns TCTRL0[T_EN] = 0 for timer channel 0

66.5.1.9 Timer Value (TVAL0 - TVAL3)

Contains timer values:

- In compare modes, this register selects the timeout period for the timer channels.
- In capture modes, this register is loaded with the value of the counter when the trigger asserts.

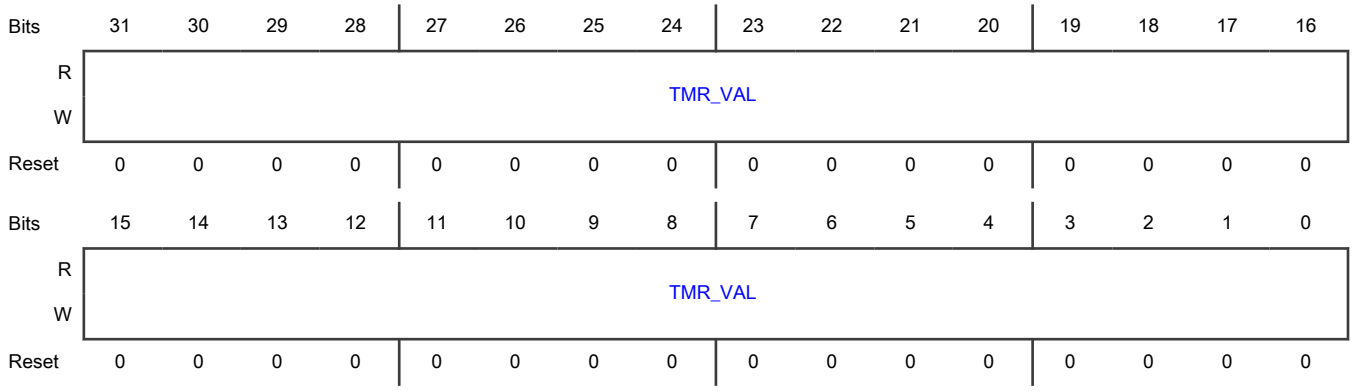
NOTE

Unless the peripheral clock to the timers is enabled (MCR[M_CEN] = 1), reading or writing to this register generates a transfer error.

Offset

Register	Offset
TVAL0	20h
TVAL1	30h
TVAL2	40h
TVAL3	50h

Diagram



Fields

Field	Description
31-0 TMR_VAL	Timer Value Specifies the timer value and whether it is valid. • In Compare mode, this field specifies the timer channel start value: — The timer counts down for (TVAL*n* + 1) cycles until the timer reaches 0, then the timer generates an interrupt and loads the TVAL*n* value again. — Writing a new value to TVAL*n* does not restart the timer channel; instead, the new value is loaded "after the timer expires." — To abort the current timer cycle and start a timer period with a new value, you must disable the timer channel and then enable it again. • In Capture mode, whenever the trigger asserts, this register stores the inverse of the counter value. 0000_0000_0000_0000_0000_0000_0000_0000,0000_0000_0000_0000_0000_0000_0000_0001 - Invalid load value in Compare mode 0000_0000_0000_0000_0000_0000_0000_0010-1111_1111_1111_1111_1111_1111_1111_1111 - In Compare mode: the value to be loaded; in Capture mode, the value of the timer

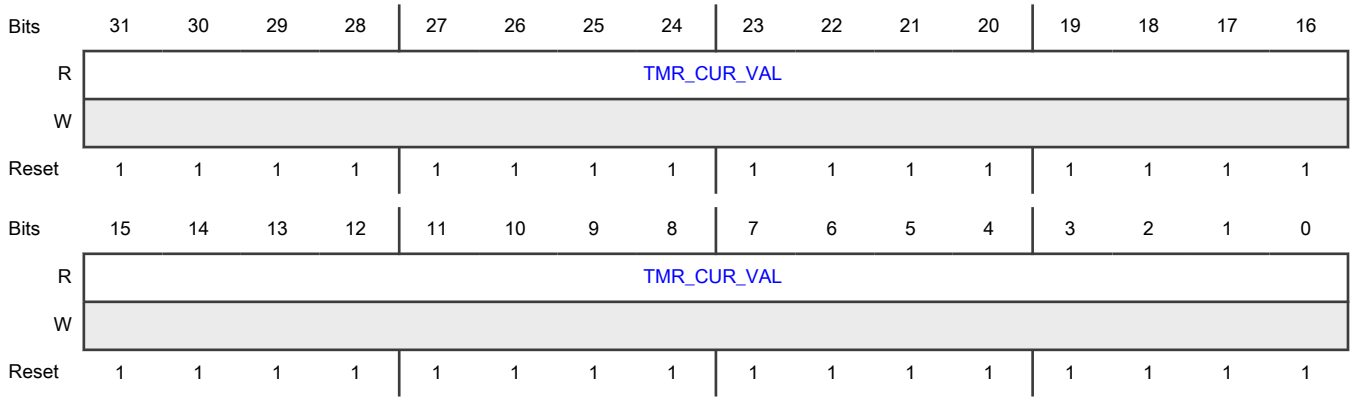
66.5.1.10 Current Timer Value (CVAL0 - CVAL3)

Indicates the current timer counter value.

Offset

Register	Offset
CVAL0	24h
CVAL1	34h
CVAL2	44h
CVAL3	54h

Diagram



Fields

Field	Description
31-0 TMR_CUR_VAL	Current Timer Value Represents the current timer value, if the timer is enabled.

66.5.1.11 Timer Control (TCTRL0 - TCTRL3)

Contains control fields for each timer channel:

- TRG_SEL for trigger selection
- TRG_SRC for trigger source selection
- TROT for timer reload
- TSOI for timer stoppage
- TSOT for timer decrementing
- MODE for timer operation mode selection
- CHAIN for channel chaining

You must update timer controls when the timer is disabled, and use either of the following ways to disable a timer:

- By writing 1 to the specific timer's CLRTEN[CLR_T_EN_*n*] field.
- By writing 0 to T_EN for that channel.

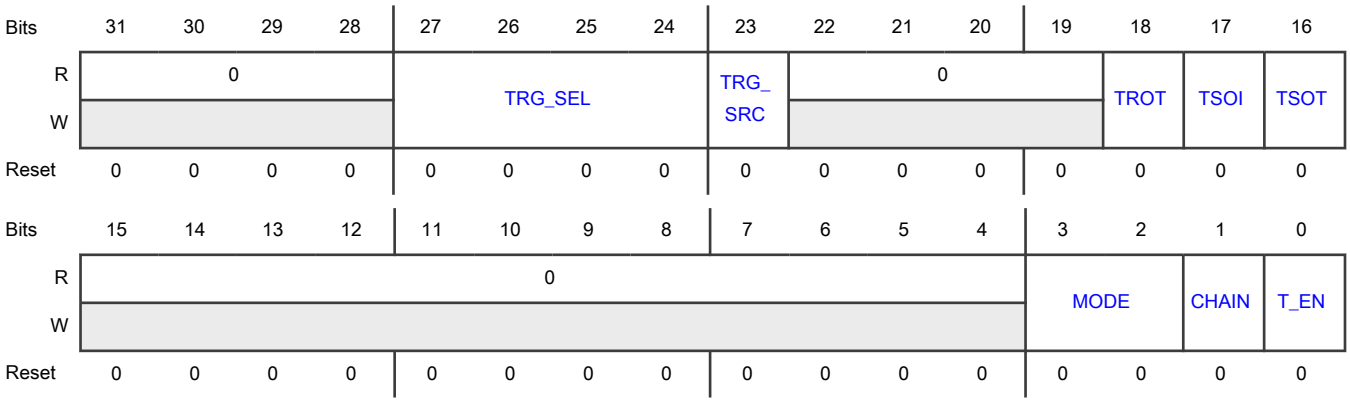
NOTE

Unless the peripheral clock to the timers is enabled (MCR[M_CEN] = 1), reading or writing to this register generates a transfer error.

Offset

Register	Offset
TCTRL0	28h
TCTRL1	38h
TCTRL2	48h
TCTRL3	58h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-24 TRG_SEL	Trigger Select Selects the trigger to use for starting and/or reloading the LPIT timer. This field selects one trigger from the set of internal or external triggers that TRG_SRC provides. TRG_SRC helps you make a choice between internal and external trigger signals for each channel. **NOTE** You must change the value of this field when the LPIT timer channel is disabled. 0000-0011 - Timer channel 0–3 trigger source 0100-1111 - Reserved
23 TRG_SRC	Trigger Source Selects whether to use internal or external trigger sources.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can select the trigger to be used by using this field or the TRG_SEL field. If a channel does not have an associated external trigger, write 1 to TRG_SRC. See LPIT chip-specific information for the available external trigger options. 0 - External 1 - Internal
22-19 —	Reserved
18 TROT	Timer Reload on Trigger Specifies whether the timer reloads after the selected trigger. If this field = 1, the LPIT timer reloads when a rising edge is detected on the selected trigger input. The trigger input is ignored if LPIT is disabled during Debug mode (MCR[DBG_EN] = 0) or Doze mode (MCR[DOZE_EN] = 0). 0 - Does not reload 1 - Reloads
17 TSOI	Timer Stop on Interrupt Controls whether the channel timer stops after being timed out. If this field = 0, the channel timer does not stop after timeout. If this field = 1, the channel timer stops after a timeout and then restarts based on the configuration of TSOT. If TSOT = 0, the channel timer restarts after a rising edge on T_EN is detected, which means that the timer channel is disabled and then enabled. If TSOT = 1, the channel timer restarts after a rising edge on the selected trigger is detected. 0 - Does not stop 1 - Stops
16 TSOT	Timer Start on Trigger Controls when the timer starts decrementing. If this field = 0, the timer starts decrementing immediately based on the restart condition (controlled by TSOI). If this field = 1, the timer starts decrementing when a rising edge on a selected trigger is detected. 0 - Immediately 1 - When a rising edge is detected
15-4 —	Reserved
3-2 MODE	Timer Operation Mode Configures the channel timer's mode of operation. This field controls how the timer decrements. 00 - 32-bit periodic counter 01 - Dual 16-bit periodic counter

Table continues on the next page...

Table continued from the previous page...

Field	Description
	10 - 32-bit trigger accumulator 11 - 32-bit trigger input capture
1 CHAIN	Chain Channel Specifies whether channel chaining is enabled. If channel chaining is enabled, the timer channel decrements after the previous channel's timeout (when timer channel N-1 trigger asserts). Timer channel 0 cannot be chained. If channel chaining is disabled, the channel timer runs independently. 0 - Disabled 1 - Enabled
0 T_EN	Timer Enable Enables the timer channel. 0 - Disable 1 - Enable

Chapter 67
Timer/PWM Module (TPM)

67.1 Chip-specific TPM information

Table 614. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The chip includes six instances of TPM timers. LPTPM1, LPTPM2 are in AONMIX and LPTPM3, 4, 5, 6 are in WAKEUPMIX. Each TPM has its own clock root.

NOTE

TPM1_CH1, TPM1_CH3 are only supported as outputs, not inputs.

NOTE

TPM1 is clocked by AON bus and TPM3 is clocked by WAKEUP bus. TPM1 and TPM3 have their trigger output 0 used as input by CAN1 and CAN2 respectively. Because of this, these instances are clocked by the bus clock - the same clock used to clock the respective CAN module (bus_aon_clk_root and bus_wakeup_clk_root respectively) in order to ensure that these signals are timed synchronously between these modules.

NOTE

All the output triggers and input triggers are unused in this chip. So the descriptions related to output triggers and input triggers are not applicable.

67.2 Overview

TPM is a 4-channel timer that controls electric motor and power management applications by supporting:

- Input capture
- Output comparison
- Generation of PWM signals

An asynchronous clock that can remain enabled in low-power modes clocks the counter, compare, and capture registers.

67.2.1 Block diagram

TPM uses one input/output (I/O) pin per channel: CH*n* (TPM channel *n*) where *n* is the channel number.

The central component of TPM is a 32-bit counter with a programmable final value. Its counting can be up or up-down.

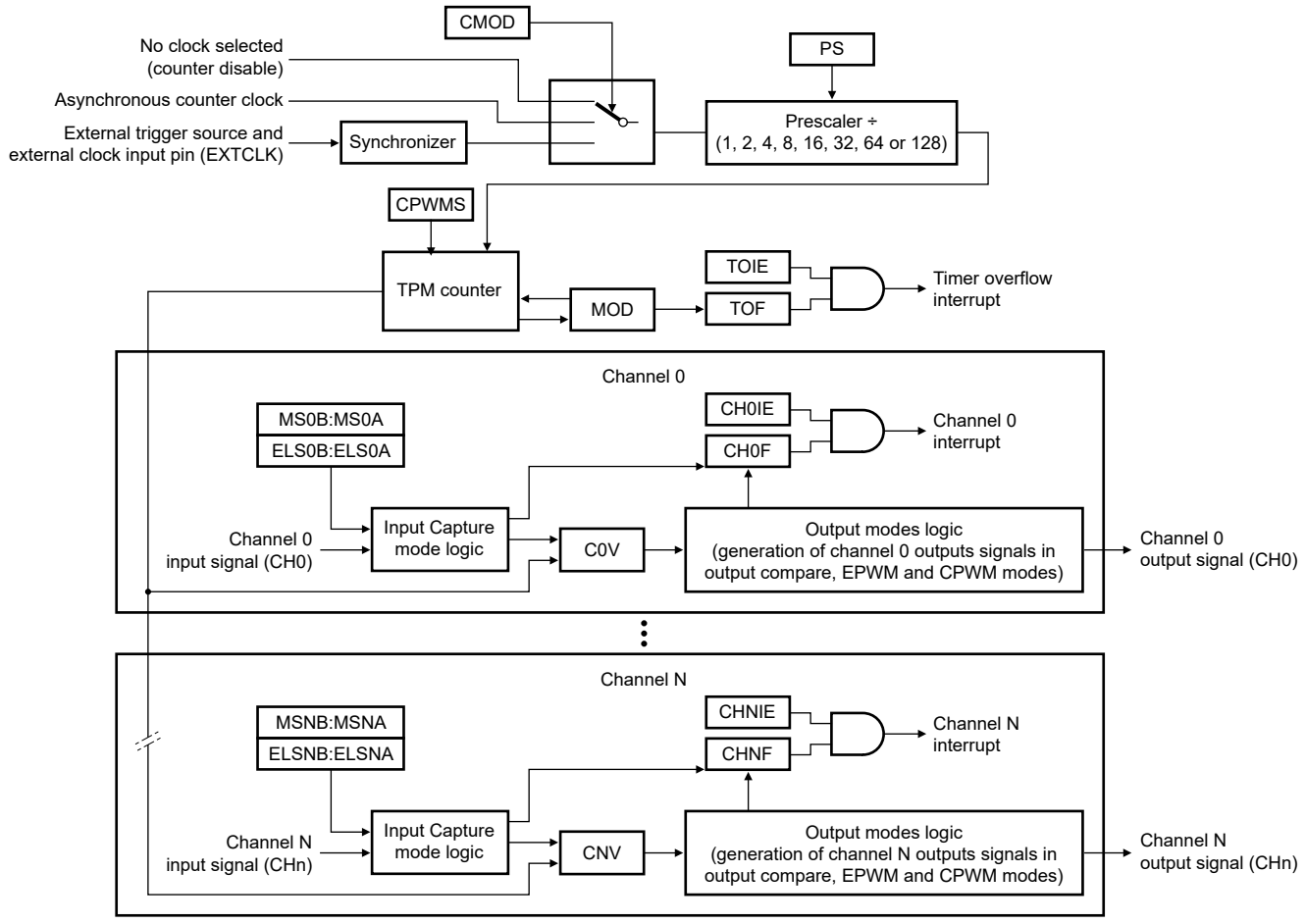


Figure 392. Block diagram

67.2.2 Features

- Supports selectable clock modes:
  - Can increment on every edge of the asynchronous counter clock
  - Can increment on the rising edge of an external clock input that is synchronized to the asynchronous counter clock
- Includes a prescaler divided by 1, 2, 4, 8, 16, 32, 64, or 128
- Includes a 32-bit TPM counter:
  - Can be a free-running counter or a modulo counter
  - The counting can be up or up-down
- Includes 4 channels that can be configured as follows:
  - Input Capture mode: the capture can occur on rising edges, falling edges, or both edges.
  - Output Compare mode: the output signal can be set, cleared, pulsed, or toggled on the match.
  - Edge-Aligned or Center-Aligned PWM mode for all channels.
- Supports the generation of an interrupt and DMA request per channel
- Supports the generation of an interrupt and DMA request when the counter overflows

- Supports selectable trigger input to either reset the counter to 0 or else cause the counter to start incrementing
  - The counter can also optionally stop incrementing on counter overflow.
- Supports the generation of hardware triggers per channel when the channel-interrupt-status flag is set
- Supports the generation of a hardware trigger when the counter overflows

67.3 Functional description

67.3.1 Counter

TPM has a 32-bit counter that the channels use for either input or output modes. The counter updates from the selected clock after it has been divided by the prescaler. The TPM counter can perform one of the following modes of operation:

- Up counting
- Up-down counting

67.3.1.1 Counter Clock mode

SC[CMOD] either disables the TPM counter or enables one of the three possible clock sources for the TPM counter. After any reset, SC[CMOD] becomes 0, which disables the TPM counter. You can configure SC[CMOD] for one of the following counter clock sources:

- Asynchronous counter clock
- External clock input pin
- External trigger source

You can read or write to SC[CMOD] at any time. Disabling the TPM counter by writing 0 to SC[CMOD] does not affect the TPM counter value or other registers, but the TPM counter clock domain must acknowledge this disabling action before SC[CMOD] becomes 0.

The external clock input and external trigger source pass through a synchronizer. The TPM counter clock clocks this synchronizer to ensure that counter transitions are properly aligned to counter clock transitions. Therefore, to meet Nyquist criteria, and also considering jitter, the frequency of the external clock source must be less than half of the counter clock frequency.

67.3.1.2 Prescaler

The counter clock source that you select passes through a prescaler that is a 7-bit counter. SC[PS] defines the value of the prescaler. The following figure shows an example of the prescaler counter and TPM counter.

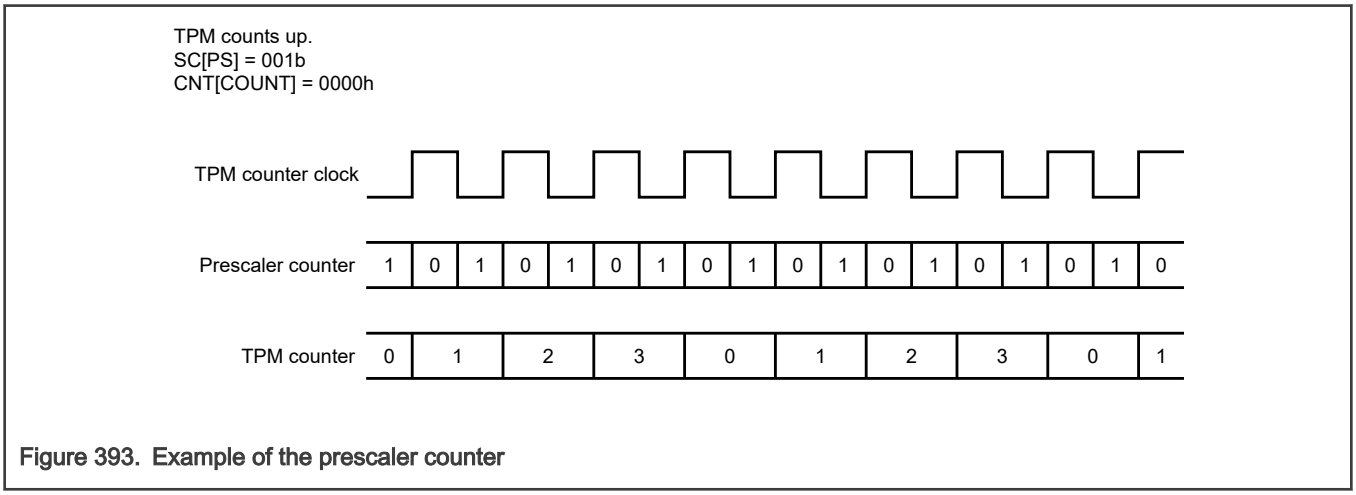


Figure 393. Example of the prescaler counter

67.3.1.3 Up counting

Up counting is enabled when you write 0 to SC[CPWMS].

In this mode, the TPM counter loads with value 0 and increments until it reaches the value defined in MOD[MOD], then the counter reloads with 0.

The TPM period when using up counting is (MOD[MOD] + 0001h) × period of the TPM counter clock.

SC[TOF] becomes 1 when the TPM counter changes from the value defined in MOD[MOD] to 0.

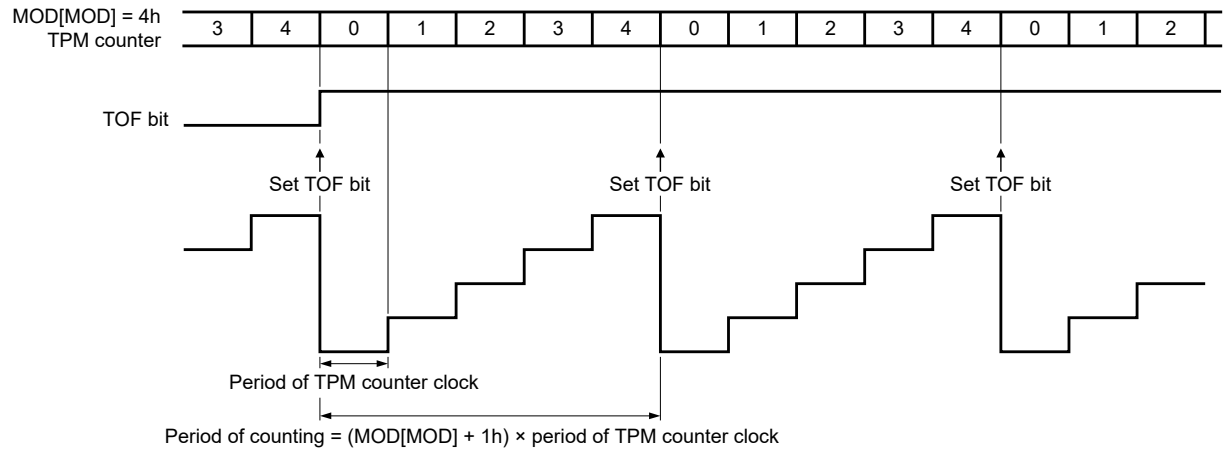


Figure 394. Example of TPM up counting

NOTE

- Configuring MOD[MOD] = 0 is a redundant condition. In this case, the TPM counter is always equal to the value of MOD[MOD], and SC[TOF] remains 1 for each rising edge of the TPM counter clock.
- Configuring MOD[MOD] = 1 and SC[PS] = 0 causes TPM to attempt to set SC[TOF] every second TPM counter clock. Because of synchronization delays between the TPM counter clock and the bus clock, you must write 1 to SC[TOF] to clear it two counter clock cycles before SC[TOF] can become 1 again.

67.3.1.4 Up-down counting

You enable up-down counting when you write 1 to SC[CPWMS]. This mode does not support configuring the value of Modulo (MOD) to be less than 2.

In this mode, the TPM counter loads with value 0 and increments until it reaches the value defined in MOD[MOD]. Then the counter starts decrementing until it returns to 0 and the up-down counting restarts.

The TPM period when using up-down counting is 2 × MOD[MOD] × period of the TPM counter clock.

TPM writes 1 to SC[TOF] when the TPM counter changes from the value defined in MOD[MOD] to (MOD[MOD] – 1).

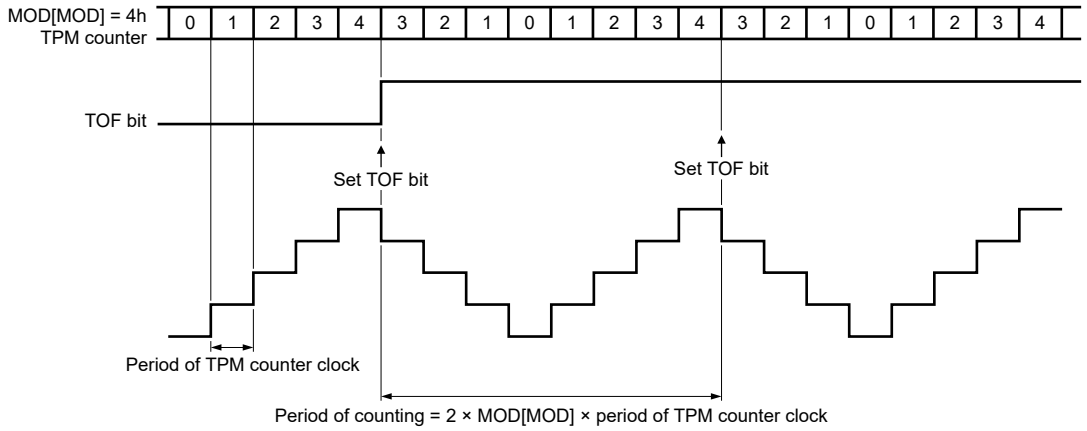


Figure 395. Example of up-down counting

67.3.1.5 Counter reset

Any write to CNT[COUNT] resets the TPM counter and forces the channel outputs to their initial values (except for channels in Output Compare mode).

67.3.1.6 Counter trigger

You can configure the TPM counter to start, stop, or reset in response to a hardware trigger input. The trigger input synchronizes with the asynchronous counter clock, so there is a three-counter-clock delay between the trigger assertion and counter response.

- When CONF[CSOT] = 1, the counter does not start incrementing until the trigger input detects a rising edge.
- When CONF[CSOO] = 1, the counter stops incrementing whenever SC[TOF] = 1. The counter does not increment again unless:
  - It is disabled.
  - CONF[CSOT] = 1 and the trigger input detects a rising edge.
- When CONF[CROT] = 1, the counter resets to 0, as if an overflow occurred, whenever the trigger input detects a rising edge. Any PWM channels are updated to their reload state.
- When CONF[CPOT] = 1, the counter pauses incrementing whenever you assert the trigger input. The counter continues incrementing when the trigger input negates. PWM output channels are forced to their default state.

You can configure the polarity of the external input trigger with CONF[TRGPOL].

When you enable an internal trigger source, the trigger input is selected from one or more channel input capture events. The input capture filters are used with the internal trigger sources. You can use POL[POL*n*] fields to invert the polarity of the input channels.

The following restrictions apply with input capture channel sources:

- When CONF[CSOT] = 1, the counter starts incrementing only on a rising edge on the channel input, if C*n*SC[ELS*n*A] = 1.
- When CONF[CROT] = 1, the counter resets to 0 on either edge of the channel input, as you configure it using C*n*SC[ELS*n*B] and C*n*SC[ELS*n*A].
- When CONF[CPOT] = 1, the counter pauses incrementing whenever you assert the channel input. PWM output channels are forced to their default state.

67.3.2 Registers updated from write buffers

67.3.2.1 MOD register update

If SC[CMOD] = 0, then Modulo (MOD) updates whenever you write to MOD[MOD].

If SC[CMOD] ≠ 0, then the MOD register updates accordingly as below:

- CPWM mode is disabled (SC[CPWMS] = 0), then the MOD register updates whenever you write to MOD[MOD], and the TPM counter value changes from the value defined in MOD[MOD] to 0.
- CPWM mode is enabled (SC[CPWMS] = 1), then the MOD register updates whenever you write to MOD[MOD], and the TPM counter value changes from the value defined in MOD[MOD] to (MOD[MOD] – 1).

67.3.2.2 C*n*V register update

If SC[CMOD] = 0, then the C*n*V register updates whenever you write to C*n*V[VAL].

If SC[CMOD] ≠ 0, then the C*n*V register updates according to the selected mode:

- If you select Output Compare mode, then the C*n*V register updates on the next TPM counter increment (end of the prescaler counting) after you write to C*n*V[VAL].
- If you select EPWM mode, then the C*n*V register updates after you write to C*n*V[VAL], and the TPM counter value changes from the value defined in MOD[MOD] to 0.
- If you select CPWM mode, then the C*n*V register updates after you write to C*n*V[VAL], and the TPM counter value changes from the value defined in MOD[MOD] to (MOD[MOD] – 1).

67.3.3 Peripheral triggers

See the chip-specific TPM information for more details.

67.3.3.1 Output triggers

TPM generates output triggers for the counter and for each channel that you can use to trigger events in other peripherals. The counter trigger asserts whenever SC[TOF] is 1 and remains asserted until the next increment.

Each TPM channel generates both a pre-trigger output and a trigger output.

1. The pre-trigger output asserts whenever C*n*SC[CH*n*F] sets.
2. The trigger output asserts on the first counter increment after the pre-trigger asserts.
3. Both the trigger and pre-trigger negate on the first counter increment after the trigger asserts.

When (COMBINE[COMBINE*n*] = 1) in Output Compare mode, the pre-trigger output for both channel *n* and channel *n* + 1 asserts when the CH*n*F flag is set and negates when the CH(*n* + 1)F flag is set. The trigger continues to assert on the first counter increment, after the pre-trigger asserts and negates at the same time as the pre-trigger negation.

67.3.3.2 Input trigger

You can configure the TPM input trigger to perform the following operations with configurable polarity:

- Increment the counter on trigger rising edge
- Start incrementing the counter on trigger rising edge
- Reset or reload the counter on trigger rising edge
- Pause the counter while trigger high

The TPM input trigger is synchronized and must remain asserted for at least two counter clock cycles, so that the input trigger can be detected.

67.3.4 Modes of operations

67.3.4.1 Input Capture mode

Input Capture mode is enabled when:

- (SC[CPWMS] = 0)
- (MS*n*B:MS*n*A = 0:0)
- (ELS*n*B:ELS*n*A ≠ 0:0)

When a selected edge occurs on the channel input:

- The C*n*V register captures the current value of the TPM counter.
- It also writes 1 to C*n*SC[CH*n*F] and generates a channel interrupt if C*n*SC[CH*n*IE] = 1 (see Figure 396).

When you configure a channel for input capture, the CH*n* pin is an edge-sensitive input. ELS*n*B:ELS*n*A control fields determine which edge, falling or rising, triggers the input-capture event. To detect the input signal correctly and to meet Nyquist criteria for signal sampling, the maximum frequency for the channel input signal is the counter clock divided by 4.

Input Capture mode ignores any writes to the C*n*V register.

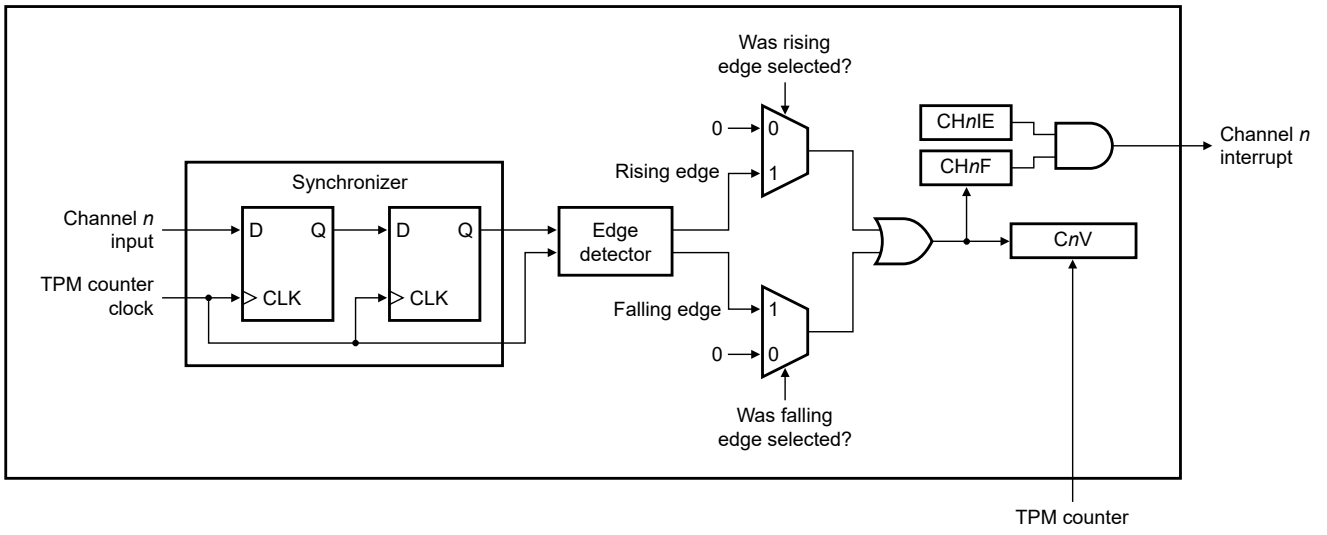


Figure 396. Input Capture mode

The CH*n*F flag is set on the third rising edge of the TPM counter clock after a valid edge occurs on the channel input.

67.3.4.1.1 Input capture filter

The input capture filter function is only available in Input Capture or Software Compare mode when Quadrature Decoder mode is enabled.

First, the TPM counter clock synchronizes the input signal. After synchronization, the input signal enters the filter block (see Figure 397).

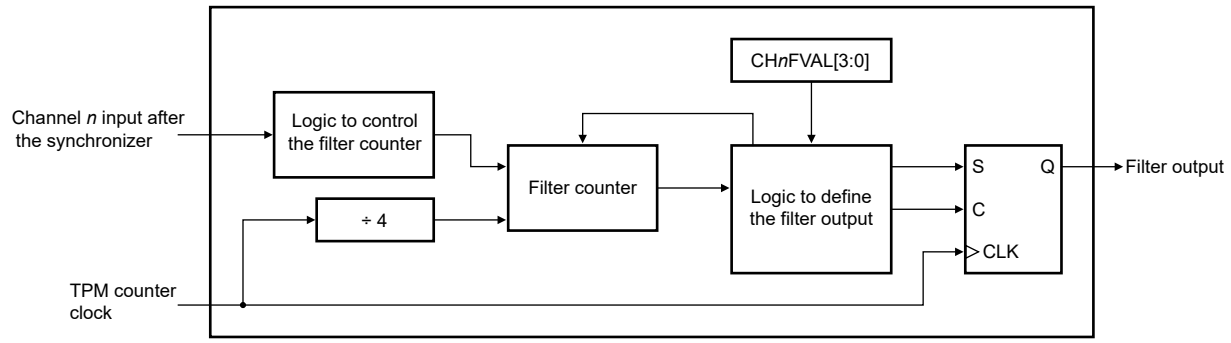


Figure 397. Channel input filter

When there is a state change in the input signal, TPM resets the counter, and it starts counting up. As long as the new state is stable on the input, the counter continues to increment. When the counter is equal to (CH*n*FVAL[3:0] × 4), the state change of the input signal is validated.

TPM resets the counter if the opposite edge appears on the input signal before it is validated. At the next input transition, the counter starts counting again. Any pulse that is less than the minimum value selected by (CH*n*FVAL[3:0] × 4 counter clocks) is considered as a glitch and does not pass through the filter. Figure 398 shows a timing diagram of the input filter.

The filter function is disabled when CH*n*FVAL[3:0] = 0. In this case, the input signal is delayed by 2 rising edges of the counter clock. If CH*n*FVAL[3:0] ≠ 0000, then the input signal is delayed by the minimum pulse width (CH*n*FVAL[3:0] × 4 system clocks) plus a further 3 rising edges of the system clock: two rising edges to the synchronizer, plus one more to the edge detector. In other words, the CH*n*F flag is set for (3 + (4 × CH*n*FVAL[3:0])) counter clock periods after a valid edge occurs on the channel input (see Figure 398).

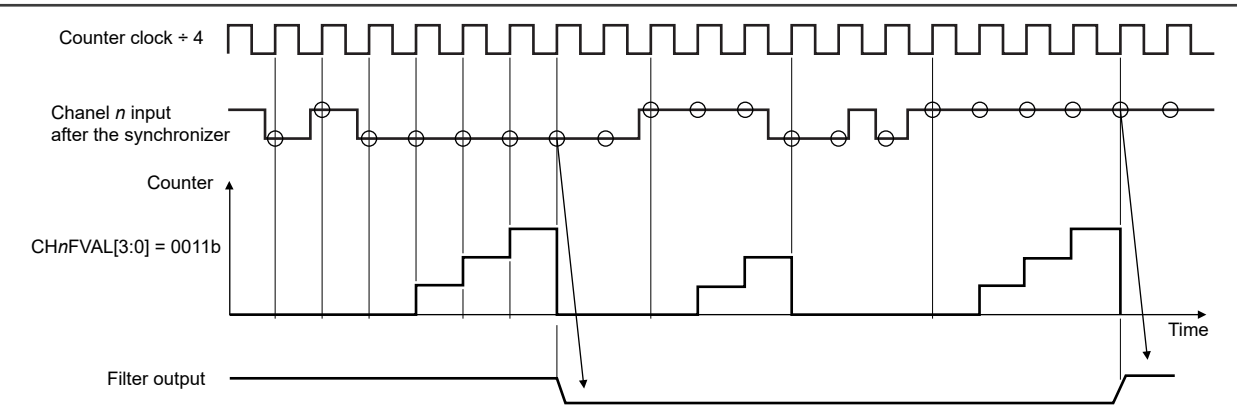


Figure 398. Channel input filter example

67.3.4.2 Output Compare mode

Output Compare mode is enabled when SC[CPWMS] = 0 and MS*n*B:MS*n*A = X:1.

In this mode, TPM can generate timed pulses with programmable:

- Position
- Polarity
- Duration
- Frequency

When the TPM counter matches the value defined in C*n*V[VAL] of an output compare channel, then channel *n* output can be set, cleared, or toggled if MS*n*B = 0. If MS*n*B = 1, then channel *n* output is pulsed high or low (as long as the TPM counter matches the value defined in C*n*V[VAL]).

When you initially configure a channel to Output Compare mode, the channel output is updated with its negated value (logic 0 for set/toggle/pulse high and logic 1 for clear/pulse low).

The channel *n* match (TPM counter = C*n*V[VAL]) writes 1 to C*n*SC[CH*n*F] and generates a channel *n* interrupt (if C*n*SC[CH*n*IE] = 1).

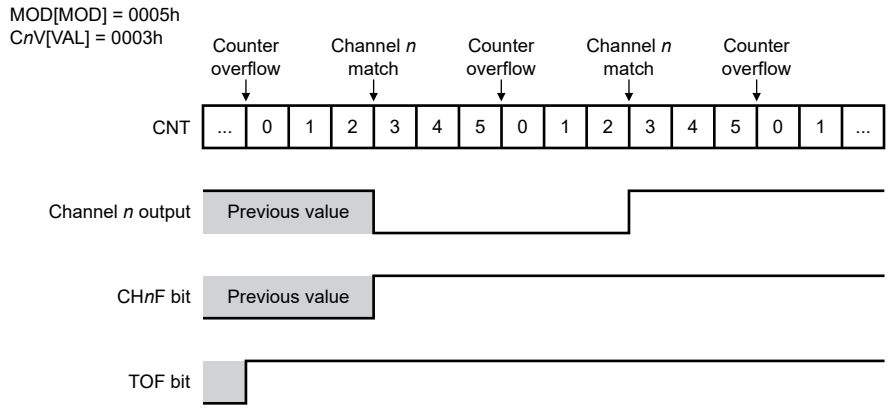


Figure 399. Example of Output Compare mode when a match toggles the channel output

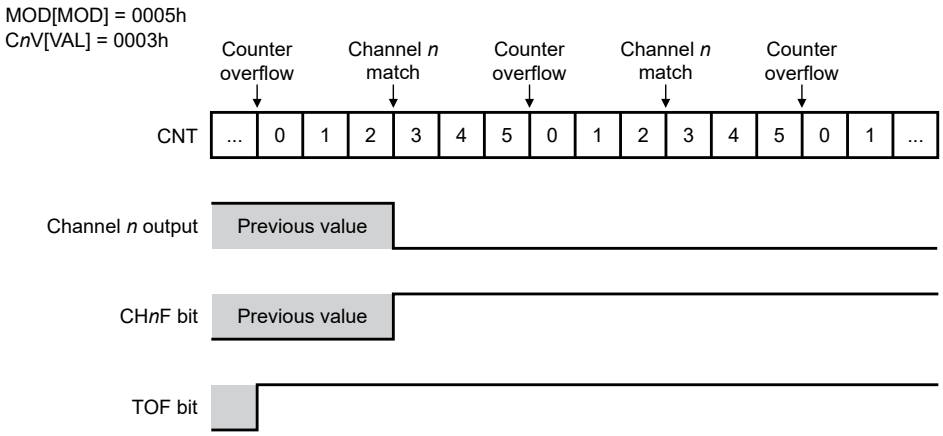


Figure 400. Example of Output Compare mode when a match clears the channel output

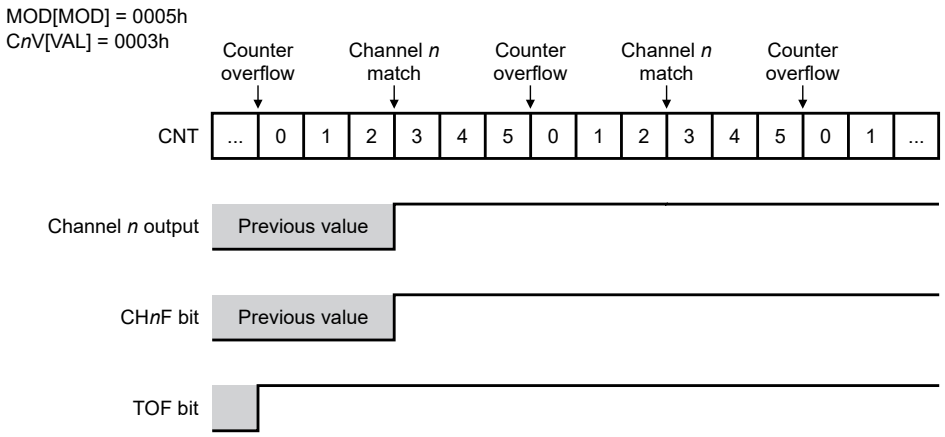


Figure 401. Example of Output Compare mode when a match sets the channel output

You can also use Output Compare mode with:

- SC[CPWMS] = 0 and MS*n*B:MS*n*A = 0:1
- SC[CPWMS] = 0 and ELS*n*B:ELS*n*A = 0:0

This condition is known as Software Compare mode. This mode sets the CH*n*F flag and generates a channel *n* interrupt (if CH*n*IE = 1) when the TPM counter reaches the value defined in the C*n*V register; however, TPM does not modify or control the channel *n* output.

67.3.4.3 Edge-Aligned PWM (EPWM) mode

EPWM mode is enabled when SC[CPWMS] = 0 and MS*n*B:MS*n*A = 1:0. The EPWM period is determined by (MOD[MOD] + 0001h) and the pulse width (duty cycle) is determined by C*n*V[VAL].

The channel *n* match (TPM counter = C*n*V[VAL]) writes 1 to C*n*SC[CH*n*F] and generates a channel *n* interrupt (if C*n*SC[CH*n*IE] = 1) at the end of the pulse width (see Figure 402).

This type of PWM signal is called edge-aligned because the leading edges of all PWM signals are aligned with the beginning of the period, which is the same for all channels within TPM.

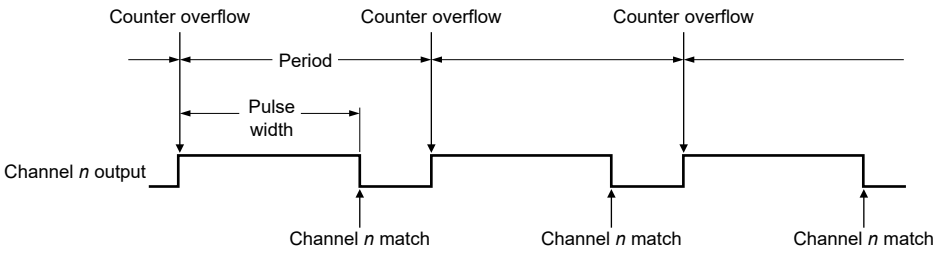


Figure 402. EPWM period and pulse width with ELS*n*B:ELS*n*A = 1:0

If ELS*n*B:ELS*n*A = 1:0, then the channel *n* output is forced high at the counter overflow or reload (when the TPM counter loads with 0), and it is forced low at the channel *n* match (TPM counter = C*n*V[VAL]). The channel *n* output is forced high when you first enable or whenever you pause the TPM counter (see Figure 403).

If ELS*n*B:ELS*n*A = 0:0, then the channel *n* output is forced high at the counter overflow or reload (when the TPM counter loads with 0), and it is forced low at the channel *n* match (TPM counter = C*n*V[VAL]). The channel *n* output is forced low when you first enable or whenever you pause the TPM counter.

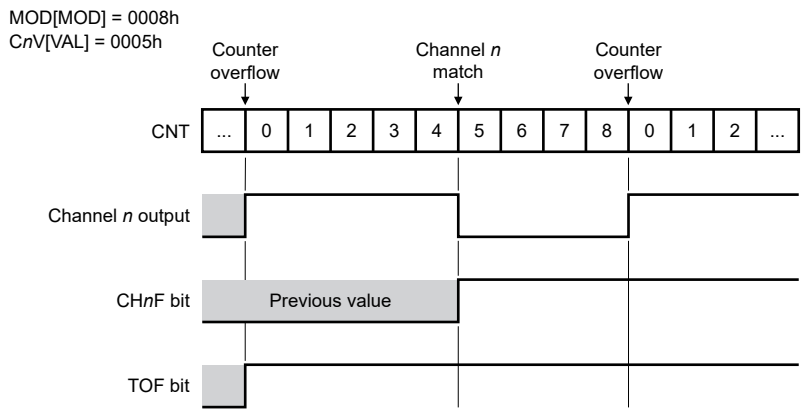


Figure 403. EPWM signal with ELS*n*B:ELS*n*A = 1:0

If ELS*n*B:ELS*n*A = 0:1, then the channel *n* output is forced low at the counter overflow or reload (when the TPM counter loads with 0), and it is forced high at channel *n* match (TPM counter = C*n*V[VAL]). Channel *n* output is forced low when you first enable or whenever you pause the TPM counter (see Figure 404).

If ELS*n*B:ELS*n*A = 1:1, then the channel *n* output is forced low at the counter overflow or reload (when the TPM counter loads with 0), and it is forced high at the channel *n* match (TPM counter = C*n*V[VAL]). Channel *n* output is forced high when you first enable or whenever you pause the TPM counter.

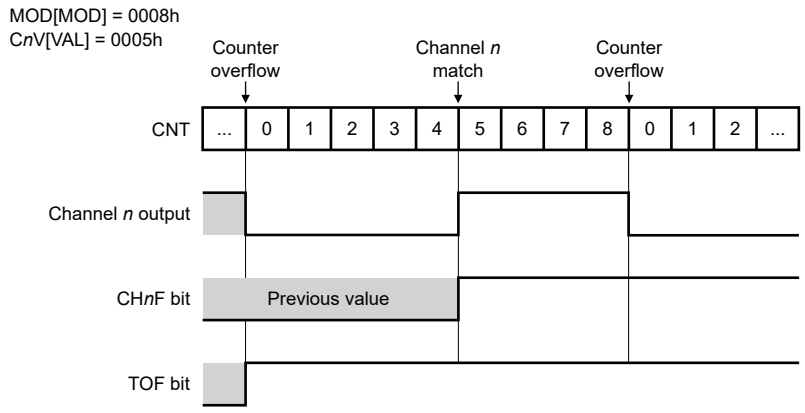


Figure 404. EPWM signal with ELS*n*B:ELS*n*A = 0:1

If C*n*V[VAL] = 0000h, then the channel *n* output is a 0% duty cycle EPWM signal.

If C*n*V[VAL] > MOD[MOD], then the channel *n* output is a 100% duty cycle EPWM signal and the CH*n*F flag is not set because there is never a channel *n* match. Therefore, MOD[MOD] must be less than FFFFFFFFh to get a 100% duty cycle EPWM signal.

67.3.4.4 Center-Aligned PWM (CPWM) mode

CPWM mode is enabled when (SC[CPWMS] = 1) and (MS*n*B:MS*n*A = 1:0).

The CPWM pulse width (duty cycle) is determined by 2 × C*n*V[VAL] and the period is determined by 2 × MOD[MOD] (see Figure 405). You must keep MOD[MOD] in the range of 0001h to 7FFFFFFF because values outside this range can produce ambiguous results.

In CPWM mode, the TPM counter counts up until it reaches the value defined in MOD[MOD], and then counts down until it reaches 0.

The channel *n* match (TPM counter = C*n*V[VAL]) sets C*n*SC[CH*n*F] and generates a channel *n* interrupt (if C*n*SC[CH*n*IE] = 1) when TPM is counting down (at the begin of the pulse width), and when TPM is counting up (at the end of the pulse width).

This type of PWM signal is called center-aligned because the pulse width centers for all channels are at the same place when the TPM counter is 0.

The other channel modes are not designed to be used with the up-down counter when SC[CPWMS] = 1. Therefore, you must use all TPM channels in CPWM mode when SC[CPWMS] = 1.

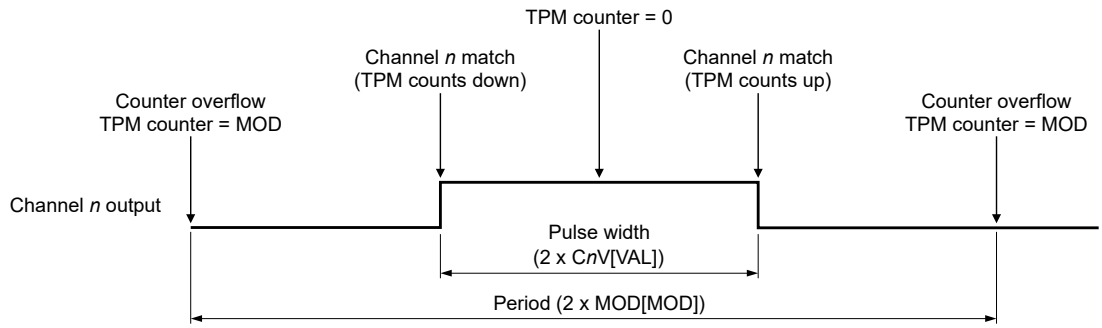


Figure 405. CPWM period and pulse width with ELS*n*B:ELS*n*A = 1:0

If (ELS*n*B:ELS*n*A = 1:0) then the channel *n* output is forced high at the channel *n* match (TPM counter = C*n*V) when counting down, and it is forced low at the channel *n* match when counting up. The channel *n* output is forced high when you first enable or reload the TPM counter, or whenever you pause it (see Figure 406).

If (ELS*n*B:ELS*n*A = 0:0) then the channel *n* output is forced high at the channel *n* match (TPM counter = C*n*V) when counting down, and it is forced low at the channel *n* match when counting up. The channel *n* output is forced low when you first enable or reload the TPM counter, or whenever you pause it.

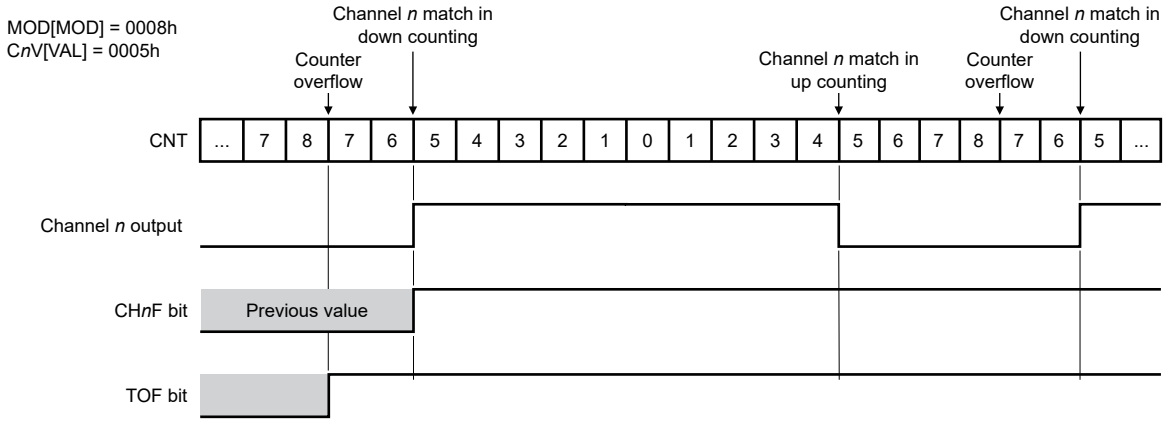


Figure 406. CPWM signal with ELS*n*B:ELS*n*A = 1:0

If (ELS*n*B:ELS*n*A = 0:1) then the channel *n* output is forced low at the channel *n* match (TPM counter = C*n*V) when counting down, and it is forced high at the channel *n* match when counting up. The channel *n* output is forced low when you first enable or reload the TPM counter, or whenever you pause it (see Figure 407).

If (ELS*n*B:ELS*n*A = 1:1) then the channel *n* output is forced low at the channel *n* match (TPM counter = C*n*V) when counting down, and it is forced high at the channel *n* match when counting up. The channel *n* output is forced high when you first enable or reload the TPM counter, or whenever you pause it.

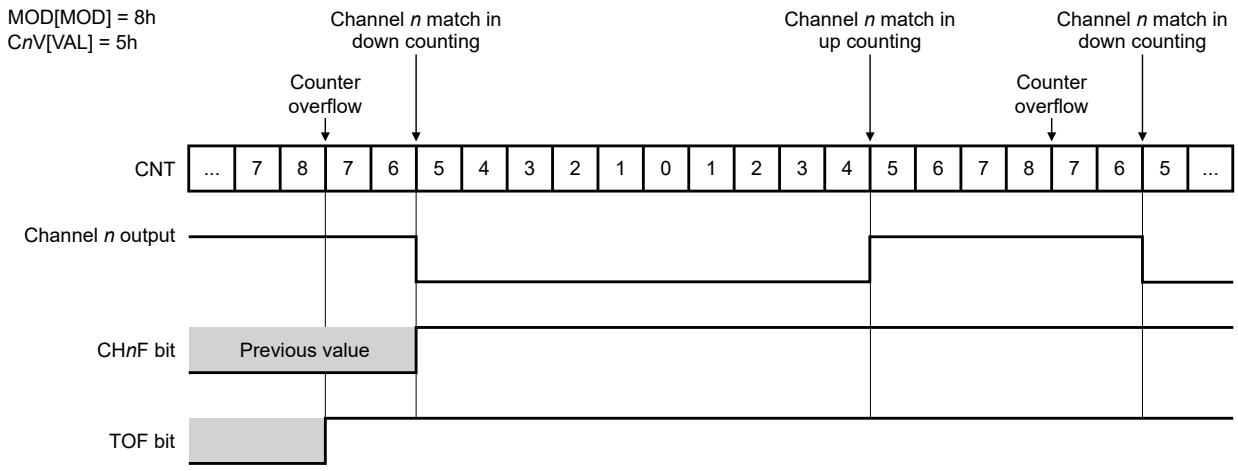


Figure 407. CPWM signal with ELS*n*B:ELS*n*A = 0:1

If (C*n*V[VAL] = 0000h) then the channel *n* output is a 0% duty cycle CPWM signal.

If (C*n*V[VAL] > MOD[MOD]) then the channel *n* output is a 100% duty cycle CPWM signal, although the CH*n*F flag is set when the counter changes from incrementing to decrementing. Therefore, MOD[MOD] must be less than 7FFFFFFF in order to get a 100% duty cycle CPWM signal.

67.3.4.5 Combine PWM mode

Combine PWM mode is enabled when:

- MS*n*B:MS*n*A = 1:0
- COMBINE[COMBINE*n*] = 1
- QDCTRL[QUADEN] = 0
- SC[CPWMS] = 0

In Combine PWM mode, an even channel *n* and adjacent odd channel *n* + 1 are combined to generate a PWM signal in the channel *n* output.

Determine the PWM period and pulse width according to Equation 49.

NOTE

Do not configure the second compare to a value that is lower than the first compare.

```
PWM period = MOD[MOD] + 0001h
PWM pulse width (duty cycle) = |C(n + 1)V[VAL] - CnV[VAL]|
```

Equation 49. PWM period and pulse width

- If (C*n*V[VAL] = C(*n* + 1)V[VAL] = 0000h), then the channel *n* output has a 0% duty cycle.
- If (C*n*V[VAL] = C(*n* + 1)V[VAL] > MOD[MOD]), then the channel *n* output has a 100% duty cycle. CH*n*F flag is not set because there is never a channel *n* match.

Therefore, MOD[MOD] must be less than FFFFh in order to get a 100% duty cycle signal.

The channel *n* match (TPM counter = C*n*V[VAL]) writes 1 to C*n*SC[CH*n*F] and generates a channel *n* interrupt (if C*n*SC[CH*n*IE] = 1). The channel *n* + 1 match (TPM counter = C(*n* + 1)V[VAL]), sets the CH(*n* + 1)F flag and generates a channel *n* + 1 interrupt, if CH(*n* + 1)IE = 1.

If channel *n* (ELS*n*B:ELS*n*A = 0:1), then the channel *n* output is forced low at the beginning of the period (TPM counter = 0) and at the channel *n* + 1 match (TPM counter = C(*n* + 1)V[VAL]). It is forced high at the channel *n* match (TPM counter = C*n*V[VAL]).

If channel *n* (ELS*n*B:ELS*n*A = 1:0), then the channel *n* output is forced high at the beginning of the period (TPM counter = 0) and at the channel *n* + 1 match (TPM counter = C(*n* + 1)V[VAL]). It is forced low at the channel *n* match (TPM counter = C*n*V). See Figure 408 for pictorial representation.

When COMBINE[COMSWAP*n*] = 1, then the channel *n* output is forced low or high at the beginning of the period (TPM counter = 0) and at the channel *n* match (TPM counter = C*n*V[VAL]). It is forced high or low at the channel *n* + 1 match (TPM counter = C(*n* + 1)V[VAL]).

Channel *n* + 1 generates the same output as the channel *n* output, but the channel *n* + 1 (ELS*n*B:ELS*n*A) configuration controls the output polarity.

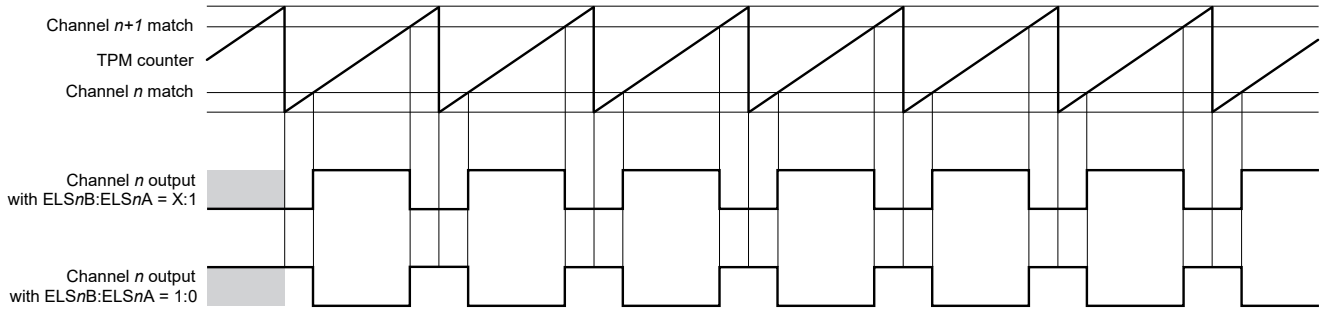


Figure 408. Combine mode

The following figures illustrate generation of PWM signals using the Combine mode.

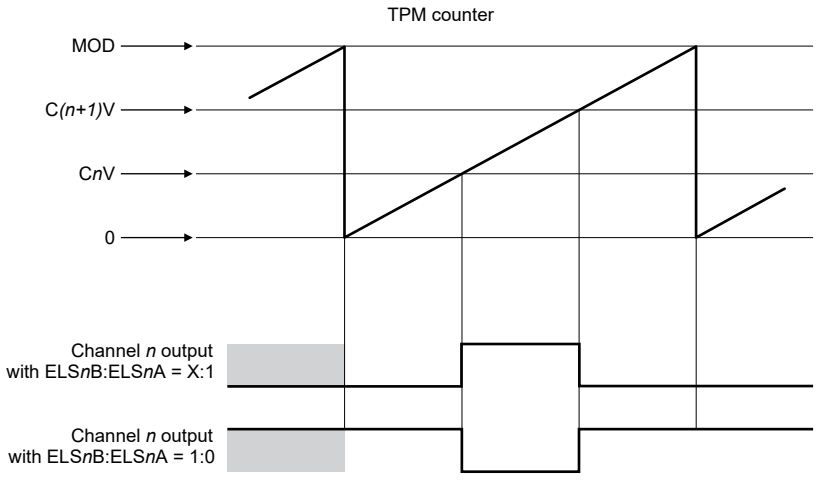


Figure 409. Channel *n* output if (C*n*V < MOD) and (C(*n* + 1)V < MOD) and (C*n*V < C(*n* + 1)V)

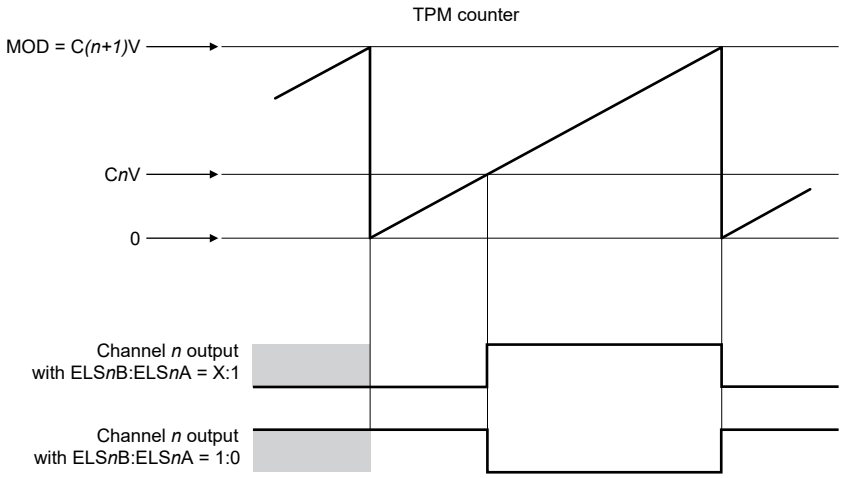


Figure 410. Channel *n* output if (C*n*V < MOD) and (C(*n* + 1)V = MOD)

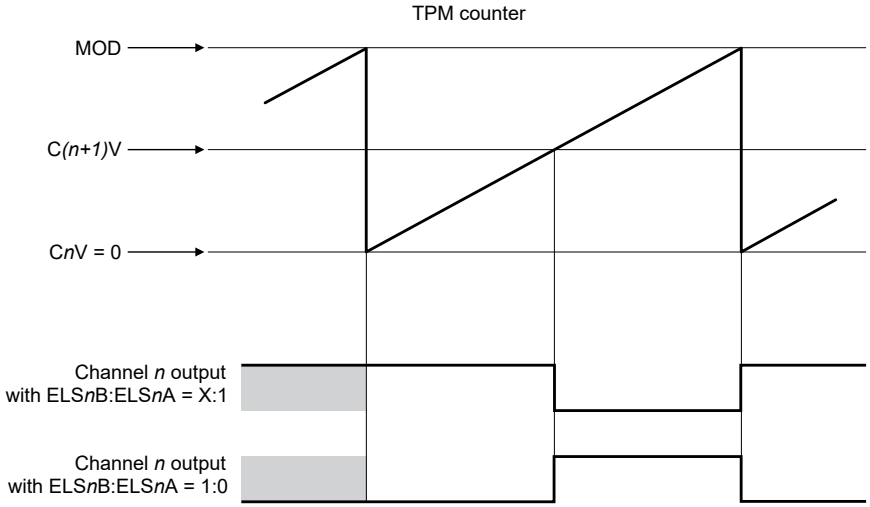


Figure 411. Channel *n* output if (C*n*V = 0) and (C(*n* + 1)V < MOD)

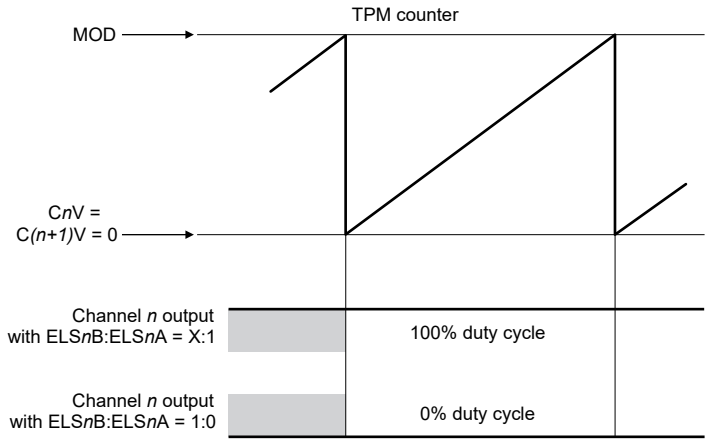


Figure 412. Channel (n) output if (C*n*V = C(*n* + 1)V = 0)

67.3.4.5.1 Dead time insertion

Dead time insertion is enabled in PWM Combine mode for each pair of channels when CH*n*FVAL and CH(*n* + 1)FVAL are non-zero. The dead time delay that is used for each TPM channel is defined as (CH(*n* + 1)FVAL[3:0] × 4) + (CH*n*FVAL[3:0] × 4).

The dead time delay insertion ensures that two complementary signals (even channel *n* and odd channel *n* + 1) do not drive the active state at the same time.

If ELS*n*B:ELS*n*A = 0:1, ELS(*n* + 1)B:ELS(*n* + 1)A = 1:0, COMBINE[COMSWAP*n*] = 0 and dead time is enabled, then:

- When a channel *n* match (TPM counter = C*n*V) occurs, the channel *n* output remains at the low value until the end of the dead time delay, when the channel *n* output is set.
- When a channel *n* + 1 match (TPM counter = C(*n* + 1)V) occurs, the channel *n* + 1 output remains at the low value until the end of the dead time delay, when the channel *n* + 1 output is set (see Figure 413).

If ELS*n*B:ELS*n*A = 1:0, ELS(*n*+1)B:ELS(*n*+1)A = 0:1, COMBINE[COMSWAP*n*] = 1, and dead time is enabled, then:

- When a channel *n* + 1 match (TPM counter = C(*n* + 1)V) occurs, the channel (*n* + 1) output remains at the low value until the end of the dead time delay, when channel *n* + 1 output is set.
- When a channel *n* match (TPM counter = C*n*V) occurs, the channel *n* output remains at the low value until the end of the dead time delay, when channel *n* output is set (see Figure 414).

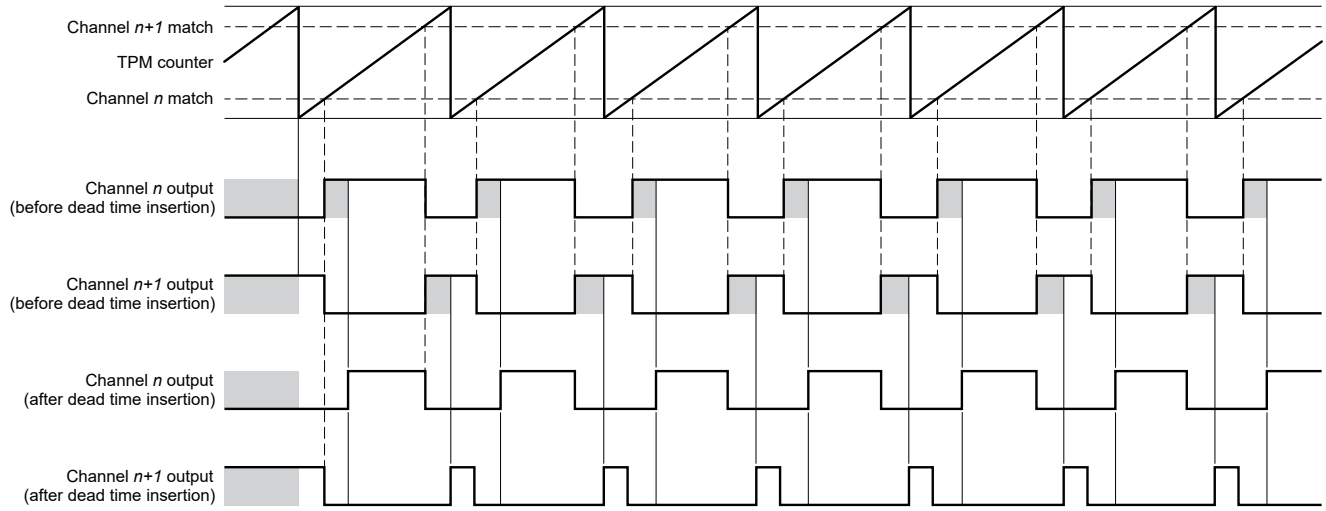


Figure 413. Dead time insertion with ELS*n*B:ELS*n*A = 0:1, ELS(*n* + 1)B:ELS(*n* + 1)A = 1:0, COMBINE[COMSWAP*n*] = 0

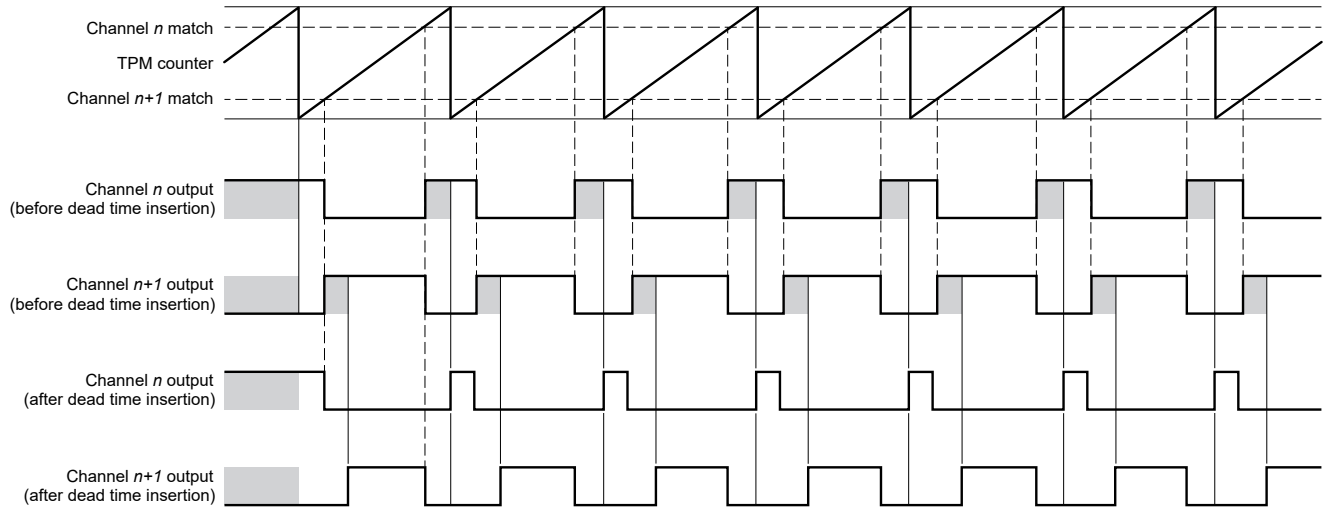


Figure 414. Dead time insertion with ELS*n*B:ELS*n*A = 1:0, ELS(*n* + 1)B:ELS(*n* + 1)A = 0:1, COMBINE[COMSWAP*n*] = 1

67.3.4.6 Combine Input Capture mode

Combine Input Capture mode is enabled if:

- (COMBINE[COMBINE*n*] = 1)
- (MS*n*B:MS*n*A = 0:0)
- (ELS*n*B:ELS*n*A ≠ 0:0)

This mode allows you to measure the pulse width of the signal at the input of channel *n* of a channel pair. The channel *n* filter can be active in this mode.

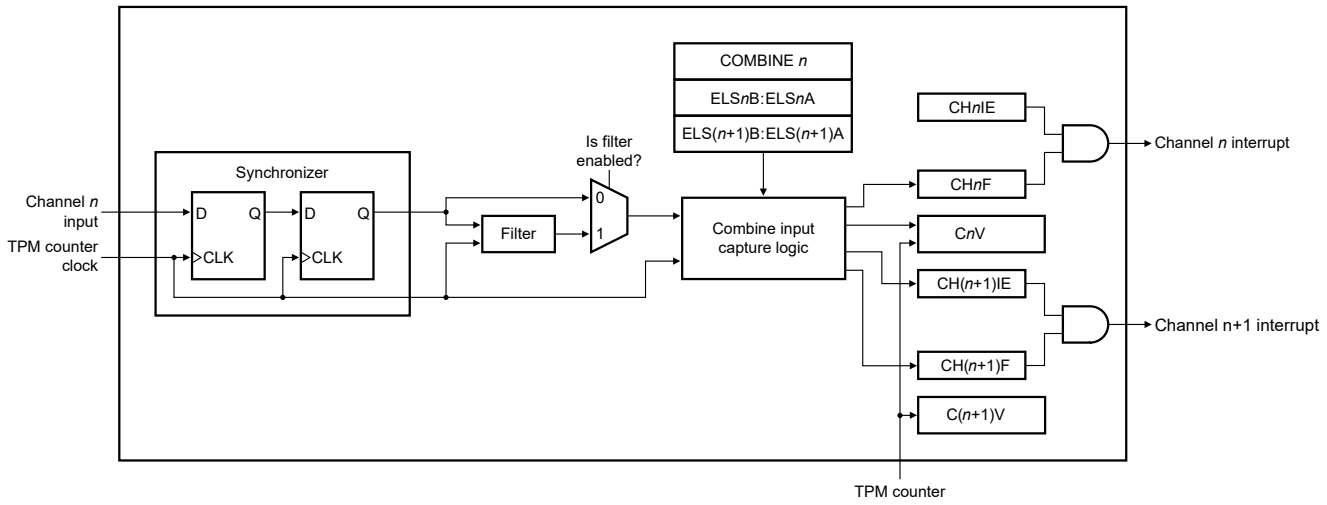


Figure 415. Block diagram

ELS*n*B:ELS*n*A fields select the edge that channel *n* captures. ELS(*n* + 1)B:ELS(*n* + 1)A fields select the edge that channel *n* + 1 captures.

This mode uses only the channel *n* input and ignores the channel *n* + 1 input. If COMBINE[COMSWAP*n*] = 1, then it uses only the channel *n* + 1 input and ignores the channel *n* input.

Detection of the edge selected by channel *n* fields at channel *n* input writes 1 to C*n*SC[CH*n*F] and generates a channel *n* interrupt (if C*n*SC[CH*n*IE] = 1). Detection of the edge selected by channel *n* + 1 fields at channel *n* input sets the CH(*n* + 1)F flag and generates the channel *n* + 1 interrupt (if CH(*n* + 1)IE = 1).

The C*n*V register stores the value of the TPM counter when the channel *n* input detects the edge selected by channel *n*. The C(*n* + 1)V register stores the value of the TPM counter when a channel *n* input detects the edge selected by channel *n* + 1.

NOTE

- Channel *n* fields are:
  - CH*n*F
  - CH*n*IE
  - MS*n*A
  - ELS*n*B
  - ELS*n*A
- Channel *n* + 1 fields are:
  - CH(*n* + 1)F
  - CH(*n* + 1)IE
  - MS(*n* + 1)A
  - ELS(*n* + 1)B
  - ELS(*n* + 1)A
- Use the Combine Input Capture mode with (ELS*n*B:ELS*n*A = 0:1 or 1:0) and (ELS(*n* + 1)B:ELS(*n* + 1)A = 0:1 or 1:0).

67.3.4.7 Quadrature Decoder mode

Quadrature Decoder mode is enabled if (QDCTRL[QUADEN] = 1). This mode uses channel 0 (phase A) and channel 1 (phase B) input signals to control the TPM counter increment and decrement (see Figure 416).

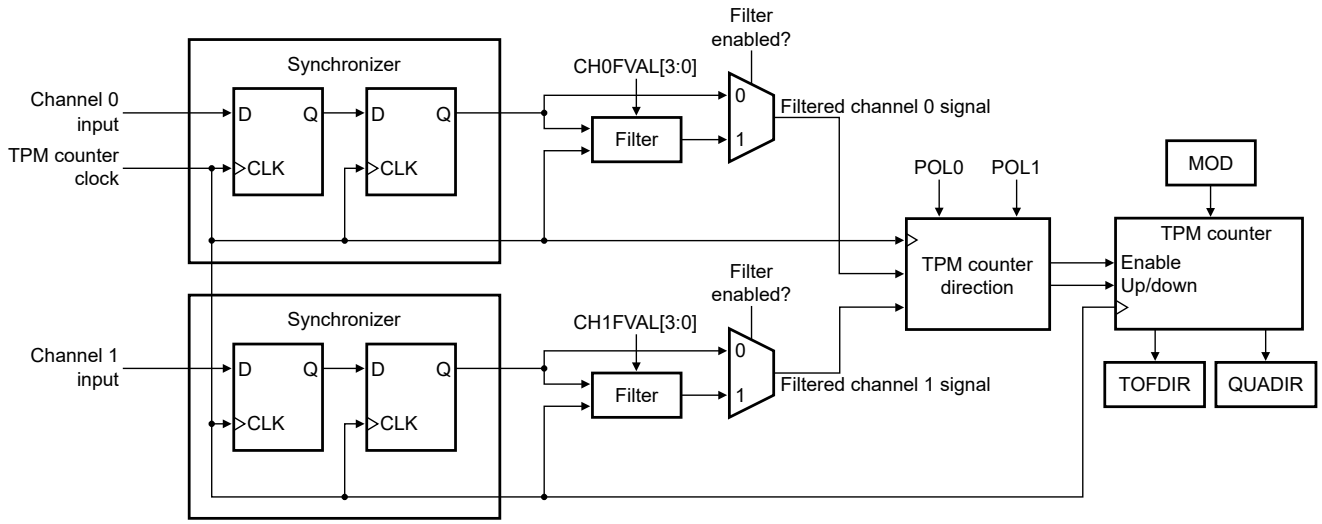


Figure 416. Block diagram

You can configure the input filter and polarity for channel 0 and channel 1 inputs in Quadrature Decode mode using Filter Control (FILTER) and Channel Polarity (POL).

NOTE

When Quadrature Decoder mode is enabled, then channel 0 and channel 1 input signals clock the TPM counter. Therefore, in this mode, you can only use Software Compare mode for channel 0 and channel 1, and you can only use Input Capture or Output Compare mode for other TPM channels.

QDCTRL[QUADMODE] defines the encoding mode that you use in Quadrature Decoder mode.

- If QDCTRL[QUADMODE] = 1, then count and direction encoding mode is enabled.
- If QDCTRL[QUADMODE] = 0, then phase encoding mode is enabled.

In count and direction encoding mode, the channel 1 input value indicates the counting direction, and the channel 0 input value defines the counting rate. A rising edge at the channel 0 input signal updates the TPM counter (see Figure 417).

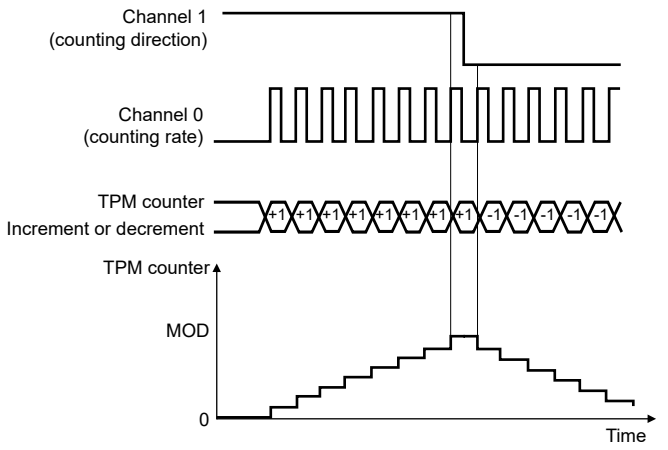


Figure 417. Quadrature Decoder—count and direction encoding mode

In phase encoding mode, the relationship between channel 0 and channel 1 signals indicates the counting direction, and channel 0 and channel 1 signals define the counting rate. An edge at either channel 0 or channel 1 signals updates the TPM counter (see Figure 418).

If POL[POL0] = 0 and POL[POL1] = 0, then:

- The TPM counter increment happens at:
  - A rising edge at channel 0 signal, and channel 1 signal is at logic 0
  - A rising edge at channel 1 signal, and channel 0 signal is at logic 1
  - A falling edge at channel 1 signal, and channel 0 signal is at logic 0
  - A falling edge at channel 0 signal, and channel 1 signal is at logic 1
- The TPM counter decrement happens at:
  - A falling edge at channel 0 signal, and channel 1 signal is at logic 0
  - A falling edge at channel 1 signal, and channel 0 signal is at logic 1
  - A rising edge at channel 1 signal, and channel 0 signal is at logic 0
  - A rising edge at channel 0 signal, and channel 1 signal is at logic 1

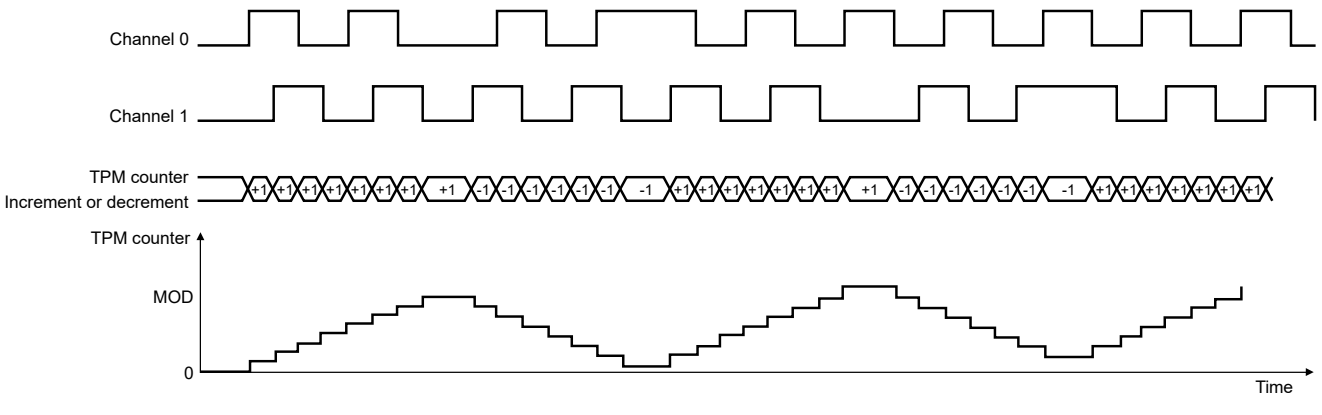


Figure 418. Quadrature Decoder—phase encoding mode

In Figure 419, when the TPM counter changes from the value defined in MOD[MOD] to 0, then SC[TOF] is 1 and QDCTRL[TOFDIR] becomes 1. SC[TOF] indicates that a TPM counter overflow has occurred. QDCTRL[TOFDIR] indicates that TPM was counting up when the TPM counter overflow occurred.

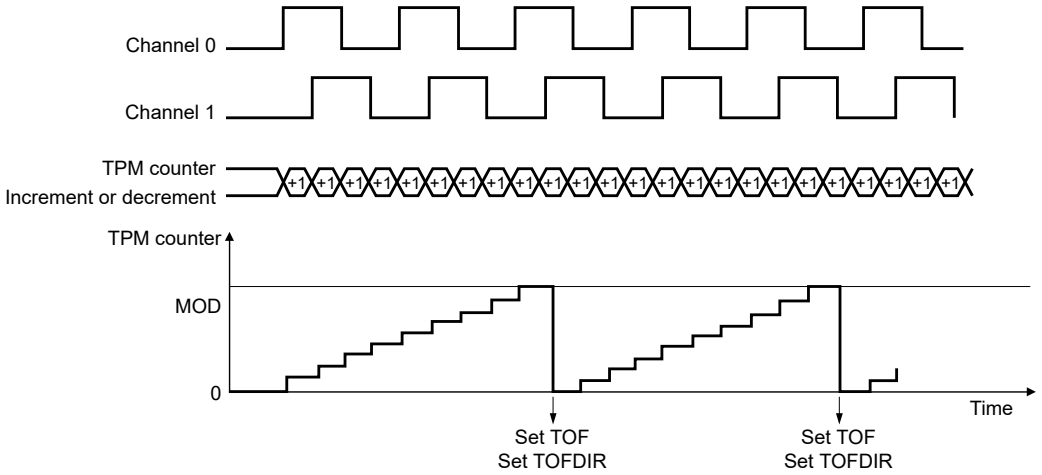


Figure 419. TPM counter overflow in up counting for Quadrature Decoder mode

In Figure 420, when the TPM counter changes from 0 to MOD[MOD], then SC[TOF] is 1 and QDCTRL[TOFDIR] becomes 0. SC[TOF] indicates that a TPM counter overflow has occurred. QDCTRL[TOFDIR] indicates that TPM was counting down when the TPM counter overflow occurred.

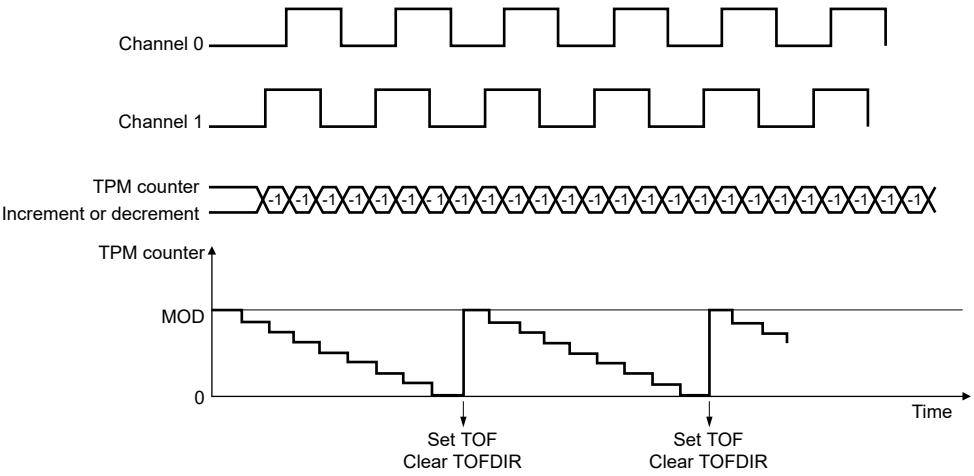


Figure 420. TPM counter overflow in down counting for Quadrature Decoder mode

67.3.5 Low-Power mode

During chip Low-Power mode, you can configure TPM to operate normally or to pause all counting.

Chip mode	TPM operation
Low-Power mode	• When CONF[DOZEEN] = 0, and TPM uses an external or internal clock source that remains functional, then counting continues. TPM can generate an asynchronous interrupt to exit the chip from Low-Power mode. • When CONF[DOZEEN] = 1, then counting pauses during Low-Power mode. Trigger inputs and input capture events are ignored, and PWM outputs are forced to their initial state.

67.3.6 Debug mode

During chip Debug mode, you can configure TPM to operate normally or to pause all counting temporarily.

Chip mode	TPM operation
Debug (core is in Debug or Halted mode)	• When CONF[DBGMODE] = 00b, then counting pauses until the core returns to normal user operating mode. Trigger inputs and input capture events are ignored, and PWM outputs are forced to their initial state. • When CONF[DBGMODE] = 11b, then counting continues.

67.3.7 Clocking

TPM supports two clock domains.

The bus clock domain is used by the register interface and for synchronizing interrupts and DMA requests.

The TPM counter clock domain clocks the counter and prescaler along with the output compare and input capture logic. The TPM counter clock is considered asynchronous to the bus clock. It can be a higher or lower frequency than the bus clock. It can optionally remain operational in low-power modes. When the global timebase feature is enabled, you must clock any TPM instance that is sharing the global timebase by the same TPM counter clock.

67.3.8 Reset

TPM resets whenever any chip reset occurs.

When TPM exits from reset:

- The TPM counter and the prescaler counter are 0 and are stopped (SC[CMOD] = 0).
- The timer overflow interrupt is 0.
- The channel interrupts are 0.
- The channels are in Input Capture mode.
- The channel outputs are 0.
- TPM does not control the channel pins (ELS*n*B:ELS*n*A = 0:0).

67.3.9 Interrupts

67.3.9.1 Timer overflow interrupt

TPM generates the timer overflow interrupt when (SC[TOIE] = 1) and (SC[TOF] = 1).

67.3.9.2 Channel *n* interrupt

TPM generates the channel *n* interrupt when (C*n*SC[CH*n*IE] = 1) and (C*n*SC[CH*n*F] = 1).

67.3.10 DMA

The channel and overflow flags generate a DMA transfer request according to SC[DMA] and C*n*SC[CH*n*IE]/SC[TOIE] fields. See the following table for more information.

Table 615. DMA transfer request

DMA	CH*n*IE/ TOIE	Channel and overflow DMA transfer request	Channel and overflow interrupt
0	0	Not generated	Not generated
0	1	Not generated	Generated if (CH*n*F/TOF = 1)
1	0	Generated if (CH*n*F/TOF = 1)	Not generated
1	1	Generated if (CH*n*F/TOF = 1)	Generated if (CH*n*F/TOF = 1)

If DMA = 1, the CH*n*F/TOF field can be cleared either by completion of the DMA transfer or by writing 1 to CH*n*F/TOF (see Table 616).

Table 616. Clear CH*n*F/TOF field

DMA	How to clear CH*n*F/TOF
0	CH*n*F/TOF field is cleared by writing 1 to it.
1	CH*n*F/TOF field is cleared either when the DMA transfer is completed or by writing 1 to it.

67.4 External signals

Table 617 shows the user-accessible signals for TPM.

Table 617. TPM signal descriptions

Signal	Description	Direction
EXTCLK	External clock If selected in SC[CMOD], EXTCLK increments the TPM counter on every rising edge synchronized to the TPM counter clock. EXTCLK must be less than half of the TPM counter clock frequency. The TPM counter prescaler selection and settings are also used when EXTCLK is selected.	I
CH*n*	Channel *n* I/O pin (where *n* = 3 to 0) You can configure each CH*n* to operate either as input or output. CH*n* is an output when configured in Output Compare or PWM mode and the TPM counter is enabled; otherwise, CH*n* is an input.	I/O

67.5 Initialization

To initialize TPM:

1. Write to the following registers, in any order, to configure the counter mode of operation:
   - Configuration (CONF)
   - Quadrature Decoder Control and Status (QDCTRL)
   - Modulo (MOD)
2. Write to the following registers, in any order, to configure the mode of operation for each channel:
   - Channel n Value (C0V - C3V)
   - Combine Channel (COMBINE)
   - Channel Trigger (TRIG)
   - Channel Polarity (POL)
   - Filter Control (FILTER)
3. Use Channel n Status and Control (C0SC - C3SC) to configure the mode of operation for each channel and to enable the channel.
4. Use Status and Control (SC) to configure the counter mode of operation and to enable the TPM (TPM is enabled and counting when SC[CMOD] ≠ 00).

67.6 Application information

To configure TPM counter to generate an overflow interrupt every 64,000 cycles:

1. Write any data to CNT to reset the counter.
2. Write MOD[MOD] = F9FFh to configure counter overflow.
3. Write SC = 48h to enable the counter and overflow interrupt.
4. Write STATUS = 100h to clear the flag during the interrupt routine.

To configure TPM channel 0 for input capture on either edge generating an interrupt:

1. Write C0SC = 4Ch to configure channel 0.
2. Optionally, write CONF = 840000h to configure counter to reset on each channel 0 input capture. Write CONF = 850000h to delay starting the counter until the first edge.
3. Write any data to CNT to reset the counter.
4. Write MOD[MOD] = FFFFFFFFh to configure the counter overflow to maximum.
5. Write SC = 48h to enable the counter and overflow interrupt.
6. During the interrupt routine, first check the setting for the STATUS flag:
   - If STATUS[TOF] is set, clear flag by writing STATUS = 100h. Note that software is responsible for tracking number of overflows between consecutive input capture events.
   - If STATUS[CH0F] is set, first read C0V to obtain the input capture value and then clear flag by writing STATUS = 0001h.

To configure TPM channel 0 for output compare to toggle the output pin using an array of counter values:

1. Write C0SC = 54h for interrupt or write C0SC = 15h for DMA.
2. Write C0V = array[0] to configure initial compare value.
3. Write any data to CNT to reset the counter.
4. Write MOD[MOD] = FFFFFFFFh to configure the counter overflow to maximum.
5. Write SC = 08h to enable the counter.
6. Write the next compare value C0V = array[N] during the interrupt routine and then write STATUS = 0001h to clear the flag.

7. Write the next compare value C0V = array[N] during the DMA transfer. DMA transfer will automatically clear the channel flag.

The array is an array of counter values where the first output compare occurs at “array[0]” cycles after the counter is enabled and subsequent output compares occur at “array[N] – array[N-1]” cycles after the last compare. If the output waveform repeats, then MOD can be set to an output compare value where the waveform starts to repeat.

An example of using TPM with the asynchronous DMA is described in AN4631:Using the Asynchronous DMA features of the Kinetis L Series.

67.7 Memory map and register definition

This section provides a detailed description of all TPM registers.

An attempt to access a reserved register location in the TPM memory map generates a bus error.

67.7.1 TPM register descriptions

67.7.1.1 TPM memory map

LPTPM1 base address: 4431_0000h

LPTPM2 base address: 4432_0000h

LPTPM3 base address: 424E_0000h

LPTPM4 base address: 424F_0000h

LPTPM5 base address: 4250_0000h

LPTPM6 base address: 4251_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VERID)	32	R	0600_0007
4	Parameter (PARAM)	32	R	0020_0404
8	TPM Global (GLOBAL)	32	RW	0000_0000
10	Status and Control (SC)	32	RW	0000_0000
14	Counter (CNT)	32	RW	0000_0000
18	Modulo (MOD)	32	RW	0000_FFFF
1C	Capture and Compare Status (STATUS)	32	RW	0000_0000
20	Channel n Status and Control (C0SC)	32	RW	0000_0000
24	Channel n Value (C0V)	32	RW	0000_0000
28	Channel n Status and Control (C1SC)	32	RW	0000_0000
2C	Channel n Value (C1V)	32	RW	0000_0000
30	Channel n Status and Control (C2SC)	32	RW	0000_0000
34	Channel n Value (C2V)	32	RW	0000_0000
38	Channel n Status and Control (C3SC)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
3C	Channel n Value (C3V)	32	RW	0000_0000
64	Combine Channel (COMBINE)	32	RW	0000_0000
6C	Channel Trigger (TRIG)	32	RW	0000_0000
70	Channel Polarity (POL)	32	RW	0000_0000
78	Filter Control (FILTER)	32	RW	0000_0000
80	Quadrature Decoder Control and Status (QDCTRL)	32	RW	0000_0000
84	Configuration (CONF)	32	RW	0000_0000

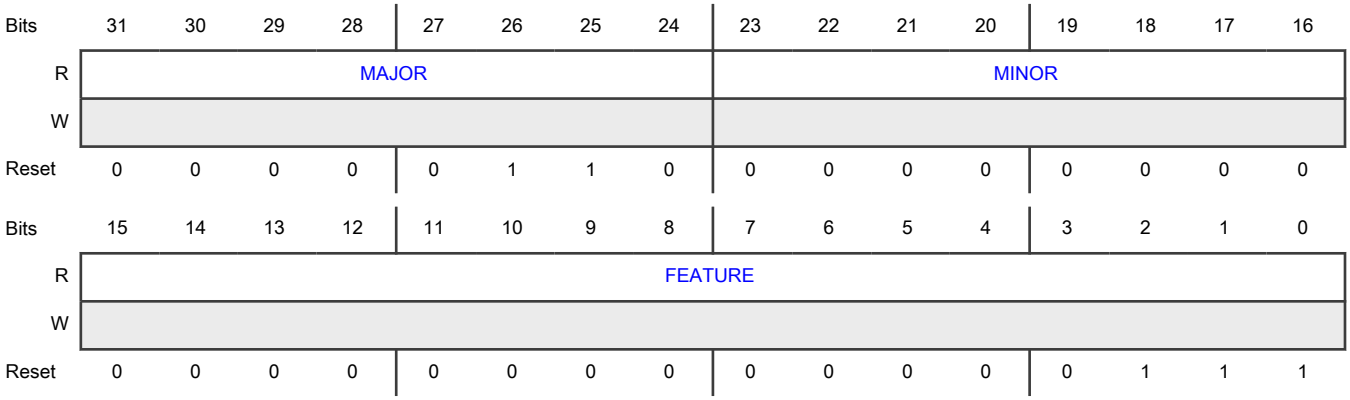
67.7.1.2 Version ID (VERID)

Contains the module version ID and feature information.

Offset

Register	Offset
VERID	0h

Diagram



Fields

Field	Description
31-24 MAJOR	Major Version Number Specifies the major version number for the module specification.
23-16	Minor Version Number

Table continues on the next page...

Table continued from the previous page...

Field	Description
MINOR	Specifies the minor version number for the module specification.
15-0 FEATURE	Feature Identification Number Configures different feature sets for TPM that implements a standard feature set with additional registers/ functions. 0000_0000_0000_0001 - Standard feature set 0000_0000_0000_0011 - Standard feature set with the filter and combine registers implemented 0000_0000_0000_0101 - Standard feature set with the quadrature register implemented 0000_0000_0000_0111 - Standard feature set with the filter, combine, and quadrature registers implemented

67.7.1.3 Parameter (PARAM)

Contains values of the module parameters.

Offset

Register	Offset
PARAM	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								WIDTH							
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TRIG								CHAN							
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

Fields

Field	Description
31-24 —	Reserved
23-16	Counter Width

Table continues on the next page...

Table continued from the previous page...

Field	Description
WIDTH	Specifies the width of the counter and timer channels.
15-8 TRIG	Trigger Count Specifies the number of trigger inputs that TPM implements.
7-0 CHAN	Channel Count Specifies the number of timer channels that TPM implements.

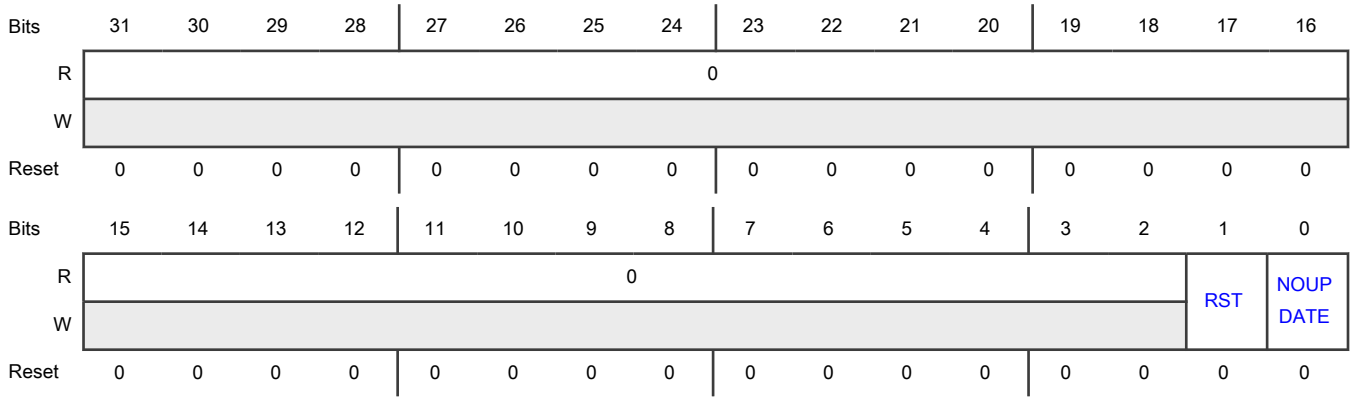
67.7.1.4 TPM Global (GLOBAL)

Includes the software reset control field.

Offset

Register	Offset
GLOBAL	8h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 RST	Software Reset Resets all internal logic and registers, except TPM Global (GLOBAL). This field remains 1 until you write 0 to it. 0 - Module is not reset

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Module is reset
0 NOUPDATE	No Update Blocks updates to all internal double-buffered registers while this field remains 1. You can still update the software visible registers, but changes cannot take effect until you write 0 to this field. At that point they update as normal. 0 - Internal double-buffered registers update as normal 1 - Internal double-buffered registers do not update

67.7.1.5 Status and Control (SC)

Contains the overflow status flag and control fields that you can use to configure:

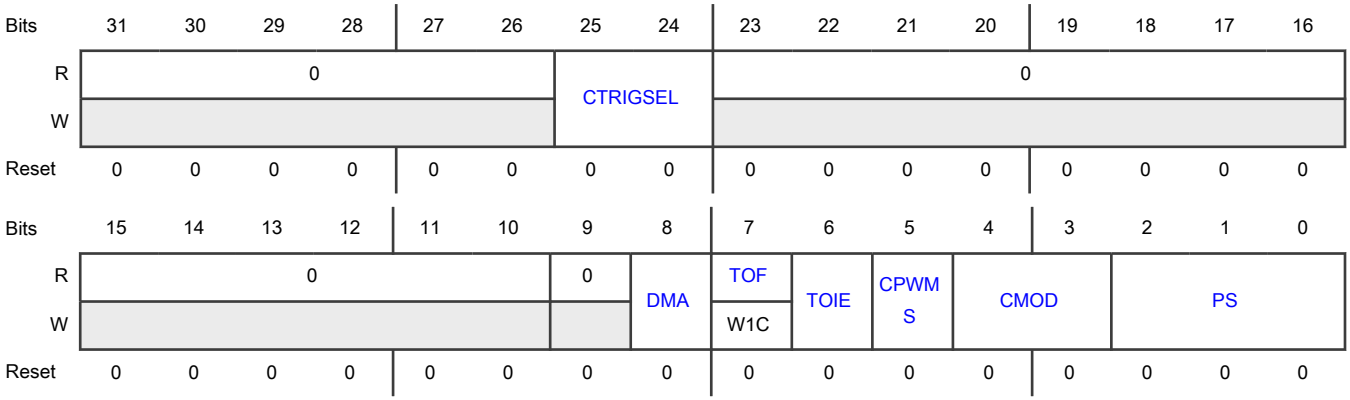
- Interrupt enable
- Module configuration
- Prescaler factor

These controls relate to all channels within this module.

Offset

Register	Offset
SC	10h

Diagram



Fields

Field	Description
31-26	Reserved

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>—</td><td></td></tr>
<tr><td>25-24
CTRIGSEL</td><td>Clock Trigger Select
External clock trigger select from trigger connections.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>LPTPM1</td><td>—</td><td>SC</td></tr><tr><td>LPTPM2</td><td>SC</td><td>—</td></tr><tr><td>LPTPM3</td><td>SC</td><td>—</td></tr><tr><td>LPTPM4</td><td>SC</td><td>—</td></tr><tr><td>LPTPM5</td><td>—</td><td>SC</td></tr><tr><td>LPTPM6</td><td>SC</td><td>—</td></tr></table></td></tr>
<tr><td>23-10
—</td><td>Reserved</td></tr>
<tr><td>9
—</td><td>Reserved</td></tr>
<tr><td>8
DMA</td><td>DMA Enable
Enables DMA transfers for the overflow flag.
0 - Disable
1 - Enable</td></tr>
<tr><td>7
TOF</td><td>Timer Overflow Flag
Indicates whether a TPM overflow has occurred. Hardware sets this flag when the TPM counter value is equal to the value defined in MOD[MOD], and increments after this value.
If another TPM overflow occurs between the flag setting and the flag clearing, then the write operation has no effect. Therefore, TOF remains set, which indicates that another overflow has occurred. In this case, a TOF interrupt request is not lost due to a delay in clearing the previous TOF.
0 - No overflow
1 - Overflow</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>6
TOIE</td><td>Timer Overflow Interrupt Enable
Enables TPM overflow interrupts. A TOF interrupt is generated when SC[TOF] = 1. If TOIE = 0, use software polling or DMA request.
0 - Disable
1 - Enable</td></tr>
<tr><td>5
CPWMS</td><td>Center-Aligned PWM Select
Enables CPWM mode. This mode configures TPM to operate in up-down counting mode.
This field is write-protected. Disable the TPM counter before you write to this field.
0 - Up counting mode
1 - Up-down counting mode</td></tr>
<tr><td>4-3
CMOD</td><td>Clock Mode Selection
Selects the TPM counter clock modes. When disabling the counter, this field remains 1 until acknowledged in the TPM clock domain.
NOTE
The descriptions of the field settings vary by module instance.
<table>
<tr><th>Instance</th><th>Field value and description</th></tr>
<tr><td>LPTPM1
LPTPM5</td><td>00 - TPM counter is disabled
01 - TPM counter increments on every TPM counter clock
10 - TPM counter increments on the rising edge of EXTCLK synchronized to the TPM counter clock
11 - TPM counter increments on the rising edge of the selected external input trigger</td></tr>
<tr><td>LPTPM2
LPTPM3
LPTPM4
LPTPM6</td><td>00 - TPM counter is disabled
01 - TPM counter increments on every TPM counter clock
10 - TPM counter increments on the rising edge of EXTCLK synchronized to the TPM counter clock
11 - TPM counter increments on both edges of EXTCLK synchronized to the TPM counter clock</td></tr>
</table></td></tr>
<tr><td>2-0
PS</td><td>Prescale Factor Selection
Selects one of 8 division factors for the clock mode selected by SC[CMOD].
This field is write-protected. Disable the TPM counter before you write to this field.
000 - Divide by 1</td></tr>
</table>

Table continues on the next page...

Table continued from the previous page...

Field	Description
	001 - Divide by 2 010 - Divide by 4 011 - Divide by 8 100 - Divide by 16 101 - Divide by 32 110 - Divide by 64 111 - Divide by 128

67.7.1.6 Counter (CNT)

Contains the TPM counter value. You can write 0 to this register with reset. Writing any value to CNT[COUNT] triggers a reload of the counter and forces PWM outputs to their reload state.

When debug is active, the TPM counter does not increment unless you configure it otherwise. Reading this register adds two wait states to the register access, because of synchronization delays.

Offset

Register	Offset
CNT	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 COUNT	Counter Value Specifies the TPM counter value.

67.7.1.7 Modulo (MOD)

Contains the modulo value for the TPM counter. When the TPM counter reaches the modulo value and increments, SC[TOF] sets and the next value of the TPM counter depends on the selected counting method (see Counter).

Writing to the MOD register latches the value into a buffer. This register updates with the value of its write buffer according to MOD register update. Additional writes to the MOD write buffer are ignored until the register has been updated.

To avoid confusion about occurrence of the first counter overflow, initialize the TPM counter (write to CNT[COUNT]) before writing to the MOD register.

Offset

Register	Offset
MOD	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MOD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MOD															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 MOD	Modulo Value Specifies the modulo value for the TPM counter. Write a single 16-bit or 32-bit access value to this field.

67.7.1.8 Capture and Compare Status (STATUS)

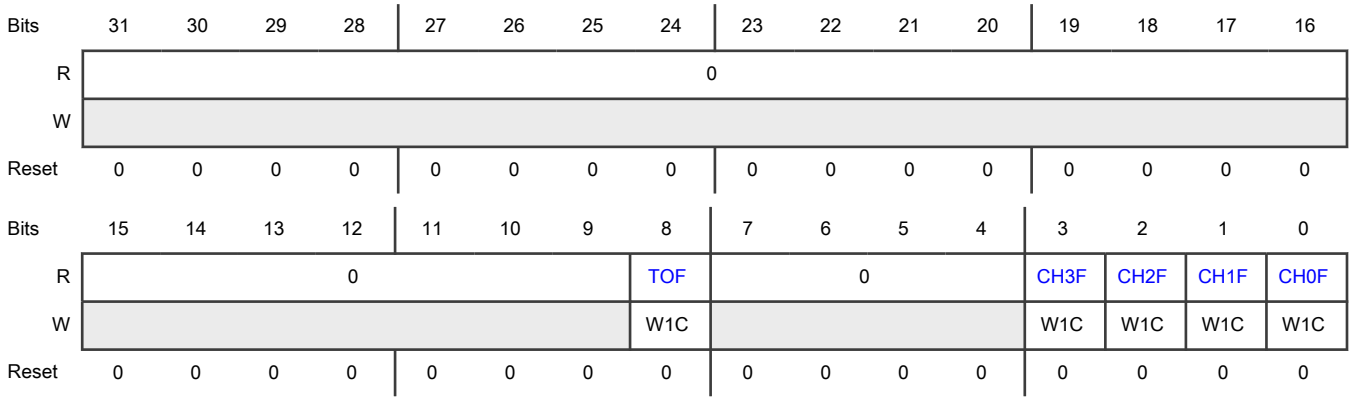
Contains a copy of the status flag (C*n*SC[CHF]) for each TPM channel, as well as SC[TOF], for software convenience. Each CH*n*F field in STATUS is a mirror of the corresponding C*n*SC[CHF]. You can check all C*n*SC[CHF] fields using only one read of STATUS. Writing all ones to STATUS writes 0 to all C*n*SC[CHF] fields.

Hardware sets the individual channel flags when an event occurs on the channel. If another event occurs between the flag setting and the write operation, the write operation has no effect; therefore, CHF remains set, which indicates that another event has occurred. In this case, a CHF interrupt request is not lost because of the clearing sequence for a previous CHF.

Offset

Register	Offset
STATUS	1Ch

Diagram



Fields

Field	Description
31-9 —	Reserved
8 TOF	Timer Overflow Flag Contains a copy of SC[TOF]. 0 - No overflow 1 - Overflow
7-4 —	Reserved
3 CH3F	Channel 3 Flag Contains a copy of C3SC[CHF]. Sets the flag when an event occurs on the channel (see the register description for more). 0 - Event not occurred 1 - Event occurred
2 CH2F	Channel 2 Flag Contains a copy of C2SC[CHF]. Sets the flag when an event occurs on the channel (see the register description for more). 0 - Event not occurred 1 - Event occurred
1 CH1F	Channel 1 Flag Contains a copy of C1SC[CHF]. Sets the flag when an event occurs on the channel (see the register description for more). 0 - Event not occurred

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Event occurred
0 CH0F	Channel 0 Flag Contains a copy of C0SC[CHF]. Sets the flag when an event occurs on the channel (see the register description for more). 0 - Event not occurred 1 - Event occurred

67.7.1.9 Channel n Status and Control (C0SC - C3SC)

Contains the channel-interrupt-status flag and control fields used to configure:

- Interrupt enable
- Channel configuration
- Pin function

Table 618. Mode, edge, and level selection

SC[CPWMS]	MS*n*B:MS*n*A	ELS*n*B:ELS*n*A	Mode	Configuration
0	00	00	None	Channel disabled
0	01	00	Software Compare	Pin not used for TPM
0	00	01	Input Capture	Capture on rising edge only
		10		Capture on falling edge only
		11		Capture on rising or falling edge
	01	01	Output Compare	Toggle output on match
		10		Clear output on match
		11		Set output on match
	10	10	Edge-Aligned PWM	High-true pulses (clear output on counter match, set output on counter reload, set output when counter first enabled or paused)
		00		High-true pulses (clear output on counter match, set output on counter reload, clear output when counter first enabled or paused)
		01		Low-true pulses (set output on counter match, clear output on counter reload, clear output when counter first enabled or paused)

Table continues on the next page...

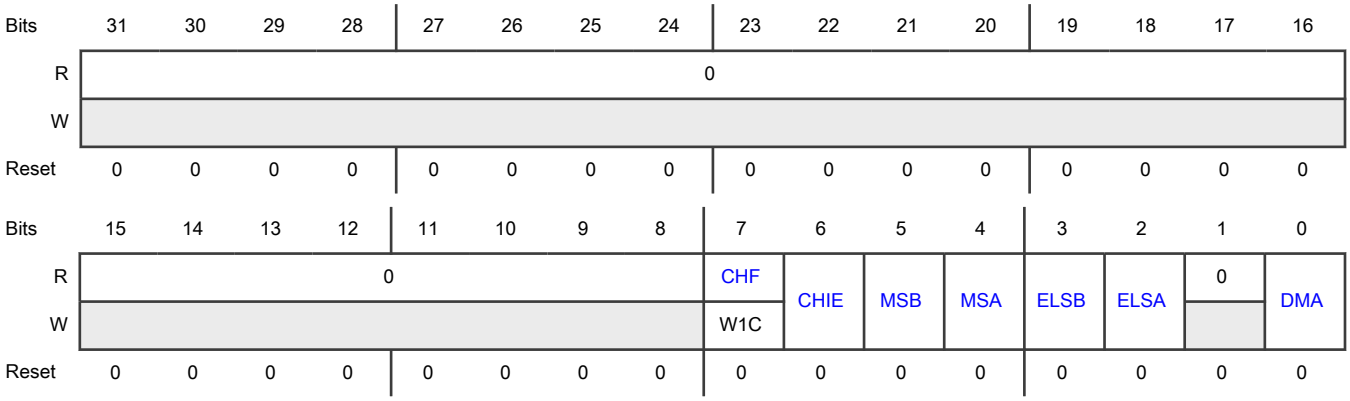
Table 618. Mode, edge, and level selection (continued)

SC[CPWMS]	MS*n*B:MS*n*A	ELS*n*B:ELS*n*A	Mode	Configuration
		11		Low-true pulses (set output on counter match, clear output on counter reload, set output when counter first enabled or paused)
	11	10	Output Compare	Pulse output low on match
		01		Pulse output high on match
1	10	10	Center-Aligned PWM	High-true pulses (clear output on counter match-up, set output on counter match-down, set output on counter reload, or when counter first enabled or paused)
		00		High-true pulses (clear output on counter match-up, set output on counter match-down, clear output on counter reload, or when counter first enabled or paused)
		01		Low-true pulses (set output on match-up, clear output on match-down, clear output on counter reload, or when counter first enabled or paused)
		11		Low-true pulses (set output on match-up, clear output on match-down, set output on counter reload, or when counter first enabled or paused)

Offset

Register	Offset
C0SC	20h
C1SC	28h
C2SC	30h
C3SC	38h

Diagram



Fields

Field	Description
31-8 —	Reserved
7 CHF	Channel Flag Indicates whether an event has occurred on the channel. TPM writes 1 to this field when an event occurs on the channel. If another event occurs between the CHF being set and the write operation, the write operation has no effect; therefore, CHF remains set, which indicates that another event has occurred. In this case, a CHF interrupt request is not lost due to the delay in clearing the previous CHF. 0 - Event not occurred 1 - Event occurred
6 CHIE	Channel Interrupt Enable Enables channel interrupts. 0 - Disable 1 - Enable
5 MSB	Channel Mode Select B Configures further selections in the channel logic. Its functionality depends on the channel mode (see Table 618).
4 MSA	Channel Mode Select A Configures further selections in the channel logic. Its functionality depends on the channel mode (see Table 618).
3 ELSB	Edge or Level Select B Performs the function that depends on the channel mode (see Table 618).
2	Edge or Level Select A

Table continues on the next page...

Table continued from the previous page...

Field	Description
ELSA	Performs the function that depends on the channel mode (see Table 618).
1 —	Reserved
0 DMA	DMA Enable Enables DMA transfers for the channel. 0 - Disable 1 - Enable

67.7.1.10 Channel n Value (C0V - C3V)

Contains the captured TPM counter value for the input modes or the match value for the output modes.

Input Capture mode ignores any write to a C*n*V register. In compare modes, writing to a C*n*V register latches the value into a buffer. A C*n*V register updates with the value of its write buffer according to C*n*V register update.

NOTE

Additional writes to the C*n*V write buffer are ignored until the register has been updated.

Offset

Register	Offset
C0V	24h
C1V	2Ch
C2V	34h
C3V	3Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 VAL	Channel Value Contains the captured TPM counter value for the input modes, or the match value for the output modes. Write a single 16-bit or 32-bit access value to this field.

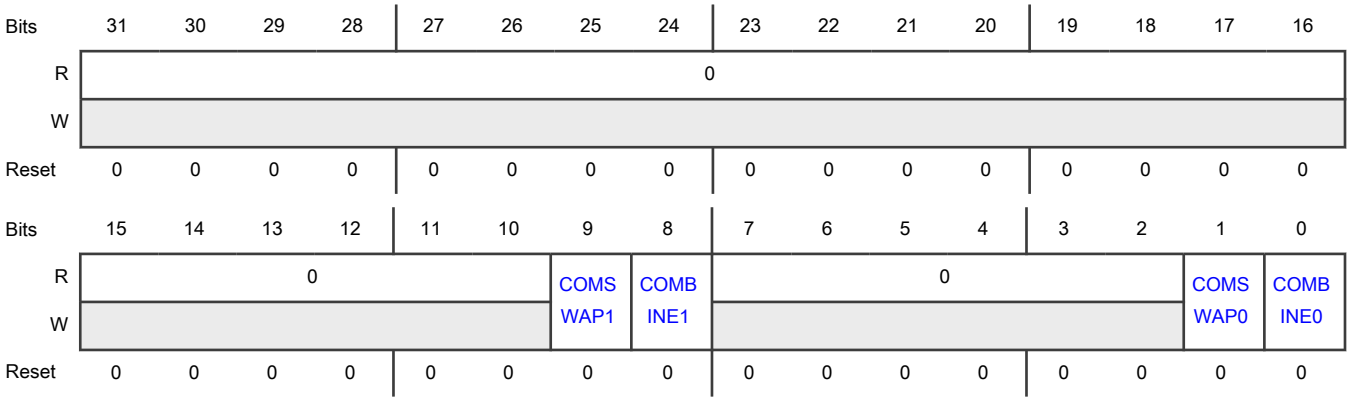
67.7.1.11 Combine Channel (COMBINE)

Contains the control fields used to configure the combine channel modes for each pair of channels *n* and *n*+1, where n is all the even-numbered channels.

Offset

Register	Offset
COMBINE	64h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-10 —	Reserved
9 COMSWAP1	Combine Channels 2 and 3 Swap When set in combine modes, the odd channel is used for the input capture and first compare, while the even channel is used for the second compare.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE The value of the first compare must not be greater than that of the second compare. 0 - Even channel 1 - Odd channel
8 COMBINE1	Combine Channels 2 and 3 Enables the combine feature for channels 2 and 3. In Input Capture mode, the combined channels use the even channel input. In Software Compare mode, the even channel match asserts the output trigger, and the odd channel match negates the output trigger. In PWM mode, the even channel match is used for the first compare, and odd channel match for the second compare. 0 - Independent 1 - Combined
7-2 —	Reserved
1 COMSWAP0	Combine Channel 0 and 1 Swap When set in combine modes, the odd channel is used for the input capture and first compare, while the even channel is used for the second compare. **NOTE** The value of the first compare must not be greater than that of the second compare. 0 - Even channel 1 - Odd channel
0 COMBINE0	Combine Channels 0 and 1 Enables the combine feature for channels 0 and 1. In Input Capture mode, the combined channels use the even channel input. In Software Compare mode, the even channel match asserts the output trigger, and the odd channel match negates the output trigger. In PWM mode, the even channel match is used for the first compare and odd channel match for the second compare. 0 - Independent 1 - Combined

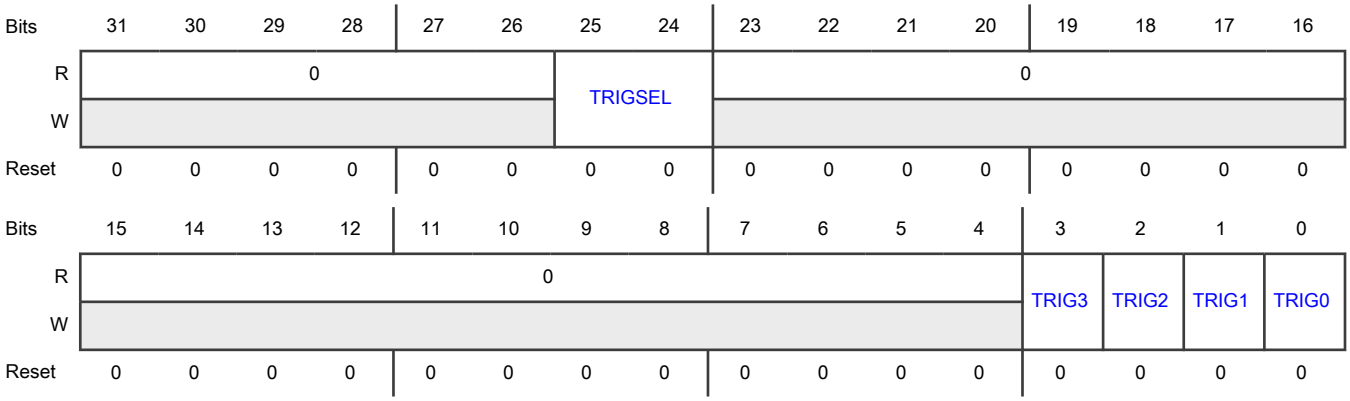
67.7.1.12 Channel Trigger (TRIG)

Configures the trigger input that is used by the channel to capture the counter value in Input Capture mode. In Output Compare or PWM mode, configures the trigger input used to modulate the channel output. When modulating the output, the output is forced to the channel initial value whenever the trigger is not asserted. The even-numbered channels share the first input trigger source, and the odd-numbered channels share the second input trigger source.

Offset

Register	Offset
TRIG	6Ch

Diagram



Fields

Field	Description
31-26 —	Reserved
25-24 TRIGSEL	Channel Trigger Select Channel trigger source select from trigger connections. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
LPTPM1	—	TRIG
LPTPM2	TRIG	—
LPTPM3	TRIG	—
LPTPM4	TRIG	—
LPTPM5	—	TRIG
LPTPM6	TRIG	—

23-4	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
3 TRIG3	Channel 3 Trigger 0 - No effect 1 - Configures trigger input 1 to be used by channel 3
2 TRIG2	Channel 2 Trigger 0 - No effect 1 - Configures trigger input 0 to be used by channel 2
1 TRIG1	Channel 1 Trigger 0 - No effect 1 - Configures trigger input 1 to be used by channel 1
0 TRIG0	Channel 0 Trigger 0 - No effect 1 - Configures trigger input 0 to be used by channel 0

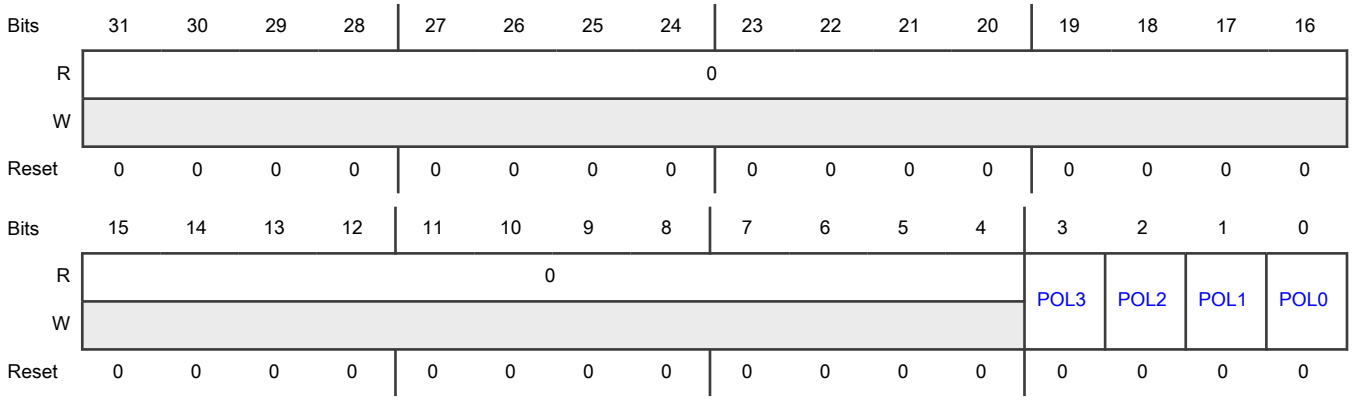
67.7.1.13 Channel Polarity (POL)

Defines the input and output polarity of each of the channels.

Offset

Register	Offset
POL	70h

Diagram



Fields

Field	Description
31-4 —	Reserved
3 POL3	Channel 3 Polarity 0 - Active high 1 - Active low
2 POL2	Channel 2 Polarity 0 - Active high 1 - Active low
1 POL1	Channel 1 Polarity 0 - Active high 1 - Active low
0 POL0	Channel 0 Polarity 0 - Active high 1 - Active low

67.7.1.14 Filter Control (FILTER)

Selects the filter value of the channel inputs, and an additional output delay value for the channel outputs. In PWM Combine mode, you can use this field to implement dead-time insertion.

NOTE

To write values to CH*n*FVAL, ensure that the corresponding channel-interrupt status flag and control (C*n*SC[5:2]) fields are set to 0.

Offset

Register	Offset
FILTER	78h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH3FVAL				CH2FVAL				CH1FVAL				CH0FVAL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-12 CH3FVAL	Channel 3 Filter Value When configured for Input Capture or Software Compare mode, configures the filter value for the channel 3 input. The filter is disabled when the value is zero. Otherwise, the filter is configured as (CH3FVAL×4)clock cycles. Sets the dead-time insertion time for channel 3 in PWM modes. Dead time insertion is disabled when CH3FVAL is zero. Otherwise, dead time insertion for channel 3 is configured as (CH3FVAL×4)clock cycles.
11-8 CH2FVAL	Channel 2 Filter Value When configured for Input Capture or Software Compare mode, configures the filter value for the channel 2 input. The filter is disabled when the value is zero. Otherwise, the filter is configured as (CH2FVAL×4)clock cycles. Sets the dead-time insertion time for channel 2 in PWM modes. Dead time insertion is disabled when CH2FVAL is zero. Otherwise, dead time insertion for channel 2 is configured as (CH2FVAL×4)clock cycles.
7-4 CH1FVAL	Channel 1 Filter Value When configured for Input Capture or Software Compare mode, configures the filter value for the channel 1 input. The filter is disabled when the value is zero. Otherwise, the filter is configured as (CH1FVAL×4)clock cycles. Sets the dead-time insertion time for channel 1 in PWM modes. Dead time insertion is disabled when CH1FVAL is zero. Otherwise, dead time insertion for channel 1 is configured as (CH1FVAL×4)clock cycles.
3-0 CH0FVAL	Channel 0 Filter Value When configured for Input Capture or Software Compare mode, configures the filter value for the channel 0 input. The filter is disabled when the value is zero. Otherwise, the filter is configured as (CH0FVAL×4)clock cycles. Sets the dead time insertion time for channel 0 in PWM modes. Dead time insertion is disabled when CH0FVAL is zero. Otherwise, dead time insertion for channel 0 is configured as (CH0FVAL×4)clock cycles.

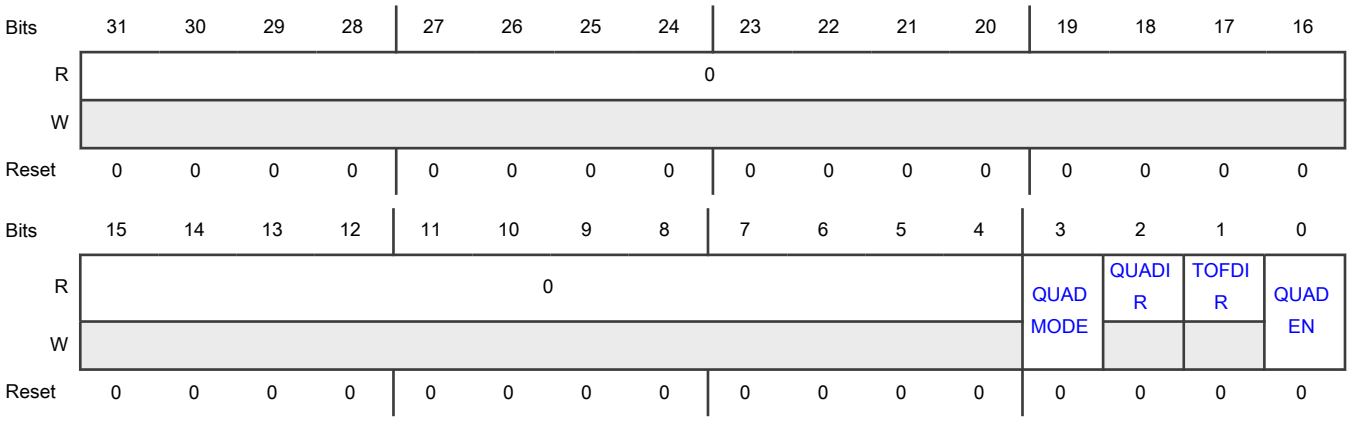
67.7.1.15 Quadrature Decoder Control and Status (QDCTRL)

Contains the control and status fields for Quadrature Decoder mode.

Offset

Register	Offset
QDCTRL	80h

Diagram



Fields

Field	Description
31-4 —	Reserved
3 QUADMODE	Quadrature Decoder Mode Selects the encoding mode used in Quadrature Decoder mode. 0 - Phase encoding mode 1 - Count and direction encoding mode
2 QUADIR	Counter Direction in Quadrature Decode Mode Indicates whether the counting direction is decreasing or increasing. 0 - Decreasing (counter decrement) 1 - Increasing (counter increment)
1 TOFDIR	Timer Overflow Direction Indicates if SC[TOF] was set on the top or bottom of counting. • Field value 0 indicates that SC[TOF] was set on the bottom of counting. There was a TPM counter decrement and TPM counter changes from its minimum value (0) to its maximum value (MOD[MOD]).

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Field value 1 indicates that SC[TOF] was set on the top of counting. There was a TPM counter increment and TPM counter changes from its maximum value (MOD[MOD]) to its minimum value (0). 0 - Bottom of counting 1 - Top of counting
0 QUADEN	Quadrature Decoder Enable Enables Quadrature Decoder mode. In this mode, the channel 0 and channel 1 inputs control the TPM counter direction and can only be used for Software Compare mode. Quadrature Decoder mode has precedence over the other modes. 0 - Disable 1 - Enable

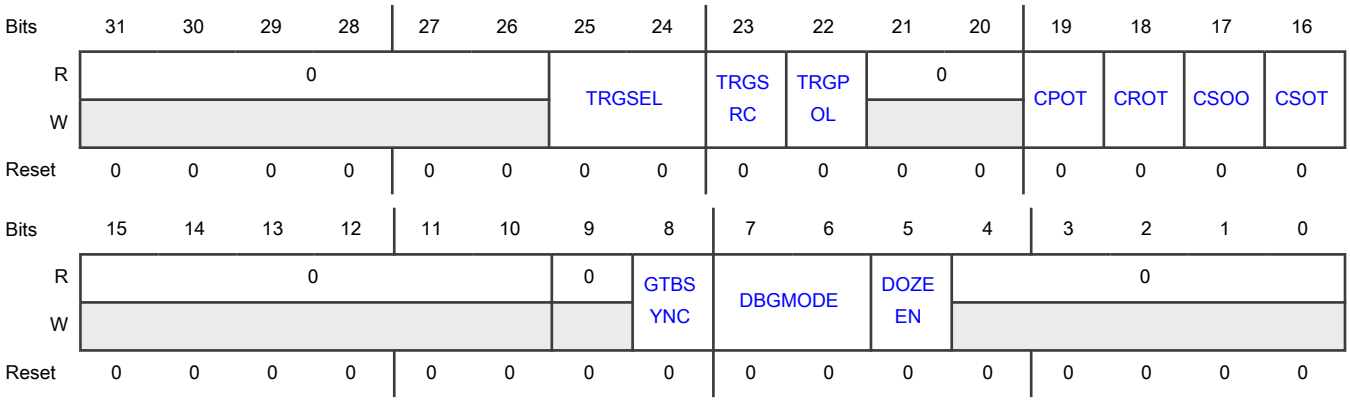
67.7.1.16 Configuration (CONF)

Selects the TPM behavior in Debug mode and Low-Power mode.

Offset

Register	Offset
CONF	84h

Diagram



Fields

Field	Description
31-26 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
25-24 TRGSEL	Trigger Select Selects the input trigger that you use to start, reload, and pause the counter. You can configure the source of the trigger (external or internal to TPM) by CONF[TRGSRC]. Disable the TPM counter before you change this field. See the chip-specific TPM information for available external trigger options. The below list shows available internal trigger sources. 01 - Channel 0 pin input capture 10 - Channel 1 pin input capture 11 - Channel 0 or channel 1 pin input capture
23 TRGSRC	Trigger Source Selects between internal (channel pin input capture) or external trigger sources. CONF[TRGSEL] selects this trigger source. When selecting an internal trigger, configure the selected channel for input captures as follows: • CONF[CSOT] enables you to start the counter initially using only a rising edge input capture. • CONF[CROT] enables you to reload the counter using either a rising edge or falling edge input capture. • CONF[CPOT] enables you to pause the counter using the state of the channel input pin. You can use Channel Polarity (POL) to invert the polarity of the channel input pins. Disable the TPM counter before you change this field. 0 - External 1 - Internal (channel pin input capture)
22 TRGPOL	Trigger Polarity Selects the polarity of the external trigger source. Disable the TPM counter before you change this field. 0 - Active high 1 - Active low
21-20 —	Reserved
19 CPOT	Counter Pause on Trigger Specifies that while the trigger remains asserted (level sensitive), and this field is 1, the counter pauses incrementing. While being paused, the counter ignores the input capture events and forces PWM outputs to their default state. If the TPM counter pauses during Debug mode or Low-Power mode, then the counter ignores the trigger input. Disable the TPM counter before you change this field. 0 - TPM counter continues 1 - TPM counter pauses

Table continues on the next page...

Table continued from the previous page...

Field	Description
18 CROT	Counter Reload on Trigger Specifies that when you detect a rising edge on the selected trigger input and this field is 1, then the TPM counter reloads with 0 (and sets PWM outputs to their reload state). If the TPM counter pauses during Debug mode or Low-Power mode, then the counter ignores the trigger input. Disable the TPM counter before you change this field. 0 - No reload 1 - Reload
17 CSOO	Counter Stop on Overflow Specifies that the TPM counter pauses when SC[TOF] = 1 and this field is 1. Reloading the counter with 0 by writing to Counter (CNT), or by a trigger input, does not cause the counter to stop incrementing. When stopped, the counter does not start incrementing unless you first disable and then enable it, or if you detect a rising edge on the selected trigger input when CONF[CSOT] is 1. While the counter is paused, it ignores the input capture events and forces PWM outputs to their default state. Disable the TPM counter before you change this field. 0 - TPM counter continues 1 - TPM counter stops
16 CSOT	Counter Start on Trigger Specifies that until you detect a rising edge on the selected trigger input, and this field is 1, the TPM counter pauses incrementing when it is enabled. If the TPM counter pauses due to an overflow, a rising edge on the selected trigger input also causes the TPM counter to start incrementing again. While being paused, the counter ignores the input capture events, and forces PWM outputs to their default state. If the TPM counter pauses during Debug mode or Low-Power mode, then the counter ignores the trigger input. Disable the TPM counter before you change this field. 0 - Counter starts immediately 1 - Counter starts after detection of a rising edge on the selected input trigger
15-10 —	Reserved
9 —	Reserved
8 GTBSYNC	GTB Synchronization Enables the TPM counter to synchronize with GTB. It uses the GTB enable, trigger, paused state and overflow to ensure that the TPM counter starts incrementing, stops incrementing, and resets at the same time as GTB. Disable the TPM counter before you change this field. **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Table continues on the next page...

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>LPTPM1</td><td>—</td><td>CONF</td></tr>
<tr><td>LPTPM2</td><td>CONF</td><td>—</td></tr>
<tr><td>LPTPM3</td><td>CONF</td><td>—</td></tr>
<tr><td>LPTPM4</td><td>CONF</td><td>—</td></tr>
<tr><td>LPTPM5</td><td>—</td><td>CONF</td></tr>
<tr><td>LPTPM6</td><td>CONF</td><td>—</td></tr>
</table>
0 - Disable
1 - Enable</td></tr>
<tr><td>7-6
DBGMODE</td><td>Debug Mode
Configures the TPM behavior in Debug mode (see Debug mode). All other configurations are reserved.
00 - TPM counter pauses
11 - TPM counter continues</td></tr>
<tr><td>5
DOZEEN</td><td>Doze Enable
Configures the TPM behavior in Low-Power mode (see Low-Power mode).
0 - TPM counter continues
1 - TPM counter pauses</td></tr>
<tr><td>4-0
—</td><td>Reserved</td></tr>
</table>

Chapter 68
Timer (TSTMR)

68.1 Chip-specific TSTMR information

Table 619. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing

The chip includes two satellite read-only Timestamp Timer Modules (TSTMR) (one in AONMIX and one in WAKEUPMIX). The TSTMR instances each provide a 64-bit read-only snapshot of the value of the free-running incrementing counter generated by the SCTR.

68.2 Overview

TSTMR is a free-running, 56-bit incrementing clock cycle counter that requires two 32-bit reads to read the full timestamp value.

TSTMR:

- Resets after every system reset.
- Is accessible through the LOW and HIGH registers.
- Stops only when the clock to TSTMR is disabled.

68.2.1 Block diagram

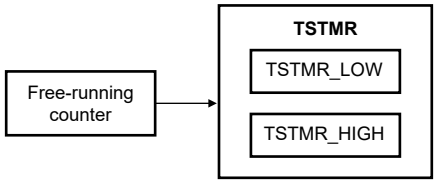


Figure 421. Block diagram

68.2.2 Features

TSTMR has the following features:

- Provides a free-running timer
- Counts the number of clock cycles since the last chip reset

68.3 Functional description

TSTMR starts running after system reset deassertion. You can read the counter at any time to determine software ticks. However, you must follow the read sequence as mentioned in the TSTMR register descriptions for correctly reading TSTMR values.

See the chip-specific TSTMR information for implementation details of this module's instances.

68.3.1 Dual mode

Dual Timer mode is used to implement a second virtual TSTMR for CPU2. The original TSTMR is assumed to be present in the CPU1 domain as a free running timer that resets only when the CPU1 domain power cycles. The CPU2 domain can power cycle independently of the CPU1 domain but it must do that while the CPU1 domain is power cycling. The second virtual TSTMR always runs in sync with the original TSTMR but starts from 0 when the CPU2 domain power cycles.

68.3.2 Clocking

This module has no clocking considerations. It uses the system bus clock to access the registers.

68.3.3 Interrupts

TSTMR has no interrupts.

68.4 External signals

TSTMR has no external signals.

68.5 Initialization

You can initialize TSTMR after the chip is powered up. No additional steps or considerations are necessary.

68.6 TSTMR_A memory map and register definition

NOTE

You can read TSTMR registers with 32-bit accesses only.

68.6.1 Accessing registers

You must always perform the following procedure in full to receive consistent timer values:

- Use a 32-bit access to read Timestamp Timer Low (LOW) register.
- Use a 32-bit access to read Timestamp Timer High (HIGH) register.
- A read to Timestamp Timer Low (LOW) register returns the low-order 32b, and samples the entire 64b.
- A read to Timestamp Timer High (HIGH) register returns the last sampled upper 32b.

68.6.2 TSTMR register descriptions

This section contains detailed register descriptions for the TSTMR registers.

68.6.2.1 TSTMR memory map

TSTMR1.TSTMRA base address: 442C_0000h

TSTMR2.TSTMRA base address: 4248_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Timestamp Timer Low (LOW)	32	R	0000_0000
4	Timestamp Timer High (HIGH)	32	R	0000_0000

68.6.2.2 Timestamp Timer Low (LOW)

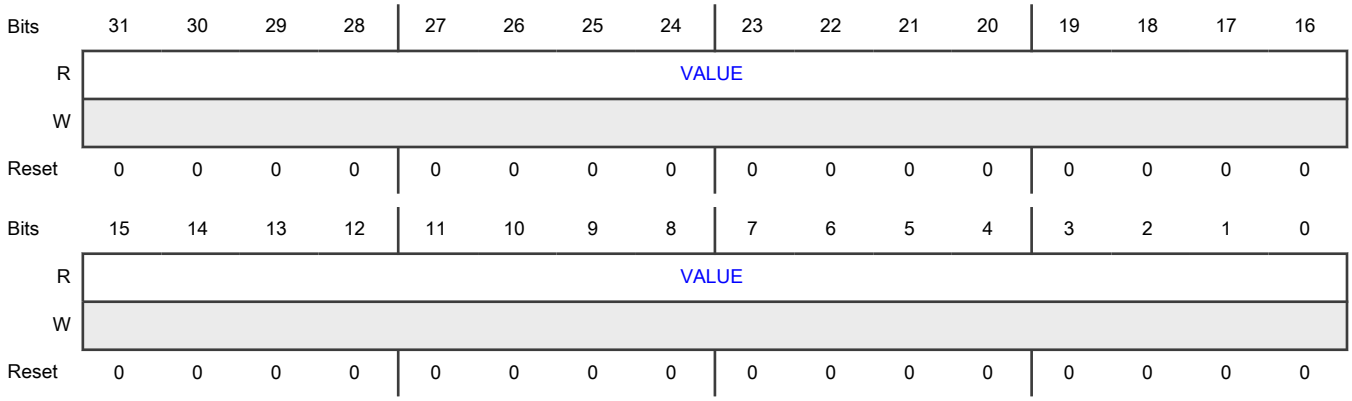
Indicates the lower 32 bits of the full 56-bit timestamp value.

See Accessing registers for information on how to use this register.

Offset

Register	Offset
LOW	0h

Diagram



Fields

Field	Description
31-0 VALUE	Timestamp Timer Low Indicates the lower 32 bits of the 56-bit timestamp value. You may not be able to detect the situation of VALUE = 0, because it increases before you can read it.

68.6.2.3 Timestamp Timer High (HIGH)

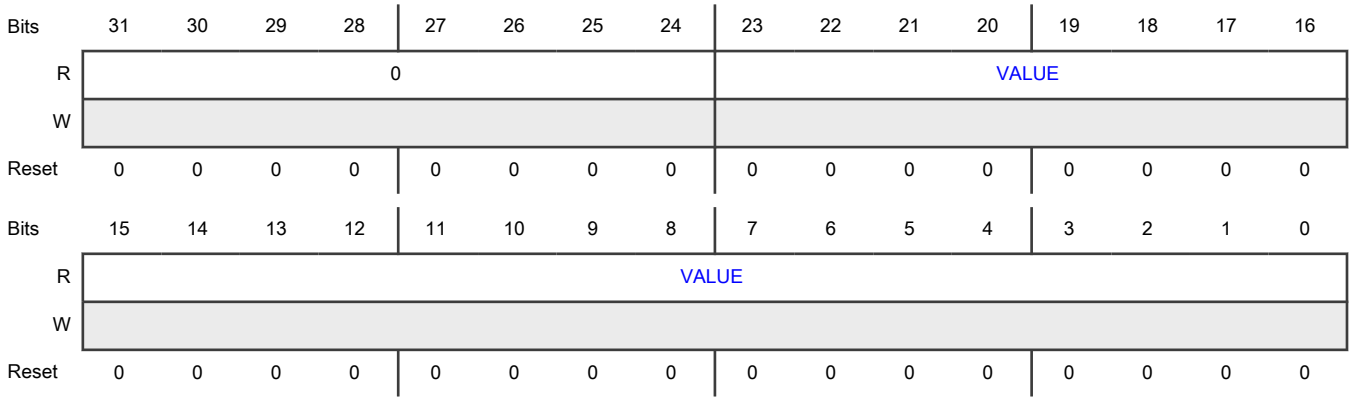
Indicates the higher 32 bits of the full 56-bit timestamp value.

See Accessing registers for information on how to use this register.

Offset

Register	Offset
HIGH	4h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-0 VALUE	Timestamp Timer High Indicates the higher 24 bits of the 56-bit timestamp value.

Chapter 69
Watchdog timer (WDOG)

69.1 Chip-specific WDOG information

Table 620. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
Full description	Arm Cortex CM33 Platform (CM33)	CM33 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

WDOG input clocks are connected as follows:

- Fixed 24MHz (ext_clk)
- Fixed 32KHz (lpo_clk)
- Fixed 133MHz from CCM (int_clk, ipg_clk_*, ipt_clk)

Table 621. Watchdog Control and Status Register CS[CLK]

Bitfield Value	Setting
2`b00	ipg_clk
2`b01	lpo_clk
2`b10	int_clk
2`b11	ext_clk

If Cortex-M33 is fuse-disabled, all 5 WDOG timers from a hardware perspective operate identically (except in low power modes where WAKEUPMIX is powergated). However, if Cortex-M33 is not fuse-disabled, then WDOG1 and WDOG2 are reset whenever the Cortex-M33 is reset. This is done to ensure that if WDOG1 and WDOG2 issue a reset to the Cortex-M33, that this reset will not only reset the Cortex-M33, but also the WDOG timers that may have been monitoring the Cortex-M33. This will prevent issues that may otherwise have been caused by a WDOG1 or WDOG2 reset permanently keeping the Cortex-M33 in reset.

69.1.1 WDOG interrupt connections

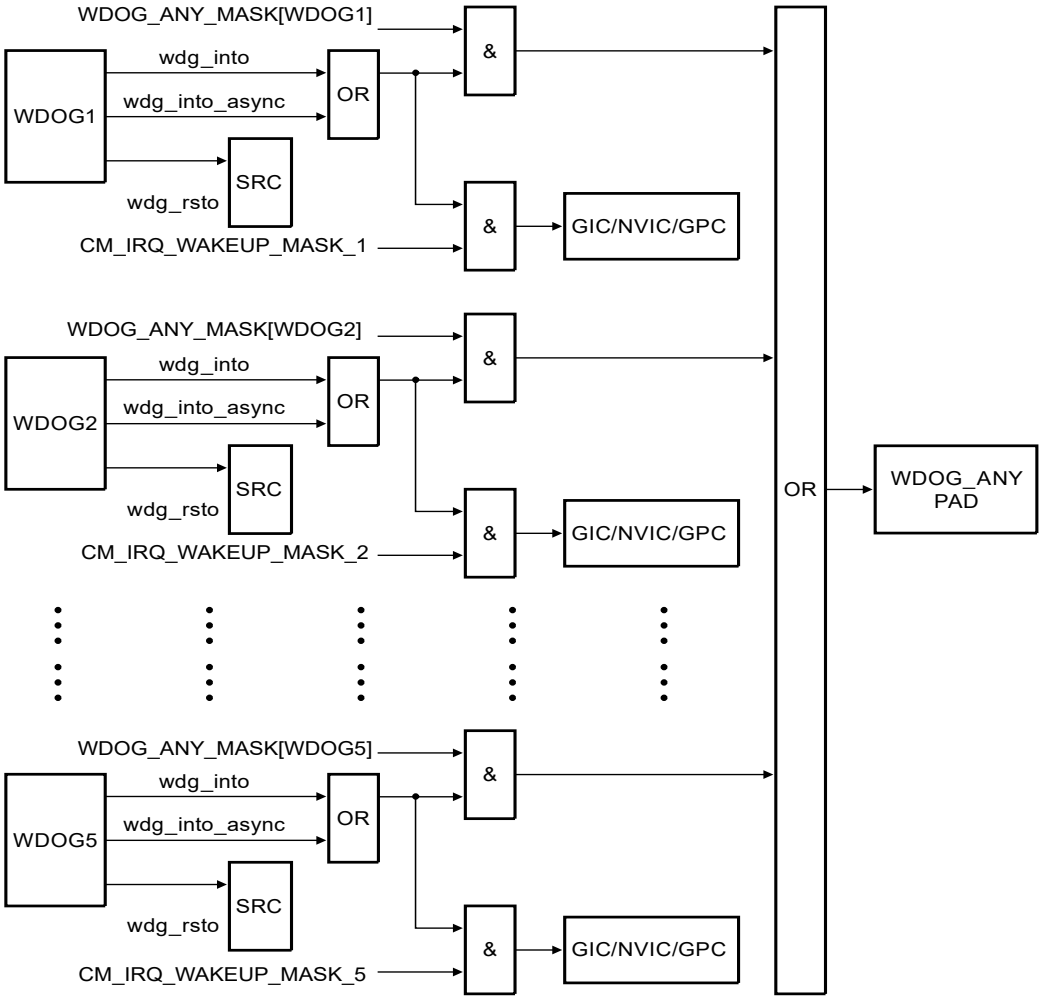


Figure 422. WDOG interrupt connections

69.2 Overview

WDOG is an independent timer that is available for system use. It provides a safety feature to ensure that the software is executing as planned and that the CPU is not stuck in an infinite loop or executing unintended code. If WDOG is not serviced (refreshed) within a certain period, it resets the chip.

69.2.1 Block diagram

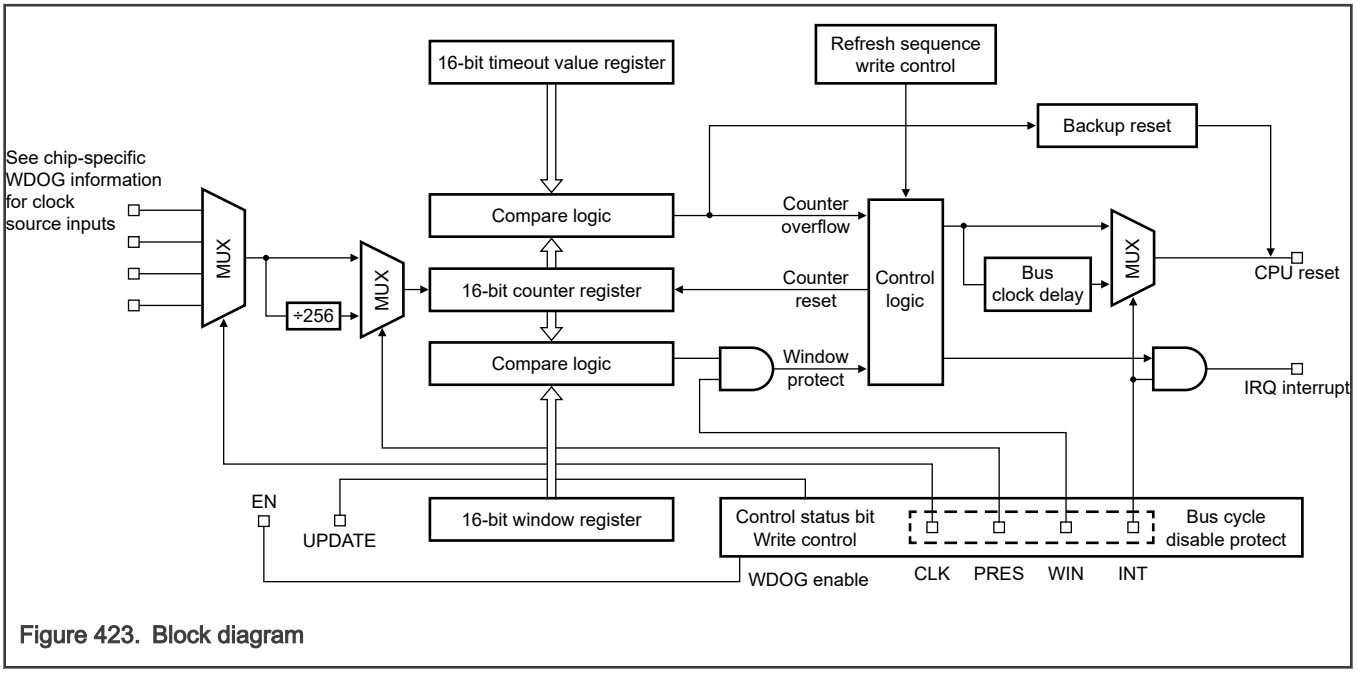


Figure 423. Block diagram

69.2.2 Features

- Configurable clock source inputs independent of the bus clock
- Programmable timeout period
  - Programmable 16-bit timeout value
  - Optional, fixed 256 clock prescaler when longer timeout periods are needed
- Robust write sequences for counter refresh: provision to refresh the sequence of writing to WDOG Counter (CNT)
- Window mode option for the refresh mechanism
  - Programmable 16-bit window value
  - Robust check to ensure that program flow is faster than expected
  - Early refresh attempts that trigger a reset
- Optional timeout interrupt to allow post-processing diagnostics
  - Interrupt request to CPU with an interrupt vector for an interrupt service routine (ISR)
  - Forced reset that occurs 128 bus clocks after the interrupt vector fetch
- Write-once-after-reset configuration fields to ensure that WDOG configuration is not altered mistakenly
- Robust write sequence for unlocking write-once configuration fields
  - Unlock sequence of writing to WDOG Counter (CNT), for allowing updates to write-once configuration fields
  - You must make updates within 1024 bus clocks after unlocking and before WDOG closing unlock window

69.3 Functional description

WDOG provides a fail-safe mechanism to ensure that you can reset the system to a known state of operation in case of system failure, such as the CPU clock stopping or there being a runaway condition in the software code. The WDOG counter runs continuously off a selectable clock source and expects to be serviced (refreshed) periodically. If it is not refreshed, it generates a reset triggering event.

69.3.1 Refresh mechanism

WDOG resets the chip if the WDOG counter is not refreshed. A robust refresh mechanism makes it very unlikely for a runaway code to refresh WDOG.

To refresh the WDOG counter, you must execute a refresh write sequence before the timeout period expires. In addition, in case of Window mode, you must not start the refresh sequence until you set the time value in Watchdog Window (WIN). See the following figure for more information.

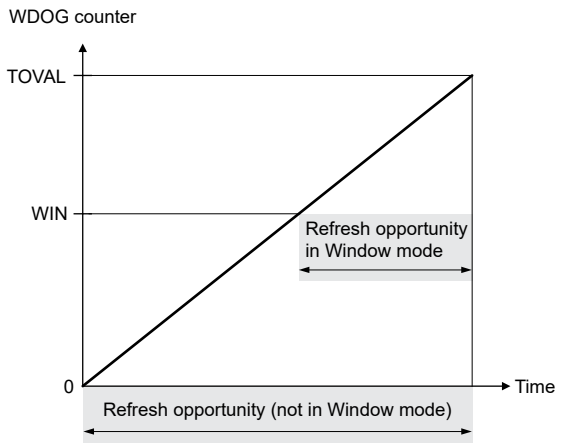


Figure 424. Refresh opportunity for the WDOG counter

69.3.1.1 Using Window mode

Software finishing its main control loop faster than expected could be an indication of a problem. Depending on the requirements of the application, you can program WDOG to force a reset when refresh attempts are early.

When Window mode is enabled, you must refresh WDOG after the counter has reached a minimum expected time value; otherwise, WDOG resets the chip. The minimum expected time value is specified in Watchdog Window (WIN). Setting CS[WIN] enables Window mode.

69.3.1.2 Refreshing WDOG

The refresh write sequence is based on the following methods:

- Either two 16-bit writes (A602h, B480h) or four 8-bit writes (A6h, 02h, B4h, 80h) to WDOG Counter (CNT) if CS[CMD32EN] = 0 [6]
- One 32-bit write (B480_A602h) to WDOG Counter (CNT) if CS[CMD32EN] = 1

You must apply these methods before WDOG times out; otherwise, it resets the chip.

[6] The example sequence of four 8-bit writes is as below:

— write A6h to WDOG_CNT[CNTHIGH], write 02h to WDOG_CNT[CNTLOW];

— write B4h to WDOG_CNT[CNTHIGH], write 80h to WDOG_CNT[CNTLOW].

Before starting the refresh sequence, disable the global interrupts. Otherwise, an interrupt could effectively invalidate the refresh sequence, if the interrupt occurs before the refresh writes finish. After the sequence finishes, restore the global interrupt control state.

See Application information for example code.

69.3.2 Configuring WDOG

69.3.2.1 Configuring WDOG once

All WDOG control bitfields, timeout value, and window value are write-once after reset. This means that after a write has occurred, they cannot be changed unless a reset occurs. You can ensure this by configuring the window and timeout values first, followed by the other control bitfields, when CS[UPDATE] = 0.

This provides a robust mechanism to configure WDOG and ensure that a runaway condition cannot mistakenly disable or modify the WDOG configuration, after the module is configured.

The new configuration takes effect only after you write to all registers except WDOG Counter (CNT) after reset. Otherwise, WDOG uses the reset values by default. If the Window mode is disabled (CS[WIN] is 0), you do not need to write to Watchdog Window (WIN) to bring the new configuration into effect.

69.3.2.2 Reconfiguring WDOG

In some cases (for example, when supporting a bootloader function), you may want to reconfigure or disable WDOG, without forcing a reset first:

- By writing 1 to CS[UPDATE] on the initial configuration of WDOG after a reset, you can reconfigure WDOG at any time by executing an unlock sequence.
- Conversely, if CS[UPDATE] remains 0, the only way to reconfigure WDOG is by initiating a reset.

The unlock sequence is similar to the refresh sequence but uses different values.

69.3.2.3 Unlocking WDOG

The unlock sequence is based on the following two methods:

- Either two 16-bit writes (C520h, D928h) or four 8-bit writes (C5h, 20h, D9h, 28h) to WDOG Counter (CNT), after WDOG is configured, if CS[CMD32EN] is 0 [7]
- One 32-bit write (D928_C520h) to WDOG Counter (CNT), after WDOG is configured, if CS[CMD32EN] is 1

An improper unlock sequence causes WDOG to reset. On completing the unlock sequence, you must reconfigure WDOG within 1024 bus clocks; otherwise, WDOG closes the unlock window.

NOTE

Because it requires 1024 bus clocks to reconfigure WDOG, you must insert some delays before executing STOP or WAIT instructions after reconfiguring WDOG. This ensures that WDOG's new configuration takes effect before the chip enters Low-Power mode. Otherwise, the chip may not wake up from Low-Power mode.

69.3.3 Functionality in Debug and Low-Power modes

By default, WDOG is not functional in Debug, Wait, or Stop modes. However, it can remain functional in these modes as follows:

- For Debug mode, write 1 to CS[DBG] (WDOG is functional in Debug mode even when Debug mode holds the CPU).
- For Wait mode, write 1 to CS[WAIT].

[7] The example sequence of four 8-bit writes is as below:

— write C5h to WDOG_CNT[CNTHIGH], write 20h to WDOG_CNT[CNTLOW];

— write D9h to WDOG_CNT[CNTHIGH], write 28h to WDOG_CNT[CNTLOW].

- For Stop mode, write 1 to CS[STOP] and CS[WAIT], and ensure that the clock source is active in Stop mode.

NOTE

WDOG can generate an interrupt in Stop mode.

Do not write 1 to CS[INT] in Stop mode. Otherwise, WDOG reset after a delay of 128 bus clocks (lose bus clock) will not occur, but backup reset will take effect.

For Debug and Stop modes, in addition to the aforementioned configuration, you must use a clock source other than the bus clock as the reference clock for the counter; otherwise, WDOG cannot function.

69.3.4 Fast testing of WDOG

Before executing the application code in safety-critical applications, you must test that WDOG works as expected and resets the chip. Testing every bit of a 16-bit counter by letting it run to the overflow value takes a relatively long time (64 k clocks).

To help minimize the startup delay for application code after reset, WDOG implements a feature that tests its functioning more quickly by splitting the counter into its constituent byte-wide stages. The low and high bytes are run independently and tested for a timeout against the corresponding byte of WDOG Timeout Value (TOVAL). For a complete coverage when testing the high byte of the counter, the test feature feeds the input clock via the 8th bit of the low byte, thus ensuring that the overflow connection from the low byte to the high byte is tested.

Using this test feature reduces the test time to 512 clocks (not including overhead, such as, user configuration and reset vector fetches). To further speed testing, use a faster clock (such as the bus clock) for the counter reference.

In case POR bitfield in the system reset register is set, indicating a power-on reset occurs, you must perform the WDOG fast test.

69.3.4.1 Testing each byte of the counter

For procedure of test, follow these steps:

1. Program the preferred WDOG timeout value in WDOG Timeout Value (TOVAL) during the WDOG configuration period.
2. Select a byte of the counter to test by using the configuration CS[TST] = 10b for the low byte, and CS[TST] = 11b for the high byte.
3. Wait for WDOG to timeout. Optionally, in the idle loop, increment RAM locations as a parallel software counter for later comparison. Because RAM is not affected by a WDOG reset, the timeout period of the WDOG counter can be compared with the software counter to verify whether the timeout period occurred as expected.

   The WDOG counter then times out and forces a reset.
4. Confirm that the WDOG flag in the system reset register is set, indicating that WDOG caused the reset (the POR flag remains clear).
5. Confirm that CS[TST] showing a test (10b or 11b) was performed.

If confirmed, the count and compare functions work for the selected byte. Repeat the procedure, and then select the next byte in step 2.

NOTE

Only a POR writes 0 to TST, which is not affected by other resets.

69.3.4.2 Entering User mode

After successfully testing the low and high bytes of the WDOG counter, you can configure TST to 01b to indicate that WDOG is ready for use in application User mode. Therefore, if a reset occurs again, you can recognize the reset trigger as a real WDOG reset caused by runaway or faulty application code.

As an ongoing test when using the default clock source, you can periodically read WDOG Counter (CNT) to ensure that the counter is being incremented.

69.3.5 Clocking

You can program CLK to select clock source options in the WDOG counter. See the chip-specific WDOG information for available clock inputs and the default option for this chip.

The option allows you to select a clock source that is independent of the bus clock for applications that need to meet more robust safety requirements. Using a clock source other than the bus clock ensures that the WDOG counter continues to run if the bus clock is somehow halted (see Backup reset).

For WDOG to function properly, you must enable the default WDOG clock source after its functional reset is deasserted.

An optional fixed prescaler for all clock sources allows longer timeout periods. When CS[PRES] is set to 1, the clock source is prescaled by 256 before clocking the WDOG counter.

The following table summarizes examples of the different available WDOG timeout periods. In the table, RCP means "reference clock period".

Table 622. WDOG timeout availability

Reference clock	Prescaler	WDOG timeout availability
REF_CLK	Pass through	1 × RCP to 65535 × RCP
	Enable	256 × RCP to 16776960 × RCP

NOTE

When you switch clock sources during reconfiguration, WDOG holds the counter at zero for 2.5 periods of the previous clock source and 2.5 periods of the new clock source after the configuration period (1024 bus clocks) ends. This delay ensures a smooth transition before restarting the counter with the new configuration.

69.3.6 Backup reset

The backup reset function is a safeguard feature that independently generates a reset in case the main WDOG logic loses its clock (the bus clock) and can no longer monitor the counter. If the WDOG counter overflows twice in succession (without an intervening reset), the backup reset function takes effect and generates a reset.

The backup reset becomes valid when an interrupt is enabled and the WDOG clock is not from a bus clock. If an interrupt is enabled, once the bus clock is cut off before exiting the interrupt routine, the normal watchdog reset is blocked. In this case, the second overflow causes a backup reset directly.

NOTE

You must use a clock source other than the bus clock as a reference clock for the counter; otherwise, the backup reset function becomes unavailable.

69.3.7 Interrupts

WDOG can generate an interrupt request to delay resets.

When interrupts are enabled (CS[INT] = 1), and after a reset-triggering event (such as a counter timeout or invalid refresh attempt), WDOG:

1. Generates an interrupt request.
2. Waits 128 bus clocks (from the interrupt vector fetch, not the reset-triggering event).
3. Forces a reset.

This process allows the ISR to perform tasks such as analyzing the stack to debug code.

When interrupts are disabled (CS[INT] = 0), WDOG does not wait before forcing a reset.

69.4 External signals

This module has no external signals.

69.5 Initialization

See Configuring WDOG once.

69.6 Application information

To disable or reconfigure WDOG, it should be completed before WDOG timeout. Disabling or reconfiguring WDOG should occur at the very beginning of the software code, for example, at the beginning of the startup or main function.

NOTE

- After you configure WDOG, it needs at least 2.5 periods of WDOG clock to take effect. This means a gap must have at least 2.5 clocks between two configurations.
- When the chip starts from boot ROM and then jumps to flash memory, you must disable WDOG at the beginning of the bootloader and enable it after the bootloader exits. To reconfigure WDOG using the flash memory program, it also needs the interval of at least 2.5 WDOG clocks after the bootloader exits.

To disable or reconfigure WDOG without forcing a reset, you must write 1 to CS[UPDATE] during the initial WDOG configuration. You can use the unlock sequence at any time, within the timeout limit, to reconfigure WDOG.

69.6.1 Disabling WDOG

To disable WDOG, you must first perform the unlock sequence. Then, write 0 to clear CS[EN]. The following code snippet shows an example of a 32-bit write.

```
DisableInterrupts; // disable global interrupt
WDOG_CNT = 0xD928C520; //unlock watchdog
WDOG_CS &= ~WDOG_CS_EN_MASK | WDOG_CS_UPDATE(1); //disable watchdog
EnableInterrupts; //enable global interrupt
```

69.6.2 Disabling WDOG after reset

All WDOG registers are unlocked by reset. Therefore, an unlock sequence is unnecessary but it needs to be written to all WDOG registers to make the new configuration take effect. The following code snippet shows an example of disabling WDOG after reset.

```
DisableInterrupts; // disable global interrupt
WDOG_CS &= ~WDOG_CS_EN_MASK; // disable watchdog
WDOG_TOVAL= 0xFFFF;
while(WDOG_CS[ULK]);  // waiting for lock
while(~WDOG_CS[RCS]); // waiting for new configuration to take effect
EnableInterrupts; // enable global interrupt
```

69.6.3 Configuring WDOG

WDOG can be configured once by clearing CS[UPDATE] = 0. After that, yWDOG cannot be reconfigured until a reset. To reconfigure without forcing a reset, write 1 to set CS[UPDATE] when configuring WDOG. The following example code shows how to configure WDOG without Window mode, clock source as LPO, interrupt enabled, and timeout value to 256 clocks. The following code snippet shows an example of a 32-bit write.

69.6.3.1 Configuring once

```
DisableInterrupts; // disable global interrupt
WDOG_CNT = 0xD928C520; //unlock watchdog
while(WDOG_CS[ULK]==0);  //wait until registers are unlocked
```

```
WDOG_TOVAL = 256; //set timeout value
WDOG_CS = WDOG_CS_EN(1) | WDOG_CS_CLK(1) | WDOG_CS_INT(1) |
          WDOG_CS_WIN(0) | WDOG_CS_UPDATE(0);
while(WDOG_CS[RCS]==0); //wait until new configuration takes effect
EnableInterrupts; //enable global interrupt
```

69.6.3.2 Configuring for Reconfigurable mode

```
DisableInterrupts; //disable global interrupt
WDOG_CNT = 0xD928C520; //unlock watchdog
while(WDOG_CS[ULK]==0);  //wait until registers are unlocked
WDOG_TOVAL = 256; //set timeout value
WDOG_CS = WDOG_CS_EN(1) | WDOG_CS_CLK(1) | WDOG_CS_INT(1) |
          WDOG_CS_WIN(0) | WDOG_CS_UPDATE(1);
while(WDOG_CS[RCS]==0); //wait until new configuration takes effect
EnableInterrupts; //enable global interrupt
```

69.6.4 Refreshing WDOG

To refresh WDOG and reset the WDOG counter to zero, a refresh sequence is required. The following code snippet shows an example of a 32-bit write.

```
DisableInterrupts; // disable global interrupt
WDOG_CNT = 0xB480A602; // refresh watchdog
EnableInterrupts; // enable global interrupt
```

69.7 Memory map and register definition

69.7.1 WDOG register descriptions

69.7.1.1 WDOG memory map

LPWDOG1 base address: 442D_0000h

LPWDOG2 base address: 442E_0000h

LPWDOG3 base address: 4249_0000h

LPWDOG4 base address: 424A_0000h

LPWDOG5 base address: 424B_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	WDOG Control and Status (CS)	32	RW	0000_2900
4	WDOG Counter (CNT)	32	RW	0000_0000
8	WDOG Timeout Value (TOVAL)	32	RW	0000_0400
C	Watchdog Window (WIN)	32	RW	0000_0000

69.7.1.2 WDOG Control and Status (CS)

Describes watchdog control and status.

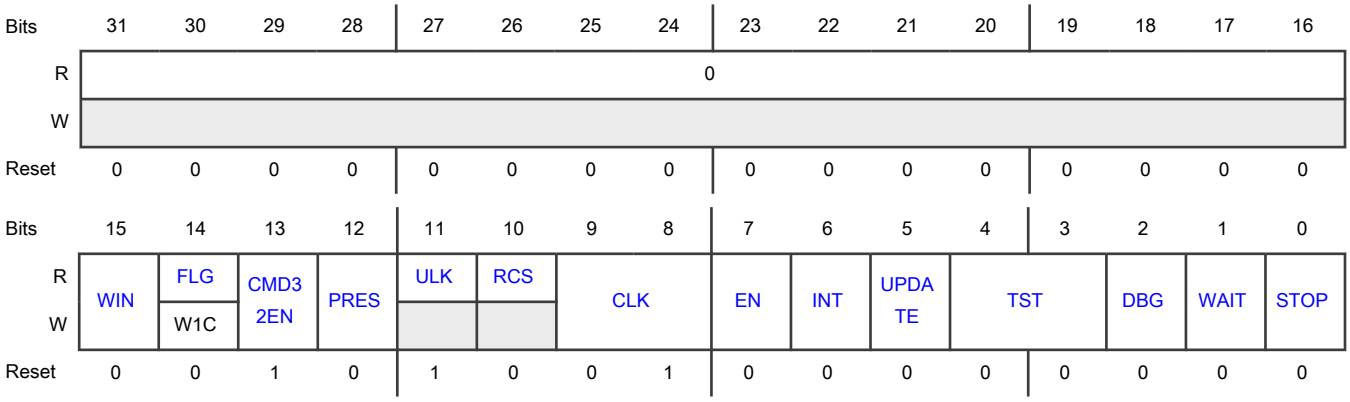
NOTE

TST is cleared (0:0) on POR only. Any other reset does not affect the value of this field.

Offset

Register	Offset
CS	0h

Diagram



Fields

Field	Description
31-16 —	Reserved
15 WIN	WDOG Window This write-once bitfield enables Window mode. 0 - Disable 1 - Enable
14 FLG	WDOG Interrupt Flag Acts as an interrupt indicator when CS[INT] = 1. 0 - No interrupt occurred 1 - An interrupt occurred
13 CMD32EN	Command 32 Enable This write-once bitfield enables or disables WDOG support for 32-bit (otherwise 16-bit or 8-bit) refresh or unlock command write words. If this field = 0, it disables support for 32-bit refresh or unlock command write words. Only a 16-bit or 8-bit write is supported. If this field = 1, it enables support for 32-bit refresh or unlock command write words. A 16-bit or 8-bit write is not supported.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You must unlock WDOG after writing to this field for reconfiguration. 0 - Disable 1 - Enable
12 PRES	WDOG Prescaler This write-once bitfield enables a fixed 256 prescaling of the WDOG counter reference clock. See Block diagram that shows the clock divider option. 0 - Disable 1 - Enable
11 ULK	Unlock Status Indicates whether WDOG is unlocked. 0 - Locked 1 - Unlocked
10 RCS	Reconfiguration Success Indicates whether the reconfiguration is successful. This field becomes 1 when new configuration takes effect, and becomes 0 with a successful unlock command. 0 - Unsuccessful 1 - Successful
9-8 CLK	WDOG Clock Selects the clock source that feeds the WDOG counter. You can write to this field only once. **NOTE** See chip-specific WDOG information for details of bitfield settings.
7 EN	WDOG Enable Enables the WDOG counter to start counting. You can write to this field only once. 0 - Disable 1 - Enable
6 INT	WDOG Interrupt Configures WDOG to immediately generate an interrupt request upon a reset-triggering event (timeout or illegal write to WDOG), before forcing a reset. After the interrupt vector fetch (that takes place after the reset-triggering event), the reset occurs after a delay of 128 bus clocks. If this field = 0, it disables WDOG interrupts (WDOG resets are not delayed). If this field = 1, it enables WDOG interrupts (WDOG interrupts are delayed by 128 bus clocks from the interrupt vector fetch).

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can write to this field only once. 0 - Disable 1 - Enable
5 UPDATE	Updates Allowed Allows you to reconfigure WDOG without a reset. If this field = 0, you cannot update WDOG after the initial configuration, without forcing a reset. If this field = 1, you can modify the WDOG configuration registers within 1024 bus clocks after performing the unlock write sequence. You can write to this field only once. 0 - Updates not allowed 1 - Updates allowed
4-3 TST	WDOG Test Enables Fast Test mode, which allows you to exercise all bits of the counter to demonstrate that WDOG is functioning properly. See Fast testing of WDOG for more information. If this field = 0, it disables WDOG Test mode. If this field = 1, it enables WDOG User mode and disables WDOG Test mode. After testing WDOG, you must use this setting to indicate that WDOG is functioning normally in User mode. If this field = 10, it enables WDOG Test mode; only the low byte is used. CNTLOW is compared with TOVALLOW. If this field = 11, it enables WDOG Test mode; only the high byte is used. CNTHIGH is compared with TOVALHIGH. This write-once field is cleared (0:0) on POR only. Any other reset does not affect the value of this field. 00 - Disable WDOG Test mode 01 - Enable WDOG User mode 10-11 - Enable WDOG Test mode
2 DBG	Debug Enable Enables WDOG to operate when the chip is in Debug mode. You can write to this field only once. 0 - Disable 1 - Enable
1 WAIT	Wait Enable Enables WDOG to operate when the chip is in Wait mode. You can write to this field only once. 0 - Disable 1 - Enable
0 STOP	Stop Enable Enables WDOG to operate when the chip is in Stop mode.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	You can write to this field only once. 0 - Disable 1 - Enable

69.7.1.3 WDOG Counter (CNT)

Provides access to the value of the free-running WDOG counter. You can read the counter register at any time but cannot write directly to it. However, the following write sequences to this register have special functions:

1. The refresh sequence resets WDOG counter to 0000h. See Refreshing WDOG for more information.
2. The unlock sequence allows WDOG to be reconfigured without forcing a reset (when CS[UPDATE] = 1). See Configuring for Reconfigurable mode for more information.

NOTE

All other writes to this register are illegal and force a reset.

Offset

Register	Offset
CNT	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNTHIGH								CNTLOW							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-8	Counter High Byte

Table continues on the next page...

Table continued from the previous page...

Field	Description
CNTHIGH	Contains the WDOG high byte.
7-0 CNTLOW	Counter Low Byte Contains the WDOG low byte.

69.7.1.4 WDOG Timeout Value (TOVAL)

Contains the 16-bit value used to set the timeout period of WDOG.

The WDOG counter (CNT) is continuously compared with the timeout value (TOVAL). If the counter reaches the timeout value, WDOG forces a reset triggering event.

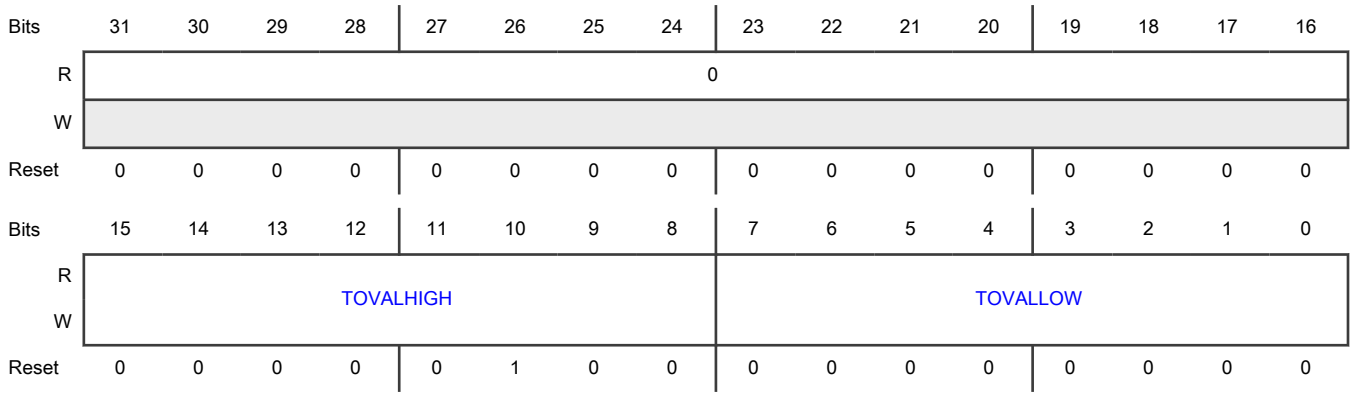
NOTE

Do not write 0 to this register (if CS[TST] = 11b, then TOVALHIGH cannot be written as 0; if CS[TST] = 10b, then TOVALLOW cannot be 0); otherwise, WDOG always generates a reset.

Offset

Register	Offset
TOVAL	8h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 TOVALHIGH	Timeout Value High Contains the high byte of the timeout value.

Table continues on the next page...

Table continued from the previous page...

Field	Description
7-0 TOVALLOW	Timeout Value Low Contains the low byte of the timeout value.

69.7.1.5 Watchdog Window (WIN)

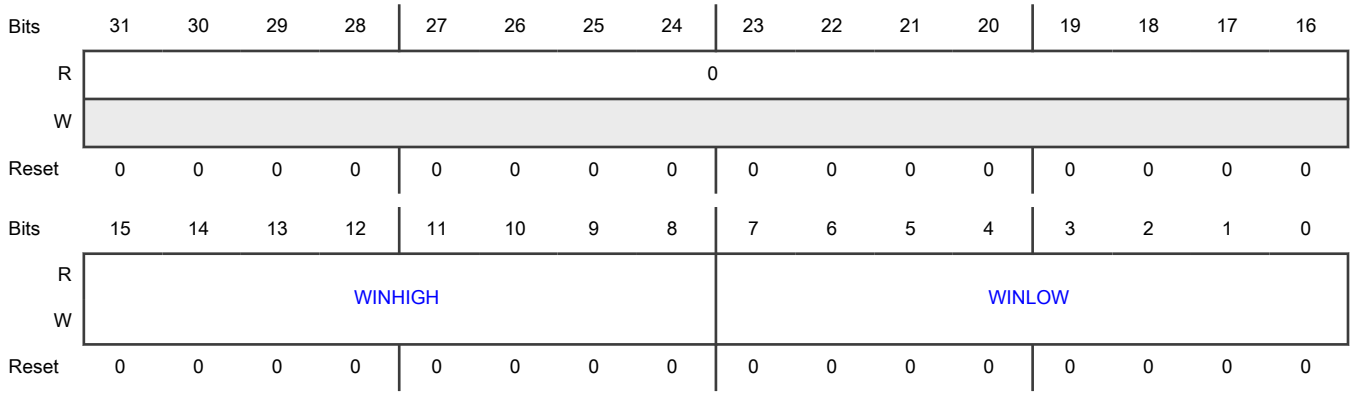
Determines the earliest time that a refresh sequence is considered valid, if CS[WIN] = 1. See Refresh mechanism for more information.

The WIN register value must be less than the TOVAL register value.

Offset

Register	Offset
WIN	Ch

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 WINHIGH	High Byte Contains the high byte of the WDOG window.
7-0 WINLOW	Low Byte Contains the low byte of the WDOG window.

Chapter 70
Analog Overview

70.1 Overview

The following analog modules are integrated on the chip:

- Analog-Digital-Converter (ADC)
- Temperature sensor
- XTAL

70.2 Analog-to-Digital Converter (ADC)

i.MX93 integrates 1 ADC. The main features are:

- It includes eight channels, four of them connected to pins in the package.
- Support the 1MS/s frequency of operation
- Multiple modes of starting conversion (Normal, Injected)

  Normal mode supports One-Shot and Scan (continuous) conversions

  Injected mode supports One-Shot conversions only
- Support TRGMUX to allow 16 trigger channels to be used by any ADC channel

70.3 Temperature Sensor

The temoerature sensor is included to monitor the die temperature. It has a digital controller which allows software to enable/disable it and read the temperature. The digital controller also provides capability to generate alarm IRQ to the CPU as follows:

- The temperature sensor has two programmable trip points: normal and critical. Each trip point has its corresponding interrupt output.
- When the temperature is above the “normal” point, an interrupt for normal trip point is generated.
- When the temperature is above the “critical” point, an interrupt for critical trip point is generated.

70.4 XTAL

The chip has two XTAL modules:

- 24MHz XTAL module for system clock
- 32KHz XTAL module for RTC

Each XTAL includes:

- Crystal oscillator to generate reference clock;
- Digital control logics for the XTAL.

70.5 PLL

The chip has multiple PLLs to meet the clock system requirement:

- Integer PLL for ARM PLL.
- Fractional PLL for SYSTEM_PLL1, DRAM PLL, AUDIO_PLL and VIDEO_PLL. Support spread spectrum.
- Each PLL has independent control to be clocked from either 24MHz XTAL, or CLKIN1 or CLKIN2 or CLKIN1 XOR CLKIN2.

70.6 DIGPROG Device ID

The chip has a 32-bit DIGPROG Device ID in OSC register space. Please refer DIGPROG DEVICE ID in OSC register descriptions.

Field	Description
31-24	**0x0** Reserved
23-16 DIGPROG_MAJOR_UPPER	Bit[7:4] is **0x9** Bit[3:0] is **0x0**, reserved
15-8 DIGPROG_MAJOR_LOWER	Bit[7:4] is the number of Cortex-A cores physically implemented. **0x20**
7-0 DIGPROG_MINOR	Bit[7:4] is the base layer revision Bit[3:0] is the metal layer revision. **0x10**

Chapter 71
Trigger Mux (TRGMUX)

71.1 Chip-specific TRGMUX information

Table 623. Reference links to related information

Topic	Related module	Reference
Full description	Arm Cortex CM33 Platform (CM33)	CM33 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

This device has one instance of the TRGMUX module. It allows any of the 16 trigger channels to be used by any ADC channel.

Please refer Hardware Triggers for details on the triggers supported and the connections between TRGMUX and SAR_ADC.

71.2 Overview

TRGMUX allows you to configure the trigger inputs for various peripherals.

71.2.1 Block diagram

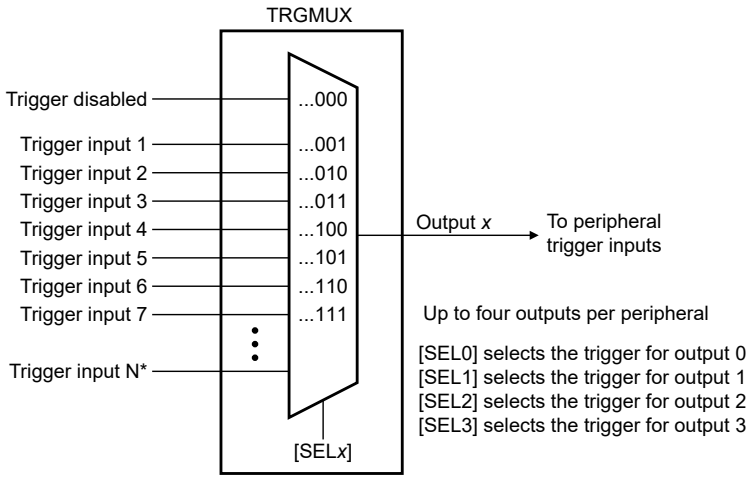


Figure 425. Block diagram

NOTE

Up to 255 trigger inputs may be available for SEL0, SEL1, and SEL2. For SEL3, up to 127 trigger inputs may be available. When the number of trigger inputs is 255, SEL3 is not available and becomes reserved. See the chip-specific TRGMUX information for details about trigger input and output configuration on this chip.

71.2.2 Features

- Configurable trigger sources for peripherals

- Dedicated TRGMUX register for each peripheral

71.3 Functional description

71.3.1 Clocking

This module has no clocking considerations.

71.3.2 Interrupts

This module has no interrupts.

71.4 External signals

This module has no external signals.

71.5 Initialization

This module does not require initialization.

71.6 TRGMUX register descriptions

71.6.1 TRGMUX memory map

You can only write to TRGMUX registers in Supervisor mode.

Table 624. Select bit fields

Field	Description
SEL*x*	Specifies the MUX select for the peripheral trigger inputs. Use this field to select the trigger sources for peripheral modules. 0h - TPM1 1h - TPM2 2h - TPM3 3h - TPM4 4h - TPM5 5h - TPM6 6h - LPIT1 7h - LPIT2 8h - LPTMR1 9h - LPTMR2 Ah - GPIO1 Bh - GPIO2 Ch - GPIO3 Dh - FLEXIO1 Eh - FLEXIO2 Fh - M33.TXEV

trgmux base address: 4453_1000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	TRGMUX REG0 (REG0)	32	RW	0000_0000

71.6.2 TRGMUX REG0 (REG0)

Configures the REG0 module.

Offset

Register	Offset
REG0	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	0							0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				SEL1				0				SEL0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 LK	TRGMUX Register Lock Disables writing to the register. You can write to this field once after system reset. When this field is 1, you cannot write to SEL*x* until the next system reset. This field becomes 0 after system reset. 0 - Register is writable 1 - Register is not writable until the next system reset
30-24 —	Reserved
23-16 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-12 —	Reserved
11-8 SEL1	TRGMUX Source Select 1 Specifies the source select for output 1. See Table 624 for field values.
7-4 —	Reserved
3-0 SEL0	TRGMUX Source Select 0 Specifies the source select for output 0. See Table 624 for field values.

Chapter 72
Analog-to-Digital Converter (SAR_ADC)

72.1 Chip-specific ADC information

Table 625. Reference links to related information

Topic	Related module	Reference
Full description	Arm Cortex CM33 Platform (CM33)	CM33 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

The chip supports a 8-channel 12-bit ADC with the lower 4 channels (CH0-CH3) connected to pins in the package.

NOTE

Channels 32, 35-38, are mapped for internal purpose.

NOTE

CH4-CH7 are not exposed to pins and connected to shield circuit.

Table 626. ADC internal channels

Channel number	Group	Channel name
0	0	ADC_CH_00
1	0	ADC_CH_01
2	0	ADC_CH_02
3	0	ADC_CH_03
4	0	Reserved/Shield
5	0	Reserved/Shield
6	0	Reserved/Shield
7	0	Reserved/Shield
8-39	Reserved	Reserved

72.1.1 Module instances

This device has one instance of the ADC module. It has its own dedicated pads and is not muxed with GPIO.

72.1.2 Hardware Triggers

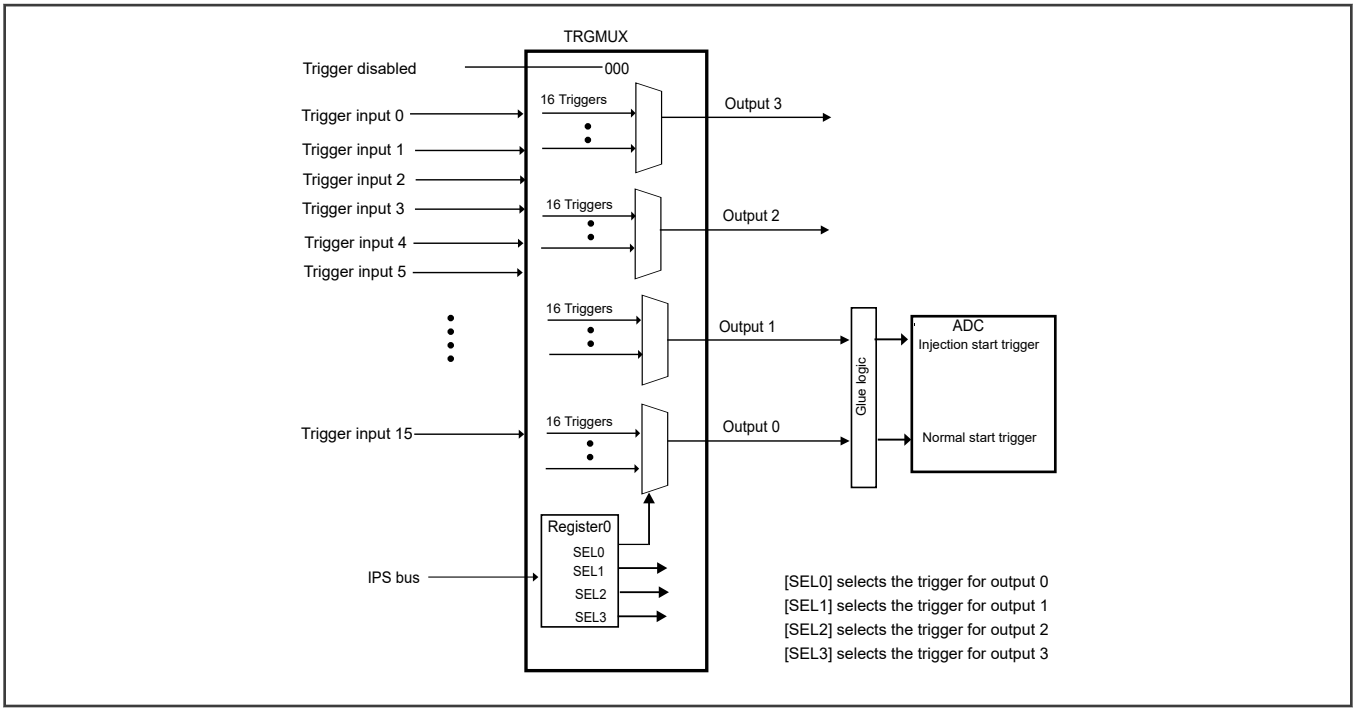
The i.MX93 supports a TRGMUX to allow any of the 16 trigger channels shown below to be used by any ADC channel. Hardware triggers available on the chip are as below:

Trigger #	Trigger Source	Comment
0	TPM1	TPM global interrupt or trigger output
1	TPM2	TPM global interrupt or trigger output
2	TPM3	TPM global interrupt or trigger output
3	TPM4	TPM global interrupt or trigger output
4	TPM5	TPM global interrupt or trigger output
5	TPM6	TPM global interrupt or trigger output
6	LPIT1	LPIT global interrupt or trigger output
7	LPIT2	LPIT global interrupt or trigger output
8	LPTMR1	LPTMR async interrupt or trigger output
9	LPTMR2	LPTMR async interrupt or trigger output
10	GPIO1	Combined interrupt indication for all GPIO1 signals
11	GPIO2	Combined interrupt indication for all GPIO2 signals
12	GPIO3	Combined interrupt indication for all GPIO3 signals
13	FLEXIO1	Trigger from FlexIO1
14	FLEXIO2	Trigger from FlexIO2
15	M33_TXEV	Event transmitted as a result of SEV instruction

The figure below shows the connections between ADC and TRGMUX.

NOTE

Injection start trigger is connected to output 1 and Normal start trigger is connected to output 0 of TRGMUX.



For more information on the FLEXIO triggers, please refer Chip-specific FLEXIO information.

72.2 Overview

SAR_ADC provides fast and accurate analog-to-digital conversion using the Successive Approximation Register (SAR) method. It has 12-bit resolution and is highly configurable for a wide range of applications.

72.2.1 Block diagram

The following figure shows the high-level interface diagram of SAR_ADC.

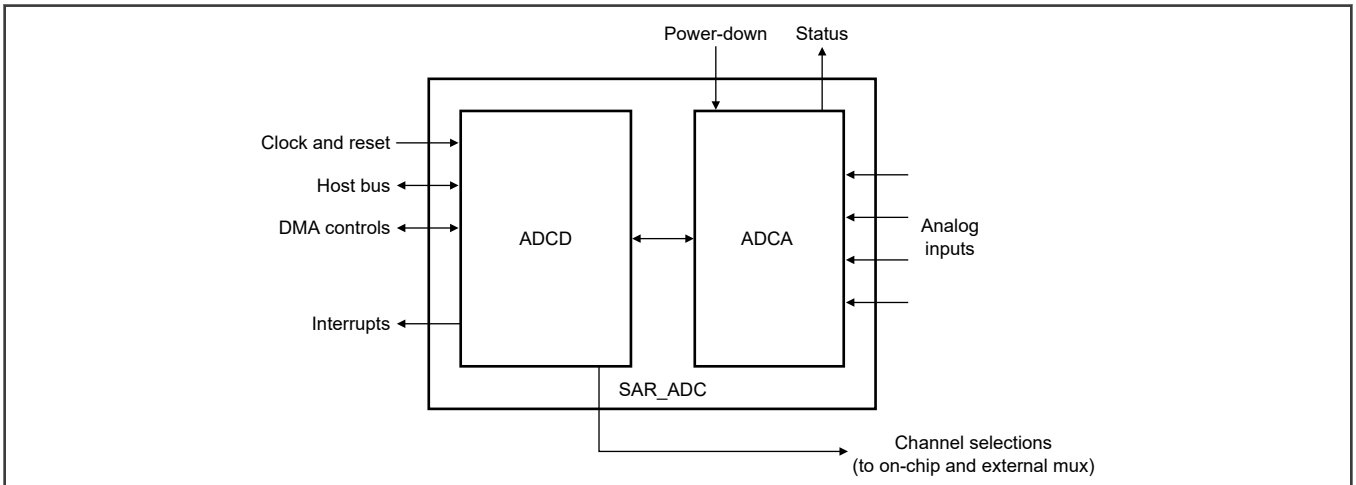


Figure 426. SAR_ADC high-level interface diagram

SAR_ADC interfaces as shown above are as follows:

- Analog inputs to sample the channels to be converted.
- Channel selection controls are provided to use internal or external analog multiplexers.

- Host Bus allows access to configuration, control, and status registers which are used to start/stop conversions, monitor the conversion results, errors, violations, and so on.
- DMA interface transfers conversion results by the DMA controller.
- Interrupts can be generated for various events depending on configuration (data availability, threshold violation, error, and so on).
- External hardware triggers are used to start conversion from sources external to SAR_ADC.
- Power-down controls are used to place the analog block into a deactivated state.

72.2.2 Features

The general feature list of SAR_ADC is shown below. Feature availability is chip-specific and is defined during module integration. See chip-specific SAR_ADC information for details.

- 4'd12-bit resolution
- Multiple modes of starting conversion (Normal, Injected)
  - Normal mode supports One-Shot and Scan (continuous) conversions
  - Injected mode supports One-Shot conversions only
- Software-initiated conversions in Normal and Injected modes, or external hardware trigger
- Two different abort features for either a single or chain conversion in Normal and Injected modes
- Independent data registers for each channel contain information about mode of conversion, data validity, overwrite status, and conversion data
- Alternate analog watchdog thresholds (threshold selected through input ports)
- Programmable DMA enables for each channel
- Individual interrupt flags for the following conditions:
  - End of conversion of a single channel for Normal and Injected modes
  - End of chain conversion for both Normal and Injected modes
  - Watchdog threshold violations
- Programmable presampling for channels
- Auto-Clock-Off feature for improved power performance
- Power-Down mode to place SAR_ADC in power-down state
- Programmable clock prescaler for SAR_ADC (bus clock, or bus clock divided by two or four)
- Software-initiated calibration
- Self-test feature

72.3 Functional description

The ADCD interfaces are shown in Figure 426. The available functionality in the ADCD is described in the following sections (see chip-specific SAR_ADC information for details).

72.3.1 Submodules

SAR_ADC consists of two major functional parts:

- ADC Digital (ADCD)
- ADC Analog hard macro cell (ADCA)

ADCD controls ADCA. ADCD holds the configuration, control, and status registers that are accessible through software using the host interface. ADCD executes the SAR Control algorithm. The selection of channel multiplexers and analog gates is driven by ADCD based on the configuration you perform. Conversion results are stored in registers, and the results can be transferred to memory by DMA or read by host access.

ADCA holds all analog circuitry which consists of a comparator, capacitor bank, bias generation, and so on. Only one channel is processed at a time. Multiple channels are processed sequentially by multiplexing the analog inputs. You can configure the duration for which each channel is sampled. The hold value is used to calculate a channel's digital code by successively comparing it with an analog voltage generated by a Digital to Analog Converter (DAC) inside ADCA.

72.3.2 Special channels

Reference voltages and supply voltages are available at predefined SAR_ADC channels.

72.3.3 Conversion

Conversions are software or external hardware initiated.

The following list shows the methods of initiating conversions in order of priority (highest to lowest):

1. Injection (see Injected Conversion mode)
2. Normal (see Normal Conversion mode)

Each type of conversion method has different processing features.

Injection can start a single chain of conversions one at a time, where a chain may have from one up to the maximum number of channels available (each channel appears once in the chain).

Normal conversion can start a continuous or single chain of conversion one at a time. In continuous mode, the chain of conversions is repeated unless you stop them.

When a normal conversion is in process, the starting injected conversions are done by software or external hardware. This allows a chain of channels that cannot wait to be converted (as selected by JCMR) to be handled immediately.

Analog watchdogs allow continuous hardware monitoring of input channel conversion results.

A built-in self-test function enables periodic checking of ADCA health. If a fault is detected, the FCCU can be signaled. The severity of the fault is programmable (critical/non-critical).

NOTE

ADC configurations must be changed only when ADC is in idle or power-down state (MSR[NSTART]=0 and MSR[ADCSTATUS] = idle/power-down).

72.3.3.1 Normal Conversion mode

The conversion method which is generally used to convert channels is Normal Conversion mode. To select channels for conversion, you program the Normal Conversion Mask Register (NCMR). Each channel can be individually enabled by setting the corresponding field in NCMR:

- NCMR0 selects channels in Group0.
- NCMR1 selects channels in Group32.

The mask registers must be programmed before starting the conversion and cannot be modified until all selected channel conversions have completed. A conversion always starts from the lowest channel number and continues sequentially to the highest channel number.

72.3.3.1.1 Starting conversions in Normal Conversion mode

After programming the required fields of the Main Configuration Register (MCR) and NCMR, conversion starts as follows:

- By software:

— Conversion chain starts when one is written to MCR[NSTART].

- By hardware trigger:
  — If external trigger is enabled (MCR[TRGEN] = 1), and an external trigger enable is detected, the conversion will start. The enable is checked only during the start of conversion. The hardware trigger needs to be active for a least 2 ADC bus clock cycles in order to recognized and start a conversion.
    ◦ Detection of an active edge is defined by MCR[EDGE] (rising = 1, falling = 0). When an external trigger is recognized, MSR[NSTART] is set and normal conversion starts.

NOTE

An external trigger, or software-initiated conversion, in Normal Conversion mode is ignored during any ongoing conversion chain.

NOTE

There must be a gap of three cycles between configuration of the NCMR and the start of conversion (setting of MCR[NSTART] or arrival of hardware start trigger) for proper SAR_ADC operation. SAR_ADC calculates the total number of channels to be converted during this period.

NOTE

There must be a gap of three cycles between configuration of the NCMR and the start of conversion (setting of MCR[NSTART] or arrival of hardware start trigger) for proper SAR_ADC operation. SAR_ADC calculates the total number of channels to be converted during this period.

72.3.3.1.2 Operation modes in Normal Conversion mode

Two operation modes are available for normal conversions:

- One-Shot Operation mode
- Scan Operation mode

The MCR[MODE] field determines the operating mode in normal conversion. The first phase of any conversion process involves sampling the analog channel. Which is followed by the conversion phase, when the sampled analog value is converted to digital as shown in Figure 427.

Figure 427. Normal conversion flow

In **One-Shot Operation mode** (MCR[MODE] = 0), a sequential conversion specified in the NCMR mask registers is performed once. At the end of each conversion, the digital result of the conversion is stored in the corresponding data register (CDR*n*; *n* = channel number).

For example: Channels A-B-C-D-E-F-G-H are available on the chip, where channels B-D-E are to be converted in One-Shot Operation mode. Conversion starts from channel B, followed by conversion of channels D-E. At the end of conversion of channel E the channel scanning stops.

MSR[NSTART] is automatically set when the normal conversion starts. At that time MCR[NSTART] is reset by hardware, allowing the software to program a new start of conversion in advance. In this case, the new requested conversion starts after completion of the currently running conversion. However, for correct device function, it is required to program MCR[NSTART] for the new conversion only after the current conversion is complete. Software can wait for the end of conversion (ISR[ECH] = 1) prior to setting MCR[NSTART] again.

If the conversion chain starts by external trigger (if MCR[TRGEN] = 1), the ADC cannot monitor the trigger input until the conversion chain finished. After the chain is finished, the next chain starts by subsequent hardware trigger edge.

In **Scan Operation mode** (MCR[MODE] = 1), a sequential conversion of *n* channels specified in the NCMR registers is continuously performed. At the end of each conversion the converted result is stored into the corresponding channel data register, as it is in One-Shot Operation mode.

MSR[NSTART] is automatically set when normal conversion starts. Unlike One-Shot Operation mode, MCR[NSTART] is not reset in Scan Operation mode. It can be reset by software when you need to stop the Scan Operation mode. In that case, the ADC will complete the current chain, and after the last conversion, it clears MSR[NSTART].

For example: Channels A-B-C-D-E-F-G-H are available on the chip, where channels B-D-E are to be converted in Scan Operation mode. Conversion starts from channel B followed by conversion of channels D-E. At the end of conversion of channel E the scanning of channel B starts again, followed by conversion of channels D-E. This sequence repeats itself until MCR[NSTART] is reset by software.

This scan chain can be stopped by writing zero to MCR[NSTART]. The conversion stops when the ongoing chain is finished.

If the conversion is started by an external trigger, MCR[NSTART] is not set. As a consequence, after it is started, the only way to stop a Scan Operation mode conversion started by external trigger is to write MCR[MODE] = 0.

End of conversion:

In both modes, at the end of each conversion an End of Conversion (EOC) interrupt is issued (if enabled by the corresponding mask bit in CIMR and IMR).

After conversion of all the selected channels in NCMR has completed, the conversion operation is considered finished. ISR[ECH] is then set and End of Chain (ECH) interrupt is issued (if enabled in IMR[MSKECH]).

The corresponding channel bit of CEOCFR is updated to indicate data is available in data register CDR*n* of the corresponding channel.

If there is no channel selected in NCMR, and there is a start of conversion trigger, ISR[ECH] is set and an ECH interrupt is issued immediately, if enabled.

NOTE

In scan (continuous) mode, ADC conversions happen very regularly as per timing derived by sample time and resolution. The host reads the result registers after checking ECH flag. The reading time of the host after setting ECH could vary based on the load on it. So, it may occasionally get invalid data (valid bit 0b in data CDR register) in channels. To avoid such scenarios, check the ECH flag twice before reading the conversion results, to ensure that the host gets valid data always for all channels. The host needs to clear the ECH flag each time it finds it set. The sequence will be as below:

1. Host waits for ECH. On ECH, it clears it.
2. Host waits for second ECH. On ECH, it clears it.
3. Host reads data for all channels.
4. Repeat steps 1–3.

72.3.3.2 Injected Conversion mode

An injected conversion is a conversion of a programmed chain that can be injected anytime into an ongoing conversion chain processed in Normal Conversion mode. You require to configure Injected Mask Register(s) (JCMR*n*). In Normal Conversion mode each internal or external channel can be individually selected:

- JCMR0 for channels in Group0
- JCMR1 for channels in Group32

An injected conversion chain can only be run in One-Shot Operation mode and suspends the ongoing conversion chain processed in Normal Conversion mode. When an injected conversion chain is inserted into an ongoing chain processed in Normal Conversion mode, the ongoing chain is suspended and the injected conversion request is processed. After conversion completion of the last channel of the injected chain, conversion resumes from the channel at the chain which was suspended in Normal Conversion mode, as shown in Figure 428.

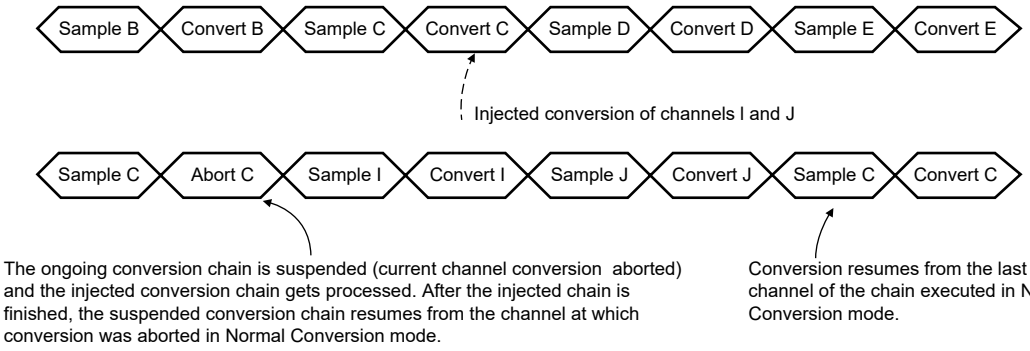


Figure 428. Injected sample/conversion sequence

72.3.3.2.1 Starting conversions in Injected Conversion mode

The injected conversion chain can be started by using one of the following methods:

- By software: When external triggering is disabled (MCR[JTRGEN] = 0), MCR[JSTART] can be set. Setting MCR[JSTART] causes any Normal Conversion mode chain to be suspended and the injected chain to start processing.
- By external trigger: When external triggering is enabled (MCR[JTRGEN] = 1), a programmed event on the external injection input (high or low depending on MCR[JEDGE] setting) starts the Injected Conversion mode. The external trigger needs to be active for a least 2 ADC bus clock cycles in order to recognized and start a conversion.

MSR[JSTART] is automatically set when the injected conversion chain starts, and MCR[JSTART] is cleared. This allows the software to program a new start of conversion in advance. If a new injected conversion request occurs while an injected conversion chain is in progress (for example, setting MCR[JSTART] while MSR[JSTART] is set), then conversion will start after the current conversion has completed.

At the end of each conversion, an End Of Injected Conversion (JEOC) (ISR[JEOC] = 1) interrupt is issued (that is, IMR[MSKJEOC] = 1), and at the end of the chain an End of Injected Chain (JECH) (ISR[JECH] = 1) interrupt is issued (that is, IMR[MSKJECH] = 1).

The corresponding channel bit of CEOCFR*n* is set, indicating that data is available in the CDR of the respective channel.

After it is started, an injected chain conversion cannot be interrupted.

NOTE

If there is no channel selected in JCMR*n*, and an injected conversion is started, then ISR[JECH] is immediately set and the corresponding interrupt is asserted (if enabled). This behavior is independent of normal conversion status.

For correct operation of SAR_ADC, there must be a gap of at least three cycles between configuration of JCMR*n* and the start of a conversion (setting MCR[JSTART] or receiving an external hardware trigger). SAR_ADC calculates the total number of channels to be converted during this period.

72.3.3.3 Abort of conversion

Two different abort functions are provided:

- Abort of a single channel
- Abort of chain

You can abort the ongoing conversion by setting MCR[ABORT]. SAR_ADC then aborts the current conversion and immediately starts the conversion of the next channel of the chain. The abort action may take 1–4 cycles of bus clock to abort the current conversion, depending on the state of the conversion. At the start of any conversion and at the end of a particular conversion, when internal state counters are changing, then the abort get delay by a maximum of three cycles. This allows all states to be placed in stable condition before aborting.

So, you are advised not to change the abort field status for the next three cycles to achieve this effect.

Abort also not to be programmed along with and within three cycles of programming MCR[NSTART] or MCR[JSTART].

In the case of an abort operation, MSR[NSTART/JSTART] remains set if not in the last channel of the chain, and MCR[ABORT] is reset as soon as the channel is aborted. The EOC corresponding to the aborted channel is not generated. This behavior is true for Normal or Injected Conversion modes. If the last channel of a chain is aborted, the end of chain is reported, generating an ECH interrupt.

It is also possible to abort the current chain of conversions by setting MCR[ABORTCHAIN]. In that case, the behavior of the ADC depends on MCR[MODE] (One-Shot/Scan Operation modes). If Scan Operation mode is disabled, MSR[NSTART] is automatically reset together with MCR[ABORTCHAIN]. Otherwise, in Scan Operation mode, a new chain is started. The EOC of the current aborted conversion is not generated but an ECH interrupt is generated to signal the end of the chain.

NOTE

For a single channel chain, ECH is not generated in the case of an abort or abort chain.

When an ABORTCHAIN is requested while an injected chain is running over a suspended normal chain, both injected and normal chains are aborted. Both ISR[ECH] and ISR[JECH] are set. MSR[JSTART] is reset to mark the end of the injected chain. MSR[NSTART] is reset if normal conversion is running in One-Shot mode. Otherwise, in Scan mode it remains set and a new chain is started.

72.3.3.4 Conversion time

Total conversion time depends on SAR_ADC clock (AD_CLK). This clock can be prescaled to half of bus clock. SAR_ADC operating clock (AD_CLK) is always synchronous with the bus clock. The frequency of AD_CLK depends on MCR[ADCLKSE] settings.

- When MCR[ADCLKSE] = 1, AD_CLK = bus clock.
- When MCR[ADCLKSE] = 0, AD_CLK = (bus clock / 2).

Conversion timing detail is shown in the figure below. The description of each time segment follows the figure.

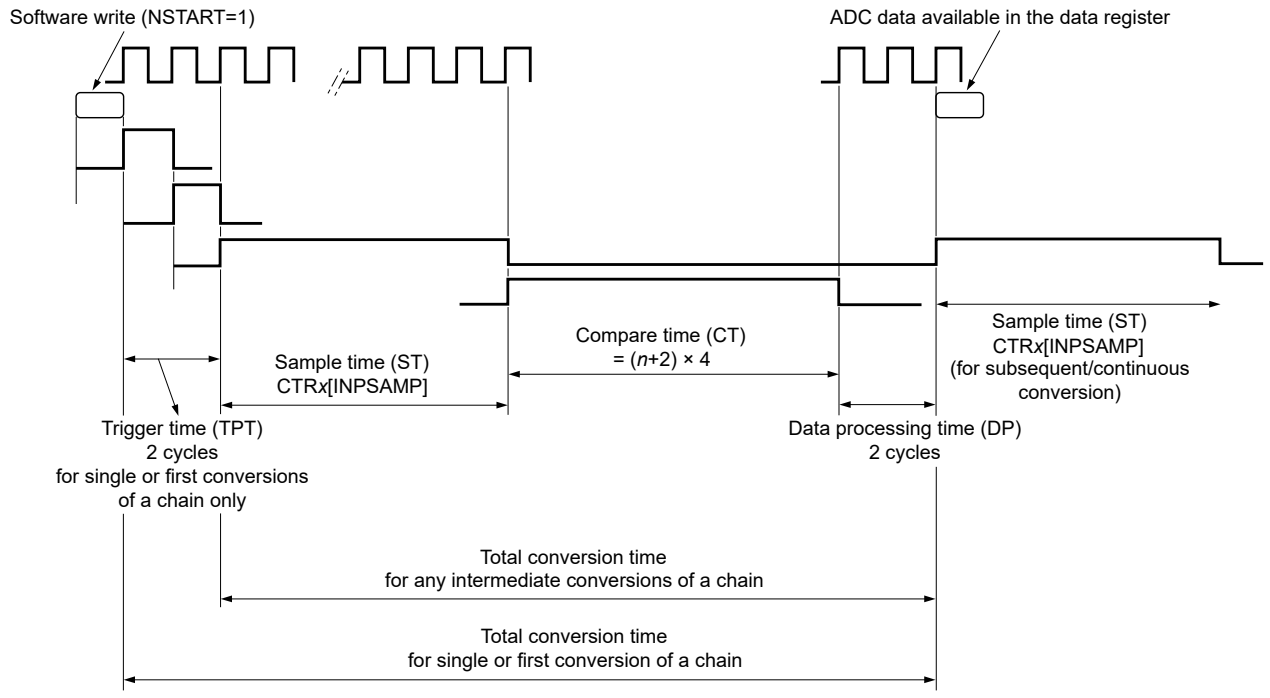


Figure 429. Conversion time

Trigger time (TPT):

To prepare a channel for conversion, two bus clock cycles are required by SAR_ADC for the single conversion and for the first conversion in a chain. For continuous conversions and subsequent conversions in a chain, this time is not required because it is hidden in the pipeline operation.

Sample phase time (ST):

ST duration is controlled by the CTR*n*[INPSAMP[7:0]] configuration. The value in the register field represents units of AD_CLK cycles. The minimum value of sample time is eight AD_CLK cycles. If the value programmed in CTR*n*[INPSAMP[7:0]] is less than eight, it has no effect on sample time duration and will default to eight AD_CLK cycles.

Compare phase time (CT):

CT depends on the resolution of SAR_ADC. The CT is ((x + 2) × 4) cycles of AD_CLK, where x = 4'd12 is the resolution of SAR_ADC.

Data processing time (DP):

DP takes two AD_CLK cycles to post and load data into the data register.

Total conversion time:

The total time required for a single or first conversion is (TPT + ST + CT + DP) AD_CLK cycles. The time for subsequent conversions in a chain or continuous conversions is (ST + CT + DP) AD_CLK cycles.

Examples:

- CTR*n*[INPSAMP] = 20d (software or hardware triggered, MCR[NSTART]):
  - Single or first conversion: 2 + 20 + ((12 + 2) × 4) + 2 = 2 + 20 + 56 +2 = 80 AD_CLK cycles
  - Subsequent or continuous conversions: 20 + ((12 + 2)× 4) + 2 = 20 + 56 + 2 = 78 AD_CLK cycles

Presample phase time (PST):

When presampling is enabled, SAR_ADC presamples prior to starting the actual sampling, as described in Presampling. This phase takes the number of AD_CLK cycles defined in CTR*n*[INPSAMP]. Additionally, to switch from presample phase to actual channel sampling phase, it takes another two AD_CLK cycles. So, the total presampling AD_CLK cycle is CTR*n*[INPSAMP] + 2 additional cycles that are required for a conversion to complete. If presampling is implemented, all equations and examples above are valid only with the additional (CTRN*n*[INPSAMP] + 2) AD_CLK cycles.

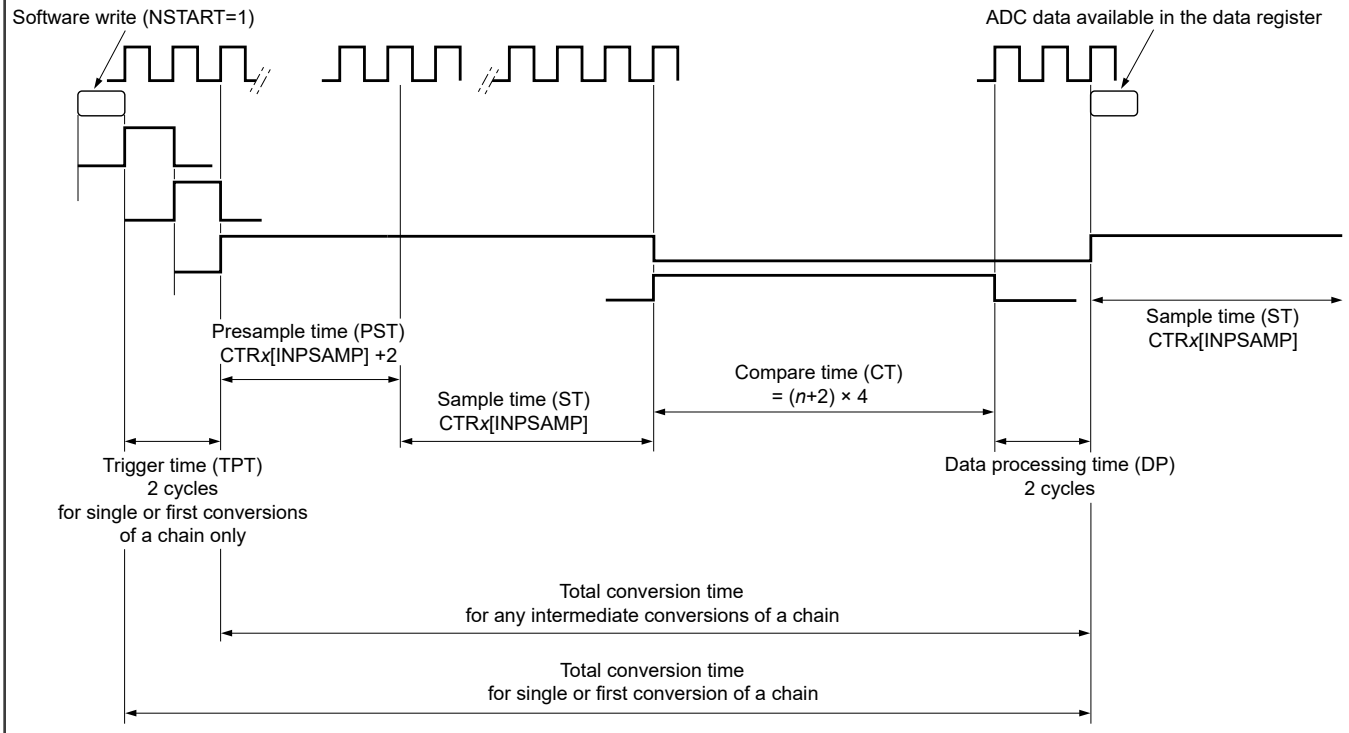


Figure 430. Conversion timing with presampling

Conversion time for calibration:

- Single conversion time = Sample time (MCR[TSAMP]) + CT (11 × 4) + DP (2)
- Interconversion gap = 1 cycle
- Total number of tests = 12
- Averaging = 32

One test duration = (single conversion time + interconversion gap) × averaging samples

Total test duration = (one test duration × total number of tests)

Example: Sample time = 22

Total test time = ((((22 + (11× 4) + 2) + 1) × 32) × 12) = 26496 cycles × 25 ns (40 MHz) = 662.4 µs

Conversion time for self-test conversion:

The time for a self-test conversion is one test period multiplied by the number of consecutive (atomic) steps. For supply self-tests, the atomic step is three and for other tests it is one.

One test period for self-test is:

- For supply tests: (ST + (14 × 4) + 2)
- For other algorithms: (ST + (11 × 4) + 2)

NOTE

Steps (conversions) of the supply algorithm always run in atomic operation, required for algorithmic procedure. So, in Scan Operation mode when all algorithms are selected, three steps (conversions) of supply algorithm run together after end of the chain. Steps (conversions) of algorithm C are not required to be run together, so after every chain one step (conversion) is executed serially. Conversion time in self-test also needs to be calculated accordingly.

Example:

- Channels in chain A→B→C
- Scan Operation mode
- All self-test algorithms selected

The resulting conversion order will be:

[A→B→C→AS_step0→AS_step1→AS_step2→A→B→C→AC_step0→A→B→C→AC_step1→A→B→C→AC_step2→ .....→A→B→C→AC_step11] →[same sequence repeats]

NOTE

AS = Algorithm S (supply self-test)

AC = Algorithm C (capacitive self-test)

72.3.3.5 Conversion data processing

The raw converted data of SAR_ADC contains many types of errors like offset, gain, and so on. In order to reduce or eliminate errors, the raw conversion data is processed inside SAR_ADC controller before sending to result register/output. The process of error correction happens in parallel during bit-by-bit evaluation, with the correction values generated during offset determination and calibration process.

The error correction value is subtracted from the raw conversion result whenever the comparator output for one of the calibrated capacitors evaluates high during the conversion period.

In order to reduce data processing time, the error correction is done in parallel during each bit evaluation of a conversion. All lower-weighted capacitors are not calibrated and no error correction is performed for these capacitors, but they are self-tested to guarantee error contribution is below specified accuracy limits.

The final result is checked for any overflow or underflow. If the processed data is above the maximum value that can be represented by ADC resolution, then the output data is forced to all ones (FFFh for 12-bit resolution). Similarly, if the processed data is negative, then output data is forced to all zeros (000h for 12-bit resolution).

72.3.4 ADC clock prescaler and sample time settings

The AD_CLK frequency can be scaled by programming MCR[ADCLKSE]. If MCR[ADCLKSE] is set, then the AD_CLK frequency is the same as the bus clock. Otherwise, AD_CLK frequency is half of bus clock, MCR[ADCLKSE] bit can be written in power-down only when MCR[PWDN] = 1.

In order to support different sampling times for different type of channels the different conversion timing registers are used (CTR0, CTR1). CTR0 is associated to channels (0 to 31), CTR1 is associated to channels (32 to 63). See the register description for details.

72.3.5 Presampling

The presampling feature allows precharging or discharging SAR_ADC internal sample capacitor node to a defined level before sampling of the selected analog input channel starts. This is useful for resetting information (history effect/offset) regarding the last converted data. During presampling, ADC samples the internally generated voltage, while in the sampling SAR_ADC samples analog input coming from pads.

Presampling can be enabled/disabled on a per-channel basis by setting the corresponding bits in the PSR0, PSR1 registers. PSR0 serves channels 0–31 , PSR1 serves channels 32–63 .

After enabling the presampling for a channel, the normal sequence of operation will be Presampling → Sampling → Evaluation for that channel. Sampling of all channels can be bypassed by setting PSCR[PRECONV]. When sampling of a channel is bypassed, the sampled and stored internal voltage applied during the presampling phase get converted (see Figure 431). See Conversion time for the timing equation for conversion.

Presampling does not apply to self-test channels.

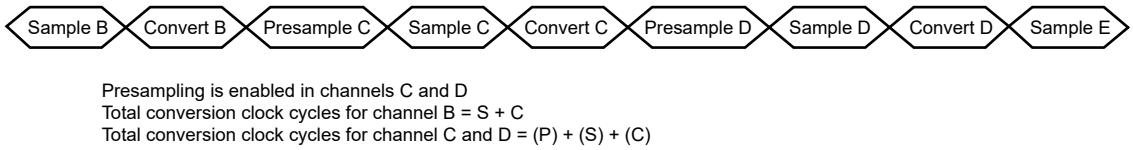


Figure 431. Presampling sequence

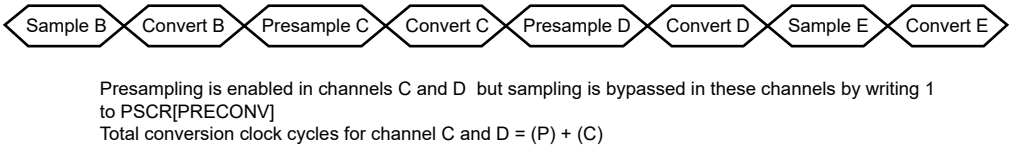


Figure 432. Presampling sequence with PRECONV=1

72.3.5.1 Presampling channel enable

The presampling channels are selected by programming the PSCR. It is possible to select between four internally generated voltages depending on the value of PSCR[PREVAL*n*], as shown in the following table.

Table 627. Presampling voltage selection

PSCR[PREVAL*n*]	Presampling voltage
00	Presample voltage - 1: (DVDD0P8/2)
01	Presample voltage - 2: (AVDD1p8/4)
10	Presample voltage - 3: VREFL_1p8
11	Presample voltage - 4: VREFH_1p8

NOTE

The voltage for presampling is (AVDD1p8/4) when PSCR[PREVAL]=01b. However, when this value is measured in sample bypass mode (PSCR[PRECONV]=1), the raw conversion result is shifted by two bits to the left (multiplied by 4) before error compensation. So the expected value of the final conversion result is AVDD1p8 (deviation may be in the range 0–3 LSB, due to left shift).

The PSCR[PREVAL*n*] fields are associated with SAR_ADC channels as follows:

- PREVAL0 for channels 0–31
- PREVAL1 for channels 32–63

72.3.6 Programmable analog watchdog

The analog watchdogs are used to monitor the conversion result to see if it is within defined limits (as shown in the figure below), specified by an upper and a lower threshold value named THRH and THRL, respectively.

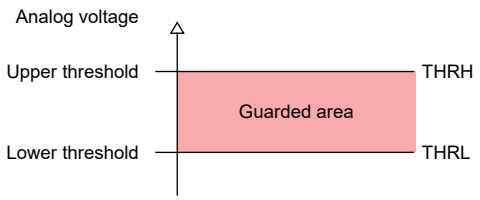


Figure 433. Guarded area

After the conversion of the selected channel, a comparison is performed between the converted value and the threshold values. If the converted value is outside the threshold values, then a corresponding threshold violation interrupt is generated.

The comparison result is stored as WDG*n*H and WDG*n*L fields in WTISR as explained in the following table. Depending on the values of WTIMR[MSKWDG*n*L] and WTIMR[MSKWDG*n*H] mask bits, an interrupt is generated on a threshold violation.

Table 628. Values of WDGxH and WDGxL fields

WDG*n*H	WDG*n*L	Converted data
1	0	Converted data > THRH
0	1	Converted data < THRL
0	0	THRH >= converted data >= THRL

Depending on the device settings, the number of analog watchdogs can vary for each instance of SAR_ADC (see chip-specific SAR_ADC information for details).

Analog watchdog setting

For each watchdog there is one threshold value (low and high) register (THRHLR*n* [*n*=0..15]).

The analog watchdog for each channel can be enabled independently by programming the corresponding CWENR fields.

- CWENR0 for Group0 channels
- CWENR1 for Group32 channels

Watchdog selection of each channel is done by configuring CWSELR*n*[WSEL_CH*m*] (where *n* = CWSELR instance and *m* = channel number).

Each CWSELR*n* holds a selection for a maximum of eight consecutive channels (for example, CWSELR0 holds eight WSEL_CH fields for channels 0–7, CWSELR1 is for channels 8–15, and so on).

If the conversion result of a selected channel is outside the range specified by threshold values, then the corresponding bit is set in the Analog Watchdog Out of Range Register (AWORR) (for example, if channel number 12 is to be monitored with the threshold values in register THRHLR3, then WSEL_CH4 of CWSELR1 is to be programmed with 3). Enabling the watchdog is done by setting the corresponding bit for channel 12 in register CWENR0.

In this setting, a set of threshold values (THRHLR*n*) can be linked to several SAR_ADC channels. The threshold values to be selected for a channel must be programmed only once in CWSELR*n* register.

NOTE

If the higher threshold of the analog watchdog is programmed to be below the lower threshold and the conversion result is below the lower threshold, then the WTISR[WDG*n*L] interrupt for the low threshold violation is set. If the conversion result is greater than the lower threshold (consequently also greater than the higher threshold), then the interrupt WTISR[WDG*n*H] for high threshold violation is set. Thus, you should take care to avoid that situation as it could lead to misinterpretation of the watchdog interrupts.

72.3.7 Clocking

SAR_ADC has only one clock input (bus clock).

This clock is used for accessing registers through the Host Bus and operation of all interfaces.

The conversion clock (AD_CLK), is derived from the bus clock input. The AD_CLK can be equal to, or half of the bus clock frequency based on configuration (see register bit descriptions of MCR[ADCLKSE]).

If MCR[ADCLKSE] = 1, then the AD_CLK is equal to the bus clock. If MCR[ADCLKSE] = 0, then the AD_CLK is half of the bus clock frequency. However, depending on configuration and performance requirements, MCR[ADCLKSE] may always need to be 1.

72.3.8 Reset

SAR_ADC has two resets, Functional (normal) and Calibration. Calibration reset is applied only when calibration is needed along, and is asserted with a functional reset. A functional reset can happen any time without a calibration reset.

Calibration reset is provided to hold internal calibration and status values if a functional reset occurs. The following register and field values are maintained after a functional reset:

- Main Status (MSR)
  - CALIBRTD
  - CALFAIL
- User OFFSET and Gain (USROFSGN) (Full register)
  - GAINUSER
  - OFFSUSER
- Calibration Status (CALSTAT) (Full register)
  - TEST_RESULT
  - STAT[1:14]
- Calibration Configuration 0 (CALCFG0) (Full register) (Also loaded from DCF/fuse bits)
  - BNDFLCNT
  - LDFAIL

72.3.9 Interrupts

SAR_ADC generates the following maskable interrupt signals:

- EOC (End of Conversion) interrupt request
- ECH (End of Chain) interrupt request
- JEOC (End of Injected Conversion mode) interrupt request
- JECH (End of injected Chain) interrupt request
- WDG*n*L and WDG*n*H (watchdog threshold) interrupt requests
- Self-testing Interrupts

Interrupts are generated during the conversion process to signal events such as EOC and so on, as explained in register description for ISR. Two registers named ISR (Interrupt Status Register) and IMR (Interrupt Mask Request Register) are provided in order to check and enable the interrupt request to external interrupt controller.

Interrupts can be individually enabled on a channel-by-channel basis by programming CIMR (Channel Interrupt Mask).

For pending EOC interrupts, registers with one bit per channel are provided to indicate completed conversions on a channel basis (CEOCFR*n*).

Interrupts generated due to the analog watchdogs are handled by two 32-bit registers named WTISR (Watchdog Threshold Interrupt Status Register) and WTIMR (Interrupt Mask for the Watchdog Threshold Register). These registers are used to check and enable an interrupt request . A watchdog interrupt causes two corresponding bits to be set for each channel monitored. One bit is set in the WTISR[WDG*n*H] register and the other bit in the WTISR[WDG*n*L] register.

Generated events are logged into status registers and used to generate an interrupt if enabled in corresponding mask registers.

In the below list, there are different interrupt outputs of ADC. The interrupt structure is shown in Figure 434.

Each interrupt is generated by combining a group of events.

- Interrupts for EOC, ECH, JEOC, JECH , and End of self-test algorithm (WDG_EOA_S and WDG_EOA_C) in STSR1 are combined (logic OR) into one interrupt output. .

- Interrupts of WDG*n*H and WDG*n*L are combined on the second interrupt output.
- Interrupts of all self-test errors (watchdog threshold, sequence, and timer, that is, WDSERR, WDTERR, ERR_S0, ERR_S1, ERR_S2, and ERR_C in STSR1) are combined on the third interrupt output.

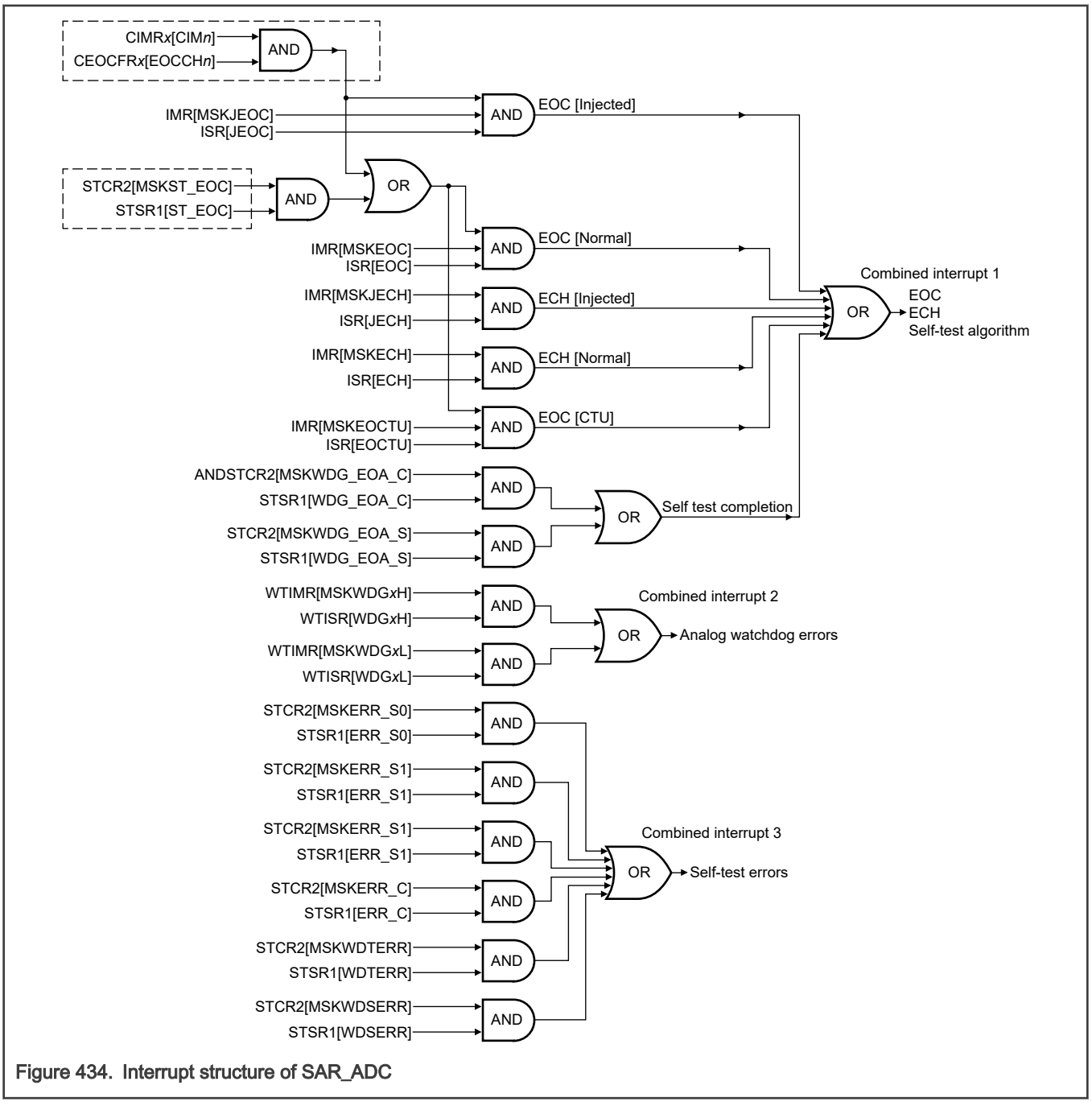


Figure 434. Interrupt structure of SAR_ADC

The ISR register contains the interrupt pending request status. If you want to clear a particular interrupt event status, then writing one to the corresponding status bit clears the pending interrupt flag (during this write operation all other bits of the ISR register must be maintained at zero).

The ISR must check the following for three different types of interrupts.

For interrupt 1:

- Read ISR to determine if the type of event falls in:

- EOC
- JEOC
- ECH
- JECH

- Read STSR1 to determine the end of self-test algorithm events (WDG_EOA_C / S) (if present).
- If the interrupt is EOC or JEOC:
  - Clear the channel pending bit of respective channel in CEOCFR*n* for EOC or JEOC in ISR.
  - Clear ISR[EOC] or ISR[JEOC].
- If the interrupt is ECH or JECH, clear ISR[ECH] or ISR[JECH].
- If the interrupt is for self-test end of algorithm, clear STSR1[WDG_EOA_C/S].

For interrupt 2:

- Read WTISR to determine which watchdog is giving the interrupt.
- Read AWORR*n* to determine which channel's data is creating the interrupt (if present).
- Clear the respective flag of WTISR [WDG*n*H / WDG*n*L] (*n* = watchdog number).
- Clear the respective flag of AWORR*n*[AWOR_CH*m*] (*n*=register, *m*=channel).

For Interrupt 3:

- Read STSR1 to determine the type of error that has occurred in self-test (ERR_S0/S1/S2/C, WDTERR, WDSERR, and so on) (if present).
- Clear the respective flag in STSR1.

The connectivity of interrupt lines at the chip level (external to SAR_ADC) determines whether software needs to read all the registers for all the interrupts or any one or a combination of any two.

72.3.10 DMA

Conversion data of any channel can be transferred from a register to system memory through Direct Memory Access (DMA). The DMA transfers can be enabled by setting DMAE[DMAEN]. When enabled, the on-chip DMA controller can receive a DMA request after the conversion of every channel by setting the respective masking bit in the DMAR*n* registers. The DMA masking registers must be programmed before starting any conversion.

The DMA request to DMA controller can be cleared at different times in two modes:

MODE 1: Clearing of DMA request on acknowledgment from DMA controller (setting DMAE [DCLR] = 0).

MODE 2: Clearing of DMA request on read to data registers (DMAE[DCLR] = 1).

Figure 435 and Figure 436 show the operation of DMA in two modes (cycle counts are for a typical case).

NOTE

The DMA should complete the read of SAR_ADC data within the throughput rate of SAR_ADC. For example, SAR_ADC can support a maximum throughput rate of 1 MSPS. Thus, DMA should complete a data read within 1 µs after assertion of DMA requests .

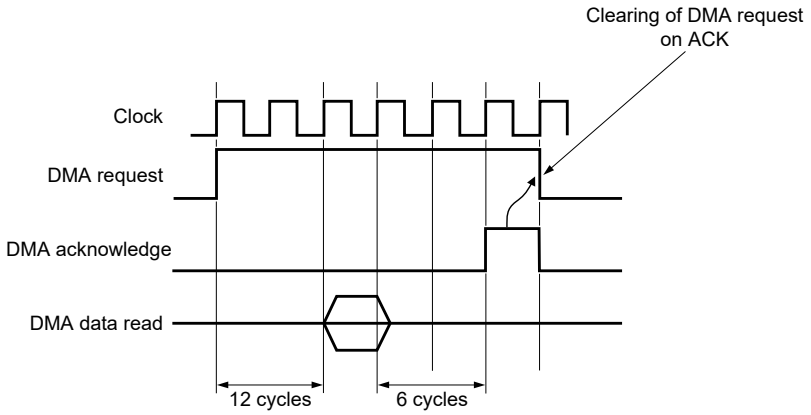


Figure 435. DMA operation MODE 1 (DMAE [DCLR] = 0)

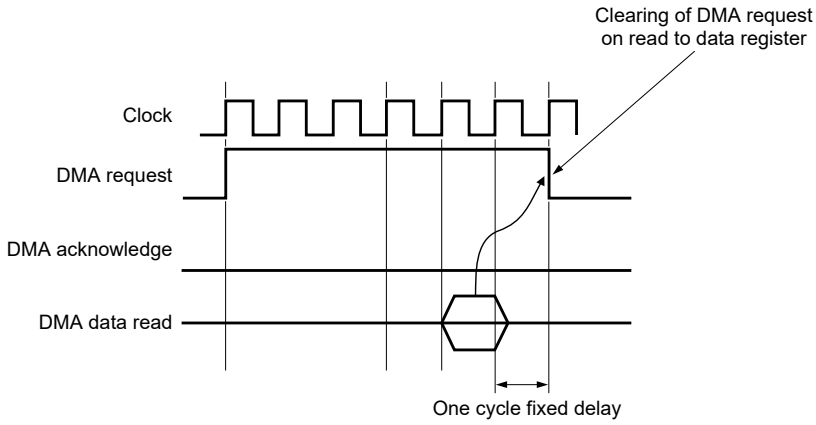


Figure 436. DMA operation MODE 2 (DMAE [DCLR] = 1)

72.3.11 Power-Down mode

The analog part along with SAR controller of the ADC can be put in Power-Down mode by setting MCR[PWDN] to shut down ADC hard macro and to stop the clock to SAR controller. After release of reset MCR[PWDN] remains one, so ADC analog module is kept in Power-Down mode by default. This state must be exited before starting any operation by resetting MCR[PWDN].

In Power-Down mode, no conversion can be started.

MCR[PWDN] can be set anytime. If a conversion is ongoing, the ADC cannot move into Power-down mode immediately. SAR_ADC enters Power-Down mode only after completion of the ongoing conversion. In Scan Operation mode the ongoing operation should be aborted manually by resetting MCR[NSTART] before or after setting MCR[PWDN].

The ADC Power-Down mode status is indicated by MSR[ADCSTATUS] when ADC enters into this mode.

If Power-Down mode is entered by setting MCR[PWDN], the process running before entry into the Power-Down mode must be restarted manually (by setting the appropriate START bit in MCR) after exiting Power-Down mode.

NOTE

Resetting MCR[PWDN] and setting MCR[NSTART] or MCR[JSTART] during the same cycle is forbidden.

72.3.12 Auto-Clock-Off mode

To reduce the power consumption during Idle state (without going into Power-down mode), an 'Auto-Clock-Off' feature can be enabled by setting MCR[ACKO]. When enabled, the internal SAR operating clock (AD_CLK) is automatically switched off during idle periods (for example, you cannot program conversions).

72.3.13 Calibration

The accuracy of ADC depends on the manufacturing precision of the devices used to build the module (for instance, capacitor size matching, amplifier gain, and so on). Due to variations during manufacturing and various runtime environmental influences (like temperature), the conversion result contains errors.

To reduce or eliminate the various error contribution effects, the ADC provides self-test and calibration features.

In the calibration process, a known reference voltage is sampled and converted under controlled conditions to determine the correction values (calibration values) for offset, gain, and capacitor mismatch.

These calibration values (except gain calibration) are used in a result post-processing step to reduce or eliminate the various error contribution effects. The gain calibration is used during sample phase to define the additional charge to be loaded in order to compensate for gain failure.

Calibration must be performed after every power-up reset (check chip-specific reset connectivity for appropriate reset applied) and whenever required in runtime operation. It is also recommended to run calibration if the operating conditions (particularly VrefH) change. Never apply functional reset during the calibration process. If applied, calibration must be rerun after exiting a reset condition; otherwise, the calibration generated values and conversion results may be unspecified.

Configuring and starting of calibration is done by programming the calibration attributes in MCR register. During calibration, the internal hardware averaging function should be enabled. Maximum averaging is recommended for best performance.

Calibration time

Clock 40 MHz. (AD_Clk == (bus_clock / 2)) (MCR[ADCLKSE] = 0)

Average samples (MCR[NRSMPL])

- 16–327 µs
- 32–654 µs
- 128–2.612 ms
- 512–10.446 ms

The calibration status is captured by status flag (MSR[CALFAIL]) to indicate if any fault is detected during calibration execution, or if the calibration has aborted at a premature state (calibration is aborted if any normal conversion is initiated during the execution of calibration, which is called premature termination). MSR[CALFAIL] = 1 means a calibration fail and a determined calibration value is out of internal defined limits. It also means that the ADC health may not be good and may not meet specifications.

The results of individual steps are also updated in CALSTAT register (CALSTAT[STAT_*n*]). The result of the last failed step is dynamically updated in the same register.

To execute the calibration algorithm, perform the following steps:

1. Wait for deassertion of both calibration and functional reset.
2. Configure desired parameters for calibration in MCR:
   - MCR[TSAMP]
   - MCR[NRSMPL]
   - MCR[AVGEN]
   - MCR[ADCLKSE]
3. Set the ADC internal operating clock to 40 MHz (MCR[ADCLKSE] = 0). (This setting assumes that ADC input clock is set to 80 MHz.)

4. Exit power-down state (MCR[PWDN] = 0).
5. Start calibration (MCR[CALSTART] = 1). Calibration starts immediately.
6. Check the status of MSR[CALBUSY] periodically. Wait until it becomes zero (alternatively, MSR[ADCSTATUS] can be monitored until it indicates an idle condition).
7. Check the fail flag (MSR[CALFAIL]) to learn the final status. If it is set to one, then calibration failed.
8. Check the pass/fail status of each calibration step in the CALSTAT register to look for failures.
9. At the end, check the status of MSR[CALIBRTD]. If calibration is successful this bit sets to one.

72.3.14 Self-test

For safety applications, it is important to check at regular intervals whether SAR_ADC is operating correctly. For this purpose, a self-test feature is provided. When self-test is enabled, SAR_ADC automatically checks its components and flags any errors it finds.

The test can be enabled to check the supply voltage (V_{DD}), reference voltage (V_{refH}), and calibrated values.

The following test algorithms have been implemented:

- Supply self-test (algorithm S): It includes the conversion of the band gap, supply, and VREF voltages. It includes a sequence of 3 test conversions (steps). The supply test conversions must be an atomic operation (that is, all supply algorithm conversions must be performed one after another with no functional conversions in between).
- Capacitive self-test (algorithm C): It includes a sequence of test steps per definition of algorithm C (See Table 629) which executes the capacitive matrix of the CDAC used for sampling and conversion.

Individual steps can take up to 1 µs at 80 MHz SAR_ADC clock frequency.

Table 629. Self-test steps

STCR3[ALG]	STCR3 [Step] (per algorithm definition)	Description	Outcome	Comment
'00'(algorithm S)	0	Supply self-test (band gap voltage)	Measures the internal band gap voltage	Band gap is considered as a constant with very little variation across temperature and process.
	1	Supply self-test (analog supply)	Measures the analog supply for SAR_ADC	
	2	Supply self-test (reference voltage high)	Measures the high reference voltage of SAR_ADC	
'01' (Reserved)	Reserved			
'10'(algorithm C)	0–11	Capacitive self-test: One of the calibration steps is being run	The difference/error of individual offset value from the previously calibrated value is being returned as SAR_ADC result that is compared with programmed value in analog watchdog register(s) to detect any fault.	The result value should be compared with analog watchdog values. See Self-test analog watchdog for more detail.

Table continues on the next page...

Table 629. Self-test steps (continued)

STCR3[ALG]	STCR3 [Step] (per algorithm definition)	Description	Outcome	Comment
			The difference or error may be caused by any runtime fault, runtime accuracy shift, or by device noise.	
'11'(algorithm S+C)	0–2 (S) 0–11 (C)	Supply for One-Shot and (Supply + Capacitive) for Scan mode only	Same as above	

In case some failure occurs in any step of the algorithm, ADC: Schedules self-testing algorithms using configuration registers Monitors the conversion result using analog watchdog registers Flags the error to a fault control unit (FCCU), if it exists.

Self-test steps can be activated from software (CPU).

ADC mode	TEST algorithm (CPU)
CPU mode	Yes • one-shot Mode • Scan Mode

NOTE

ADC self-test should be run with MCR[ADCLKSE] bit set to 1. Self-test with ADCLKSE bit set to 0 can give erroneous results.

72.3.14.1 CPU mode

In this case, test conversion works similar to normal conversion. The self-test is enabled by setting STCR2[EN].

The self-test conversions are run with the functional conversions. The sequencing of steps of the selected algorithm depends on the operating mode of normal conversions as selected by MCR[MODE].

In **One-Shot mode**, if self-test is enabled, only one step of the selected self-testing algorithm is executed at the end of the chain. The step number and algorithm to be executed are programmed in STCR3. So, in One-Shot mode the sequence is as follows:

1. Program NCMR*n* to select channels to be converted for normal conversion.
2. Program MCR[MODE] = 0 to select One-Shot mode.
3. Program sampling duration values in STCR1[INPSAMP*n*] field. (x = S,C).
4. Select the self-testing algorithm in STCR3[ALG] and step STCR3[MSTEP]. Default is Algorithm S step 0.
5. Enable self-testing by setting STCR2[EN].
6. Start the normal conversion by setting MCR[NSTART].
7. All normal conversions are executed as usual.
8. At the end of all the normal conversions in the chain, step number programmed in STCR3[MSTEP] field of self-testing algorithm selected by STCR3[ALG] is executed, similar to a normal functional channel.

9. On the receiving end of the conversion for test channel, the result is written in STDR1[TCDATA], and STDR1[VALID] is set. ISR[EOC], ISR[ECH] are set, and so is STSR1[ST_EOC].
10. ADC returns to Idle state.

For example: Channels A-B-C-D-E-F-G-H are present in the device where channels B-D-E are to be converted in One-Shot mode. At the end of conversion for channels B-D-E, self-test conversion is done and ISR[ECH] and ISR[EOC] are set. The sequence is B→D→E→self-test step.

MSR[NSTART] is automatically set when the normal conversion starts and is reset at the end of conversion for test channel.

In **Scan mode**, consecutive steps of the selected self-test algorithm are converted continuously at the end of each chain of normal conversions. The number of channels converted at the end of each chain is one (except for Algorithm S, in which all the steps are performed at once without any functional conversion interleaved). So, in Scan mode the sequence is as follows.

1. Program NCMRx to select channels to be converted for normal conversion.
2. Program MCR[MODE] = 1 to select Scan mode.
3. Select the self-testing algorithm in STCR3[ALG]. By default, all algorithms (supply and capacitive) are selected that is, all algorithms will be executed step-by-step one after the other.
4. Enable self-testing by setting STCR2[EN].
5. Start the normal conversion by setting MCR[NSTART].
6. All normal conversions are executed as usual.
7. At the end of chain of the normal conversions (assuming default value of STCR3[ALG]) all steps of Algorithm S are performed (as Algorithm S is always atomic). MSR[SELF_TEST_S] is set.
8. On receiving end of conversion of test channel for last step of Algorithm S:
   - The result is written in STDR1[TCDATA] and STDR1[VALID] is set.
   - At the same time, MSR[SELF_TEST_S] is reset.
   - For Step 1, the integral part and fractional part are written in STDR2[IDATA] and STDR2[FDATA].
   - Also, EOC and ECH bits are set in ISR register and ST_EOC is set in STSR1.
9. The next chain of normal conversion then starts.
10. At end of the normal conversion chain, Step0 of algorithm C is executed.
11. On the receiving end of conversion of test channel for Step0 of algorithm C:
    - The result is stored in STDR1[TCDATA] and STDR1[VALID] is set (if MCR[OVERWR] is set).
    - Also, EOC and ECH bits are set in ISR register and ST_EOC is set in STSR1.
12. Then the next chain of normal conversion starts.
13. At end of the normal conversion chain, Step1 of algorithm C is executed.
14. This process continues for all the steps of all three algorithms.
15. ADC returns to Idle state when MCR[NSTART] is written to 0.

For example: channels A-B-C-D-E-F-G-H are present in the device where channels B-D-E are to be converted in Scan mode. At the end of every conversion chain for channels B-D-E, one step of self-test conversion is done and ISR[ECH] and ISR[EOC] are set.

The sequence is:

1. B→D→E→ST-S0→ST-S1→ST-S2→B→D→E→ST-C0→B→D→E→ST-C1→B→D→E→ST-C2→ . . . B→D→E→ST-C11→B→D→E→ST-S0 . . .
2. ST-Sx - Self-test supply step x (x = 0,1,2)
3. ST-Cx - Self-test capacitive step x (x = 0 to 11)

MCR[NSTART] is automatically set when the normal conversion starts.

NOTE

The self-test behavior is still the same if normal conversion is started by software (by setting MCR[NSTART]) instead of by external trigger.

In case of Injected Conversion modes, test channel conversion is not performed. It is performed only during normal conversions.

If during a test channel conversion, injection conversion begins, then the test conversion is aborted (as with a normal functional channel) and Injected Conversion modes are completed. After Injected Conversion modes are completed, the test conversion resumes from the step at which it was aborted. In this case, MSR[SELF_TEST_S] remains high during Injected Conversion mode.

For self-testing, the mode bit must be programmed at least one cycle before setting MCR[NSTART], and should not be changed thereafter until the conversion is finished or terminated.

72.3.14.2 Abort and abort chain for self-testing channel

Setting MCR[ABORT] during self-test channel conversion has no effect.

In One-Shot Operation mode, if MCR[ABORTCHAIN] is set when test channel conversion is ongoing, the test channel conversion is aborted and ISR[ECH] is set. In this case, ISR[EOC] for test channel is not generated.

In Scan Operation mode, if MCR[ABORTCHAIN] is set when test channel Step n is ongoing, the test channel Step n is aborted and next the chain conversion starts. At the end of this chain, Step n conversion is performed again. (In the case of Algorithm S, the full algorithm is executed again).

72.3.14.3 Self-test analog watchdog

The ADC digital interface also provides monitors for the conversion data for self-test algorithms. The analog watchdogs are used to determine whether the result of conversion for self-test algorithms is in the range of a particular guard area. For this purpose, separate self-test analog watchdog registers have been provided for each algorithm.

After the conversion of each step of an algorithm, a comparison is performed between the converted value and the threshold values if the analog watchdog feature is enabled by setting STAW*n*R[AWDE]. If the converted value does not lie between the upper and lower threshold values specified by Analog Watchdog Register of the particular algorithm, corresponding error bit STSR1[ERR_S*n*] is set and the step number in which the error occurred is updated in STSR1[STEP_C] (in case of algorithm C). Also, erroneous data is written in the STSR4[DATA*n*] field. The STSR1[ERR_S*n*] fields generate an interrupt if enabled by the corresponding mask bit in STCR2. The fault indication is also given to the fault control unit, if present, through critical and non-critical fault lines, so that necessary action can be taken at the chip level. Configuration in STCR2 register decides the choice between critical and noncritical type.

The analog watchdog feature works differently in algorithm S. As already mentioned, algorithm S is always an atomic operation. So, separate error bits are provided in STSR1 for each Step of algorithm S to avoid overwriting if error occurs in more than one Step. Hence, there are separate mask bits for each step in STCR1. For the same reason, separate fields exist in status registers (STSR2 and STSR3) to store erroneous data for each step.

In step0 of the supply algorithm, ADC measures the band gap voltage (+1.15 V), which is assumed to be stable. The conversion result of Step0 is measured with high (THRH) and low (THRL) thresholds defined in STAW0R if enabled (STAW0R[AWDE] = 1). STSR1[ERR_S0] is set to one if either threshold is violated. After set, it is cleared by writing a one to it.

For algorithm S Step1, a fixed-point division is performed, which takes about 26 cycles after EOC for Step1 to receive the divided value. It is used for analog watchdog checks. So, the value to be compared for Step1 contains the integer as well as fractional part—thus, two registers (STAW1AR and STAW1BR) are provided.

The comparison is done first for the integer part using the threshold values programmed in STAW1AR. If the integer part lies in the range, the fractional part comparison is skipped; otherwise it is compared to the values programmed in STAW1BR. The following table summarizes this feature for Step1.

Table 630. Algorithm S (Step1) threshold comparison

STDR2[IDATA] (integer part)	STDR2[FDATA] (fractional part)	STSR1[ERR_S1]
> STAW1AR[THRH]	Any value	Set
< STAW1AR[THRL]	Any value	Set
== STAW1AR[THRH]	> STAW1BR[THRH]	Set
== STAW1AR[THRL]	< STAW1BR[THRL]	Set

For algorithm S Step2, VREF/VREF is measured in order to check the integrity of the sampling signal. For this particular conversion, no higher threshold value is required, as the ideal value is FFFh. Only the lower threshold value is programmed in STAW2R.

For algorithm C, a separate register is provided for Step0. In Step0, an offset for other steps is measured. The converted data is compared with the threshold values provided by STAW5R if STAW4R[AWDE] is set. For other steps, this offset is subtracted from converted data before performing watchdog checks.

The recommended analog watchdog threshold values are set according to the expected and tested results with a clean environment (noise introduction from external environment minimized) for ADC operation. In a noisier setup, running the self-test with these analog watchdog values may cause failures. To overcome these failures, you can relax the threshold values if the needs of the application can tolerate the existing noise level. The more the threshold values are relaxed, the more the magnitude of true error that can pass. For example, looking at the values for the STAW4R and STAW5R registers, THRH of 010h (16d) and THRL of FF0h (–16d), means that the algorithm C fails for an error of magnitude greater than (16/8 = 2) 2LSB@12b. This is the recommended value for a clean environment.

Similarly, THRH of 020h (32d) and THRL of FE0h (–32d) means that the algorithm C fails for an error of a magnitude greater than 4LSB@12b. Thus to pass the self test in a noisy environment where the application can live with the noise level, the thresholds can be relaxed from ±40 to ±64 or as needed. However, it is preferable that the environment is cleaned up to match the expectation for better operation instead of relaxing the thresholds. Relaxing the thresholds too much (more than ±64) leads to a false pass, and is not recommended for safety applications.

Relaxation of the self-test thresholds due to noise does not directly relate to normal use conversions, and it is not possible to tell what the noise impact to those measurements is with respect to the threshold relaxation. The noise is application-specific and not under control/judgment of the chip. In this scenario, not passing with the default threshold values is caused by external noise and not by the constituent capacitors. Limits are relaxed knowing that there is noise in the system that rides on top of the actual capacitor error, and the magnitude of the noise component is so high that limits should be relaxed as a result. Thus in reality whenever tests are run environment noise will be there, and if there is any true shift of the capacitor error noise it will be added to this environment noise, and should result in a violation of the relaxed threshold limits. The self-test is designed to catch and flag runtime and catastrophic errors. It is not able to, and not meant to, test for specification limits.

72.3.14.4 S1 algorithm: calculated voltage comparison method

The software can implement a method of calculating S1 algorithm STDR2[IDATA], STDR2[FDATA], THRHLRn[THRH] and THRHLRn[THRL] voltage values, then perform a comparison to the calculated THRH and THRL voltages.

The steps involved in this method are as follows:

1. Calculate the high threshold voltage:

   (STAW1AR[THRH] + (STAW1BR[THRH] / 4096)) × $V_{BANDGAP}$

2. Calculate the low threshold voltage:

   (STAW1AR[THRL] + (STAW1BR[THRL] / 4096)) × $V_{BANDGAP}$

3. Calculate the converted results to voltage:

   (STDR2[IDATA] + (STDR2[FDATA] / 4096)) × $V_{BANDGAP}$

4. Compare result from step 3 to THRH and THRL. S1 algorithm passes if low threshold voltage ≤ converted voltage ≤ high threshold voltage.

72.3.14.5 Watchdog timer

The watchdog timer is an additional check which monitors the sequence of the self-testing algorithm implemented. It also verifies that the algorithm is completed within a safe time period. The watchdog timers can be enabled for CPU conversions.

Each algorithm has a different watchdog timer which runs independently of the other. The watchdog timer for a particular algorithm can be enabled by setting STAW*n*R[WDTE]. The safe time value can be programmed in STBRR[WDT] (default value is 10 ms assuming an 80 MHz clock).

The safe time is measured starting from Step0 of the algorithm (including all normal chain conversions in between) to the point where Step0 of the same algorithm starts again.

The programming sequence is as follows:

1. Program NCMR*n* to select the channels on which conversion needs to be performed for normal conversion in Scan Operation mode (MCR[MODE] = 1).
2. Select the self-testing algorithm in STCR3[ALG]. By default, all algorithms (supply and capacitive) are selected, that is, all algorithms will be executed step-by-step one after the other.
3. Enable self-testing channel by setting STCR2[EN].
4. Program safe period value in STBRR[WDT].
5. Enable watchdog timer by setting STAW*n*R[WDTE]. Assume setting of STAW*n*R[WDTE] to be 't0'. It is important to do all the programming first and then enable STAWnR[WDTE], as the safe time check is also performed between setting STAWnR[WDTE] and the start of Step0. This step checks that the algorithm has started within the safe time.
6. Start the normal conversion by setting MCR[NSTART].
7. At the end of the first conversion chain, three steps (Step 0/1/2) of algorithm S are executed in sequence. Let us assume the start of Step0 to be 't1'.
8. After completion of algorithm S, conversion of the next chain is performed.
9. At the end of the conversion chain, the first Step (Step0) of algorithm C is performed.
10. In this way, one Step of algorithm C is performed after completion of one conversion chain until all the steps have been completed. Steps are executed in sequence (Step 0, 1, 2, . . 11).
11. After the last step of algorithm C, another chain conversion is executed. At the end of this chain conversion, Step0 of algorithm S is started again, thus repeating the whole sequence. Let's assume this time (starting of Step0 of Supply algorithm) to be 't2'.
12. For S algorithm, if (t1–t0) > Safe Period or (t2–t1) > Safe Period, the watchdog timer flags an error and STSR1[WDTERR] is set. A fault is asserted on the specified module port as per criticality configuration (Critical or Non-critical), and an interrupt is also generated if enabled by STCR2[MSKWDTERR]. Otherwise, the watchdog timer counter is reset and starts again to monitor the same for the next sequence.
13. A similar sequence is followed for watchdog timers for algorithm C.

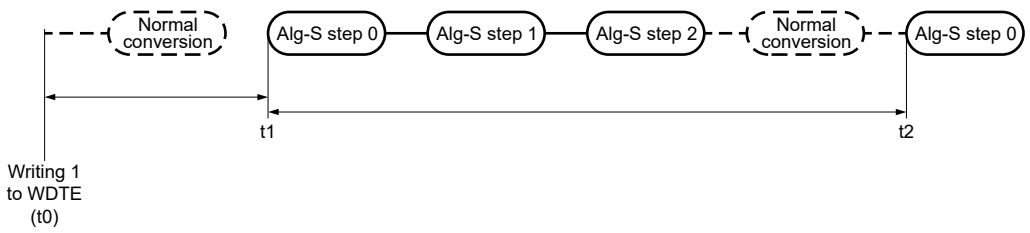


Figure 437. Watchdog timer monitor for Alg-S

NOTE

You must ensure that watchdog timers are disabled for self-test algorithms which are not executed.

72.3.14.5.1 Watchdog sequence checking

The watchdog timer also incorporates sequence checking features that check that the steps of a particular algorithm are in correct order. If steps are not in correct order, an error is flagged by setting STSR1[WDSERR]. A fault is indicated by an interrupt flag if enabled by STCR2[MSKWDSERR]. A fault is also asserted on the specified module port as per criticality configuration (Critical or Non-critical) and an interrupt generated if enabled by STCR2[MSKWDSERR].

A watchdog sequence error is flagged in the following cases:

- If steps of any algorithm are not executed in proper order.
- If abort chain occurs during test channel conversion. That step has to be repeated at the end of the next chain. This gives a sequence error when test channel conversion starts again.
  - Exception: If abort chain occurs during the last step of Algorithm S, then a sequence error is not flagged as the whole algorithm has to be repeated again.

If injected conversion occurs during the test channel conversion, a watchdog sequence error is NOT flagged although the ongoing step number is aborted and is repeated again.

NOTE

Watchdog timer feature is applicable only for Scan Operation mode and not for One-Shot Operation mode.

72.3.14.6 Baud rate control for test channel

Baud rate control defines the scheduling of the test channel between the normal conversion chain. STBRR[BR] specify the scheduling rate.

By default, if a test channel is enabled, one step of the selected algorithm is executed after every chain of normal conversion. The bandwidth consumed by a test channel depends on the number of channels in a normal chain. For example, if we have 50 normal conversions in a chain, then a test channel consumes only 2% of total bandwidth. But if the number decreases to just 3 channels, then the bandwidth consumed by a test channel is 25%, which is significant (and may not be desirable as it slows down the normal conversion rate).

STBRR[BR] provides flexibility by scheduling the test channel conversion to be performed not at the end of every chain, but at the end of BR+1 number of chains. For example, if BR = 5, a single step of the selected algorithm for a test channel is performed after six chain conversions, then the next step is performed at the end of the next six chain conversions, and so on. By default, the value of BR is zero.

NOTE

This feature is applicable only for Scan Operation mode of operation and not for One-Shot Operation mode. STBRR[BR] should be set to zero for One-Shot Operation mode.

To use the baud rate feature in Scan Operation mode, NCMR should have a non-zero value.

72.3.14.6.1 Abort chain when baud rate is non-zero

As already described, for a non-zero value of STBRR[BR], the test channel conversion is performed at the end of (BR+1) number of chains. If an abort chain occurs during the chain in which test conversion is scheduled to be converted, then test conversion is done after the next (BR+1) number of chains.

For example, if STBRR[BR] is programmed to two, the sequence will be two normal chains (without any test channel conversion) followed by a chain with test channel converted at the end. Now, if an abort chain occurs during the first two chains, it is treated as a normal chain abort and test channel is converted at the end of the third chain only (as is the case without any abort chain). But if an abort chain occurs during the third chain in which the test channel is scheduled to be converted, then the test channel is converted after the next three chains, that is at the end of sixth chain (counting from the beginning).

72.3.15 Offset and gain values

Offset and gain errors in ADC are compensated by its calibration mechanism. However, there could be a source of errors outside the ADC module (for example, at system and/or board level). These errors can potentially contribute to ADC static errors and lower the overall TUE observed when an application runs. To compensate for these errors, there is a set of offset and gain values that you can program in User OFFSET and Gain (USROFSGN). By programming this register appropriately, the complete system's TUE can be enhanced (calibrated), when impacted by errors other than ADC itself.

These values (GAIN_USER and OFFSET_USER) must be stored in 2's complement format as described in the register section (see User OFFSET and Gain (USROFSGN)). The values stored here are being considered by SAR_ADC engine and being computed/adjusted for every output. Using this register to compensate offset and gain is described below.

72.3.15.1 Writing for 12-bit ADC

72.3.15.1.1 Gain compensation (GAIN_USER)

The ADC can compensate a maximum of ±63 LSB @12b gain by programming GAIN_USER field in USROFSGN register. The field is in 2's complement format and stored with an 8x multiplication. Thus, if a value of ± Y is stored in this field, it leads to a final result adjusted with ±(Y/8) LSB from its unprogrammed version. It is required to keep the 3-LSBs zero to avoid any fractional subtraction/multiplication related errors.

ADC_OUTPUT_NEW = ADC_OUTPUT_OLD × (1 + (GAIN_USER / (8 × 4096))).

Example:

For GAIN_USER = 0, result is ADC_OUTPUT_NEW = ADC_OUTPUT_OLD = P

For GAIN _USER = +Y, result is ADC_OUTPUT_NEW = P × (1 + (Y / 32768))

For GAIN _USER = −Y, result is ADC_OUTPUT_NEW = P × (1 − (Y / 32768))

72.3.15.1.2 Offset compensation (OFFSET_USER)

The ADC can compensate a maximum of ±15LSB @12b offset by programming OFFSET_USER field in USROFSGN register. The field is in 2's complement format and stored with an 8x multiplication. Thus, if a value of ±X is stored in this field, it leads to a final result adjusted with ±(X/8) from its unprogrammed version. It is required to keep the 3-LSBs zero to avoid any fractional subtraction-related errors.

ADC_OUTPUT_NEW = ADC_OUTPUT_OLD − (OFFSET_USER/8)

Example:

For OFFSET_USER = 0, result is ADC_OUTPUT_NEW = ADC_OUTPUT_OLD = K

For OFFSET_USER = +X, result is ADC_OUTPUT_NEW = K − (X / 8)

For OFFSET_USER = −X, result is ADC_OUTPUT_NEW = K − (−X / 8) = K + (X / 8)

72.3.15.1.3 Using OFFSET_USER and GAIN_USER together

When both the fields are programmed together, gain is computed first and then offset is computed.

ADC_OUTPUT_NEW = (ADC_OUTPUT_OLD × (1 + (GAIN_USER / (8 × 4096)))) − ((OFFSET_USER / 8)).

Example:

For OFFSET_USER = 0 and GAIN_USER = 0, result is ADC_OUTPUT_NEW = ADC_OUTPUT_OLD = M

For OFFSET_USER = X and GAIN_USER = Y, result is ADC_OUTPUT_NEW = (M x (1 + (Y / 32768))) − (X / 8)

NOTE

Negative values are stored in 2's complement in these fields. Example:

GAIN_USER = 1F0h means +496d, and 210h means −496d

OFFSET _USER = 70h means +112d, and 90h means −112d

72.4 External signals

Also see the Block diagram section.

Table 631. External signals

Signal description	Direction	Width	Function
ADC_CH0 – ADC_CH(*n*-1)	Input	*n* (number of channels)	Analog inputs
Injection start trigger	Input	1	Control
Normal start trigger	Input	1	Control

72.5 Initialization

ADC must be programmed properly to function correctly. This section describes the basic programming sequences for applications to use. Application software may use them in combination by sequencing them carefully.

72.5.1 Running calibration

Calibration is the first step that should be run after every power-on reset. ADC has two types of reset:

- Calibration
- Functional

Calibration reset is used for registers that store internal calibration settings and values. This reset is done to store them during functional reset. Functional reset may be applied many times during an application for various reasons and calibration may not be required each time, so storing the settings and values in a higher level of reset avoids recalibration with every functional reset.

1. Wait for deassertion of functional reset.
2. Configure SAR controller operating clock (MCR[ADCLKSE] = 0).
3. Bring ADC out of Power-down state (MCR[PWDN] = 0).
4. Configure desired calibration settings (default values kept for highest accuracy maximum time).
   - MCR[TSAMP]: Sample time for calibration conversion
   - MCR[NRSMPL]: Number of samples in averaging
   - MCR[AVGEN]: Averaging function enable in calibration
5. Run calibration by writing a one to MCR[CALSTART].
6. Check calibration run status in MSR[CALBUSY]—wait until MSR[CALBUSY] = 0; alternatively, MSR[ADCSTAT] can be used to check status.
7. Check calibration pass/fail status in MSR[CALFAIL] field. If MSR[CALFAIL] = 1 then calibration failed. Detailed status can be checked in CALSTAT.

72.5.2 Running normal conversion

Normal conversion mode should usually be used to convert analog samples most of the time in an application, unless there is a special need. There are other modes available for special requirements as described in later sections of this document. Normal conversions can be initiated by software trigger or by hardware trigger. Calibration should be done before running any conversion to ensure an accurate result.

1. Configure channel mask registers NCMR*n*, for different channel groups.
2. Configure desired sample time in CTR*n*, for different channel groups.
3. Configure interrupt mask registers, IMR for global and CIMR*n*, for different channel groups.
4. Configure DMA registers, DMAE, and DMAR*n*, for different channel groups.

5. Configure pre-sampling registers, if needed, PSCR and PSR*n*, for different channel groups (if needed).
6. Configure watchdogs in WTIMR
7. Configure One-Shot or Scan mode in MCR[MODE] and other settings (MCR[WLSIDE], MCR[OWREN], and so on).
8. Start conversion by writing one to MCR[NSTART].
9. Check EOC status in ISR or wait for interrupt.
10. On interrupt read the data register, or DMA will read out data if configured.
11. Status of conversion for individual channel can be seen in CEOCFR*n*.
12. Clear the registers by writing one to the fields.
13. Steps 9 to 12 will continue in loop until conversion finishes or is terminated.

72.5.3 Running injected conversion

Injected conversion has a higher priority than normal conversion and it runs in One-Shot mode only. It can be started in idle condition or when normal conversion is in process. The configuration sequence is almost similar to normal conversion above, except for the registers discussed in the following steps.

1. Channel mask register JCMR*n*, is used for different channel groups.
2. Configure other channel attributes as done in normal conversion.
3. Start conversion by writing 1b to MCR[JSTART].
4. Check status of ISR[JEOC] or wait for interrupt.
5. Read out conversion data and clear interrupts as done in normal conversion.
6. Configure channel mask registers NCMR*n* for different channel groups.
7. Configure desired sample time in CTR*n* for different channel groups.
8. Configure interrupt mask registers, IMR for global and *n*, for different channel groups.
9. Configure DMA registers, DMAE and DMAR*n*, for different channel groups.
10. Configure presampling registers, if needed, PSCR and PSR*n*, for different channel groups (if needed).
11. Configure watchdogs in WTIMR.
12. Configure One-Shot or Scan mode in MCR[MODE] and other settings (MCR[WLSIDE], MCR[OWREN], and so on).
13. Start conversion by writing 1b to MCR[NSTART].
14. Check status of ISR[EOC] or wait for interrupt.
15. On interrupt, read the data register or DMA will read out data if configured.
16. Status of conversion for an individual channel can be seen in CEOCFR*n*.
17. Clear the registers by writing one to the bits.
18. Steps 9 to 12 will continue in loop until conversion finishes or is terminated.

72.5.4 Running self-test

Self-test conversions run after a programmable number of normal conversion chains. To run normal conversions, additional configurations for self-test need to be implemented before starting normal conversion (as shown in Self-test). Also, you may need to configure the self-test watchdog, timer, and baud rate as described in Self-test analog watchdog.

72.6 SAR_ADC register descriptions

Addresses outside the space defined in this section, as well as unused addresses between registers, are reserved. Do not read from or write to reserved spaces.

72.6.1 SAR_ADC memory map

ADC1 base address: 4453_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Main Configuration (MCR)	32	RW	0000_3901
4	Main Status (MSR)	32	RW	0000_0001
10	Interrupt Status (ISR)	32	RW	0000_0000
14	Channel Pending 0 (CEOCFR0)	32	RW	0000_0000
18	Channel Pending 1 (CEOCFR1)	32	RW	0000_0000
20	Interrupt Mask (IMR)	32	RW	0000_0000
24	Channel Interrupt Mask 0 (CIMR0)	32	RW	0000_0000
28	Channel Interrupt Mask 1 (CIMR1)	32	RW	0000_0000
30	Watchdog Threshold Interrupt Status (WTISR)	32	RW	0000_0000
34	Watchdog Threshold Interrupt Mask (WTIMR)	32	RW	0000_0000
40	DMAE (DMAE)	32	RW	0000_0000
44	DMA 0 (DMAR0)	32	RW	0000_0000
48	DMA 1 (DMAR1)	32	RW	0000_0000
60	Analog Watchdog Threshold 0 (THRHLR0)	32	RW	0FFF_0000
64	Analog Watchdog Threshold 1 (THRHLR1)	32	RW	0FFF_0000
68	Analog Watchdog Threshold 2 (THRHLR2)	32	RW	0FFF_0000
6C	Analog Watchdog Threshold 3 (THRHLR3)	32	RW	0FFF_0000
80	Presampling Control (PSCR)	32	RW	0000_0000
84	Presampling 0 (PSR0)	32	RW	0000_0000
88	Presampling 1 (PSR1)	32	RW	0000_0000
94	Conversion Timing 0 (CTR0)	32	RW	0000_0014
98	Conversion Timing 1 (CTR1)	32	RW	0000_0014
A4	Normal Conversion Mask 0 (NCMR0)	32	RW	0000_0000
A8	Normal Conversion Mask 1 (NCMR1)	32	RW	0000_0000
B4	Injected Conversion Mask 0 (JCMR0)	32	RW	0000_0000
B8	Injected Conversion Mask 1 (JCMR1)	32	RW	0000_0000
C0	User OFFSET and Gain (USROFSGN)	32	RW	0000_0000
C8	Power Down Exit Delay (PDEDR)	32	RW	0000_0000
100 - 11C	Precision Channel n Data (PCDR0 - PCDR7)	32	R	0000_0000
180 - 19C	Internal Channel n Data (ICDR32 - ICDR39)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
280	Analog Watchdog Threshold 4 (THRHLR4)	32	RW	0FFF_0000
284	Analog Watchdog Threshold 5 (THRHLR5)	32	RW	0FFF_0000
288	Analog Watchdog Threshold 6 (THRHLR6)	32	RW	0FFF_0000
28C	Analog Watchdog Threshold 7 (THRHLR7)	32	RW	0FFF_0000
2B0	Channel Watchdog Select 0 (CWSELR0)	32	RW	0000_0000
2C0	Channel Watchdog Select 4 (CWSELR4)	32	RW	0000_0000
2E0	Channel Watchdog Enable 0 (CWENR0)	32	RW	0000_0000
2E4	Channel Watchdog Enable 1 (CWENR1)	32	RW	0000_0000
2F0	Analog Watchdog Out of Range 0 (AWORR0)	32	RW	0000_0000
2F4	Analog Watchdog Out of Range 1 (AWORR1)	32	RW	0000_0000
340	Self-Test Configuration 1 (STCR1)	32	RW	1800_A000
344	Self-Test Configuration 2 (STCR2)	32	RW	0000_0005
348	Self-Test Configuration 3 (STCR3)	32	RW	0000_0300
34C	Self-Test Baud Rate (STBRR)	32	RW	0005_0000
350	Self-Test Status 1 (STSR1)	32	RW	0000_0000
354	Self-Test Status 2 (STSR2)	32	R	0000_0000
358	Self-Test Status 3 (STSR3)	32	R	0000_0000
35C	Self-Test Status 4 (STSR4)	32	R	0000_0000
370	Self-Test Data 1 (STDR1)	32	R	0000_0000
374	Self-Test Data 2 (STDR2)	32	R	0000_0000
380	Self-Test Analog Watchdog 0 (STAW0R)	32	RW	0B60_093F
384	Self-Test Analog Watchdog 1A (STAW1AR)	32	RW	0001_0001
388	Self-Test Analog Watchdog 1B (STAW1BR)	32	RW	0B33_0586
38C	Self-Test Analog Watchdog 2 (STAW2R)	32	RW	0000_0FF5
390	Self-Test Analog Watchdog 3 (STAW3R)	32	R	0000_0000
394	Self-Test Analog Watchdog 4 (STAW4R)	32	RW	0028_0FD8
398	Self-Test Analog Watchdog 5 (STAW5R)	32	RW	0028_0FD8
39C	Calibration Status (CALSTAT)	32	R	0000_0000
3A0	Calibration Configuration 0 (CALCFG0)	32	RW	371B_C4C8

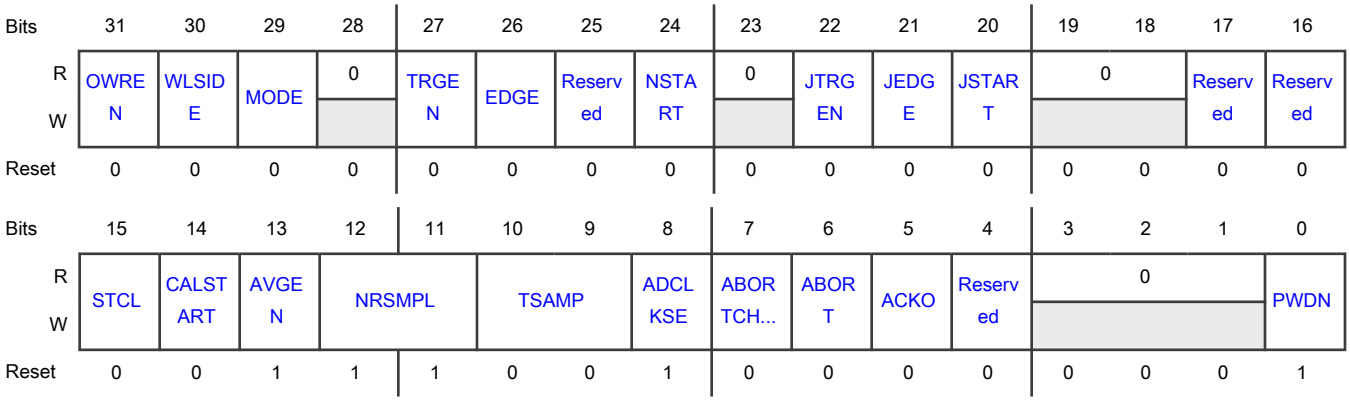
72.6.2 Main Configuration (MCR)

Contains fields for performing the main configuration.

Offset

Register	Offset
MCR	0h

Diagram



Fields

Field	Description
31 OWREN	Overwrite Enable Specifies whether the latest conversion overwrites current value in data registers. 0 - Older valid conversion data is not overwritten by newer conversion data 1 - Newer conversion result is always overwritten, irrespective of the validity of older conversion data
30 WLSIDE	Write Left Side Selects whether conversion data is left or right aligned. 0 - Write right-aligned conversion data (from 11 to 0) 1 - Write left-aligned conversion data (from 15 to 4)
29 MODE	Normal Scan Mode Select Selects between One-Shot or Scan Operation modes; MODE must be changed in ADC's Idle condition (check MSR [ADCSTATUS] field). In One-Shot Operation mode, configuration for one chain conversion in Normal Conversion mode takes place. In Scan Operation mode, configuration for continuous chain conversion in Normal Conversion mode takes place. When the chain conversion is finished, a new scan starts immediately. 0 - One-Shot Operation mode 1 - Scan Operation mode
28	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
27 TRGEN	External Trigger Enable Enables external triggers to start conversions. 0 - External trigger is disabled 1 - Enables the external trigger to start a conversion
26 EDGE	Trigger Edge Select Selects which edge of the external trigger starts a conversion. This field is only valid if MCR[TRGEN] = 1b. 0 - Falling edge is trigger 1 - Rising edge is trigger
25 —	Reserved
24 NSTART	Normal Conversion Start Starts normal conversion in One-Shot or Scan Operation mode. In Scan Operation mode, this field remains high after setting. Clearing this field in Scan Operation mode causes the current chain conversion to finish, then stop the scan. In One-shot Operation mode, this field is cleared immediately after the conversion starts. Setting this field during ongoing conversions keeps this field high until the requested conversion is started (or pending during Injected Operation mode).
23 —	Reserved
22 JTRGEN	Injection External Trigger Enable When set, enables the external trigger to start injected conversions. This field must be set if the external trigger feature is needed to start injected conversions. 0 - Injected conversion not started by external trigger 1 - Injected conversion started by external trigger
21 JEDGE	Injection Trigger Edge Selection Selects which edge of the external trigger starts an injected conversion. This field is only valid if MCR[JTRGEN] = 1b. 0 - Falling edge is trigger 1 - Rising edge is trigger
20 JSTART	Start Injection Conversion

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Starts injection conversion for analog channels to be converted. Clearing this field has no effect since the injected chain conversion cannot be interrupted. If this field is set during on ongoing injected conversion, the field will remain set until the new conversion starts.
19-18 —	Reserved
17 —	Reserved
16 —	Reserved
15 STCL	Self-Testing Configuration Lock Locks the self-test configuration registers (write-protected): 1. Self-Test Configuration 1 (STCR1) 2. Self-Test Configuration 2 (STCR2) 3. Self-Test Configuration 3 (STCR3) 4. Self-Test Baud Rate (STBRR) 5. Self-Test Analog Watchdog 0 (STAW0R) 6. Self-Test Analog Watchdog 1A (STAW1AR) 7. Self-Test Analog Watchdog 1B (STAW1BR) 8. Self-Test Analog Watchdog 2 (STAW2R) 9. Self-Test Analog Watchdog 3 (STAW3R) 10. Self-Test Analog Watchdog 4 (STAW4R) 11. Self-Test Analog Watchdog 5 (STAW5R) It can be used only in CPU and Scan operation modes. It is cleared by reset only. 0 - Not locked 1 - Locked
14 CALSTART	Calibration Start Starts calibration. It is self-clearing, and is cleared by hardware when calibration completes or is terminated prematurely. 0 - No effect 1 - Start calibration
13	Average Enable Allows averaging of calibration time.

Table continues on the next page...

Table continued from the previous page...

Field	Description
AVGEN	0 - Disable 1 - Enable (default)
12-11 NRSMPL	Number of Averaging Samples Selects the number of averaging samples to be used during Calibration. 00 - 16 01 - 32 10 - 128 11 - 512
10-9 TSAMP	Sample Time for Calibration Specifies the sample time of calibration conversions. 00 - 22 cycles of AD_CLK (default) 01 - 8 cycles of AD_CLK 10 - 16 cycle of AD_CLK 11 - 32 cycle of AD_CLK
8 ADCLKSE	Analog Clock Frequency Select Selects the clock frequency of the AD_CLK signal. It can be written in Power-down mode only. 0 - AD_CLK frequency is half 1 - AD_CLK frequency is equal to bus clock frequency
7 ABORTCHAIN	Abort Conversion Chain Aborts the chain conversion that is in process. When the chain aborts, this field is reset by hardware. Setting this field in idle state (for example, no normal/injected conversion is in process) has no effect. In this case, the field is reset immediately. During a CTU conversion, this field cannot be programmed. **NOTE** You must not apply Abort and Abort Chain simultaneously in SCAN mode operation. 0 - Chain conversion aborted or is currently not running 1 - Abort current chain conversion
6 ABORT	Abort Conversion Aborts any ongoing conversion and starts the next one. When the ongoing conversion aborts, this field is reset by hardware. This field is not written while the self-test channel conversion is in process. **NOTE** You must not apply Abort and Abort Chain simultaneously in SCAN mode operation.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Channel conversion has been aborted, or channel conversion is not currently running 1 - Abort current channel conversion
5 ACKO	Auto-Clock-Off Mode Enable Enables the auto-clock-off feature. 0 - Auto-Clock-Off feature is disabled 1 - Auto-Clock-Off feature is enabled
4 —	Reserved
3-1 —	Reserved
0 PWDN	Power-Down Enable Enables entry into Power-Down or Idle modes. 0 - When ADC status is in Power-down mode (MSR[ADCSTATUS] = 001b), start ADC transition to IDLE mode 1 - Request to enter Power-down mode

72.6.3 Main Status (MSR)

Contains main status fields.

Offset

Register	Offset
MSR	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CALIBRTD	CALFAIL	CALBUSY	0				NSTART	JABORT	0		JSTART	0	SELF_TE...	Reserved	0
W		W1C														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHADDR							0			ACKO	0		ADCSTATUS		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31 CALIBRTD	Calibration Status Updates after running high accuracy calibration (see Calibration). 0 - Uncalibrated or calibration unsuccessful 1 - Calibrated or calibration successful
30 CALFAIL	Calibration Failed Indicates the status of calibration. It is set automatically if any of the calibration step fails during the calibration process or calibration is aborted prematurely (by any normal conversion). Cleared by writing one, or on restarting calibration by writing one to MCR[CALSTART]. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Calibration passed (must be checked with CALBUSY = 0b) 1 - Calibration failed When writing 0 - No effect 1 - Clear the flag
29 CALBUSY	Calibration Busy Indicates whether SAR_ADC is executing calibration. Set when MCR[CALSTART] is set. 0 - ADC is ready for use 1 - ADC is busy in a calibration process
28-25 —	Reserved
24 NSTART	Normal Conversion Status Signals whether a normal conversion is in process. 0 - Normal conversion is not in process 1 - Normal conversion is in process
23 JABORT	Injected Conversion Abort Status Signals whether an injected conversion has been aborted. It is reset when a new injected conversion starts. 0 - Injected conversion has not been aborted 1 - Injected conversion has been aborted
22-21	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
20 JSTART	Injected Conversion Status Specifies whether an injected conversion is in process. 0 - Injected conversion is not in process 1 - Injected conversion is in process
19 —	Reserved
18 SELF_TEST_S	Self-Test Status Indicates whether a self-test conversion is in process. 0 - Self-test conversion is not in process 1 - Self-test conversion is in process
17 —	Reserved
16 —	Reserved
15-9 CHADDR	Channel Address Specifies the address of the channel that is currently being converted. 000_0000 - Channel 0 selected 000_0001 - Channel 1 selected 000_0010 - Channel 2 selected 000_0011 - Channel 3 selected 000_0100 - Channel 4 selected 000_0101 - Channel 5 selected 000_0110 - Channel 6 selected 000_0111 - Channel 7 selected 010_0000 - Bandgap input selected 010_0001 - Pre-sample voltage - 1 : DVDD1P0/2 010_0010 - Pre-sample voltage - 2 : AVDD1p8 010_0011 - Pre-sample voltage - 3 : VREFL_1p8 010_0100 - Pre-sample voltage - 4 : VREFH_1p8 All other values are reserved.

Table continues on the next page...

Table continued from the previous page...

Field	Description
8-6 —	Reserved
5 ACKO	Auto-Clock-Off Enable Signals whether the auto-clock-off feature is enabled. 0 - Auto-Clock-Off feature is not enabled 1 - Auto-Clock-Off feature is enabled
4-3 —	Reserved
2-0 ADCSTATUS	ADC Status Represents ADC status. 000 - Idle 001 - Power-down 010 - Wait state (waiting to start conversion after [external trigger]). 011 - Busy in calibration 100 - Sample 110 - Conversion

72.6.4 Interrupt Status (ISR)

Indicates interrupt status.

Offset

Register	Offset
ISR	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											0	JEOC	JECH	EOC	ECH
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-5 —	Reserved
4 —	Reserved
3 JEOC	Injected Channel End of Conversion Indicates end of conversion status for an injected channel. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Injected channel end of conversion has not occurred 1 - Injected channel end of conversion has occurred When writing 0 - No effect 1 - Clear the flag
2 JECH	Injected End of Conversion Chain Indicates end of conversion chain status for an injected channel. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Injected channel end of conversion chain has not occurred 1 - Injected channel end of conversion chain has occurred When writing 0 - No effect 1 - Clear the flag
1 EOC	End of Channel Conversion Indicates the digital end of conversion for a channel. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Channel end of conversion has not occurred 1 - Channel end of conversion has occurred When writing

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - No effect 1 - Clear the flag
0 ECH	End of Conversion Chain Indicates the digital end of conversion chain for a channel. **NOTE** This field behaves differently for register reads and writes. When reading 0 - End of conversion chain has not occurred 1 - End of conversion chain has occurred When writing 0 - No effect 1 - Clear the flag

72.6.5 Channel Pending 0 (CEOCFR0)

Indicates end of conversion pending interrupt status for SAR_ADC channels in Group0.

Offset

Register	Offset
CEOCFR0	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								EOC_CH7	EOC_CH6	EOC_CH5	EOC_CH4	EOC_CH3	EOC_CH2	EOC_CH1	EOC_CH0
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7 EOC_CH7	Channel 7 EOC Status Indicates whether channel 7 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
6 EOC_CH6	Channel 6 EOC Status Indicates whether channel 6 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
5 EOC_CH5	Channel 5 EOC Status Indicates whether channel 5 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 EOC_CH4	Channel 4 EOC Status Indicates whether channel 4 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
3 EOC_CH3	Channel 3 EOC Status Indicates whether channel 3 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
2 EOC_CH2	Channel 2 EOC Status Indicates whether channel 2 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
1 EOC_CH1	Channel 1 EOC Status Indicates whether channel 1 conversion is complete.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
0 EOC_CH0	Channel 0 EOC Status Indicates whether channel 0 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag

72.6.6 Channel Pending 1 (CEOCFR1)

Indicates end of conversion pending interrupt status for group32 channels.

Offset

Register	Offset
CEOCFR1	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	EOC_CH39	EOC_CH38	EOC_CH37	EOC_CH36	EOC_CH35	EOC_CH34	EOC_CH33	EOC_CH32
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 EOC_CH39	Channel 39 EOC Status Indicates whether channel 39 conversion is complete.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
6 EOC_CH38	Channel 38 EOC Status Indicates whether channel 38 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
5 EOC_CH37	Channel 37 EOC Status Indicates whether channel 37 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
4 EOC_CH36	Channel 36 EOC Status Indicates whether channel 36 conversion is complete. **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
3 EOC_CH35	Channel 35 EOC Status Indicates whether channel 35 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
2 EOC_CH34	Channel 34 EOC Status Indicates whether channel 34 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
1 EOC_CH33	Channel 33 EOC Status Indicates whether channel 33 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Conversion complete When writing 0 - No effect 1 - Clear the flag
0 EOC_CH32	Channel 32 EOC Status Indicates whether channel 32 conversion is complete. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Conversion not complete 1 - Conversion complete When writing 0 - No effect 1 - Clear the flag

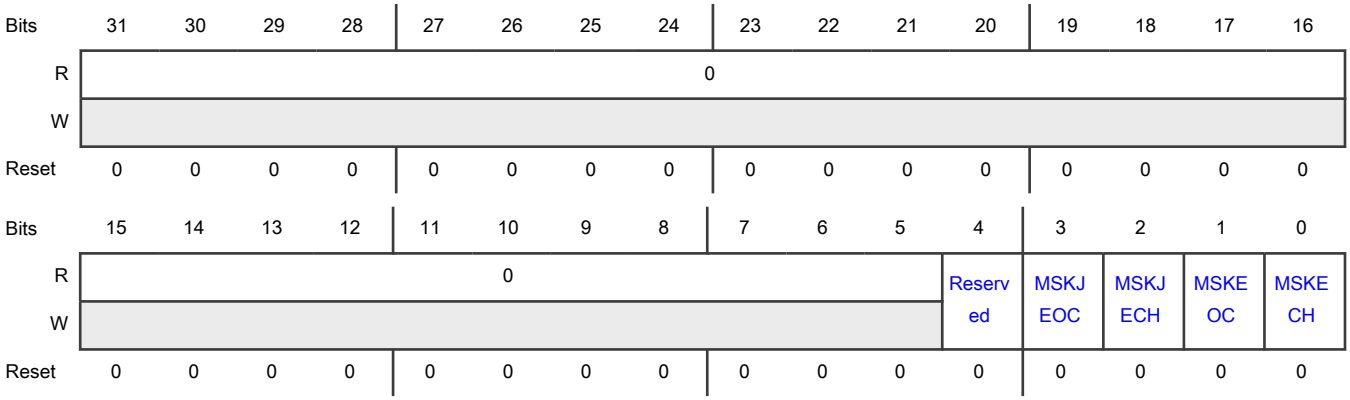
72.6.7 Interrupt Mask (IMR)

Contains fields for interrupt mask.

Offset

Register	Offset
IMR	20h

Diagram



Fields

Field	Description
31-5 —	Reserved
4 —	Reserved
3 MSKJEOC	End of Injected Conversion Interrupt Mask Enables JEOC interrupt. 0 - End of injected conversion interrupt disabled 1 - End of injected conversion interrupt enabled
2 MSKJECH	End of Injected Chain Conversion Interrupt Mask Enables JECH interrupt. 0 - End of injected chain conversion interrupt disabled 1 - End of injected chain conversion interrupt enabled
1 MSKEOC	End of Conversion Interrupt Mask Enables end of conversion interrupt. 0 - End of conversion interrupt disabled 1 - End of conversion interrupt enabled
0 MSKECH	End of Chain Conversion Interrupt Mask Enables end of chain conversion interrupt. 0 - End of chain conversion interrupt disabled 1 - End of chain conversion interrupt enabled

72.6.8 Channel Interrupt Mask 0 (CIMR0)

Contains fields for channel interrupt mask for group0.

NOTE

The available CIMR0 fields are chip-specific. See the chip-specific SAR_ADC information for details.

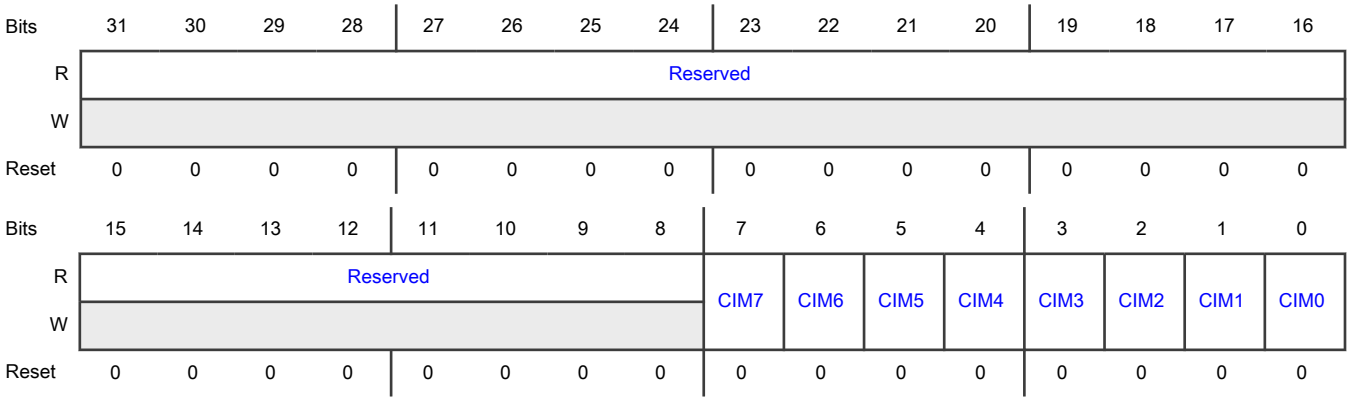
Table 632. CIM*n* field description

Field	Description
CIM0	Interrupt enable. When set (CIM0 = 1), interrupt for channel zero is enabled.
CIM*n*	Interrupt enable. When set (CIM*n* = 1), interrupt for channel *n* is enabled.

Offset

Register	Offset
CIMR0	24h

Diagram



Fields

Field	Description
31-8 —	Reserved
7 CIM7	Channel 7 Interrupt Enable Enables interrupt for channel 7. 0 - Interrupt disabled 1 - Interrupt enabled
6 CIM6	Channel 6 Interrupt Enable Enables interrupt for channel 6. 0 - Interrupt disabled 1 - Interrupt enabled
5 CIM5	Channel 5 Interrupt Enable Enables interrupt for channel 5. 0 - Interrupt disabled 1 - Interrupt enabled
4 CIM4	Channel 4 Interrupt Enable Enables interrupt for channel 4. 0 - Interrupt disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Interrupt enabled
3 CIM3	Channel 3 Interrupt Enable Enables interrupt for channel 3. 0 - Interrupt disabled 1 - Interrupt enabled
2 CIM2	Channel 2 Interrupt Enable Enables interrupt for channel 2. 0 - Interrupt disabled 1 - Interrupt enabled
1 CIM1	Channel 1 Interrupt Enable Enables interrupt for channel 1. 0 - Interrupt disabled 1 - Interrupt enabled
0 CIM0	Channel 0 Interrupt Enable Enables interrupt for channel 0. 0 - Interrupt disabled 1 - Interrupt enabled

72.6.9 Channel Interrupt Mask 1 (CIMR1)

Contains fields for channel interrupt mask for group32.

NOTE

The available CIMR1 fields are chip-specific. See chip-specific SAR_ADC information for details.

Table 633. CIM*n* field description

Field	Description
CIM0	Interrupt enable. When set (CIM0 = 1), interrupt for channel zero is enabled.
CIM*n*	Interrupt enable. When set (CIM*n* = 1), interrupt for channel *n* is enabled.

Offset

Register	Offset
CIMR1	28h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CIM39	CIM38	CIM37	CIM36	CIM35	CIM34	CIM33	CIM32
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 CIM39	Channel 39 Interrupt Enable Enables interrupt for channel 39.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Interrupt disabled 1 - Interrupt enabled
6 CIM38	Channel 38 Interrupt Enable Enables interrupt for channel 38. 0 - Interrupt disabled 1 - Interrupt enabled
5 CIM37	Channel 37 Interrupt Enable Enables interrupt for channel 37. 0 - Interrupt disabled 1 - Interrupt enabled
4 CIM36	Channel 36 Interrupt Enable Enables interrupt for channel 36. 0 - Interrupt disabled 1 - Interrupt enabled
3 CIM35	Channel 35 Interrupt Enable Enables interrupt for channel 35. 0 - Interrupt disabled 1 - Interrupt enabled
2 CIM34	Channel 34 Interrupt Enable Enables interrupt for channel 34. 0 - Interrupt disabled 1 - Interrupt enabled
1 CIM33	Channel 33 Interrupt Enable Enables interrupt for channel 33. 0 - Interrupt disabled 1 - Interrupt enabled
0 CIM32	Channel 32 Interrupt Enable Enables interrupt for channel 32. 0 - Interrupt disabled 1 - Interrupt enabled

72.6.10 Watchdog Threshold Interrupt Status (WTISR)

Indicates watchdog threshold interrupt status.

Offset

Register	Offset
WTISR	30h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		0		0		0		0		0		0		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WDG7H	WDG7L	WDG6H	WDG6L	WDG5H	WDG5L	WDG4H	WDG4L	WDG3H	WDG3L	WDG2H	WDG2L	WDG1H	WDG1L	WDG0H	WDG0L
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-30 —	Reserved
29-28 —	Reserved
27-26 —	Reserved
25-24 —	Reserved
23-22 —	Reserved
21-20 —	Reserved
19-18	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
17-16 —	Reserved
15 WDG7H	Channel 7 Watchdog High Threshold Interrupt Corresponds to the interrupt generated on the converted value being higher than the programmed higher threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted When writing 0 - No effect 1 - Clear the flag
14 WDG7L	Channel 7 Watchdog Low Threshold Interrupt Corresponds to the interrupt generated on the converted value being lower than the programmed lower threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted When writing 0 - No effect 1 - Clear the flag
13 WDG6H	Channel 6 Watchdog High Threshold Interrupt Corresponds to the interrupt generated on the converted value being higher than the programmed higher threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When writing 0 - No effect 1 - Clear the flag
12 WDG6L	Channel 6 Watchdog Low Threshold Interrupt Corresponds to the interrupt generated on the converted value being lower than the programmed lower threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted When writing 0 - No effect 1 - Clear the flag
11 WDG5H	Channel 5 Watchdog High Threshold Interrupt Corresponds to the interrupt generated on the converted value being higher than the programmed higher threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted When writing 0 - No effect 1 - Clear the flag
10 WDG5L	Channel 5 Watchdog Low Threshold Interrupt Corresponds to the interrupt generated on the converted value being lower than the programmed lower threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When writing 0 - No effect 1 - Clear the flag
9 WDG4H	Channel 4 Watchdog High Threshold Interrupt Corresponds to the interrupt generated on the converted value being higher than the programmed higher threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted When writing 0 - No effect 1 - Clear the flag
8 WDG4L	Channel 4 Watchdog Low Threshold Interrupt Corresponds to the interrupt generated on the converted value being lower than the programmed lower threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted When writing 0 - No effect 1 - Clear the flag
7 WDG3H	Channel 3 Watchdog High Threshold Interrupt Corresponds to the interrupt generated on the converted value being higher than the programmed higher threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When writing 0 - No effect 1 - Clear the flag
6 WDG3L	Channel 3 Watchdog Low Threshold Interrupt Corresponds to the interrupt generated on the converted value being lower than the programmed lower threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted When writing 0 - No effect 1 - Clear the flag
5 WDG2H	Channel 2 Watchdog High Threshold Interrupt Corresponds to the interrupt generated on the converted value being higher than the programmed higher threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted When writing 0 - No effect 1 - Clear the flag
4 WDG2L	Channel 2 Watchdog Low Threshold Interrupt Corresponds to the interrupt generated on the converted value being lower than the programmed lower threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When writing 0 - No effect 1 - Clear the flag
3 WDG1H	Channel 1 Watchdog High Threshold Interrupt Corresponds to the interrupt generated on the converted value being higher than the programmed higher threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted When writing 0 - No effect 1 - Clear the flag
2 WDG1L	Channel 1 Watchdog Low Threshold Interrupt Corresponds to the interrupt generated on the converted value being lower than the programmed lower threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted When writing 0 - No effect 1 - Clear the flag
1 WDG0H	Channel 0 Watchdog High Threshold Interrupt Corresponds to the interrupt generated on the converted value being higher than the programmed higher threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When writing 0 - No effect 1 - Clear the flag
0 WDG0L	Channel 0 Watchdog Low Threshold Interrupt Corresponds to the interrupt generated on the converted value being lower than the programmed lower threshold. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Interrupt not asserted 1 - Interrupt asserted When writing 0 - No effect 1 - Clear the flag

72.6.11 Watchdog Threshold Interrupt Mask (WTIMR)

Indicates the watchdog threshold interrupt mask.

Offset

Register	Offset
WTIMR	34h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		0		0		0		0		0		0		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSKWDG7H	MSKWDG7L	MSKWDG6H	MSKWDG6L	MSKWDG5H	MSKWDG5L	MSKWDG4H	MSKWDG4L	MSKWDG3H	MSKWDG3L	MSKWDG2H	MSKWDG2L	MSKWDG1H	MSKWDG1L	MSKWDG0H	MSKWDG0L
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-30 —	Reserved
29-28 —	Reserved
27-26 —	Reserved
25-24 —	Reserved
23-22 —	Reserved
21-20 —	Reserved
19-18 —	Reserved
17-16 —	Reserved
15 MSKWDG7H	Channel 7 Watchdog High Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being higher than the programmed higher threshold. When set, the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
14 MSKWDG7L	Channel 7 Watchdog Low Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being lower than the programmed lower threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
13 MSKWDG6H	Channel 6 Watchdog High Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being higher than the programmed higher threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
12 MSKWDG6L	Channel 6 Watchdog Low Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being lower than the programmed lower threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
11 MSKWDG5H	Channel 5 Watchdog High Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being higher than the programmed higher threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
10 MSKWDG5L	Channel 5 Watchdog Low Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being lower than the programmed lower threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
9 MSKWDG4H	Channel 4 Watchdog High Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being higher than the programmed higher threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
8 MSKWDG4L	Channel 4 Watchdog Low Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being lower than the programmed lower threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
7 MSKWDG3H	Channel 3 Watchdog High Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being higher than the programmed higher threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
6 MSKWDG3L	Channel 3 Watchdog Low Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being lower than the programmed lower threshold. When set the interrupt is enabled. 0 - Interrupt is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Interrupt is enabled
5 MSKWDG2H	Channel 2 Watchdog High Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being higher than the programmed higher threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
4 MSKWDG2L	Channel 2 Watchdog Low Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being lower than the programmed lower threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
3 MSKWDG1H	Channel 1 Watchdog High Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being higher than the programmed higher threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
2 MSKWDG1L	Channel 1 Watchdog Low Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being lower than the programmed lower threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
1 MSKWDG0H	Channel 0 Watchdog High Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being higher than the programmed higher threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled
0 MSKWDG0L	Channel 0 Watchdog Low Threshold Interrupt Mask Corresponds to the mask bit for the interrupt generated on the converted value being lower than the programmed lower threshold. When set the interrupt is enabled. 0 - Interrupt is disabled 1 - Interrupt is enabled

72.6.12 DMAE (DMAE)

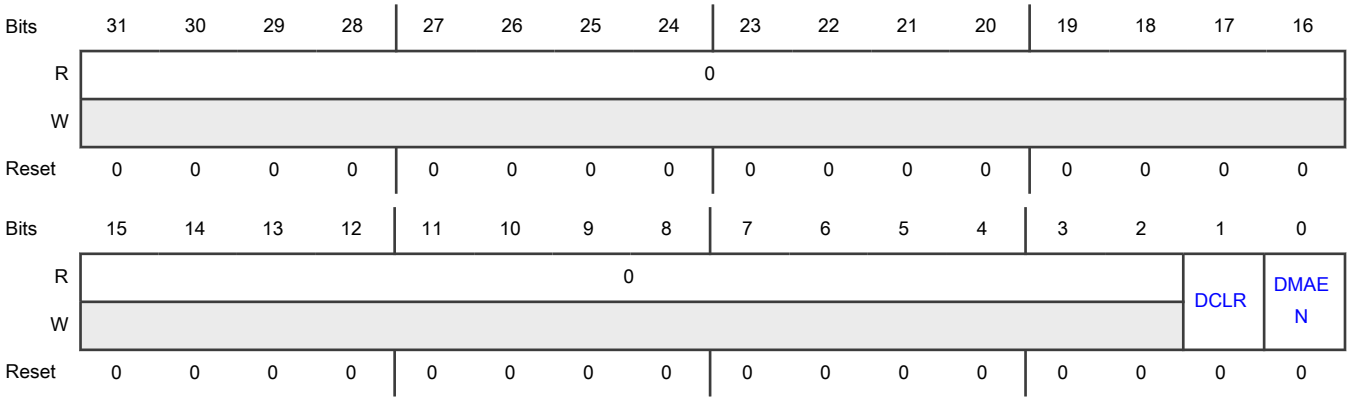
Contains fields for DMA global enable and DMA clear sequence enable.

DMA settings must be changed while ADC is in idle state (MSR[ADCSTATUS] = IDLE).

Offset

Register	Offset
DMAE	40h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 DCLR	DMA Clear Sequence Enable Selects DMA request clear either by acknowledgement from DMA controller or read of data register. 0 - DMA request cleared by acknowledge from DMA controller 1 - DMA request cleared on read of data registers
0 DMAEN	DMA Global Enable Enables DMA feature if set. 0 - DMA feature is disabled 1 - DMA feature is enabled

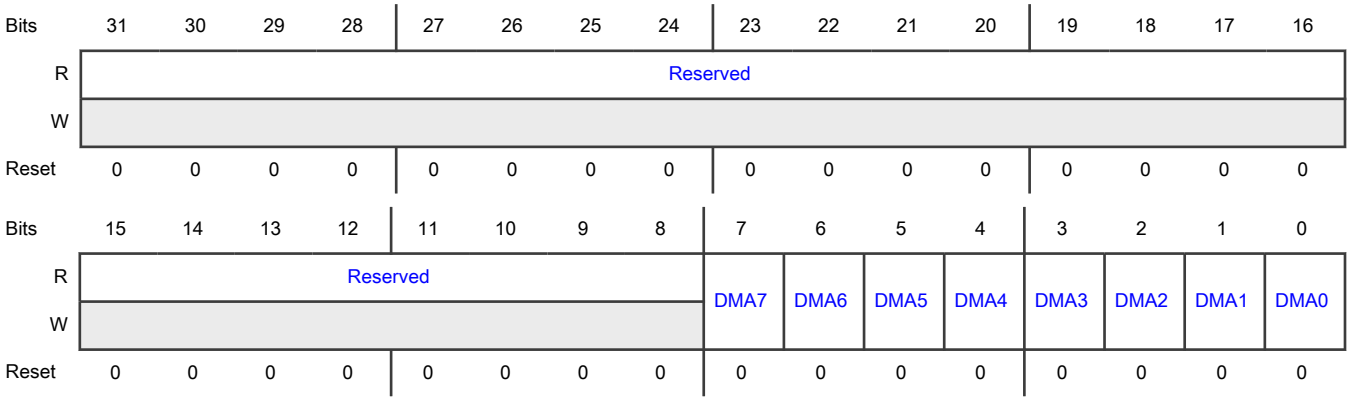
72.6.13 DMA 0 (DMAR0)

Enables transfer of data in DMA mode for Group0.

Offset

Register	Offset
DMAR0	44h

Diagram



Fields

Field	Description
31-8 —	Reserved
7 DMA7	Channel 7 DMA Enable Enables transfer of data in DMA mode for Channel 7 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
6 DMA6	Channel 6 DMA Enable Enables transfer of data in DMA mode for Channel 6 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
5 DMA5	Channel 5 DMA Enable Enables transfer of data in DMA mode for Channel 5 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
4 DMA4	Channel 4 DMA Enable Enables transfer of data in DMA mode for Channel 4 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
3 DMA3	Channel 3 DMA Enable Enables transfer of data in DMA mode for Channel 3 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 DMA2	Channel 2 DMA Enable Enables transfer of data in DMA mode for Channel 2 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
1 DMA1	Channel 1 DMA Enable Enables transfer of data in DMA mode for Channel 1 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
0 DMA0	Channel 0 DMA Enable Enables transfer of data in DMA mode for Channel 0 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled

72.6.14 DMA 1 (DMAR1)

Enables transfer of data in DMA mode for Group32.

Table 634. DMA*n* field description

Field	Description
DMA32	DMA enable. When set (DMA32 = 1), channel zero is enabled to transfer data in DMA mode.
DMA *n*	DMA enable. When set (DMA *n* = 1), channel n is enabled to transfer data in DMA mode.

Offset

Register	Offset
DMAR1	48h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DMA39	DMA38	DMA37	DMA36	DMA35	DMA34	DMA33	DMA32
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 DMA39	Channel 39 DMA Enable Enables transfer of data in DMA mode for Channel 39 if set.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
6 DMA38	Channel 38 DMA Enable Enables transfer of data in DMA mode for Channel 38 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
5 DMA37	Channel 37 DMA Enable Enables transfer of data in DMA mode for Channel 37 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
4 DMA36	Channel 36 DMA Enable Enables transfer of data in DMA mode for Channel 36 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
3 DMA35	Channel 35 DMA Enable Enables transfer of data in DMA mode for Channel 35 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
2 DMA34	Channel 34 DMA Enable Enables transfer of data in DMA mode for Channel 34 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
1 DMA33	Channel 33 DMA Enable Enables transfer of data in DMA mode for Channel 33 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled
0 DMA32	Channel 32 DMA Enable Enables transfer of data in DMA mode for Channel 32 if set. 0 - Transfer of data in DMA mode is disabled 1 - Transfer of data in DMA mode is enabled

72.6.15 Analog Watchdog Threshold 0 (THRHLR0)

Specifies high and low threshold 0 values for channel *n*.

Offset

Register	Offset
THRHLR0	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				THRH											
W																
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				THRL											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved Write of any value has no effect, read value is always zero.
27-16 THRH	High Threshold Specifies the high threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.
15-12 —	Reserved
11-0 THRL	Low Threshold Specifies the low threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.

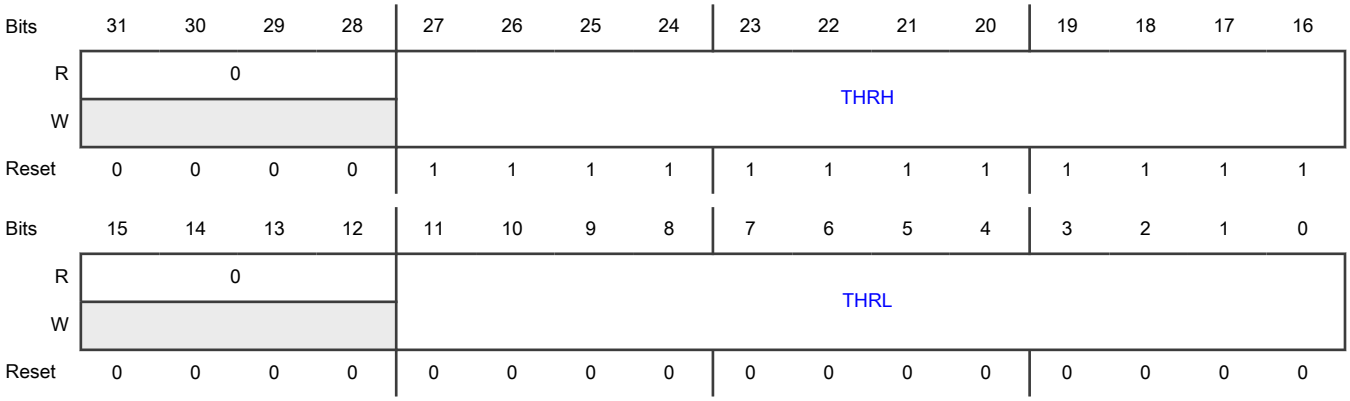
72.6.16 Analog Watchdog Threshold 1 (THRHLR1)

Specifies high and low threshold 1 values for channel *n*.

Offset

Register	Offset
THRHLR1	64h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 THRH	High Threshold Specifies the high threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.
15-12 —	Reserved
11-0 THRL	Low Threshold Specifies the low threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.

72.6.17 Analog Watchdog Threshold 2 (THRHLR2)

Specifies high and low threshold 2 values for channel *n*.

Offset

Register	Offset
THRHLR2	68h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				THRH											
W																
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				THRL											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 THRH	High Threshold Specifies the high threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.
15-12 —	Reserved
11-0 THRL	Low Threshold Specifies the low threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.

72.6.18 Analog Watchdog Threshold 3 (THRHLR3)

Specifies high and low threshold 3 values for channel *n*.

Offset

Register	Offset
THRHLR3	6Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				THRH											
W																
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				THRL											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 THRH	High Threshold Specifies the high threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.
15-12 —	Reserved
11-0 THRL	Low Threshold Specifies the low threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.

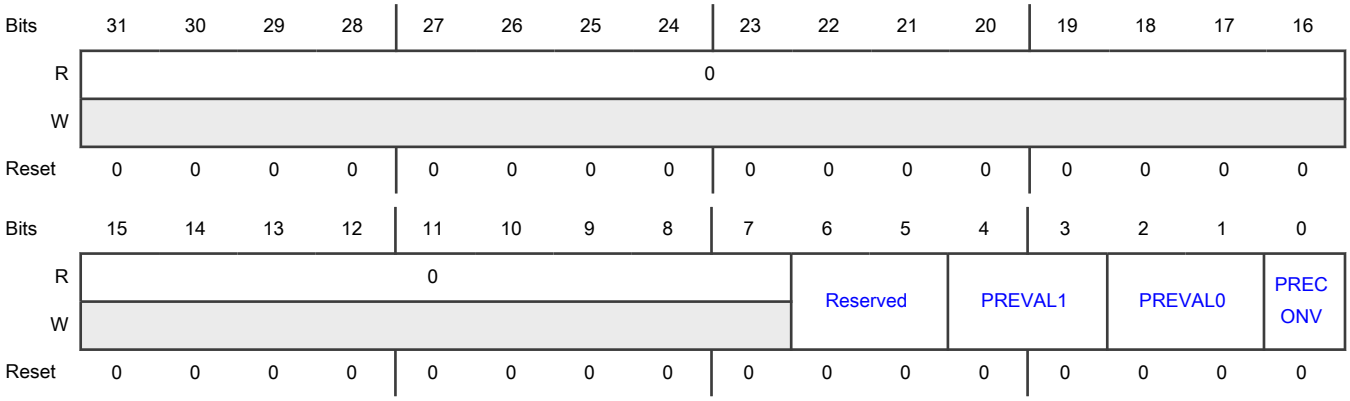
72.6.19 Presampling Control (PSCR)

Contains fields for converting presampled values and selecting analog input voltages for presampling.

Offset

Register	Offset
PSCR	80h

Diagram



Fields

Field	Description
31-7 —	Reserved
6-5 —	Reserved
4-3 PREVAL1	Internal Presampling Voltage Selection. Selects analog input voltages for presampling from four available internal voltages (channels in the range 32 to 63). See Presampling channel enable for details.
2-1 PREVAL0	Internal Presampling Voltage Selection Selects analog input voltages for presampling from four internal available voltages (channels in the range 0 to 31). See Presampling channel enable for details.
0 PRECONV	Convert Presampled Value Converts presampled values. If PRECONV is set, presampling is followed by the conversion. Sampling will be bypassed and conversion of presampled data will be done. 0 - Presampling is followed by sampling then conversion. 1 - Presampling is followed by the conversion.

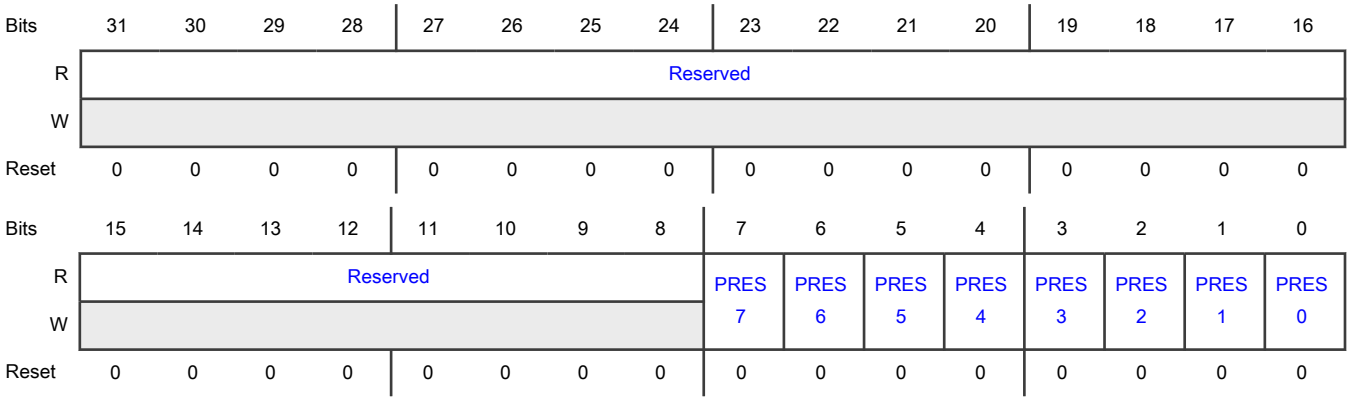
72.6.20 Presampling 0 (PSR0)

Enables presampling for channel *n*, when set (PRES *n* = 1).

Offset

Register	Offset
PSR0	84h

Diagram



Fields

Field	Description
31-8 —	Reserved
7 PRES7	Presampling Enable for Channel 7 Enables presampling for channel 7, if set. 0 - Presampling is disabled 1 - Presampling is enabled
6 PRES6	Presampling Enable for Channel 6 Enables presampling for channel 6, if set. 0 - Presampling is disabled 1 - Presampling is enabled
5 PRES5	Presampling Enable for Channel 5 Enables presampling for channel 5, if set. 0 - Presampling is disabled 1 - Presampling is enabled
4 PRES4	Presampling Enable for Channel 4 Enables presampling for channel 4, if set. 0 - Presampling is disabled 1 - Presampling is enabled
3 PRES3	Presampling Enable for Channel 3 Enables presampling for channel 3, if set. 0 - Presampling is disabled 1 - Presampling is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
2 PRES2	Presampling Enable for Channel 2 Enables presampling for channel 2, if set. 0 - Presampling is disabled 1 - Presampling is enabled
1 PRES1	Presampling Enable for Channel 1 Enables presampling for channel 1, if set. 0 - Presampling is disabled 1 - Presampling is enabled
0 PRES0	Presampling Enable for Channel 0 Enables presampling for channel 0, if set. 0 - Presampling is disabled 1 - Presampling is enabled

72.6.21 Presampling 1 (PSR1)

Enables presampling for channel *n*, when set (PRES *n* = 1).

Offset

Register	Offset
PSR1	88h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PRES39	PRES38	PRES37	PRES36	PRES35	PRES34	PRES33	PRES32
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 PRES39	Presampling Enable for Channel 39 Enables presampling for channel 39, if set. 0 - Presampling is disabled 1 - Presampling is enabled
6 PRES38	Presampling Enable for Channel 38 Enables presampling for channel 38, if set. 0 - Presampling is disabled 1 - Presampling is enabled
5 PRES37	Presampling Enable for Channel 37 Enables presampling for channel 37, if set. 0 - Presampling is disabled 1 - Presampling is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 PRES36	Presampling Enable for Channel 36 Enables presampling for channel 36, if set. 0 - Presampling is disabled 1 - Presampling is enabled
3 PRES35	Presampling Enable for Channel 35 Enables presampling for channel 35, if set. 0 - Presampling is disabled 1 - Presampling is enabled
2 PRES34	Presampling Enable for Channel 34 Enables presampling for channel 34, if set. 0 - Presampling is disabled 1 - Presampling is enabled
1 PRES33	Presampling Enable for Channel 33 Enables presampling for channel 33, if set. 0 - Presampling is disabled 1 - Presampling is enabled
0 PRES32	Presampling Enable for Channel 32 Enables presampling for channel 32, if set. 0 - Presampling is disabled 1 - Presampling is enabled

72.6.22 Conversion Timing 0 (CTR0)

Configures the sample period duration in terms of the SAR controller clock.

This register is associated with channels 0 to 31.

Offset

Register	Offset
CTR0	94h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								INPSAMP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Fields

Field	Description
31-8 —	Reserved
7-0 INPSAMP	Sampling Phase Duration Configures the sample period duration in terms of the SAR controller clock. The minimum acceptable value is 8. Configuring to a value lower than 8 sets the sample period to 8 cycles.

72.6.23 Conversion Timing 1 (CTR1)

Configures the sample period duration in terms of the SAR controller clock.

This register is associated with channels 32 to 63. An exception exists for the temperature sensor channel, which always uses the CTR1 value.

Offset

Register	Offset
CTR1	98h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								INPSAMP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Fields

Field	Description
31-8 —	Reserved
7-0 INPSAMP	Sampling Phase Duration Configures the sample period duration in terms of the SAR controller clock. The minimum acceptable value is 8. Configuring to a value lower than 8 sets the sample period to 8 cycles.

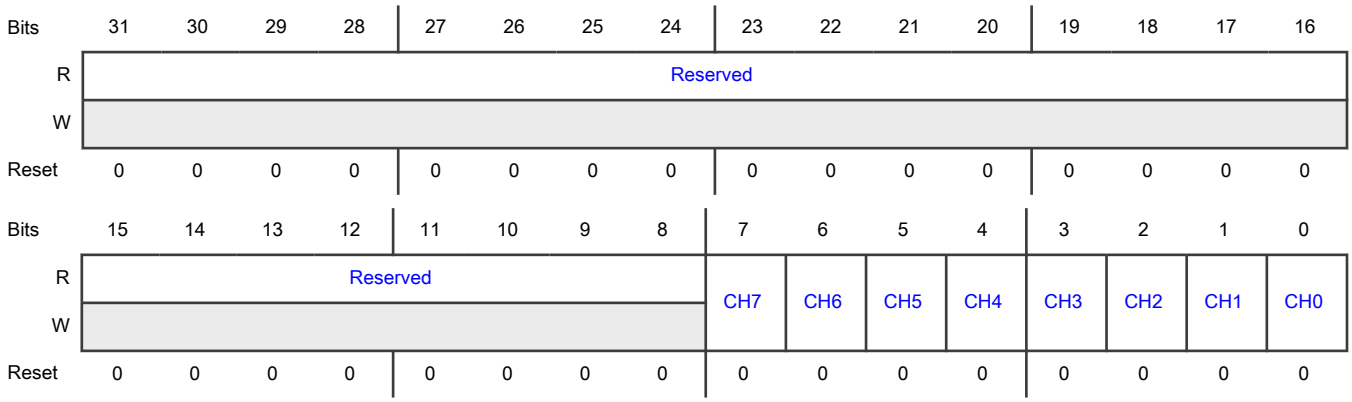
72.6.24 Normal Conversion Mask 0 (NCMR0)

Sampling enable for Group0 channel *n* (*n* = Group0 channel number).

Offset

Register	Offset
NCMR0	A4h

Diagram



Fields

Field	Description
31-8 —	Reserved
7 CH7	Normal Conversion Mask for Channel 7 Enables normal conversion for channel 7, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
6 CH6	Normal Conversion Mask for Channel 6 Enables normal conversion for channel 6, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
5 CH5	Normal Conversion Mask for Channel 5 Enables normal conversion for channel 5, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
4 CH4	Normal Conversion Mask for Channel 4 Enables normal conversion for channel 4, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
3 CH3	Normal Conversion Mask for Channel 3 Enables normal conversion for channel 3, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
2 CH2	Normal Conversion Mask for Channel 2 Enables normal conversion for channel 2, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
1 CH1	Normal Conversion Mask for Channel 1 Enables normal conversion for channel 1, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
0 CH0	Normal Conversion Mask for Channel 0 Enables normal conversion for channel 0, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled

72.6.25 Normal Conversion Mask 1 (NCMR1)

Sampling enable for Group32 channel *n* (*n* = Group32 channel number).

Offset

Register	Offset
NCMR1	A8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CH39	CH38	CH37	CH36	CH35	CH34	CH33	CH32
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 —	Reserved
7 CH39	Normal Conversion Mask for Channel 39 Enables normal conversion for channel 39, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
6 CH38	Normal Conversion Mask for Channel 38 Enables normal conversion for channel 38, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
5 CH37	Normal Conversion Mask for Channel 37 Enables normal conversion for channel 37, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
4 CH36	Normal Conversion Mask for Channel 36 Enables normal conversion for channel 36, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
3 CH35	Normal Conversion Mask for Channel 35 Enables normal conversion for channel 35, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
2 CH34	Normal Conversion Mask for Channel 34 Enables normal conversion for channel 34, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
1 CH33	Normal Conversion Mask for Channel 33 Enables normal conversion for channel 33, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled
0	Normal Conversion Mask for Channel 32

Table continues on the next page...

Table continued from the previous page...

Field	Description
CH32	Enables normal conversion for channel 32, if set. 0 - Normal Conversion is disabled 1 - Normal Conversion is enabled

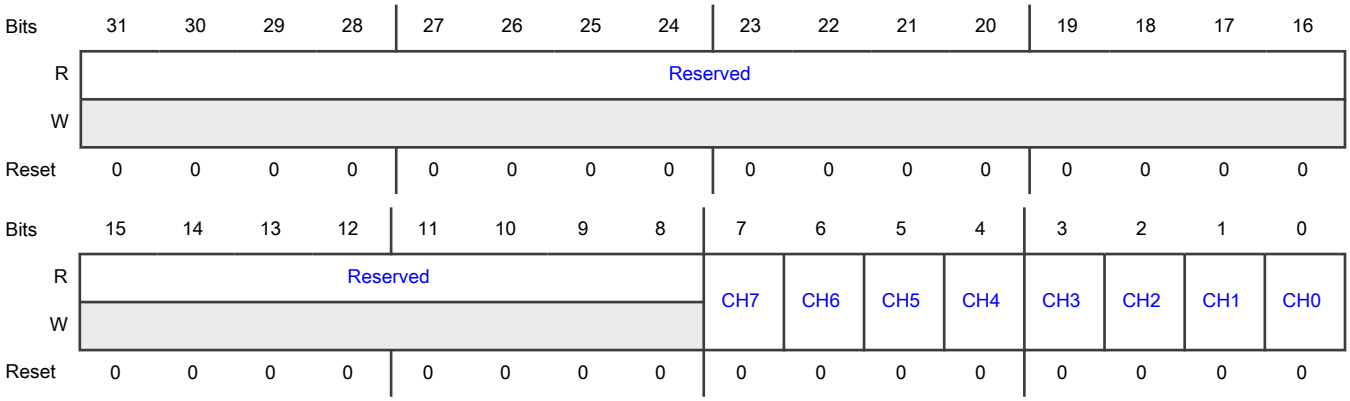
72.6.26 Injected Conversion Mask 0 (JCMR0)

Enables injected sampling for Group0 channels.

Offset

Register	Offset
JCMR0	B4h

Diagram



Fields

Field	Description
31-8 —	Reserved
7 CH7	Injected Conversion Mask for Channel 7 Enables injected conversion for channel 7, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
6 CH6	Injected Conversion Mask for Channel 6 Enables injected conversion for channel 6, if set.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Injected conversion is disabled 1 - Injected conversion is enabled
5 CH5	Injected Conversion Mask for Channel 5 Enables injected conversion for channel 5, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
4 CH4	Injected Conversion Mask for Channel 4 Enables injected conversion for channel 4, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
3 CH3	Injected Conversion Mask for Channel 3 Enables injected conversion for channel 3, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
2 CH2	Injected Conversion Mask for Channel 2 Enables injected conversion for channel 2, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
1 CH1	Injected Conversion Mask for Channel 1 Enables injected conversion for channel 1, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
0 CH0	Injected Conversion Mask for Channel 0 Enables injected conversion for channel 0, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled

72.6.27 Injected Conversion Mask 1 (JCMR1)

Enables injected sampling for Group32 channels.

Offset

Register	Offset
JCMR1	B8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CH39	CH38	CH37	CH36	CH35	CH34	CH33	CH32
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 —	Reserved
7 CH39	Injected Conversion Mask for Channel 39 Enables injected conversion for channel 39, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
6 CH38	Injected Conversion Mask for Channel 38 Enables injected conversion for channel 38, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
5 CH37	Injected Conversion Mask for Channel 37 Enables injected conversion for channel 37, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
4 CH36	Injected Conversion Mask for Channel 36 Enables injected conversion for channel 36, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
3 CH35	Injected Conversion Mask for Channel 35 Enables injected conversion for channel 35, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
2 CH34	Injected Conversion Mask for Channel 34 Enables injected conversion for channel 34, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
1 CH33	Injected Conversion Mask for Channel 33 Enables injected conversion for channel 33, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled
0	Injected Conversion Mask for Channel 32

Table continues on the next page...

Table continued from the previous page...

Field	Description
CH32	Enables injected conversion for channel 32, if set. 0 - Injected conversion is disabled 1 - Injected conversion is enabled

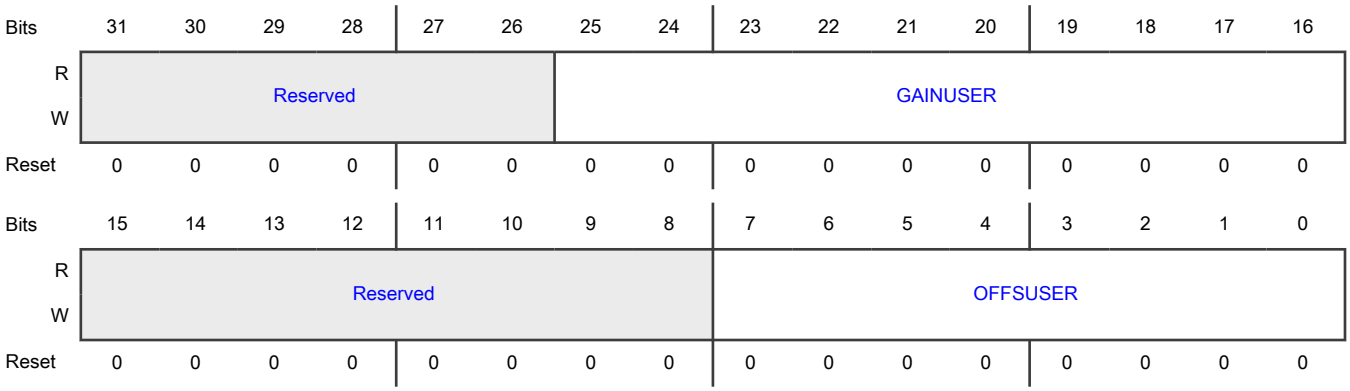
72.6.28 User OFFSET and Gain (USROFSGN)

Specifies additional offset and gain values for data processing.

Offset

Register	Offset
USROFSGN	C0h

Diagram



Fields

Field	Description
31-26 —	Reserved
25-16 GAINUSER	User-Defined Gain Value Specifies additional gain value in two's complement format used by ADC for data processing (for example, values can be either positive or negative; MSB represents sign bit).
15-8 —	Reserved
7-0 OFFSUSER	User Defined Offset Specifies additional offset of ADC in two's complement format used for data processing (values can be either positive or negative; MSB represents sign bit).

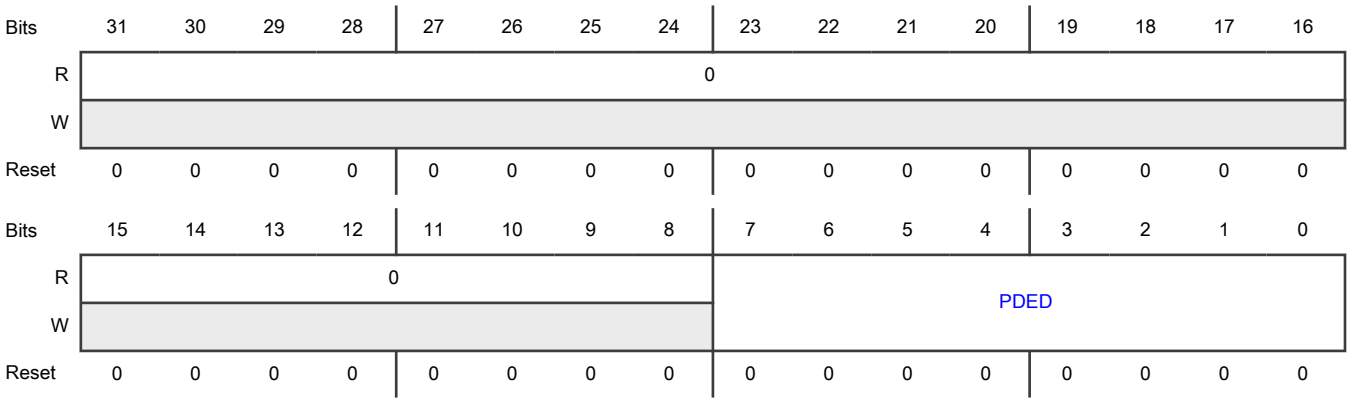
72.6.29 Power Down Exit Delay (PDEDR)

Specifies the delay between the power-down bit reset and the start of conversion.

Offset

Register	Offset
PDEDR	C8h

Diagram



Fields

Field	Description
31-8 —	Reserved
7-0 PDED	Power Down Exist Delay Specifies the delay between the power-down bit reset and the start of conversion. The power-down delay is at least (1/ADC Bus clock) + ((PDED+1) × 1/[ADC_clock_frequency]) where ADC_clock_frequency = frequency of ADC Hard Macro clock(AD_CLK) and ADC Bus clock = module interface clock for register access.

72.6.30 Precision Channel n Data (PCDR0 - PCDR7)

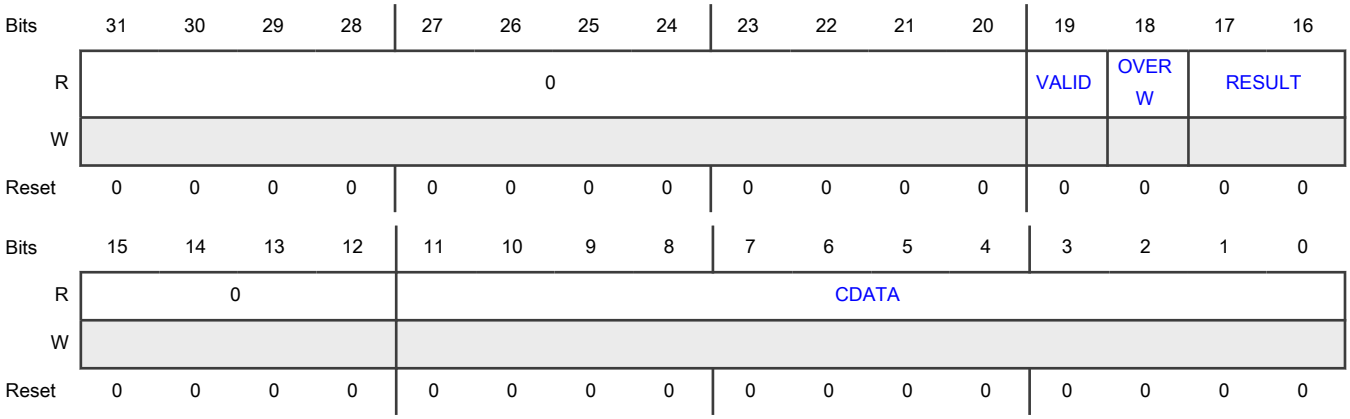
Contains conversion results for the precision channels. Each data register also contains information regarding the result.

Offset

For a = 0 to 7:

Register	Offset
PCDRa	100h + (a × 4h)

Diagram



Fields

Field	Description
31-20 —	Reserved
19 VALID	Conversion Data Valid Indicates when the data is valid (a new value has been written). It is automatically cleared when data is read. 0 - Not valid data 1 - Valid data
18 OVERW	Data Overwrite Indicates when conversion data was overwritten by a newer result. The new data is written or discarded according to MCR[OWREN]. 0 - Data not overwritten 1 - Data overwritten
17-16 RESULT	Mode of Conversion Status Indicates the mode of conversion for the corresponding channel. 00 - Data is a result of Normal conversion mode 01 - Data is a result of Injected conversion mode 10 - Data is a result of CTU conversion mode 11 - Reserved
15-12 —	Reserved
11-0 CDATA	Channel Converted Data Stores the conversion data corresponding to precision channel.

72.6.31 Internal Channel n Data (ICDR32 - ICDR39)

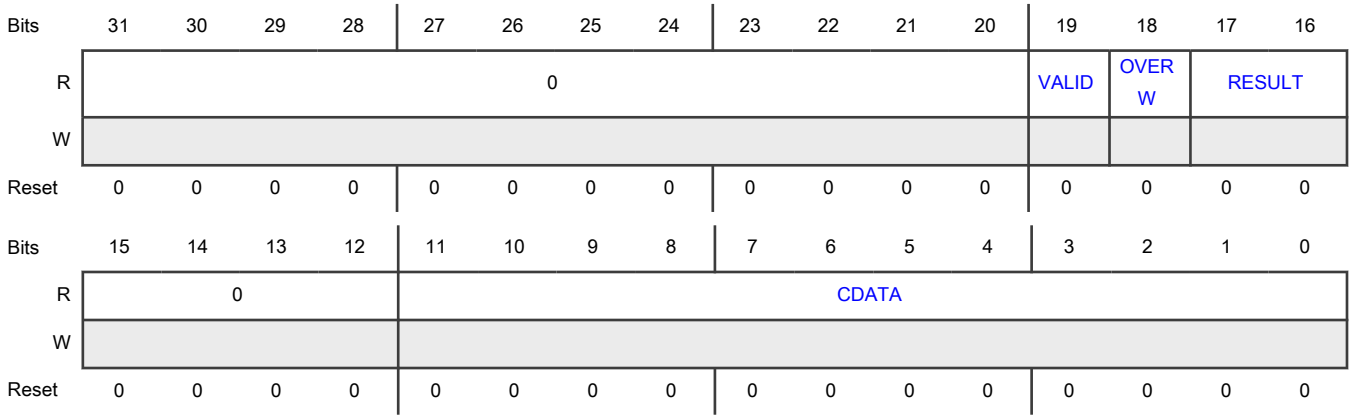
Contains conversion results for the internal channels. Each data register also contains information regarding the result.

Offset

For a = 32 to 39:

Register	Offset
ICDRa	100h + (a × 4h)

Diagram



Fields

Field	Description
31-20 —	Reserved
19 VALID	Conversion Data Valid Notifies when the data is valid (a new value has been written). It is automatically cleared when data is read. 0 - Not valid data 1 - Valid data
18 OVERW	Data Overwrite Indicates when conversion data was overwritten by a newer result. The new data is written or discarded according to MCR[OWREN]. 0 - Data not overwritten 1 - Data overwritten
17-16 RESULT	Mode of Conversion Status Indicates the mode of conversion for the corresponding channel.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	00 - Data is a result of Normal conversion mode 01 - Data is a result of Injected conversion mode 10 - Data is a result of CTU conversion mode 11 - Reserved
15-12 —	Reserved
11-0 CDATA	Channel Converted Data Stores the conversion data corresponding to internal channel.

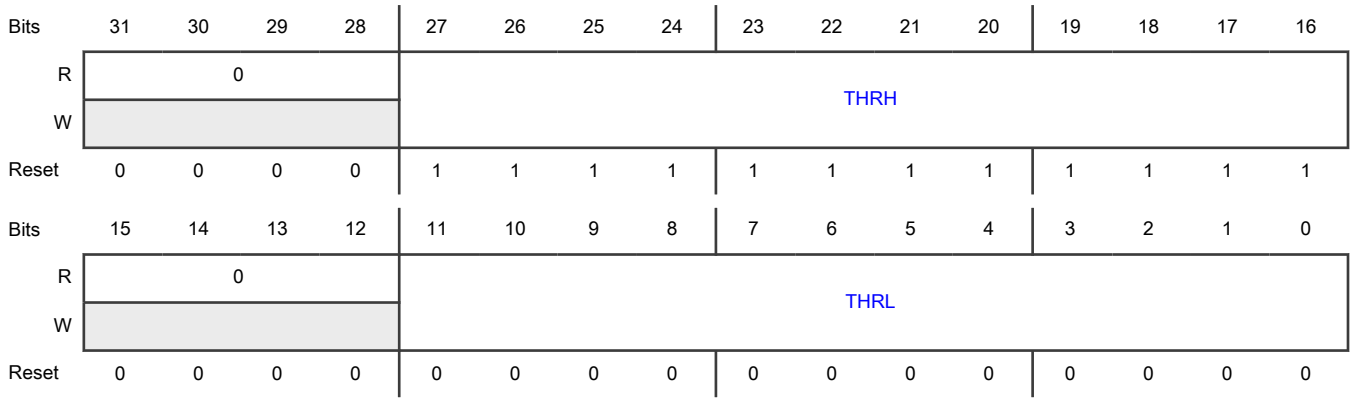
72.6.32 Analog Watchdog Threshold 4 (THRHLR4)

Specifies high and low threshold 4 values for channel *n*.

Offset

Register	Offset
THRHLR4	280h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16	High Threshold

Table continues on the next page...

Table continued from the previous page...

Field	Description
THRH	Specifies high threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.
15-12 —	Reserved
11-0 THRL	Low Threshold Specifies low threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.

72.6.33 Analog Watchdog Threshold 5 (THRHLR5)

Specifies high and low threshold 5 values for channel *n*.

Offset

Register	Offset
THRHLR5	284h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				THRH											
W																
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				THRL											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 THRH	High Threshold Specifies high threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.

Table continues on the next page...

Table continued from the previous page...

Field	Description
15-12 —	Reserved
11-0 THRL	Low Threshold Specifies low threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.

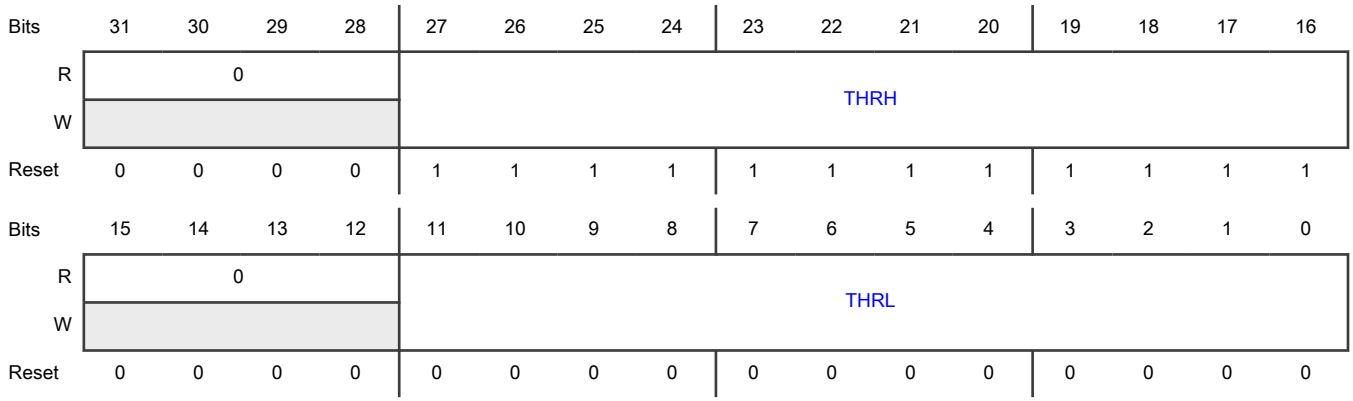
72.6.34 Analog Watchdog Threshold 6 (THRHLR6)

Specifies high and low threshold 6 values for channel *n*.

Offset

Register	Offset
THRHLR6	288h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 THRH	High Threshold Specifies high threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.
15-12 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
11-0 THRL	Low Threshold Specifies low threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.

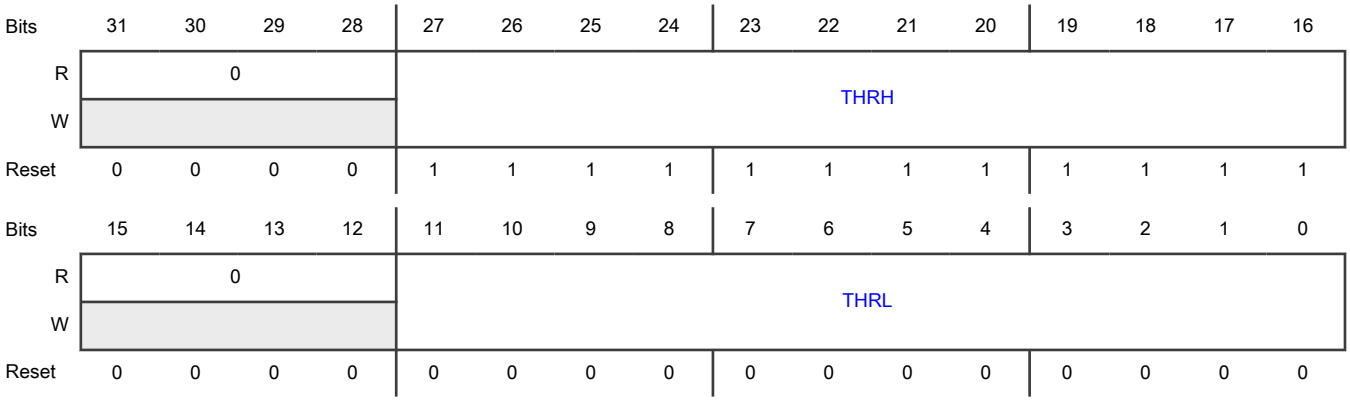
72.6.35 Analog Watchdog Threshold 7 (THRHLR7)

Specifies high and low threshold 7 values for channel *n*.

Offset

Register	Offset
THRHLR7	28Ch

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 THRH	High Threshold Specifies high threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.
15-12 —	Reserved
11-0 THRL	Low Threshold Specifies low threshold value for channel *n*. Channel *n* is selected by corresponding field in Channel Select Register.

72.6.36 Channel Watchdog Select 0 (CWSELR0)

Controls the selection of the threshold (or alternate threshold) register to provide upper and lower threshold values for channels 0 through 7.

Table 635. Channel watchdog select for channel *n*

Field	Description
WSEL_CH *n*	Channel watchdog select for channel *n*. Selects the threshold (or alternate threshold) register for channel *n*. 0h THRHLR0 (THRALT0) register is selected 1h THRHLR1 (THRALT1) register is selected 2h THRHLR2 (THRALT2) register is selected 3h THRHLR3 (THRALT3) register is selected 4h THRHLR4 register is selected 5h THRHLR5 register is selected 6h THRHLR6 register is selected 7h THRHLR7 register is selected

Offset

Register	Offset
CWSELR0	2B0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	WSEL_CH7			0	WSEL_CH6			0	WSEL_CH5			0	WSEL_CH4		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	WSEL_CH3			0	WSEL_CH2			0	WSEL_CH1			0	WSEL_CH0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
30-28 WSEL_CH7	Channel Watchdog Select for Channel 7 See Table 635 for values.
27 —	Reserved
26-24 WSEL_CH6	Channel Watchdog Select for Channel 6 See Table 635 for values.
23 —	Reserved
22-20 WSEL_CH5	Channel Watchdog Select for Channel 5 See Table 635 for values.
19 —	Reserved
18-16 WSEL_CH4	Channel Watchdog Select for Channel 4 See Table 635 for values.
15 —	Reserved
14-12 WSEL_CH3	Channel Watchdog Select for Channel 3 See Table 635 for values.
11 —	Reserved
10-8 WSEL_CH2	Channel Watchdog Select for Channel 2 See Table 635 for values.
7 —	Reserved
6-4 WSEL_CH1	Channel Watchdog Select for Channel 1 See Table 635 for values.
3 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
2-0 WSEL_CH0	Channel Watchdog Select for Channel 0 See Table 635 for values.

72.6.37 Channel Watchdog Select 4 (CWSELR4)

Controls the selection of the threshold (or alternate threshold) register to provide upper and lower threshold values for channels 32 through 39.

Table 636. Channel watchdog select for channel *n*

Field	Description
WSEL_CH *n*	Channel watchdog select for channel *n*. Selects the threshold (or alternate threshold) register for channel *n*. 0h THRHLR0 (THRALT0) register is selected 1h THRHLR1 (THRALT1) register is selected 2h THRHLR2 (THRALT2) register is selected 3h THRHLR3 (THRALT3) register is selected 4h THRHLR4 register is selected 5h THRHLR5 register is selected 6h THRHLR6 register is selected 7h THRHLR7 register is selected

Offset

Register	Offset
CWSELR4	2C0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	WSEL_CH39			0	WSEL_CH38			0	WSEL_CH37			0	WSEL_CH36		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	WSEL_CH35			0	WSEL_CH34			0	WSEL_CH33			0	WSEL_CH32		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 —	Reserved
30-28 WSEL_CH39	Channel Watchdog Select for Channel 39 See Table 636 for values.
27 —	Reserved
26-24 WSEL_CH38	Channel Watchdog Select for Channel 38 See Table 636 for values.
23 —	Reserved
22-20 WSEL_CH37	Channel Watchdog Select for Channel 37 See Table 636 for values.
19 —	Reserved
18-16 WSEL_CH36	Channel Watchdog Select for Channel 36 See Table 636 for values.
15 —	Reserved
14-12 WSEL_CH35	Channel Watchdog Select for Channel 35 See Table 636 for values.
11 —	Reserved
10-8 WSEL_CH34	Channel Watchdog Select for Channel 34 See Table 636 for values.
7 —	Reserved
6-4 WSEL_CH33	Channel Watchdog Select for Channel 33 See Table 636 for values.
3	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
2-0 WSEL_CH32	Channel Watchdog Select for Channel 32 See Table 636 for values.

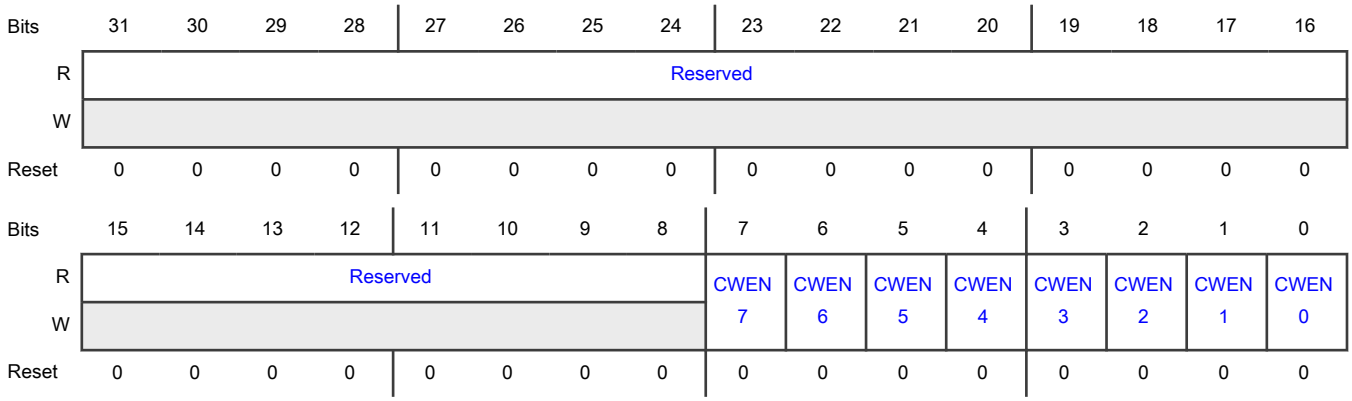
72.6.38 Channel Watchdog Enable 0 (CWENR0)

Enables the watchdog feature for channel *n*, when set (CWEN *n* = 1).

Offset

Register	Offset
CWENR0	2E0h

Diagram



Fields

Field	Description
31-8 —	Reserved
7 CWEN7	Watchdog Enable for Channel 7 Enables watchdog for channel 7, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
6 CWEN6	Watchdog Enable for Channel 6 Enables watchdog for channel 6, if set.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Watchdog is disabled 1 - Watchdog is enabled
5 CWEN5	Watchdog Enable for Channel 5 Enables watchdog for channel 5, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
4 CWEN4	Watchdog Enable for Channel 4 Enables watchdog for channel 4, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
3 CWEN3	Watchdog Enable for Channel 3 Enables watchdog for channel 3, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
2 CWEN2	Watchdog Enable for Channel 2 Enables watchdog for channel 2, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
1 CWEN1	Watchdog Enable for Channel 1 Enables watchdog for channel 1, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
0 CWEN0	Watchdog Enable for Channel 0 Enables watchdog for channel 0, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled

72.6.39 Channel Watchdog Enable 1 (CWENR1)

Enables Watchdog feature for channel *n* , when set (CWEN *n* = 1).

Offset

Register	Offset
CWENR1	2E4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CWEN39	CWEN38	CWEN37	CWEN36	CWEN35	CWEN34	CWEN33	CWEN32
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 —	Reserved
7 CWEN39	Watchdog Enable for Channel 39 Enables watchdog for channel 39, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
6 CWEN38	Watchdog Enable for Channel 38 Enables watchdog for channel 38, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
5 CWEN37	Watchdog Enable for Channel 37 Enables watchdog for channel 37, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
4 CWEN36	Watchdog Enable for Channel 36 Enables watchdog for channel 36, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
3 CWEN35	Watchdog Enable for Channel 35 Enables watchdog for channel 35, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
2 CWEN34	Watchdog Enable for Channel 34 Enables watchdog for channel 34, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
1 CWEN33	Watchdog Enable for Channel 33 Enables watchdog for channel 33, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled
0	Watchdog Enable for Channel 32

Table continues on the next page...

Table continued from the previous page...

Field	Description
CWEN32	Enables watchdog for channel 32, if set. 0 - Watchdog is disabled 1 - Watchdog is enabled

72.6.40 Analog Watchdog Out of Range 0 (AWORR0)

Indicates whether channel *n* converted data is out of range.

Offset

Register	Offset
AWORR0	2F0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								AWOR_CH7	AWOR_CH6	AWOR_CH5	AWOR_CH4	AWOR_CH3	AWOR_CH2	AWOR_CH1	AWOR_CH0
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7 AWOR_CH7	Analog Watchdog Out of Range for Channel 7 Indicates whether channel 7 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag
6 AWOR_CH6	Analog Watchdog Out of Range for Channel 6 Indicates whether channel 6 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag
5 AWOR_CH5	Analog Watchdog Out of Range for Channel 5 Indicates whether channel 5 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag
4 AWOR_CH4	Analog Watchdog Out of Range for Channel 4 Indicates whether channel 4 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - No effect 1 - Clear the flag
3 AWOR_CH3	Analog Watchdog Out of Range for Channel 3 Indicates whether channel 3 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag
2 AWOR_CH2	Analog Watchdog Out of Range for Channel 2 Indicates whether channel 2 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag
1 AWOR_CH1	Analog Watchdog Out of Range for Channel 1 Indicates whether channel 1 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 AWOR_CH0	Analog Watchdog Out of Range for Channel 0 Indicates whether channel 0 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag

72.6.41 Analog Watchdog Out of Range 1 (AWORR1)

Indicates whether channel *n* converted data is out of range.

Offset

Register	Offset
AWORR1	2F4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	AWOR_CH...	AWOR_CH...	AWOR_CH...	AWOR_CH...	AWOR_CH...	AWOR_CH...	AWOR_CH...	AWOR_CH...
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 AWOR_CH39	Analog Watchdog Out of Range for Channel 39 Indicates whether channel 39 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag
6 AWOR_CH38	Analog Watchdog Out of Range for Channel 38 Indicates whether channel 38 converted data is out of range.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag
5 AWOR_CH37	Analog Watchdog Out of Range for Channel 37 Indicates whether channel 37 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag
4 AWOR_CH36	Analog Watchdog Out of Range for Channel 36 Indicates whether channel 36 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag
3 AWOR_CH35	Analog Watchdog Out of Range for Channel 35 Indicates whether channel 35 converted data is out of range. **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag
2 AWOR_CH34	Analog Watchdog Out of Range for Channel 34 Indicates whether channel 34 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag
1 AWOR_CH33	Analog Watchdog Out of Range for Channel 33 Indicates whether channel 33 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range 1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag
0 AWOR_CH32	Analog Watchdog Out of Range for Channel 32 Indicates whether channel 32 converted data is out of range. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Converted data is in range

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Converted data is out of range When writing 0 - No effect 1 - Clear the flag

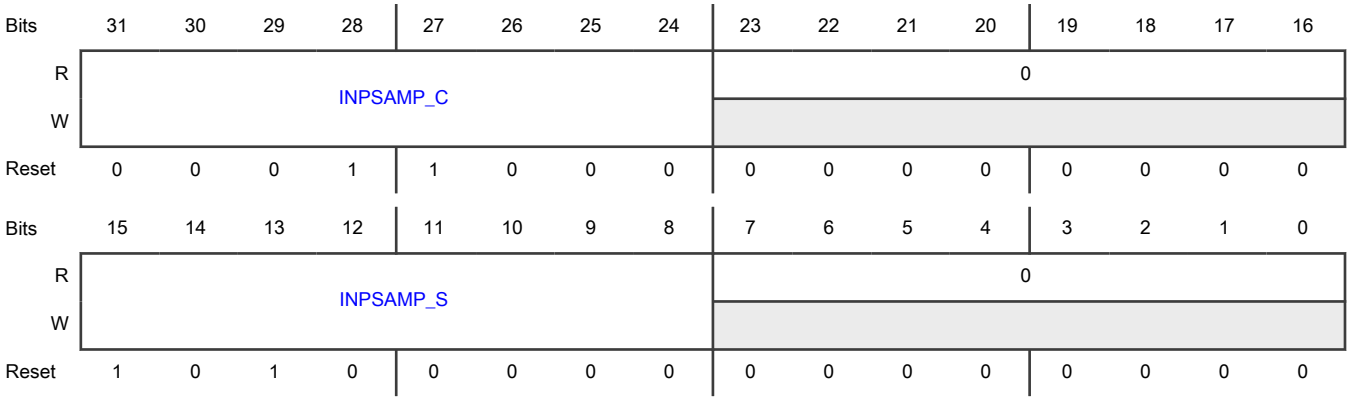
72.6.42 Self-Test Configuration 1 (STCR1)

Defines ADC self-testing capabilities in CPU mode . The ADC parameters not defined by the ADC self-testing registers, these are controlled according to the setting defined for the functional conversions.

Offset

Register	Offset
STCR1	340h

Diagram



Fields

Field	Description
31-24 INPSAMP_C	Sampling Configuration for Algorithm C Specifies the sampling phase duration for the test conversions related to Algorithm C.
23-16 —	Reserved
15-8 INPSAMP_S	Sampling Configuration for Algorithm S Specifies the sampling phase duration for the test conversions related to Algorithm S.

Table continues on the next page...

Table continued from the previous page...

Field	Description
7-0 —	Reserved

72.6.43 Self-Test Configuration 2 (STCR2)

Defines ADC self-testing capabilities in CPU mode .

Offset

Register	Offset
STCR2	344h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				MSKWDSE...	0	MSKWDTE...	0	MSKST_E...	0				MSKWDG_...	0	MSKWDG_...
W						SERR										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSKERR_C	0	MSKERR_...	MSKERR_...	MSKERR_...	0			EN	0		FMA_WDS...	FMA_WDT...	FMA_C	0	FMA_S
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Description
31-28 —	Reserved
27 MSKWDSERR	Watchdog Sequence Error Interrupt Mask Enables the Watchdog sequence error interrupt (STSR1[WDSERR] = 1). 0 - Interrupt disabled 1 - Interrupt enabled
26 SERR	Error Fault Injection Field (write-only) Sets STSR1[ERR_C/S0/S1/S2] only once if you write 1b to this field. Read always returns 0b. Setting this field will not affect any of the self-test data registers.
25	Watchdog Timer Error Interrupt Mask

Table continues on the next page...

Table continued from the previous page...

Field	Description
MSKWDTERR	Enables the Watchdog timer error interrupt (STSR1[WDTERR] = 1). 0 - Interrupt disabled 1 - Interrupt enabled
24 —	Reserved
23 MSKST_EOC	Self-Test EOC Interrupt Mask Enables the self-test EOC interrupt (STSR1[ST_EOC] = 1). Also, enables IMR[MSKEOC] for the self-test EOC interrupt to be asserted. 0 - Interrupt disabled 1 - Interrupt enabled
22-19 —	Reserved
18 MSKWDG_EOA_C	End of Algorithm C Interrupt Mask Enables the end of Algorithm C interrupt (STSR1[WDG_EOA_C] = 1). 0 - Interrupt disabled 1 - Interrupt enabled
17 —	Reserved
16 MSKWDG_EOA_S	End of Algorithm S Interrupt Mask Enables the end of Algorithm S interrupt (STSR1[WDG_EOA_S] = 1). 0 - Interrupt disabled 1 - Interrupt enabled
15 MSKERR_C	Error on Algorithm C Channel Interrupt Mask Enables the error on Algorithm C channel interrupt (STSR1[ERR_C] = 1). 0 - Interrupt disabled 1 - Interrupt enabled
14 —	Reserved
13 MSKERR_S2	Error on Algorithm S2 Channel Interrupt Mask Enables the error on Algorithm S2 channel (SUPPLY, step 2) interrupt (STSR1[ERR_S2] = 1). 0 - Interrupt disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Interrupt enabled
12 MSKERR_S1	Error on Algorithm S1 Channel Interrupt Mask Enables the error on Algorithm S1 channel (SUPPLY, step 1) interrupt (STSR1[ERR_S1] = 1). 0 - Interrupt disabled 1 - Interrupt enabled
11 MSKERR_S0	Error on Algorithm S0 Channel Interrupt Mask Enables the error on Algorithm S0 channel (SUPPLY, step0) interrupt (STSR1[ERR_S0] = 1). 0 - Interrupt disabled 1 - Interrupt enabled
10-8 —	Reserved
7 EN	Self-Testing Channel Enable Enables the TEST channel only in CPU mode. This field must be set before starting normal conversion and should not be changed while the conversion is in process. This field should only be reset after end of conversion of the last self-test channel has been received. 0 - Disable 1 - Enable
6-5 —	Reserved
4 FMA_WDSERR	Fault Mapping for Watchdog Sequence Error Specifies the type of fault mapping (noncritical fault (NCF) or critical fault (CF)) for watchdog sequence error. By default, a failure on the Watchdog Sequence is mapped to drive the NCF line. 0 - NCF mapping 1 - CF mapping
3 FMA_WDTERR	Fault Mapping for Watchdog Timer Error Specifies the type of fault mapping (noncritical fault (NCF) or critical fault (CF)) for watchdog timer error. By default, a failure on the Watchdog Timer is mapped to drive the NCF line. 0 - NCF 1 - CF
2 FMA_C	Fault Mapping for Algorithm C Specifies the type of fault mapping (noncritical fault (NCF) or critical fault (CF)) for algorithm C. By default, a failure during Algorithm C is mapped to drive the CF fault line. 0 - NCF

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - CF
1 —	Reserved
0 FMA_S	Fault Mapping for BGAP Algorithm Specifies the type of fault mapping (noncritical fault (NCF) or critical fault (CF)) for BGAP algorithm. By default, a failure during the supply algorithm is mapped to drive the Critical Fault (CF) line. 0 - NCF 1 - CF

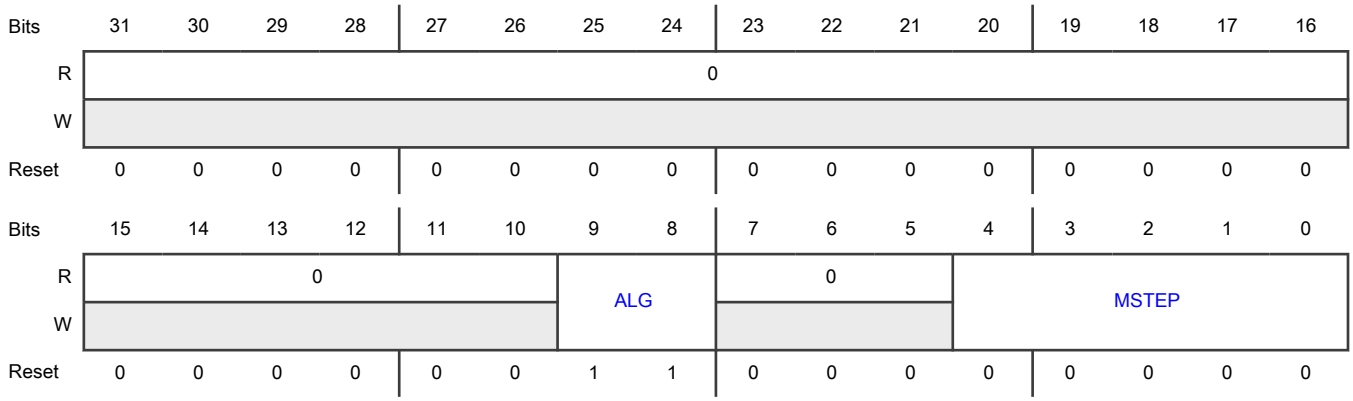
72.6.44 Self-Test Configuration 3 (STCR3)

Defines ADC self-testing capabilities in CPU mode (MCR[CTUEN] = 0). You must program this register before starting the conversion, and must not change the programming while conversion is in process.

Offset

Register	Offset
STCR3	348h

Diagram



Fields

Field	Description
31-10 —	Reserved
9-8	Self-Test Algorithm Select

Table continues on the next page...

Table continued from the previous page...

Field	Description
ALG	Specifies the algorithm for self-test. In One-Shot Operation mode, for test and debug purposes, STCR3[MSTEP] selects the step for self-test. In Scan Operation mode, the step for self-test is automatically selected based on the previous step and the value of this field. The configuration of Self-Test Baud Rate (STBRR) defines the execution baud rate of algorithm C. 00 - Algorithm S 01 - Reserved 10 - Algorithm C 11 - Algorithm S (for One-Shot Operation mode); Algorithm S + C (for Scan Operation mode)
7-5 —	Reserved
4-0 MSTEP	Self-Test Step Select Selects self-test step. For One-Shot Operation mode: Current step for Algorithm S/ C. • MSTEP = 0 to NUM_SUPPLY_STEPS (3) - 1 for Algorithm S • MSTEP = 0 to NUM_C_STEPS (12) - 1 for Algorithm C The unused codes are RESERVED (should not be used). This field is not used in Scan Operation mode as single-step execution (interleaved mode) is always performed. It should be programmed to zero in Scan Operation mode.

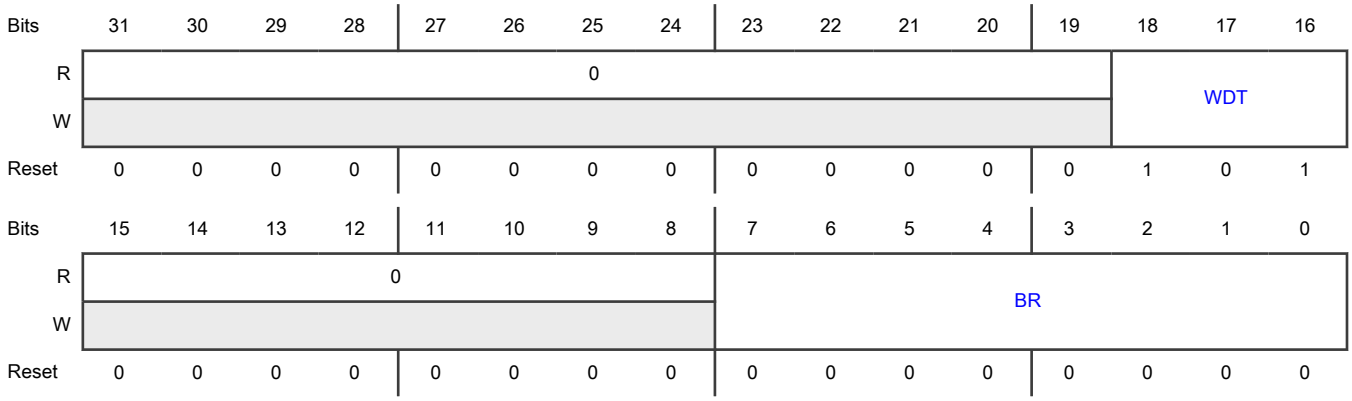
72.6.45 Self-Test Baud Rate (STBRR)

Specifies the self-test baud rate.

Offset

Register	Offset
STBRR	34Ch

Diagram



Fields

Field	Description
31-19 —	Reserved
18-16 WDT	Watchdog Timer Value Specifies the watchdog timer value that is used to monitor the algorithm sequence to verify that it is correctly executing within the safe time period. The self-testing watchdog is enabled by setting STAW *n* R[WDTE]. Default value is 10 ms. A fixed prescaler of 1024 is used to obtain the timer value from the ADCdig clock (80 MHz). Prescaler value is fixed at 1024. 000 - 0.1 ms ((0008h × Prescaler) cycles at 80 MHz) 001 - 0.5 ms ((0027h × Prescaler) cycles at 80 MHz) 010 - 1 ms ((004Eh × Prescaler) cycles at 80 MHz) 011 - 2 ms ((009Ch × Prescaler) cycles at 80 MHz) 100 - 5 ms ((0187h × Prescaler) cycles at 80 MHz) 101 - 10 ms ((030Dh × Prescaler) cycles at 80 MHz) 110 - 20 ms (061Ah × Prescaler) cycles at 80 MHz) 111 - 50 ms (0F42h ×Prescaler) cycles at 80 MHz)
15-8 —	Reserved
7-0 BR	Algorithm Baud Rate Specifies the baud rate for the selected algorithm in Scan Operation mode (MCR[MODE] = 1). 00h ≥ maximum scheduling rate (NOMINAL rate) ... FFh ≥ minimum scheduling rate (NOMINAL rate scaled by 255) You must write to this field before enabling a self-test channel.

72.6.46 Self-Test Status 1 (STSR1)

Indicates self-test status 1.

Offset

Register	Offset
STSR1	350h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				WDSERR	0	WDTERR	OVERWR	ST_EOC	0				WDG_EOA...	0	WDG_EOA...
W					W1C		W1C	W1C	W1C					W1C		W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_C	0	ERR_S2	ERR_S1	ERR_S0	0	STEP_C					0				
W	W1C		W1C	W1C	W1C		W1C									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27 WDSERR	Watchdog Sequence Errors Indicates whether a watchdog sequence error of the ADC subsystem (check for algorithm step sequence) occurred. If a failure occurs, an interrupt will be generated if enabled (STCR2[MSKWDSERR] = 1). It provides the fault indication to the FCCU, asserting a CF or NCF according to the STCR2[FMA_WDSERR] configuration. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No failure 1 - Failure occurred When writing 0 - No effect 1 - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
26 —	Reserved
25 WDTERR	Watchdog Timer Error Indicates whether a watchdog timer error of the ADC subsystem (algorithm check for completion within safe time) occurred. If a failure occurs, an interrupt will be generated if enabled (STCR2[MSKWDTERR] = 1). It provides the fault indication to the FCCU, asserting a CF or NCF according to the STCR2[FMA_WDTERR] configuration. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No failure 1 - Failure occurred When writing 0 - No effect 1 - Clear the flag
24 OVERWR	Overwrite Error Notifies when STSR1[ERR*x*] is overwritten by a new error. The new error status is written or discarded according to the MCR[OWREN] value. To avoid Overwrite Error indication, STSR1[ERR*x*] must be cleared by software. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No overwrite error 1 - Overwrite error occurred When writing 0 - No effect 1 - Clear the flag
23 ST_EOC	Self-Test EOC Indicates whether self-test end of conversion is complete. This field is set along with EOC bit when end of conversion occurs from the analog SAR_ADC for self-test channel. An interrupt is generated if STCR2[MSKST_EOC] = 1, and IMR[MSKEOC] = 1. Writing 0 has no effect. **NOTE** This field behaves differently for register reads and writes. When reading

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Self-test end of conversion is not complete 1 - Self-test end of conversion is complete When writing 0 - No effect 1 - Clear the flag
22-19 —	Reserved
18 WDG_EOA_C	Watchdog End of Algorithm C Indicates that Algorithm C has completed. This field is set after the last STEP of algorithm is executed. An interrupt is generated if STCR2[MSKWDG_EOA_C] = 1. This field will set only if STAW4R[WDTE] = 1. Writing 0 has no effect. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Self-test end of Algorithm C conversion is not complete 1 - Self-test end of Algorithm C conversion is complete When writing 0 - No effect 1 - Clear the flag
17 —	Reserved
16 WDG_EOA_S	Watchdog End of Algorithm S Indicates whether Algorithm S is complete. This field is set after the last Step of Algorithm S is executed. An interrupt is generated if STCR2[MSKWDG_EOA_S] = 1. This field is set only if STAW0R[WDTE] = 1. Writing 0 has no effect. **NOTE** This field behaves differently for register reads and writes. When reading 0 - Self-test end of Algorithm S conversion is not complete. 1 - Self-test end of Algorithm S conversion is complete. When writing 0 - No effect 1 - Clear the flag

Table continues on the next page...

Table continued from the previous page...

Field	Description
15 ERR_C	Algorithm C Error Indicates an error on the self-testing channel (Algorithm C). An interrupt is generated if STCR2[MSKERR_C] = 1. It provides the fault indication to the FCCU, asserting the programmed fault line (STCR2[FMA_ *x*]). STSR1[ERR_C] can also be set by software (fault injection) by writing 1 to STCR2[SERR]. In this case, a CF or NCF will be asserted according to the STCR2[FMA_ *x*] configuration. Writing 0 has no effect. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No Algorithm C error 1 - Algorithm C error occurred When writing 0 - No effect 1 - Clear the flag
14 —	Reserved
13 ERR_S2	Algorithm S2 Error Indicates an error on the self-test channel (Algorithm S (SUPPLY), step2). An interrupt is generated if STCR2[MSKERR_S2] = 1. It provides the fault indication to the FCCU, asserting the programmed fault line (STCR2[FMA_ *x*]). STSR1[ERR_S2] can be also set by software (fault injection) by writing 1 to STCR2[SERR]. In this case, a CF or NCF will be asserted according to the STCR2[FMA_ *x*] configuration. Writing 0 has no effect. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No error occurred on the sampled signal 1 - Error occurred on the sampled signal When writing 0 - No effect 1 - Clear the flag
12 ERR_S1	Algorithm S1 Error Indicates an error on the self-test channel (Algorithm S (SUPPLY), step1). An interrupt is generated if STCR2[MSKERR_S1] = 1. It provides the fault indication to the FCCU, asserting the programmed fault line (STCR2[FMA_ *x*]). STSR1[ERR_S1] can be also set by software (fault injection) by writing 1 to STCR2[SERR]. In this case, a CF or NCF will be asserted according to the STCR2[FMA_ *x*] configuration. Writing 0 has no effect.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE This field behaves differently for register reads and writes. When reading 0 - No VDD ERROR 1 - VDD ERROR occurred When writing 0 - No effect 1 - Clear the flag
11 ERR_S0	Algorithm S0 Error Indicates an error on the self-test channel (Algorithm S (SUPPLY), step0). An interrupt is generated if STCR2[MSKERR_S0] = 1. It provides the fault indication to the FCCU, asserting the programmed fault line (STCR2[FMA_ *x*]). STSR1[ERR_S0] can be also set by software (fault injection) by writing 1 to STCR2[SERR]. In this case, a CF or NCF will be asserted according to the STCR2[FMA_ *x*] configuration. Writing 0 has no effect. **NOTE** This field behaves differently for register reads and writes. When reading 0 - No VREF error 1 - VREF error occurred When writing 0 - No effect 1 - Clear the flag
10 —	Reserved
9-5 STEP_C	Algorithm C Step Number Error Indicates step of the Algorithm C step when an Algorithm C error occurred (as indicated by STSR1[ERR_C]). 0...(NUM_C_STEPS (12)-1) ≥ Algorithm C
4-0 —	Reserved

72.6.47 Self-Test Status 2 (STSR2)

Indicates self-test status 2.

Offset

Register	Offset
STSR2	354h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OVFL	0			DATA1											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DATA0											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 OVFL	Overflow Bit Indicates whether an overflow occurred. This flag is set when divisor is zero. Also, STSR1.ERR_S1 is set if overflow occurs. 0 - Not overflow 1 - Overflow
30-28 —	Reserved
27-16 DATA1	Test Channel Converted Data when ERR_S1 Occurred Indicates test channel converted data when the ERR_S1 error occurs. Algorithm S (step1) ≥ fractional part of the ratio TEST (step1)/TEST (step0) = VDD/VBGAP.
15-12 —	Reserved
11-0 DATA0	Test Channel Converted Data when ERR_S1 Occurred Indicates test channel converted data when the ERR_S1 error occurs. Algorithm S (step1) ≥ integer part of the ratio TEST (step1)/TEST (step0) = VDD/VBGAP.

72.6.48 Self-Test Status 3 (STSR3)

Indicates self-test status 3.

Offset

Register	Offset
STSR3	358h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				DATA1											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DATA0											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-16 DATA1	Test Channel Converted Data when ERR_C Occurred Indicates test channel converted data when the ERR_S2 error occurs. Algorithm S (step2) ≥ TEST channel data = VREF/VREF
15-12 —	Reserved
11-0 DATA0	Test Channel Converted Data when ERR_S0 Occurred Indicates test channel converted data when the ERR_S0 error occurs. Algorithm S (step0) ≥ TEST channel data = VBGAP/VREF

72.6.49 Self-Test Status 4 (STSR4)

Indicates self-test status 4.

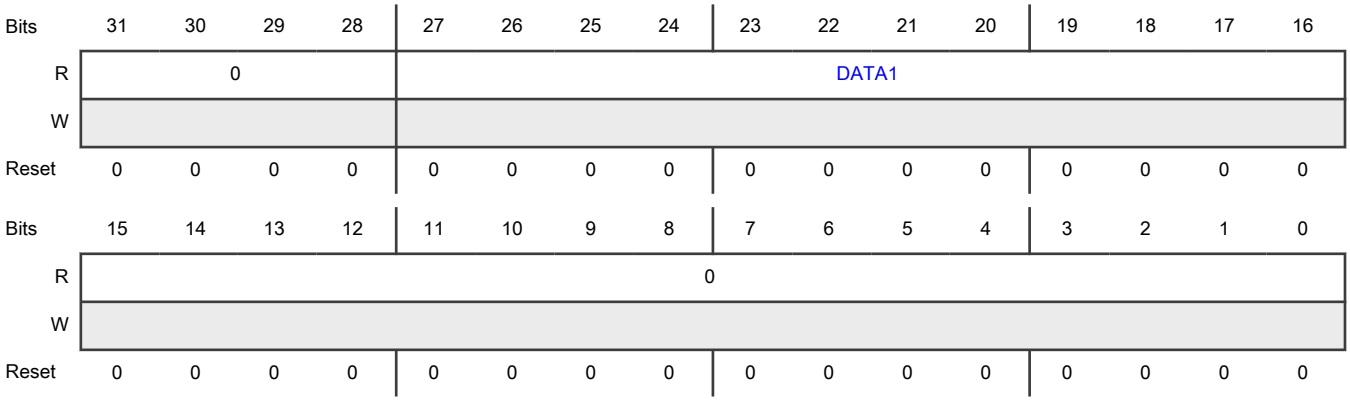
NOTE

If MCR.OVERWR is set, registers STSR[2:4] are overwritten if another error occurs.

Offset

Register	Offset
STSR4	35Ch

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 DATA1	Test Channel Converted Data When ERR_C Occurred Indicates test channel converted data when ERR_C occurs. Algorithm C ≥ TEST channel data.
15-0 —	Reserved

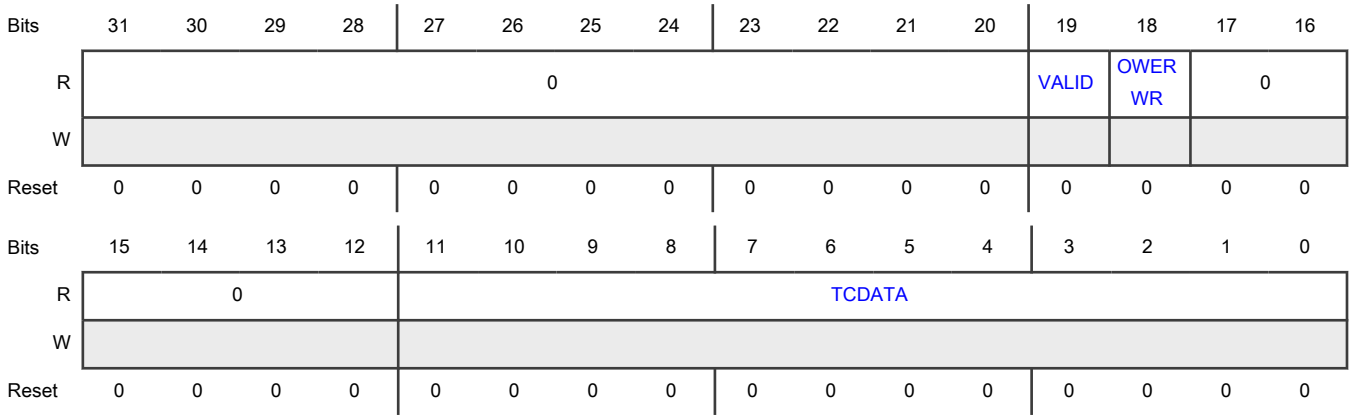
72.6.50 Self-Test Data 1 (STDR1)

Indicates self-test data 1.

Offset

Register	Offset
STDR1	370h

Diagram



Fields

Field	Description
31-20 —	Reserved
19 VALID	Valid Data Indicates whether data is valid (a new value is written). The field clears automatically when data is read. 0 - Data not valid 1 - Data valid
18 OWERWR	Overwrite Data Indicates whether conversion data is overwritten by a newer result. The new data is written or discarded according to the MCR[OWREN] value. 0 - Data not overwritten 1 - Data overwritten
17-12 —	Reserved
11-0 TCDATA	Test Channel Converted Data Indicates test channel data that is in 2's complement format.

72.6.51 Self-Test Data 2 (STDR2)

Indicates self-test data 2.

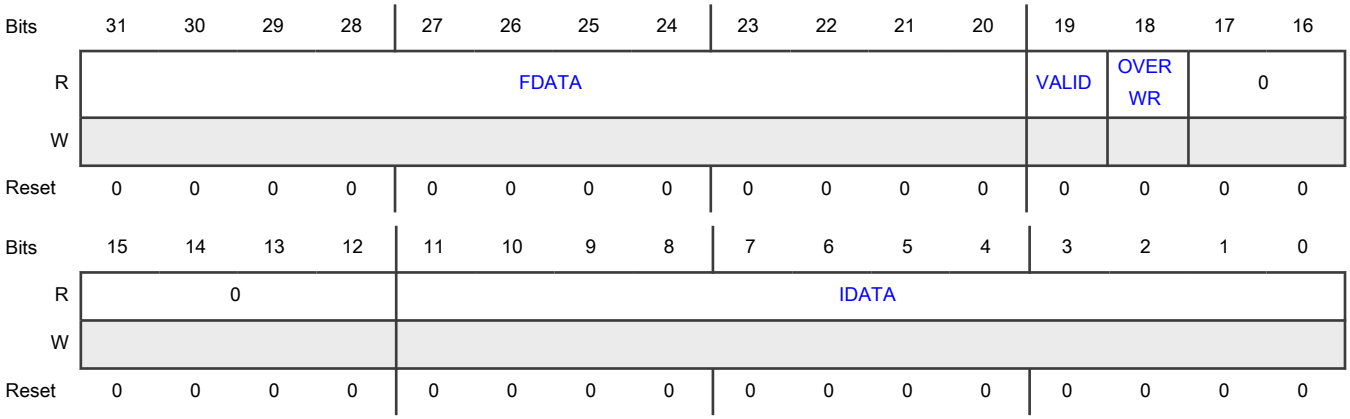
NOTE

If an overflow error occurs for Step1 of Algorithm S, STDR2 is not written.

Offset

Register	Offset
STDR2	374h

Diagram



Fields

Field	Description
31-20 FDATA	Fractional Data Indicates fractional data. The fractional part of the ratio TEST (step1)/TEST (step0) = VDD/VBGAP for Algorithm S.
19 VALID	Valid Data Indicates when data is valid (a new value is written). This field clears automatically when data is read. 0 - Data not valid 1 - Data valid
18 OVERWR	Overwrite Data Indicates when conversion data is overwritten by a newer result. The new data is written or discarded according to the MCR[OWREN] configuration. 0 - Data not overwritten 1 - Data overwritten
17-12 —	Reserved
11-0 IDATA	Integer Data Indicates integer data. The integer part of the ratio TEST (step1)/TEST (step0) = VDD/VBGAP for Algorithm S.

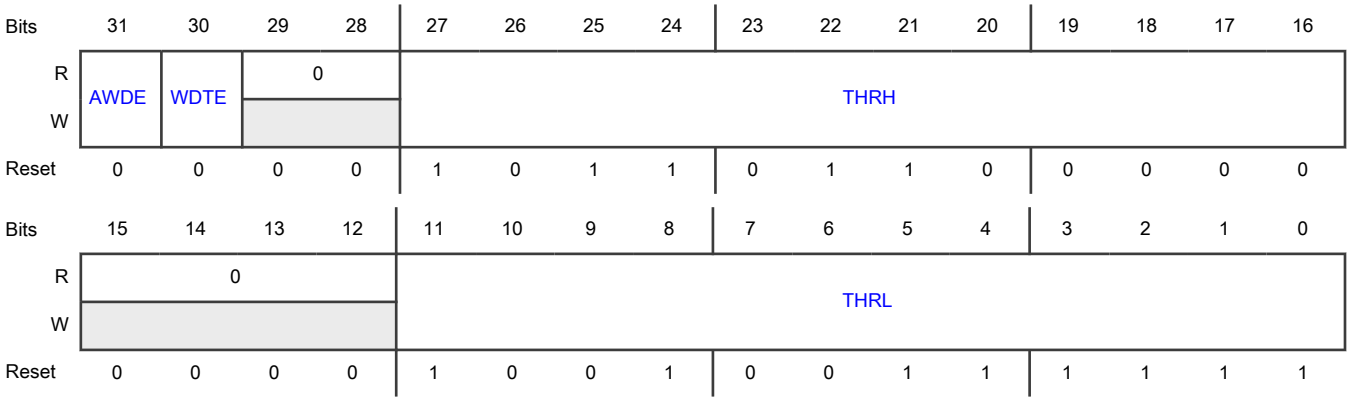
72.6.52 Self-Test Analog Watchdog 0 (STAW0R)

Contains self-test analog watchdog 0 fields.

Offset

Register	Offset
STAW0R	380h

Diagram



Fields

Field	Description
31 AWDE	Analog Watchdog Enable Enables the comparison of the conversion result from the ADC supply self-test step0 to the thresholds contained in this register (THRH and THRL). 0 - Disabled 1 - Enabled
30 WDTE	Watchdog Timer Enable Enables the watchdog timer monitoring function (related to the Algorithm S) for all ADC supply self-test steps. The watchdog timer verifies: • Correct sequence of the algorithm (step sequence) • Execution of the algorithm within the safe time period as defined by STBRR[WDT] As soon as the watchdog timer is enabled the algorithm execution must be detected within the safe time period. The watchdog timer is reset each time the algorithm restarts. Set this field only in Scan mode. 0 - Disabled 1 - Enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
29-28 —	Reserved
27-16 THRH	High Threshold Value for Algorithm S Step0 Specifies the high threshold value for Algorithm S Step0. If the analog watchdog is enabled, STSR1[ERR_ *x*] is set to 1 if STDR1[TCDATA] > THRH. The reset value of this field is calculated based on a bandgap voltage of 1.15 V. The formula for calculating the higher threshold is (Bandgap_Voltage_Maximum)*4095/(Reference_Voltage_Minimum). **NOTE** The recommended watchdog threshold values are set according to expected and tested results in a noise-controlled environment (for example, introduction of noise from outside of the device minimized). The setting may need to be adjusted to prevent a given threshold value from falsely reporting fault detections in applications operating in noisier environments.
15-12 —	Reserved
11-0 THRL	Low Threshold Value for Algorithm S Step0 Specifies the low threshold value for Algorithm S Step0. If the analog watchdog is enabled, STSR1[ERR_ *x*] is set to 1 if STDR1[TCDATA] < THRL. The reset value of this field is calculated based on a bandgap voltage of 1.15V. The formula for calculating the lower threshold is (Bandgap_Voltage_Minimum)*4095/(Reference_Voltage_Maximum). **NOTE** The recommended watchdog threshold values are set according to expected and tested results in a noise-controlled environment (for example, introduction of noise from outside of the device minimized). This value may need to be adjusted to prevent a given threshold value from falsely reporting fault detections in applications operating in noisier environments.

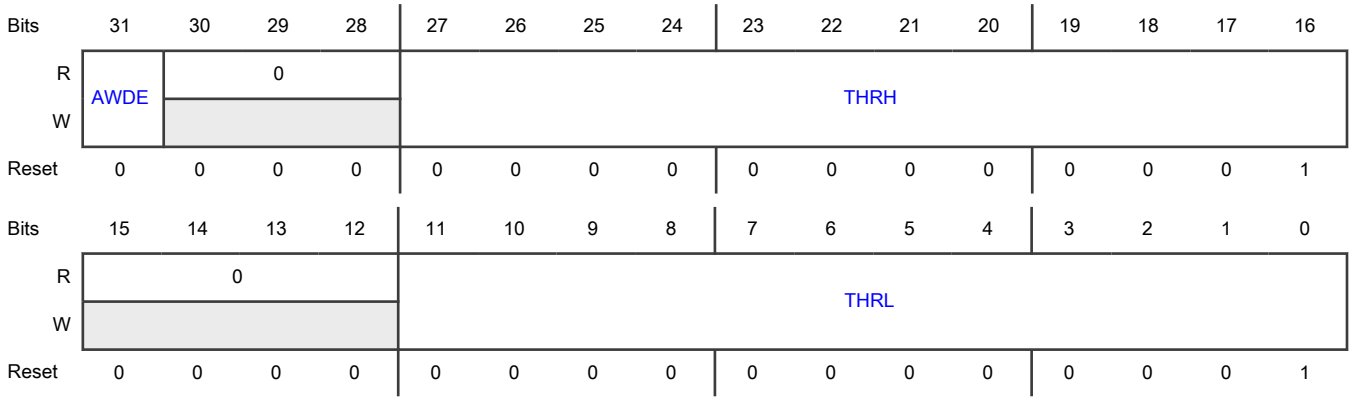
72.6.53 Self-Test Analog Watchdog 1A (STAW1AR)

Contains self-test analog watchdog 1A fields.

Offset

Register	Offset
STAW1AR	384h

Diagram



Fields

Field	Description
31 AWDE	Analog Watchdog Enable Enables analog watchdog related to Algorithm S (step1). 0 - Disabled 1 - Enabled
30-28 —	Reserved
27-16 THRH	High Threshold Value for Algorithm S Step1 Specifies the high threshold value (integer part) for Algorithm S (step1) test channel (unsigned coding).
15-12 —	Reserved
11-0 THRL	Low Threshold Value for Algorithm S Step1 Specifies the low threshold value (integer part) for Algorithm S (step1) test channel (unsigned coding).

72.6.54 Self-Test Analog Watchdog 1B (STAW1BR)

Contains fields for self-test analog watchdog 1B.

Offset

Register	Offset
STAW1BR	388h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				THRH											
W																
Reset	0	0	0	0	1	0	1	1	0	0	1	1	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				THRL											
W																
Reset	0	0	0	0	0	1	0	1	1	0	0	0	0	1	1	0

Fields

Field	Description
31-28 —	Reserved
27-16 THRH	High Threshold Value for Algorithm S Step1 Specifies the high threshold value (fractional part) for Algorithm S (step1) test channel (unsigned coding).
15-12 —	Reserved
11-0 THRL	Low Threshold Value for Algorithm S Step1 Specifies the low threshold value (fractional part) for Algorithm S (step1) test channel (unsigned coding).

72.6.55 Self-Test Analog Watchdog 2 (STAW2R)

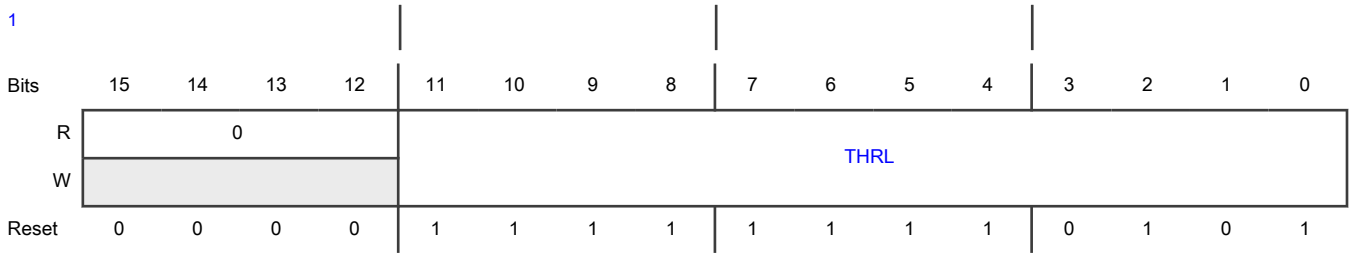
Contains fields for self-test analog watchdog 2.

Offset

Register	Offset
STAW2R	38Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	AWDE	0														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



1. 12-bit instance

Fields

Field	Description
31 AWDE	Analog Watchdog Enable Enables analog watchdog related to Algorithm S (step2). 0 - Disabled 1 - Enabled
30-12 —	Reserved
11-0 THRL	Threshold level low Specifies the low threshold value for channel *n* (unsigned coding). If the analog watchdog is enabled, STSR1[ERR_S2] is set if STDR1[TCDATA] < THRL.

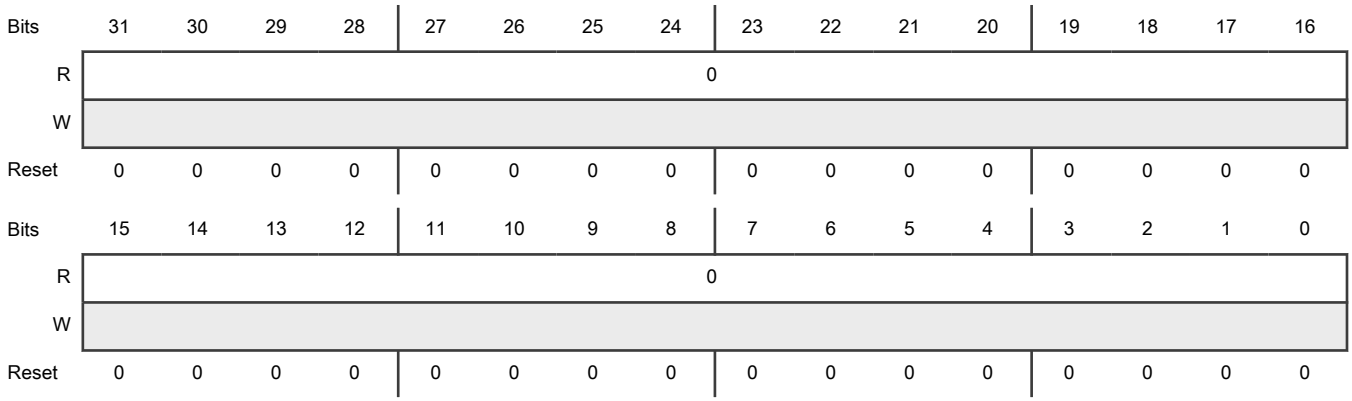
72.6.56 Self-Test Analog Watchdog 3 (STAW3R)

Contains fields for self-test analog watchdog 3.

Offset

Register	Offset
STAW3R	390h

Diagram



Fields

Field	Description
31-0 —	Reserved

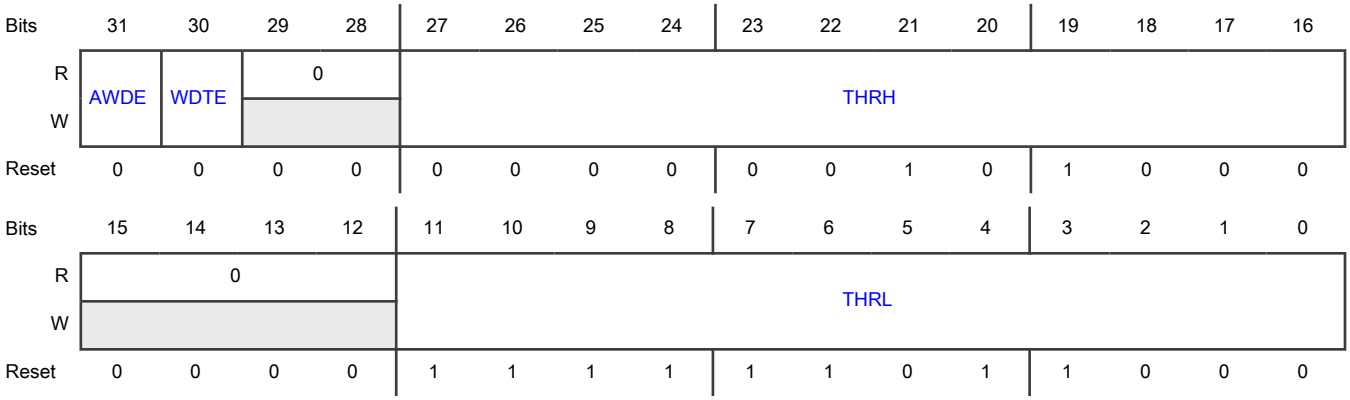
72.6.57 Self-Test Analog Watchdog 4 (STAW4R)

Contains fields for self-test analog watchdog 4.

Offset

Register	Offset
STAW4R	394h

Diagram



Fields

Field	Description
31 AWDE	Analog Watchdog Enable Enables the analog watchdog monitoring function (related to Algorithm C) for all ADC capacitive self-test steps. 0 - Disabled 1 - Enabled
30 WDTE	Watchdog Timer Enable Enables the watchdog timer monitoring function (related to Algorithm C) for all ADC capacitive self-test steps. The watchdog timer verifies:

Table continues on the next page...

Table continued from the previous page...

Field	Description
	• Correct sequence of the algorithm (step sequence) • Execution of the algorithm within the safe time period as defined by STBRR[WDT] As soon as the watchdog timer is enabled, execution must be detected within the safe time period. The watchdog timer is reset each time the algorithm restarts. This field should be set only in scan mode. 0 - Disabled 1 - Enabled
29-28 —	Reserved
27-16 THRH	High Threshold Value for Step0 of Algorithm C Specifies the high threshold value for Step0 of Algorithm C. This value is read by the ADC as two's-complement, but should be written positive. If the watchdog is enabled (STAW4R[AWDE] = 1) and STDR1[TCDATA] > STAW4R[THRH], then STSR1[ERR_C] is set. **NOTE** The recommended watchdog threshold values are set according to expected and tested results in a noise-controlled environment (for example, introduction of noise from outside of the device minimized). The setting may need to be adjusted to prevent a given threshold value from falsely reporting fault detections in applications operating in noisier environments.
15-12 —	Reserved
11-0 THRL	Low Threshold Value for Step0 of Algorithm C Specifies the low threshold value for Step0 of Algorithm C. This value is read by the ADC as two's-complement, and it should be the inverse of STAW4R[THRH]. If the watchdog is enabled (STAW4R[AWDE] = 1) and STDR1[TCDATA] < STAW4R[THRL], then STSR1[ERR_C] is set. The settings in this register are applicable only to step0 of the ADC's capacitive self-test. Refer to the STAW5R register for settings applicable to other capacitive self-test steps. The recommended watchdog threshold values are set according to expected and tested results in a noise-controlled environment (for example, introduction of noise from outside of the device minimized). The setting may need to be adjusted to prevent a given threshold value from falsely reporting fault detections in applications operating in noisier environments.

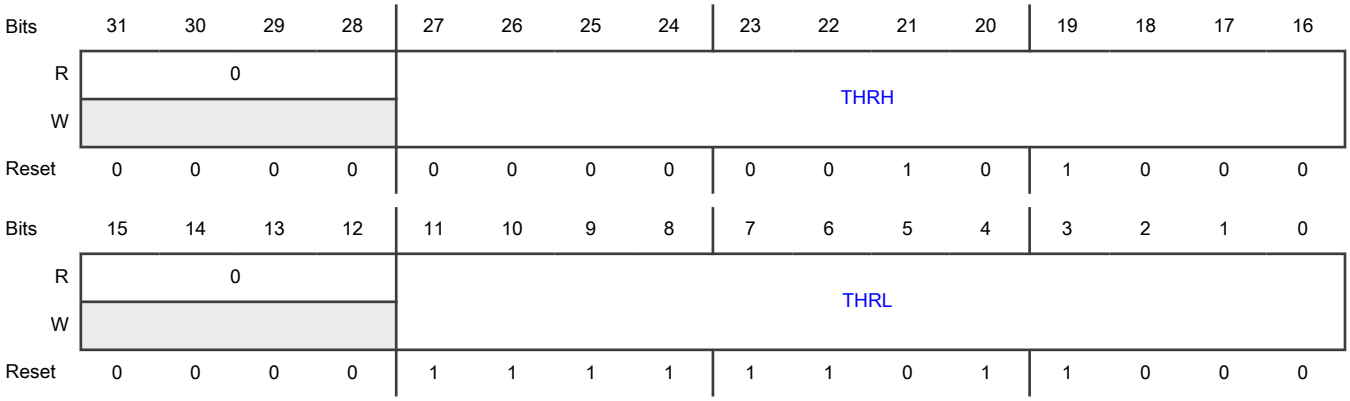
72.6.58 Self-Test Analog Watchdog 5 (STAW5R)

Contains fields for self-test analog watchdog 5.

Offset

Register	Offset
STAW5R	398h

Diagram



Fields

Field	Description
31-28 —	Reserved
27-16 THRH	High Threshold Value for Step N of Algorithm C Specifies the high threshold value Step N of Algorithm (N = 1 to CS-1). This value is read by the ADC as two's-complement, but should be written positive. If the watchdog is enabled (STAW4R[AWDE] = 1) and (STDR1[TCDATA] (step-N)–STDR1[TCDATA] (step-0)) > STAW5R[THRH], then STSR1[ERR_C] is set. **NOTE** The recommended watchdog threshold values are set according to expected and tested results in a noise-controlled environment (for example, introduction of noise from outside of the device minimized). The setting may need to be adjusted to prevent a given threshold value from falsely reporting fault detections in applications operating in noisier environments.
15-12 —	Reserved
11-0 THRL	Low Threshold Value for Step N of Algorithm C Specifies the low threshold value for Step N of Algorithm C (N = 1 to CS-1). This value is read by the ADC as two's-complement, and it should be the inverse of STAW5R[THRH]. If the watchdog is enabled (STAW4R[AWDE = 1]) and (STDR1[TCDATA] (step-N)–STDR1[TCDATA] (step-0)) < STAW5R[THRL], then STSR1[ERR_C] is set.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE The recommended watchdog threshold values are set according to expected and tested results in a noise-controlled environment (for example, introduction of noise from outside of the device minimized). The setting may need to be adjusted to prevent a given threshold value from falsely reporting fault detections in applications operating in noisier environments.

72.6.59 Calibration Status (CALSTAT)

Indicates status of calibration steps. See Calibration for more information on calibration steps.

Offset

Register	Offset
CALSTAT	39Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TEST_RESULT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0	0	STAT_12	STAT_11	STAT_10	STAT_9	STAT_8	STAT_7	STAT_6	STAT_5	STAT_4	STAT_3	STAT_2	STAT_1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 TEST_RESULT	TEST_RESULT Indicates test result for last failing test ($FIXED_VALUE–$RAW_SAR_DATA). Bit [31] is a sign bit, and negative numbers are in two's complement.
15-14 —	Reserved
13	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
12 —	Reserved
11 STAT_12	Status of Calibration Step 12 Indicates status of test *n* (where *n* = 1...12). This field is cleared when MCR[CALSTART] = 1, which initiates the test. This field updates automatically as the test progresses. Bit for each test is updated dynamically during test and the test number field reflects the last failing test. 0 - Test passed 1 - Test failed
10 STAT_11	Status of Calibration Step 11 Indicates status of test *n* (where *n* = 1...12). This field is cleared when MCR[CALSTART] = 1, which initiates the test. This field updates automatically as the test progresses. Bit for each test is updated dynamically during test and the test number field reflects the last failing test. 0 - Test passed 1 - Test failed
9 STAT_10	Status of Calibration Step 10 Indicates status of test *n* (where *n* = 1...12). This field is cleared when MCR[CALSTART] = 1, which initiates the test. This field updates automatically as the test progresses. Bit for each test is updated dynamically during test and the test number field reflects the last failing test. 0 - Test passed 1 - Test failed
8 STAT_9	Status of Calibration Step 9 Indicates status of test *n* (where *n* = 1...12). This field is cleared when MCR[CALSTART] = 1, which initiates the test. This field updates automatically as the test progresses. Bit for each test is updated dynamically during test and the test number field reflects the last failing test. 0 - Test passed 1 - Test failed
7 STAT_8	Status of Calibration Step 8 Indicates status of test *n* (where *n* = 1...12).

Table continues on the next page...

Table continued from the previous page...

Field	Description
	This field is cleared when MCR[CALSTART] = 1, which initiates the test. This field updates automatically as the test progresses. Bit for each test is updated dynamically during test and the test number field reflects the last failing test. 0 - Test passed 1 - Test failed
6 STAT_7	Status of Calibration Step 7 Indicates status of test *n* (where *n* = 1...12). This field is cleared when MCR[CALSTART] = 1, which initiates the test. This field updates automatically as the test progresses. Bit for each test is updated dynamically during test and the test number field reflects the last failing test. 0 - Test passed 1 - Test failed
5 STAT_6	Status of Calibration Step 6 Indicates status of test *n* (where *n* = 1...12). This field is cleared when MCR[CALSTART] = 1, which initiates the test. This field updates automatically as the test progresses. Bit for each test is updated dynamically during test and the test number field reflects the last failing test. 0 - Test passed 1 - Test failed
4 STAT_5	Status of Calibration Step 5 Indicates status of test *n* (where *n* = 1...12). This field is cleared when MCR[CALSTART] = 1, which initiates the test. This field updates automatically as the test progresses. Bit for each test is updated dynamically during test and the test number field reflects the last failing test. 0 - Test passed 1 - Test failed
3 STAT_4	Status of calibration step 4 Indicates status of test *n* (where *n* = 1...12). This field is cleared when MCR[CALSTART] = 1, which initiates the test. This field updates automatically as the test progresses. Bit for each test is updated dynamically during test and the test number field reflects the last failing test. 0 - Test passed 1 - Test failed
2	Status of Calibration Step 3

Table continues on the next page...

Table continued from the previous page...

Field	Description
STAT_3	Indicates status of test *n* (where *n* = 1...12). This field is cleared when MCR[CALSTART] = 1, which initiates the test. This field updates automatically as the test progresses. Bit for each test is updated dynamically during test and the test number field reflects the last failing test. 0 - Test passed 1 - Test failed
1 STAT_2	Status of Calibration Step 2 Indicates status of test *n* (where *n* = 1...12). This field is cleared when MCR[CALSTART] = 1, which initiates the test. This field updates automatically as the test progresses. Bit for each test is updated dynamically during test and the test number field reflects the last failing test. 0 - Test passed 1 - Test failed
0 STAT_1	Status of Calibration Step 1 Indicates status of test *n* (where *n* = 1...12). This field is cleared when MCR[CALSTART] = 1, which initiates the test. This field updates automatically as the test progresses. Bit for each test is updated dynamically during test and the test number field reflects the last failing test. 0 - Test passed 1 - Test failed

72.6.60 Calibration Configuration 0 (CALCFG0)

This register defines settings for calibration (RW, Reserved, non_cust, fuse bit).

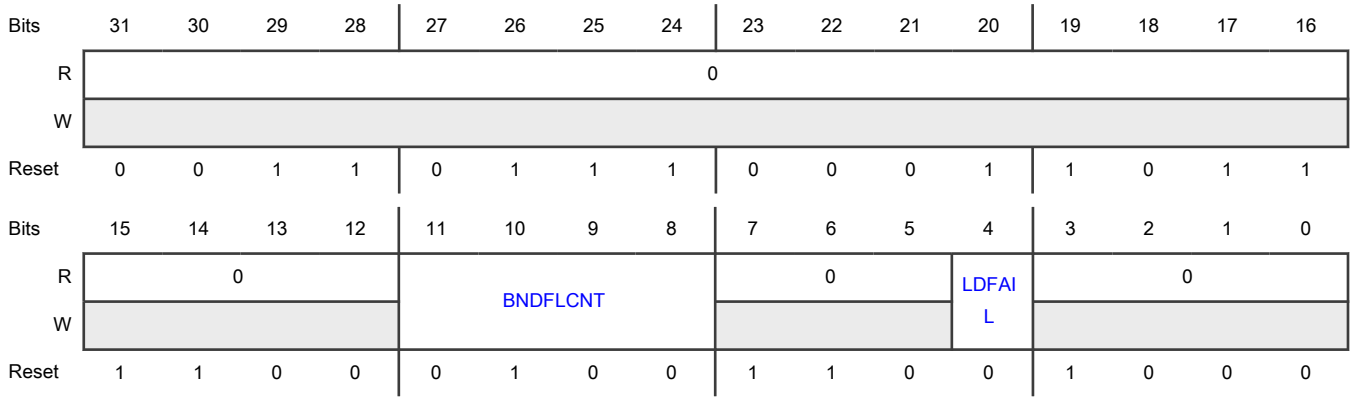
NOTE

LDFAIL and BNDFLCNT bits can be used if DC-to-DC converter is used as a source of ADC supply (reference) and calibration fails. For more details, contact the NXP representative."

Offset

Register	Offset
CALCFG0	3A0h

Diagram



Fields

Field	Description
31-12 —	Reserved
11-8 BNDFLCNT	Boundary condition fail count Boundary condition is checked in all individual conversions of calibration. The condition is (2^BNDFLCNT). 0000 - Single occurrence 0001 - Double occurrence 0010 - Four occurrence
7-5 —	Reserved
4 LDFAIL	Load calibrated values in fail condition 0 - Load calibrated values only if the calibration step passes (retain existing values in case of fail) (default) 1 - Enable loading of calibrated values in fail condition also
3-0 —	Reserved

Chapter 73
Thermal Monitoring Unit (TMU)

73.1 Chip-specific TMU information

Table 637. Reference links to related information

Topic	Related module	Reference
CM33	Arm Cortex CM33 Platform (CM33)	CM33 Platform
A55	Arm Cortex CA55 Platform (CA55)	CA55 Platform
System memory map		System memory map
Clocking		Clock generation
Signal multiplexing	Port control	External signals and pin multiplexing
System Debug		System Debug Overview

This chip has one temperature monitoring site (remote site) in the ANAMIX, next to the temperature diode.

In Block diagram only remote site 0 applies to this chip.

73.1.1 Module instances

This device has one instance of the TMU module.

73.1.2 Interrupts

This chip has one TMU Interrupt with ID 83.

When the temperature reaches one of the enabled thresholds at the enabled site, TMU sets:

- A flag that corresponds to that threshold in the Interrupt Detect register
- A flag in one of the Site Capture registers to indicate which site reached the threshold
- These registers should be examined and then cleared in the ISR to retrieve details of the interrupt and to prepare them for the next one.

73.1.3 TMU configuration

The Thermal Monitoring Unit should be properly configured before the monitoring mode can be enabled. Below is the recommended procedure and code examples for configuring TMU for the first time after reset (monitoring is disabled).

1. Clear (W1C) the 8 MSBs in the Interrupt Detect register (TIDR)

   ```
   TMU.TIDR.R |= 0xFF000000; // clear int. detect reg.
   ```

2. Clear the 3 LSBs in the interrupt site capture registers:
   - Interrupt Immediate Site Capture register (TIISCR)
   - Interrupt Average Site Capture register (TIASCR)
   - Interrupt Critical Site Capture register (TICSCR)

   ```
   TMU.TIISCR.R & 0xFFFFFFF8; // clear site capture reg.
   TMU.TIASCR.R &= 0xFFFFFFF8; // clear site capture reg.
   ```

```
TMU.TICSCR.R &= 0xFFFFFFF8; // clear site capture reg.
```

3. Enable required interrupts (8 MSBs) in the Interrupt Enable register (TIER)

```
TMU.TIER.R |= 0xFF000000; // enable all interrupts
```

4. Set and enable required thresholds in the following registers:
   - Monitor High Temperature Immediate Threshold (TMHTITR)
   - Monitor Low Temperature Immediate Threshold (TMLTITR)
   - Monitor High Temperature Average Threshold (TMHTATR)
   - Monitor Low Temperature Average Threshold (TMLTATR)
   - Monitor High Temperature Average Critical Threshold (TMHTACTR)
   - Monitor Low Temperature Average Critical Threshold (TMLTACTR)
   - Monitor Rising Temperature Rate Critical Threshold (TMRTRCTR)
   - Monitor Falling Temperature Rate Critical Threshold (TMFTRCTR)

```
TMU.TMHTITR.B.TEMP = kelvin; // set threshold
                     TMU.TMHTITR.B.EN = 1; // enable threshold
```

5. Configure monitoring interval in the Monitor Temperature Measurement Interval register (TMTMIR).

```
TMU.TMTMIR.B.TMI = 0x7; // if TMU clk = 400MHz, 0x7 = 2.7s
```

6. Configure the ALPF coefficient that is used in the average temperature calculations.

```
TMU.TMR.B.ALPF = 0x1; // 0x1 => ALPF = 0.5
```

7. Enabled the sensor site in the Monitor Site register (TMSR)

```
TMU.TMSR.B.SITE = sites; // enable sites
```

8. Monitoring mode can now be enabled in the Mode register (TMR)

```
TMU.TMR.B.MODE = 1; // enable monitoring mode
```

73.1.4 Reading measurement results

Once the monitoring mode is enabled, TMU measures signals from the temperature sensor at the frequency determined by the monitoring interval setting and updates the registers containing temperature measurement results.

The results can be read by the software from the following registers:

- Report Immediate Temperature at Site 0 (TRITSR0)
- Report Average Temperature at Site 0 (TRATSR0)
- Monitor High Temperature Capture (TMHTCR)
- Monitor Low Temperature Capture (TMLTCR)
- Monitor Rising Temperature Rate Capture (TMRTRCR)
- Monitor Falling Temperature Rate Capture (TMFTRCR)

Report Immediate Temperature registers contain the last immediate temperature measurement results. Similarly, Report Average Temperature registers contain average temperatures, the Monitor High Temperature Capture register contains the highest recorded temperature, and the Monitor Low Temperature Capture register contains the lowest temperature.

73.1.5 Calibration

The central module uses the calibration points to convert digitized temperature sensor signal into a temperature value in Kelvin. These points are generic for a family of products. NXP characterizes the calibration points shown in the following table and must not be diverged from.

Table 638. TMU range and error

Description	Min	Typ	Max	Units
Temperature monitoring range [1]	-40	-	125	°C
Temperature sensor error	-3.0	-	3.0	°C

1. Accuracy outside of operating range (-40 to 125°C) is not guaranteed.

The preliminary calibration table for the chip is as below:

TTCFGR[CAL_PNT]	Temp (C, dec)	TTRCR[TEMP[K]]	TSCFGR[SENSOR]	
Index		Temp (K, hex)	Code (dec)	Code (hex)
0	-55	0x0DA	14	0x00E
1	-40	0x0E9	41	0x029
2	-15	0x102	86	0x056
3	25	0x12A	162	0x0A2
4	85	0x166	278	0x116
5	150	0x1A7	405	0x195
6	165	0x1B6	434	0x1B2

73.2 Overview

TMU has access to multiple remote temperature measurement sites, which are strategically located on the chip. The module monitors these sites and reports the temperature from one or more of them. You can access the data related to the current and average temperature, which the module constantly captures, for each site. The module can also signal an alarm if a programmed event occurs.

73.2.1 Block diagram

The PTAT BJT Core block generates a signal proportional to absolute temperature in K. It extracts $V_{be2} – V_{be1}$ (a quantity whose behavior is PTAT) of two BJTs by injecting different currents into both and then sensing $V_{be2} – V_{be1}$.

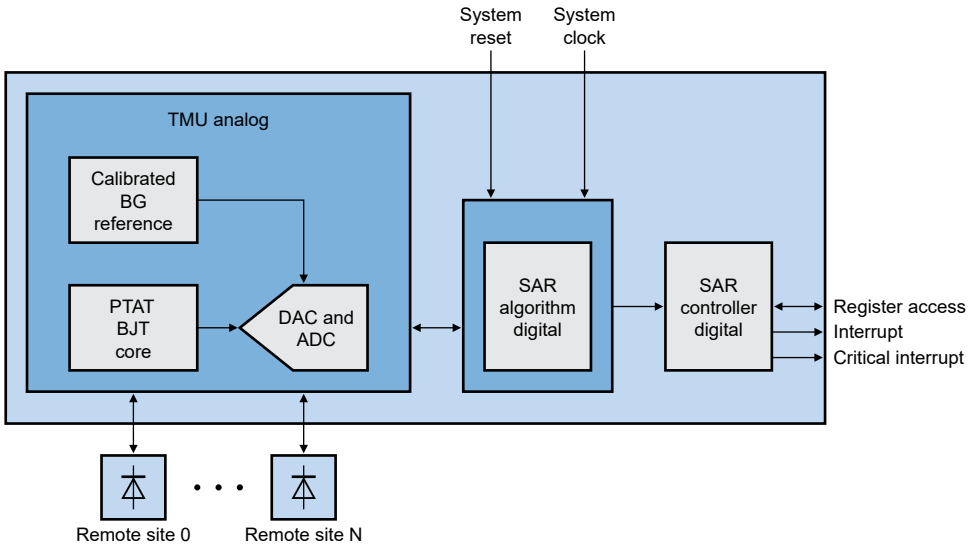


Figure 438. TMU block diagram

73.2.2 Features

TMU includes the following features:

- Circuit has the capability of measuring temperatures from 233K to 433K (-40 °C to +160 °C); actual calibration table min/max temperatures may differ due to process limitations
- Configuration
  - TMU software configuration loads and initializes the calibration data
- Monitoring
  - Single- or multi-site monitoring
  - Programmable monitoring interval
  - Out-of-range indication
  - High/low temperature range monitoring
  - Temperature rate change monitoring
  - Immediate and average temperature monitoring
  - Average temperature monitoring based on programmable low-pass filtering
  - Programmable monitoring of high/low temperature thresholds for triggering normal and/or critical alarms
  - Programmable monitoring of the rate of rising/falling temperatures for triggering critical alarms
- Reporting
  - Immediate and average temperature reporting for all monitor sites

73.3 Functional description

The following sections describe the functionality of the TMU in detail.

73.3.1 Operation

The TMU has the following modes of operation as defined by TMR[MODE]:

- TMU disabled; low-power mode

- Monitoring

73.3.2 Monitoring

Monitoring is the process of reading on-chip temperature sensor sites at regular intervals and taking appropriate actions, such as alarming the user when the temperature exceeds a programmed temperature threshold. The monitoring process only applies to those sites that are enabled.

During this process, TMU periodically measures all enabled temperature sensor sites for temperature, starting with site 0 and ending with site 15. In Monitor Temperature Measurement Interval (TMTMIR), you can set the interval at which TMU reads the sensors, and the setting must reflect the maximum interval time required to accurately capture temperature changes. If the measurement interval has not expired after the last active site is read, the sensor logic may enter the low-power mode. If the interval expires before the module reads the last active site, the module sets TSR[MIE] to 1 and reads the next active site immediately. If the interval has been exceeded, you may opt to reduce the number of sites monitored by TMU, or increase the interval, if possible. This is especially critical for capturing temperature rate change events accurately.

TMU records the current and average temperature for each site. It calculates the average temperature based on the low-pass filter function in the TMU mode register. If the module triggers any of the enabled threshold or rate event registers, it sets the corresponding interrupt detect field in Interrupt Detect (TIDR). You enable interrupts through Interrupt Enable (TIER).

The procedure to enable the monitoring mode is as follows:

1. Clear the interrupt detect register, Interrupt Detect (TIDR).
2. Clear the interrupt capture registers: Interrupt Immediate Site Capture (TIISCR), Interrupt Average Site Capture (TIASCR), and Interrupt Critical Site Capture (TICSCR).
3. Enable interrupt handling by writing 1 to the appropriate fields in the interrupt enable register, Interrupt Enable (TIER).
4. Write the appropriate values to the following temperature threshold registers:
   - Monitor High Temperature Immediate Threshold (TMHTITR)
   - Monitor Low Temperature Immediate Threshold (TMLTITR)
   - Monitor High Temperature Average Threshold (TMHTATR)
   - Monitor Low Temperature Average Threshold (TMLTATR)
   - Monitor High Temperature Average Critical Threshold (TMHTACTR)
   - Monitor Low Temperature Average Critical Threshold (TMLTACTR)
5. Write the appropriate values to these temperature rate change registers: Monitor Rising Temperature Rate Critical Threshold (TMRTRCTR) and Monitor Falling Temperature Rate Critical Threshold (TMFTRCTR).
6. Write the appropriate value to the monitoring interval register, Monitor Temperature Measurement Interval (TMTMIR).
7. Select the ALPF coefficient in the Mode (TMR). The ALPF coefficient is used in the average temperature calculations.
8. Enable the monitor mode by setting TMR[MODE] to 10b. You control the monitored sites by writing an appropriate value to TMSR[SITE]. You must enable at least one active site, and set the other mode control fields as needed.
9. If the monitoring interval is too short, as indicated by TSR[MIE], you may need to increase the interval or reduce the number of sites.

73.3.3 Reporting

The TMU can directly report the current and average temperature for a particular temperature sensor site during the monitoring mode by reading one of the site report registers. The report uses the last measurement from the monitoring process, and requires a site to be actively monitored to obtain an accurate reading. Reading a site, which last measured an invalid temperature outside the sensor range of 273 K to 368 K, will have the valid bit cleared.

If you disable monitoring, the last temperature measurement for the site(s) that the TMU previously monitored is retained by the module, and you can still read the report registers, with the knowledge that the temperature(s) reported is (are) no longer accurate. This method can be used to capture the temperatures at multiple sites in time, but does not allow for continuous monitoring.

The temperature capture registers (Monitor High Temperature Capture (TMHTCR) and Monitor Low Temperature Capture (TMLTCR)) record the history of the highest and lowest temperatures that they measure within the valid sensor range. The temperature rate change capture registers (Monitor Rising Temperature Rate Capture (TMRTRCR) and Monitor Falling Temperature Rate Capture (TMFTRCR)) record how quickly the temperature rises or falls, and you can use this information to determine programmable values for the rate change threshold registers in real applications.

You can read out-of-range indications from Status (TSR). You write 1 to clear these indications during active monitoring.

73.3.4 Using temperature thresholds

The main purpose of the TMU is to autonomously measure temperature at select sites and report to the core when a potentially dangerous operating temperature is about to be reached. This may be to protect a circuit from overheating and causing permanent damage or affecting reliability.

The TMU has a two-level (normal/critical) threshold and interrupt scheme. This allows the software to take preventive actions at a first temperature threshold and additional drastic measures at a second critical threshold. For high temperatures, these measures may include:

- Downgrading the performance of a function or core to reduce power consumption (preventive)
- Reducing the voltage or frequency to a function or core

When TMU detects a high temperature (through interrupt), the software could enable a low or normal threshold to detect when to restore functions to normal. Similarly, preventive actions taken for low temperatures could enable a high or normal temperature to restore functions.

For proper operation, you must set the temperatures in the high threshold registers to be higher than those you set in the low threshold registers.

The rising and falling temperature rate threshold registers (Monitor Rising Temperature Rate Critical Threshold (TMRTRCTR) and Monitor Falling Temperature Rate Critical Threshold (TMFTRCTR)) trigger critical interrupts. A quickly rising temperature because of heat sink separation or cooling fan issues may occur sooner than a critically high temperature. For example, immediate preventive measures can be taken to protect the chip. You could determine a nominal value for the rising threshold value by using the Monitor Rising Temperature Rate Capture (TMRTRCR) or from lab simulations. A quickly falling temperature is less common but you could use this to warn the chip about stress caused by drastic changes in temperature that may lead to cracking.

73.3.5 Clocking

Design uses a single clock which is divided for driving internal sub-modules. Register TCMCFG[DIV_CLK/DFD] provides the divider for the clocks used during successive approximation and the digital filter sampling.

73.3.6 Interrupts

The following table provides the details of TMU interrupts.

Table 639. TMU interrupts

Interrupt description	Enable field	Reported as	
		Interrupt	Critical interrupt
Immediate High Temperature Threshold Interrupt Enable	IHTTIE	Y	—
Immediate Low Temperature Threshold Interrupt Enable	ILTTIE	Y	—
Average High Temperature Threshold Interrupt Enable	AHTTIE	Y	—
Average Low Temperature Threshold Interrupt Enable	ALTTIE	Y	—

Table continues on the next page...

Table 639. TMU interrupts (continued)

Interrupt description	Enable field	Reported as	
		Interrupt	Critical interrupt
Average High Temperature Critical Threshold Interrupt Enable	AHTCTIE	—	Y
Average Low Temperature Critical Threshold Interrupt Enable	ALTCTIE	—	Y
Rising Temperature Rate Critical Threshold Interrupt Enable	RTRCTIE	—	Y
Falling Temperature Rate Critical Threshold Interrupt Enable	FTRCTIE	—	Y

73.4 Initialization

TMU calculates temperature by reading one of the on-chip temperature sensor sites. The module calculates the temperature using a table of calibration points that you must program using the procedure outlined below.

You can manually program the sensor translation table by supplying the sensor readings for all temperature points as defined by the TMU temperature range control registers. Failure to initialize all the temperature/sensor configuration table entries correctly may lead to an undefined behavior. The TMU temperature range control registers, Temperature Range Control a (TTRCR0 - TTRCR15), define the starting temperature, the ending temperature, and a number of intervening calibration points. The number of configuration points used depends on the linearity of the sensor. If a sensor reading falls between two configuration points, TMU reports the temperature based on linear interpolation between the nearest upper/lower two points in the table. If a sensor reading falls below the first calibration point, then Out-of-Range Low Temperature Measurement flag (ORL) is set in the Status (TSR). Similarly, if a sensor reading falls above the last calibration point, then Out-of-Range High Temperature Measurement (ORH) flag is set in the Status (TSR).

Following is the procedure to program the TMU calibration table:

1. Disable monitoring mode, TMR[MODE]=0 (default).
2. Configure Central Module Configuration Register (TCMCFG) according to instructions given in Chip-specific TMU information section.
3. Write the temperature configuration register with TTCFGR[CAL_PT]=*n* (where *n* is the calibration point from the table).
4. Write to the TMU configuration register, Sensor Configuration (TSCFGR), with the sensor value for the first calibration point *n*.
5. Write to the TMU temperature range control registers (TTRCR*n*) associated with calibration point *n*.

You need to repeat steps 3—5 for all calibration points from the provided calibration table. You must write 1 to the DEMA and OCM fields of Central Module Configuration (TCMCFG) for the proper operation of TMU.

The values for TTCFGR and TTRCR are chip-specific. See the "Chip-specific TMU section" for details.

73.5 External signals

This module has no external signals.

73.6 TMU register descriptions

73.6.1 TMU memory map

TMU base address: 4448_2000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Mode (TMR)	32	RW	0000_0000
4	Status (TSR)	32	RW	0000_0000
8	Monitor Site (TMSR)	32	RW	0000_0000
C	Monitor Temperature Measurement Interval (TMTMIR)	32	RW	0000_0000
20	Interrupt Enable (TIER)	32	RW	0000_0000
24	Interrupt Detect (TIDR)	32	RW	0000_0000
30	Interrupt Immediate Site Capture (TIISCR)	32	RW	0000_0000
34	Interrupt Average Site Capture (TIASCR)	32	RW	0000_0000
38	Interrupt Critical Site Capture (TICSCR)	32	RW	0000_0000
40	Monitor High Temperature Capture (TMHTCR)	32	RW	0000_0000
44	Monitor Low Temperature Capture (TMLTCR)	32	RW	0000_0000
48	Monitor Rising Temperature Rate Capture (TMRTRCR)	32	RW	0000_0000
4C	Monitor Falling Temperature Rate Capture (TMFTRCR)	32	RW	0000_0000
50	Monitor High Temperature Immediate Threshold (TMHTITR)	32	RW	0000_0000
54	Monitor High Temperature Average Threshold (TMHTATR)	32	RW	0000_0000
58	Monitor High Temperature Average Critical Threshold (TMHTACTR)	32	RW	0000_0000
60	Monitor Low Temperature Immediate Threshold (TMLTITR)	32	RW	0000_0000
64	Monitor Low Temperature Average Threshold (TMLTATR)	32	RW	0000_0000
68	Monitor Low Temperature Average Critical Threshold (TMLTACTR)	32	RW	0000_0000
70	Monitor Rising Temperature Rate Critical Threshold (TMRTRCTR)	32	RW	0000_0000
74	Monitor Falling Temperature Rate Critical Threshold (TMFTRCTR)	32	RW	0000_0000
80	Temperature Configuration (TTCFGR)	32	RW	0000_0000
84	Sensor Configuration (TSCFGR)	32	RW	0000_0000
100	Report Immediate Temperature at Site (TRITSR0)	32	R	0000_0000
104	Report Average Temperature at Site (TRATSR0)	32	R	0000_0000
110	Report Immediate Temperature at Site (TRITSR1)	32	R	0000_0000
114	Report Average Temperature at Site (TRATSR1)	32	R	0000_0000
120	Report Immediate Temperature at Site (TRITSR2)	32	R	0000_0000
124	Report Average Temperature at Site (TRATSR2)	32	R	0000_0000
130	Report Immediate Temperature at Site (TRITSR3)	32	R	0000_0000
134	Report Average Temperature at Site (TRATSR3)	32	R	0000_0000
140	Report Immediate Temperature at Site (TRITSR4)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
144	Report Average Temperature at Site (TRATSR4)	32	R	0000_0000
150	Report Immediate Temperature at Site (TRITSR5)	32	R	0000_0000
154	Report Average Temperature at Site (TRATSR5)	32	R	0000_0000
160	Report Immediate Temperature at Site (TRITSR6)	32	R	0000_0000
164	Report Average Temperature at Site (TRATSR6)	32	R	0000_0000
170	Report Immediate Temperature at Site (TRITSR7)	32	R	0000_0000
174	Report Average Temperature at Site (TRATSR7)	32	R	0000_0000
180	Report Immediate Temperature at Site (TRITSR8)	32	R	0000_0000
184	Report Average Temperature at Site (TRATSR8)	32	R	0000_0000
190	Report Immediate Temperature at Site (TRITSR9)	32	R	0000_0000
194	Report Average Temperature at Site (TRATSR9)	32	R	0000_0000
1A0	Report Immediate Temperature at Site (TRITSR10)	32	R	0000_0000
1A4	Report Average Temperature at Site (TRATSR10)	32	R	0000_0000
1B0	Report Immediate Temperature at Site (TRITSR11)	32	R	0000_0000
1B4	Report Average Temperature at Site (TRATSR11)	32	R	0000_0000
1C0	Report Immediate Temperature at Site (TRITSR12)	32	R	0000_0000
1C4	Report Average Temperature at Site (TRATSR12)	32	R	0000_0000
1D0	Report Immediate Temperature at Site (TRITSR13)	32	R	0000_0000
1D4	Report Average Temperature at Site (TRATSR13)	32	R	0000_0000
1E0	Report Immediate Temperature at Site (TRITSR14)	32	R	0000_0000
1E4	Report Average Temperature at Site (TRATSR14)	32	R	0000_0000
1F0	Report Immediate Temperature at Site (TRITSR15)	32	R	0000_0000
1F4	Report Average Temperature at Site (TRATSR15)	32	R	0000_0000
F00	Central Module Configuration (TCMCFG)	32	RW	See section
F10 - F4C	Temperature Range Control a (TTRCR0 - TTRCR15)	32	RW	0000_0000

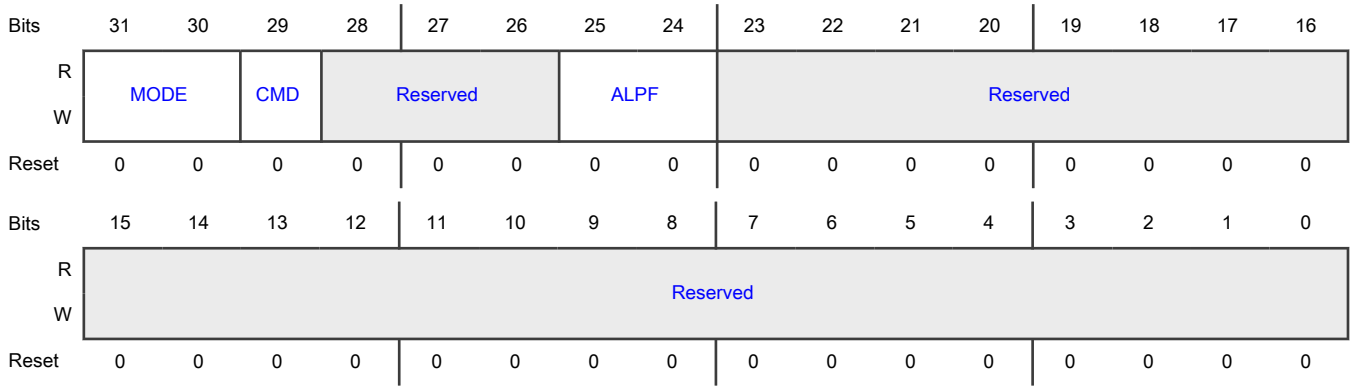
73.6.2 Mode (TMR)

This register allows the software to control the thermal monitoring operation. You must wait at least 500 µs before changing modes.

Offset

Register	Offset
TMR	0h

Diagram



Fields

Field	Description
31-30 MODE	Mode Before you enable the TMU for monitoring, you must first configure it; see Initialization. Failure to properly initialize the configuration table may result in undefined behavior. 00 - Idle; low-power mode 01 - Reserved 10 - Monitoring of sites as defined by TMSR[SITE] 11 - Reserved
29 CMD	Central Module Disable Before disabling the central module, you must first deassert the MODE field and verify that the TMU is idle. You must then enable the circuit during normal operation by writing an appropriate non-zero value in the MODE field. This field allows for statically disabling the centralized circuit and the associated digital logic to reduce power. If the central circuit is re-enabled, TMU must wait before operations can continue as defined by TCMCFG[CMET]. 0 - Enabled 1 - Disabled
28-26 —	Reserved
25-24 ALPF	Average Low Pass Filter Setting The average temperature is calculated using this formula: ALPF x Current_Temp + (1 - ALPF) x Average_Temp. If no previous (average) temperature is valid, then the module uses the current temperature for Average_Temp in this calculation. For proper operation, you must change this field only when monitoring is disabled. 00 - 1.0 01 - 0.5 10 - 0.25

Table continues on the next page...

Table continued from the previous page...

Field	Description
	11 - 0.125
23-0 —	Reserved

73.6.3 Status (TSR)

This register reports the monitoring and calibration status during operation.

Offset

Register	Offset
TSR	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TB	MIE	ORL	ORH	Reserved											
W		W1C	W1C	W1C												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 TB	TMU Busy In monitoring mode, this indicates that a temperature measurement is pending. 0 - TMU is idle 1 - TMU is busy
30 MIE	Monitoring Interval Exceeded The time required to perform measurement of all monitored sites has exceeded the monitoring interval as defined by Monitor Temperature Measurement Interval (TMTMIR). 0 - Monitoring interval not exceeded 1 - Monitoring interval exceeded

Table continues on the next page...

Table continued from the previous page...

Field	Description
29 ORL	Out-of-Range Low Temperature Measurement A temperature sensor detected a temperature reading below the lowest calibrated temperature. 0 - No out-of-range low temperature measurement detected 1 - Out-of-range low temperature measurement detected
28 ORH	Out-of-Range High Temperature Measurement A temperature sensor detected a temperature reading above the highest calibrated temperature. 0 - No out-of-range high temperature measurement detected 1 - Out-of-range high temperature measurement detected
27-0 —	Reserved

73.6.4 Monitor Site (TMSR)

This register indicates which remote site sensor is actively being monitored by TMU when TMR[MODE]=10b.

Offset

Register	Offset
TMSR	8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SITE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
15-0 SITE	Monitoring Site Select 15-0 When you set the select bit for a temperature sensor site, it is enabled and included in all monitoring functions. For proper operation, you must only change this field when monitoring is disabled. Multiple sites can be enabled at the same time. SITE[x] controls monitoring of site x, such as: • SITE[0] - controls monitoring site 0 — SITE[0]==0: Disables site 0 — SITE[0]==1: Enables site 0 • SITE[1] - controls monitoring site 1 — SITE[1]==0: Disables site 1 — SITE[1]==1: Enables site 1 • SITE[2] - controls monitoring site 2 — SITE[2]==0: Disables site 2 — SITE[2]==1: Enables site 2

73.6.5 Monitor Temperature Measurement Interval (TMTMIR)

This register allows you to control at what frequency the module reads the temperature sensors. The module reads all enabled monitored sites once in this time interval. The value of the status field, TSR[MIE], is 1 if the temperature measurement takes longer than the set interval. The software must consider increasing the interval or reducing the number of active sites if the interval exceeds the temperature monitoring interval determined by the TMI field value in this register. If you disable the interval, then monitoring is continuous.

Offset

Register	Offset
TMTMIR	Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												TMI			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-4 —	Reserved
3-0 TMI	Temperature Monitoring Interval in Seconds For proper operation, you must only change this field when monitoring is disabled, that is, when TMR[MODE]=00b. For the TMU frequencies calculation, use the following formula for the field description: • $2^{(22+TMI)}$ × TMU clock period in seconds For the monitoring interval calculation, use the following formula: • 0b – 1110b: Interval(s) = $2^{(22+TMI)}$ × TMU clock period in seconds • 1111b: Continuous monitoring mode **NOTE** When operating in continuous mode, the time between samples can be estimated using the formula: Interval = TMU_clk_per x $2^{(TCMCFG[CLK_DIV]+4)}$ x [(10 + $2^{(TCMCFG[RCTC]+3)}$) x 8 {TCMCFG[DEMA]=1} x 2 {TCMCFG[OCM]=1} x NUM_SITES]. When operating in interval mode, the time between samples should be less than the interval setting.

73.6.6 Interrupt Enable (TIER)

This register determines if a detected status condition causes a system interrupt. A system interrupt occurs if the value of any field in this register is 1 and the corresponding field in Interrupt Detect (TIDR) is 1. To clear the interrupt, you write a 1 to Interrupt Detect (TIDR).

Offset

Register	Offset
TIER	20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IHTTIE	AHTTIE	AHTCTIE	ILTTIE	ALTTIE	ALTCTIE	RTRCTIE	FTRCTIE	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 IHTTIE	Immediate High Temperature Threshold Interrupt Enable 0 - Disabled 1 - Interrupt enabled; generates an interrupt if TIDR[IHTT] is 1
30 AHTTIE	Average High Temperature Threshold Interrupt Enable 0 - Disabled 1 - Interrupt enabled; generates an interrupt if TIDR[AHTT] is 1
29 AHTCTIE	Average High Temperature Critical Threshold Interrupt Enable 0 - Disabled 1 - Interrupt enabled; generates an interrupt if TIDR[AHTCT] is 1
28 ILTTIE	Immediate Low Temperature Threshold Interrupt Enable 0 - Disabled 1 - Interrupt enabled; generates an interrupt if TIDR[ILTT] is 1
27 ALTTIE	Average Low Temperature Threshold Interrupt Enable 0 - Disabled 1 - Interrupt enabled; generates an interrupt if TIDR[ALTT] is 1
26 ALTCTIE	Average Low Temperature Critical Threshold Interrupt Enable 0 - Disabled 1 - Interrupt enabled; generates an interrupt if TIDR[ALTCT] is 1
25 RTRCTIE	Rising Temperature Rate Critical Threshold Interrupt Enable 0 - Disabled 1 - Interrupt enabled; generates an interrupt if TIDR[RTRCT] is 1
24 FTRCTIE	Falling Temperature Rate Critical Threshold Interrupt Enable 0 - Disabled 1 - Interrupt enabled; generates an interrupt if TIDR[FTRCT] is 1
23-0 —	Reserved

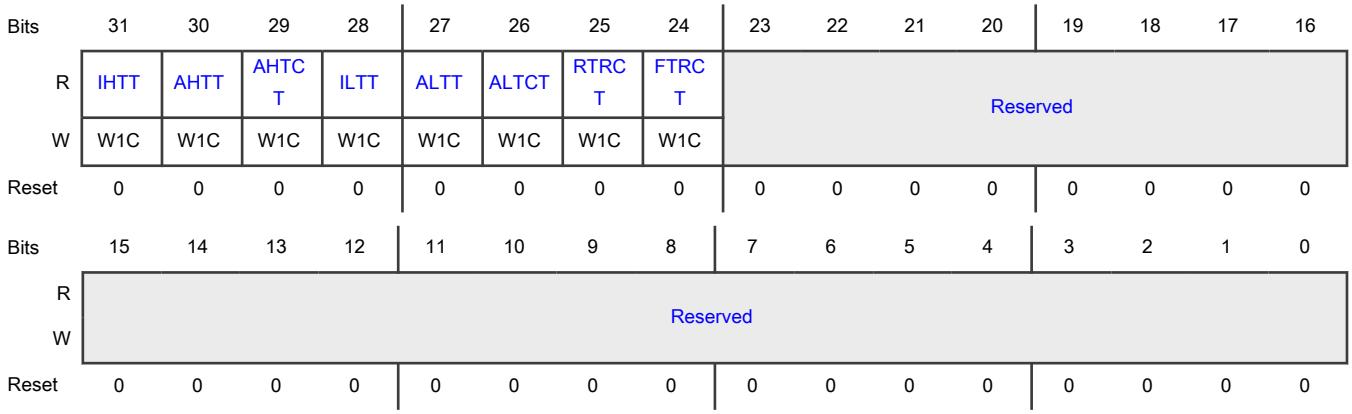
73.6.7 Interrupt Detect (TIDR)

This register indicates if a status condition is detected that could generate an interrupt. If enabled, you write a 1 to clear the detected condition and the interrupt.

Offset

Register	Offset
TIDR	24h

Diagram



Fields

Field	Description
31 IHTT	Immediate High Temperature Threshold Exceeded Write 1 to clear. 0 - No threshold exceeded 1 - One or more monitored sites has exceeded the immediate high temperature threshold, as defined by Monitor High Temperature Immediate Threshold (TMHTITR). This includes an out-of-range measured temperature above the maximum calibrated temperature. TIISCR[SITE] captures the sites that exceed the threshold.
30 AHTT	Average High Temperature Threshold Exceeded Write 1 to clear. 0 - No threshold exceeded 1 - One or more monitored sites exceed the average high temperature threshold, as defined by Monitor High Temperature Average Threshold (TMHTATR). TIASCR[SITE] captures the sites that exceed the threshold.
29 AHTCT	Average High Temperature Critical Threshold Exceeded Write 1 to clear. 0 - No threshold exceeded 1 - One or more monitored sites exceed the average high temperature critical threshold, as defined by Monitor High Temperature Average Critical Threshold (TMHTACTR). TICSCR[SITE] captures the sites that exceed the threshold.
28	Immediate Low Temperature Threshold

Table continues on the next page...

Table continued from the previous page...

Field	Description
ILTT	Write 1 to clear. 0 - No threshold exceeded 1 - One or more monitored sites has passed the immediate low temperature threshold, as defined by Monitor Low Temperature Immediate Threshold (TMLTITR). This includes an out-of-range measured temperature below the minimum calibrated temperature. TIISCR[SITE] captures the sites that pass the threshold.
27 ALTT	Average Low Temperature Threshold Write 1 to clear. 0 - No threshold exceeded 1 - One or more monitored sites pass the average low temperature threshold, as defined by Monitor Low Temperature Average Threshold (TMLTATR). TIASCR[SITE] captures the sites that pass the threshold.
26 ALTCT	Average Low Temperature Critical Threshold Write 1 to clear. 0 - No threshold exceeded 1 - One or more monitored sites pass the average low temperature threshold, as defined by Monitor Low Temperature Average Critical Threshold (TMLTACTR). TICSCR[SITE] captures the sites that pass the threshold.
25 RTRCT	Rising Temperature Rate Critical Threshold Write 1 to clear. 0 - No threshold exceeded 1 - One or more monitored sites pass the rising temperature rate critical threshold, as defined by Monitor Rising Temperature Rate Critical Threshold (TMRTRCTR). TICSCR[SITE] captures the sites that exceed the threshold.
24 FTRCT	Falling Temperature Rate Critical Threshold Write 1 to clear. 0 - No threshold exceeded 1 - One or more monitored sites exceed the falling temperature rate critical threshold, as defined by Monitor Falling Temperature Rate Critical Threshold (TMFTRCTR). TICSCR[SITE] captures the sites that exceed the threshold.
23-0 —	Reserved

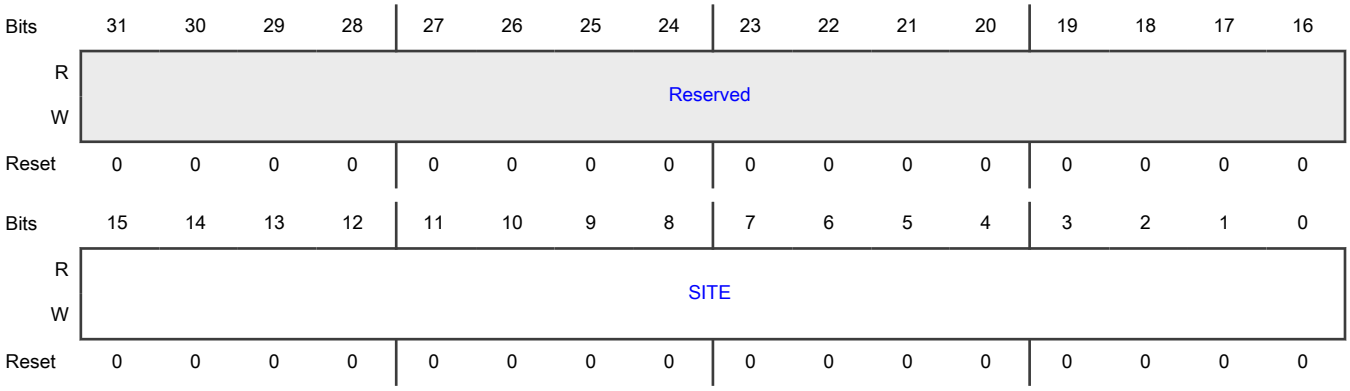
73.6.8 Interrupt Immediate Site Capture (TIISCR)

This register indicates which remote sensor sites are associated with the detection of an interrupt event related to current reported temperatures.

Offset

Register	Offset
TIISCR	30h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-0 SITE	Temperature Sensor Site Temperature sensor site associated with the setting of TIDR[IHTT] and TIDR[ILTT]. This field has the same bit representation as TMSR[SITE]. The software must clear this field after handling the detected interrupt event IHTT or ILTT.

73.6.9 Interrupt Average Site Capture (TIASCR)

This register indicates which remote sensor sites are associated with the detection of an interrupt event related to average reported temperatures.

Offset

Register	Offset
TIASCR	34h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SITE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 SITE	Temperature Sensor Site Temperature sensor site associated with the setting of TIDR[AHTT] and TIDR[ALTT]. This field has the same bit representation as TMSR[SITE]. The software must clear this field after handling the detected interrupt event AHTT or ALTT.

73.6.10 Interrupt Critical Site Capture (TICSCR)

This register indicates which remote sensor sites are associated with the detection of a critical interrupt event.

Offset

Register	Offset
TICSCR	38h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SITE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-16 —	Reserved
15-0 SITE	Temperature Sensor Site Temperature sensor site associated with the setting of TIDR[FTRCT], TIDR[RTRCT], TIDR[AHTCT], and TIDR[ALTCT]. This field has the same bit representation as TMSR[SITE]. The software must clear this field after handling the detected critical interrupt event FTRCT, RTRCT, AHTCT, or ALTCT.

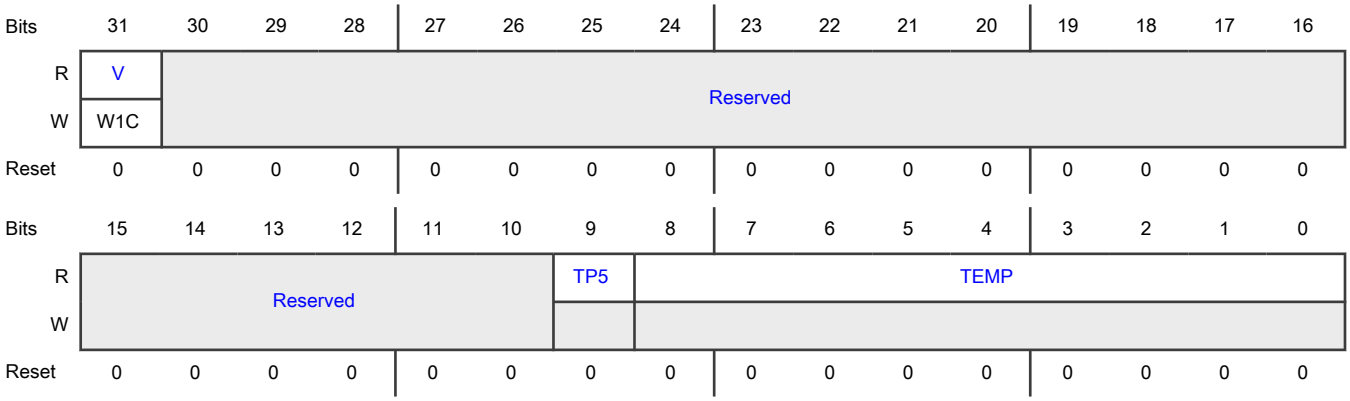
73.6.11 Monitor High Temperature Capture (TMHTCR)

This register captures and records the highest temperature for any one enabled monitored site within the temperature sensor range.

Offset

Register	Offset
TMHTCR	40h

Diagram



Fields

Field	Description
31 V	Valid Reading Re-enabling TMU automatically clears this field and starts a new search. 0 - Temperature reading is not valid due to no measured temperature within the calibrated sensor range for an enabled monitored site 1 - Temperature reading is valid
30-10	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
9 TP5	Highest Temperature Recorded in Kelvin by Any Enabled Monitored Site Valid when V=1. When this field is 1, you must add 0.5 K to the temperature that TEMP reports. For example, if TEMP is 300 K and TP5=1, then the final temperature is 300.5 K.
8-0 TEMP	Highest Temperature Recorded in Kelvin by Any Enabled Monitored Site Valid when V=1. **NOTE** Highest recordable temperature will not exceed maximum calibrated temperature.

73.6.12 Monitor Low Temperature Capture (TMLTCR)

This register captures and records the lowest temperature reached for any one enabled monitored site within the temperature sensor range.

Offset

Register	Offset
TMLTCR	44h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	V	Reserved														
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						TP5	TEMP								
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 V	Valid Reading Re-enabling TMU automatically clears this field and starts a new search.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Temperature reading is not valid because of no measured temperature within the sensor calibration range for an enabled monitored site 1 - Temperature reading is valid
30-10 —	Reserved
9 TP5	Lowest Temperature in Kelvin that Any Enabled Monitored Site Records Valid when V=1. When this field is 1, you must add 0.5 K to the temperature that TEMP reports. For example, if TEMP is 300 K and TP5=1, then the final temperature is 300.5 K.
8-0 TEMP	Lowest Temperature in Kelvin that Any Enabled Monitored Site Records Valid when V=1. **NOTE** Lowest recordable temperature will not exceed minimum calibrated temperature.

73.6.13 Monitor Rising Temperature Rate Capture (TMRTRCR)

This register captures and records the highest temperature rising rate change between two successive measurements from the same monitored site based on the monitoring interval setting.

Offset

Register	Offset
TMRTRCR	48h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	V	Reserved														
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TEMP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 V	Valid Reading Re-enabling TMU automatically clears this field and starts a new search. 0 - Temperature reading is not valid because of no two successive measured temperatures within the sensor calibration range for an enabled monitored site 1 - Temperature reading is valid
30-8 —	Reserved
7-0 TEMP	Highest Rising Temperature Rate Change in Kelvin that Any Enabled Monitored Site Records Valid when V=1. Rate change is between 0 K to 200 K.

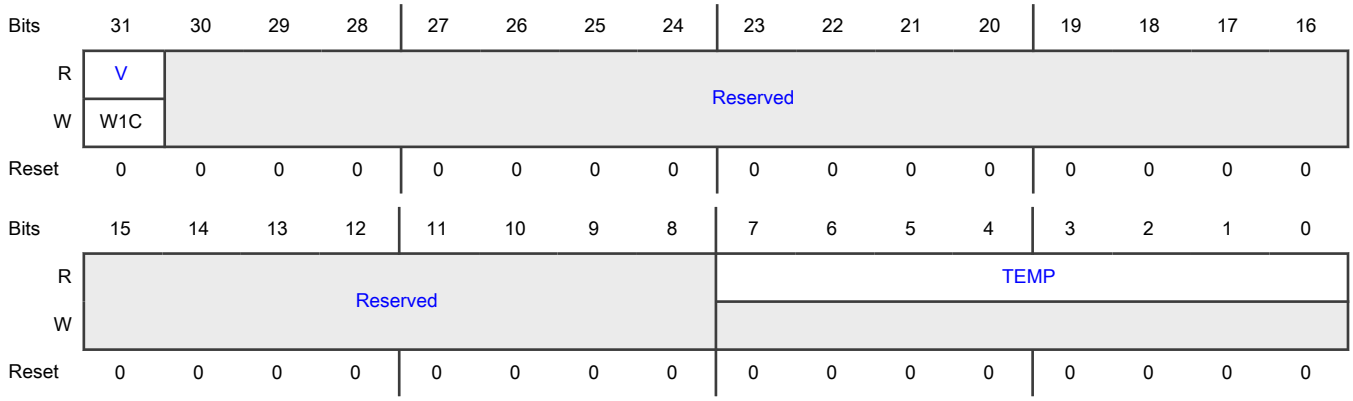
73.6.14 Monitor Falling Temperature Rate Capture (TMFTRCR)

This register captures and records the highest temperature falling rate change between two successive measurements from the same monitored site based on the monitoring interval setting.

Offset

Register	Offset
TMFTRCR	4Ch

Diagram



Fields

Field	Description
31	Valid Reading

Table continues on the next page...

Table continued from the previous page...

Field	Description
V	Re-enabling TMU automatically clears this field and starts a new search. 0 - Temperature reading is not valid because of no two successive measured temperatures within the sensor calibration range for an enabled monitored site 1 - Temperature reading is valid
30-8 —	Reserved
7-0 TEMP	Highest Falling Temperature Rate Change in Kelvin that Any Enabled Monitored Site Records Valid when V=1. Rate change is a positive number between 0 K to 200 K.

73.6.15 Monitor High Temperature Immediate Threshold (TMHTITR)

This register determines the current high-temperature threshold for generating the TIDR[IHTT] event.

Offset

Register	Offset
TMHTITR	50h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 EN	Enable Threshold 0 - Disabled 1 - Threshold enabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
30-9 —	Reserved
8-0 TEMP	High Temperature Immediate Threshold Value Determines the current high-temperature threshold for any enabled monitored site. If this threshold is exceeded, the module sets the value of TIDR[IHTT] to 1 if the value of the corresponding EN field is 1. **NOTE** Highest recordable temperature will not exceed maximum calibrated temperature.

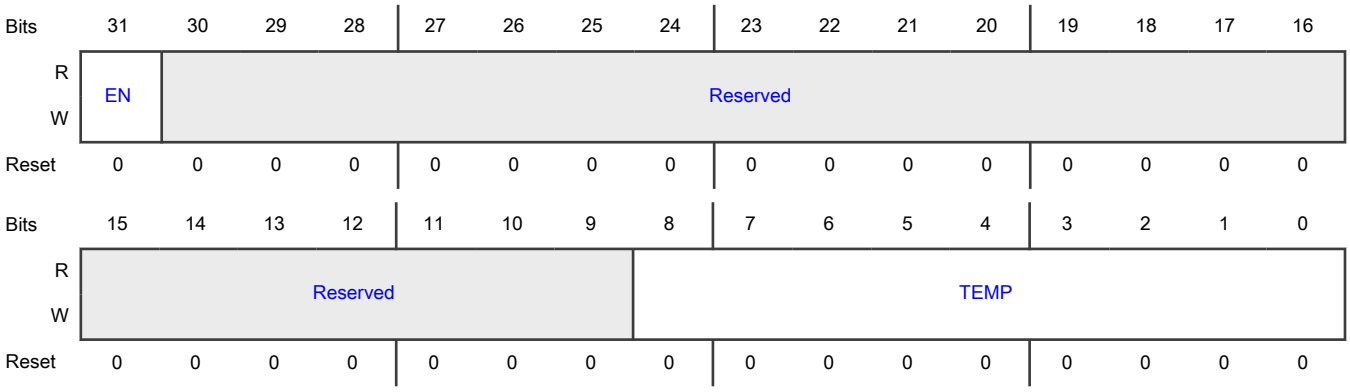
73.6.16 Monitor High Temperature Average Threshold (TMHTATR)

This register determines the average high temperature threshold for generating the TIDR[AHTT] event. The low-pass filter setting, TMR[ALPF], determines the function for calculating the average temperature.

Offset

Register	Offset
TMHTATR	54h

Diagram



Fields

Field	Description
31 EN	Enable Threshold 0 - Disabled 1 - Threshold enabled
30-9	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
8-0 TEMP	High Temperature Average Threshold Value Determines the current high-temperature threshold for any enabled monitored site. If this threshold is exceeded, the module sets the value of TIDR[AHTT] to 1 if the value of the corresponding EN field is 1. **NOTE** Highest recordable temperature will not exceed maximum calibrated temperature.

73.6.17 Monitor High Temperature Average Critical Threshold (TMHTACTR)

This register determines the average critical high temperature threshold for generating the TIDR[AHTCT] event. The low-pass filter setting, TMR[ALPF], determines the function for calculating the average temperature.

Offset

Register	Offset
TMHTACTR	58h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 EN	Enable Threshold 0 - Disabled 1 - Threshold enabled
30-9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
8-0 TEMP	High Temperature Average Critical Threshold Value Determines the average critical high temperature threshold, for any enabled monitored site, that if exceeded will cause TIDR[AHTCT] to be set when EN=1. **NOTE** Highest recordable temperature will not exceed maximum calibrated temperature.

73.6.18 Monitor Low Temperature Immediate Threshold (TMLTITR)

This register determines the current low temperature threshold for generating the TIDR[ILTT] event.

Offset

Register	Offset
TMLTITR	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 EN	Enable Threshold 0 - Disabled 1 - Threshold enabled
30-9 —	Reserved
8-0 TEMP	Low Temperature Immediate Threshold Value

Table continues on the next page...

Table continued from the previous page...

Field	Description
	Determines the current low temperature threshold for any enabled monitored site. If the temperature is below the threshold it will cause TIDR[ILTT] to be set when EN=1. **NOTE** Lowest recordable temperature will not exceed minimum calibrated temperature.

73.6.19 Monitor Low Temperature Average Threshold (TMLTATR)

This register determines the average low temperature threshold for generating the TIDR[ALTT] event. The low-pass filter setting, TMR[ALPF], determines the function for calculating the average temperature.

Offset

Register	Offset
TMLTATR	64h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 EN	Enable Threshold 0 - Disabled 1 - Threshold enabled
30-9 —	Reserved
8-0 TEMP	Low Temperature Average Threshold Value Determines the average low temperature threshold for any enabled monitor site. If the temperature is below the threshold it will cause TIDR[ALTT] to be set when EN=1.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	NOTE Lowest recordable temperature will not exceed minimum calibrated temperature.

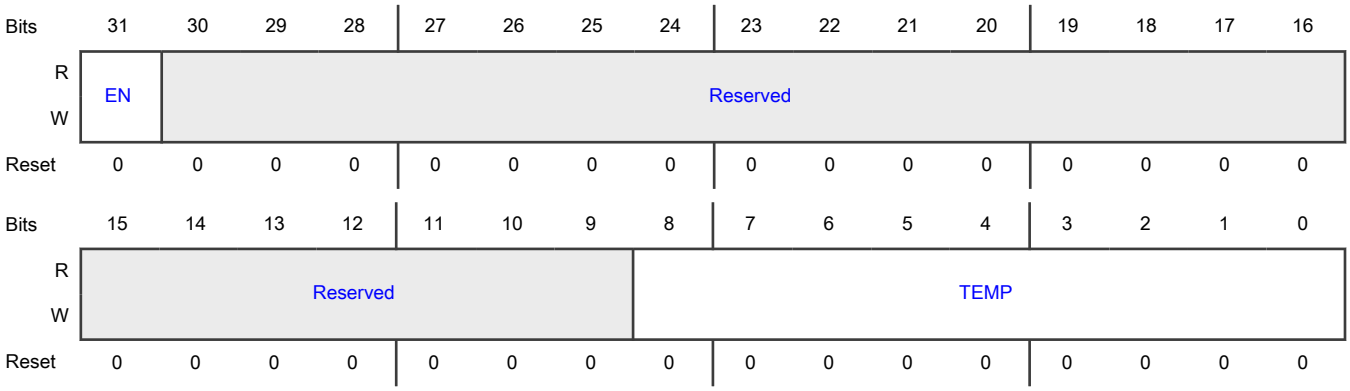
73.6.20 Monitor Low Temperature Average Critical Threshold (TMLTACTR)

This register determines the average critical low temperature threshold for generating the TIDR[ALTCT] event. The low-pass filter setting, TMR[ALPF], determines the function for calculating the average temperature.

Offset

Register	Offset
TMLTACTR	68h

Diagram



Fields

Field	Description
31 EN	Enable Threshold 0 - Disabled 1 - Threshold enabled
30-9 —	Reserved
8-0 TEMP	Low Temperature Average Critical Threshold Value Determines the average critical high temperature threshold, for any enabled monitored site, that if passed will cause TIDR[ALTCT] to be set when EN=1. **NOTE** Lowest recordable temperature will not exceed minimum calibrated temperature.

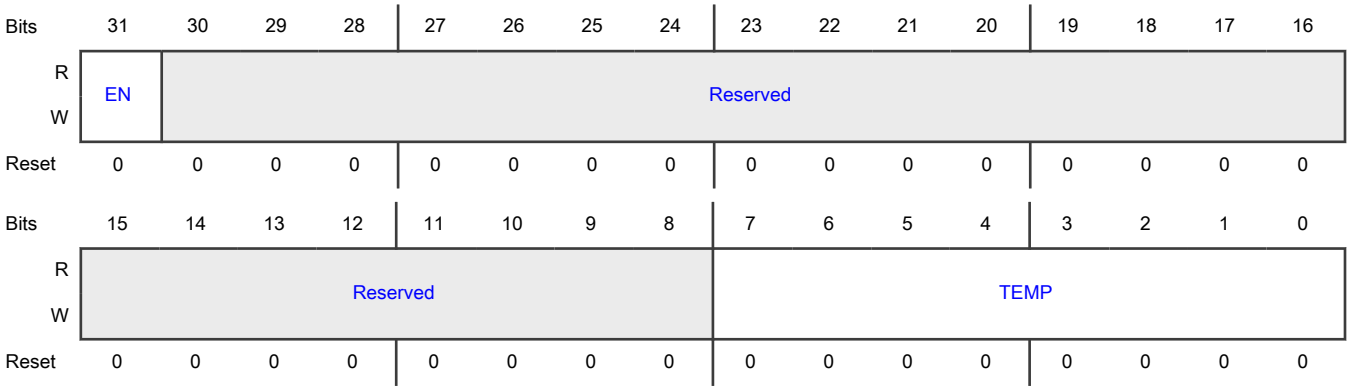
73.6.21 Monitor Rising Temperature Rate Critical Threshold (TMRTRCTR)

This register is used to detect a fast-rising temperature by calculating the difference between two valid successive measurements per site, based on the monitoring interval setting. This could indicate a critical event such as heat sink separation or cooling fan malfunction.

Offset

Register	Offset
TMRTRCTR	70h

Diagram



Fields

Field	Description
31 EN	Enable Threshold 0 - Disabled 1 - Threshold enabled
30-8 —	Reserved
7-0 TEMP	Temperature Difference Between Two Measurements From the Same Site This is a number between 0 K and 200 K. If this threshold is exceeded, the module sets the value of TIDR[RTRCT] to 1.

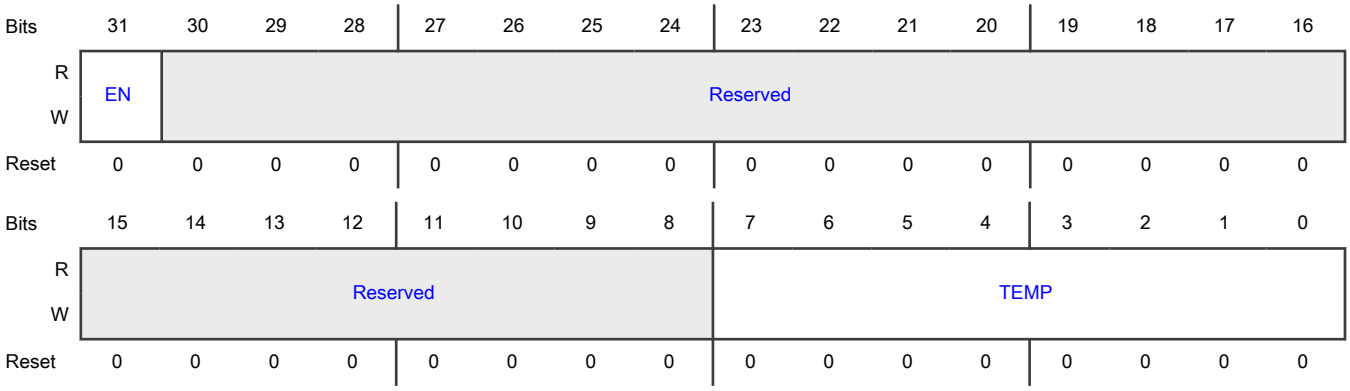
73.6.22 Monitor Falling Temperature Rate Critical Threshold (TMFTRCTR)

This register is used to detect a fast-falling temperature by calculating the difference between two valid successive measurements per site, based on the monitoring interval setting. Use case may include detecting stress to the chip because of excessive cooling, or cold-boot attacks on volatile memory.

Offset

Register	Offset
TMFTRCTR	74h

Diagram



Fields

Field	Description
31 EN	Enable threshold 0 - Disabled 1 - Threshold Enabled
30-8 —	Reserved
7-0 TEMP	Temperature Difference Between Two Measurements From the Same Site This is a (positive) number between 0 K and 200 K. If this threshold is exceeded, the module sets the value of TIDR[FTRCT] to 1.

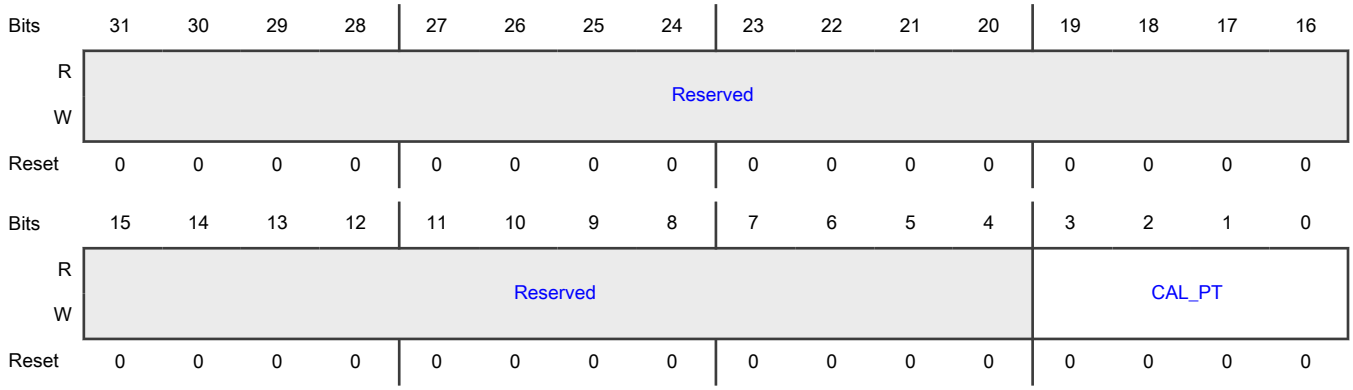
73.6.23 Temperature Configuration (TTCFGR)

This register, in conjunction with Sensor Configuration (TSCFGR), is used to initialize the internal sensor translation table used during monitoring. The register pair defines an indirect access to the table.

Offset

Register	Offset
TTCFGR	80h

Diagram



Fields

Field	Description
31-4 —	Reserved
3-0 CAL_PT	Calibration Point CAL_PT 0 is the lowest calibrated temperature. Each calibration point is related to the TTRCRn registers as shown below.

CAL_PT	Pointer to TTRCRn
0	TTRCR0
1	TTRCR1
2	TTRCR2
3	TTRCR3
4	TTRCR4
5	TTRCR5
6	TTRCR6
7	TTRCR7
8	TTRCR8
9	TTRCR9
10	TTRCR10
11	TTRCR11
12	TTRCR12
13	TTRCR13
14	TTRCR14
15	TTRCR15

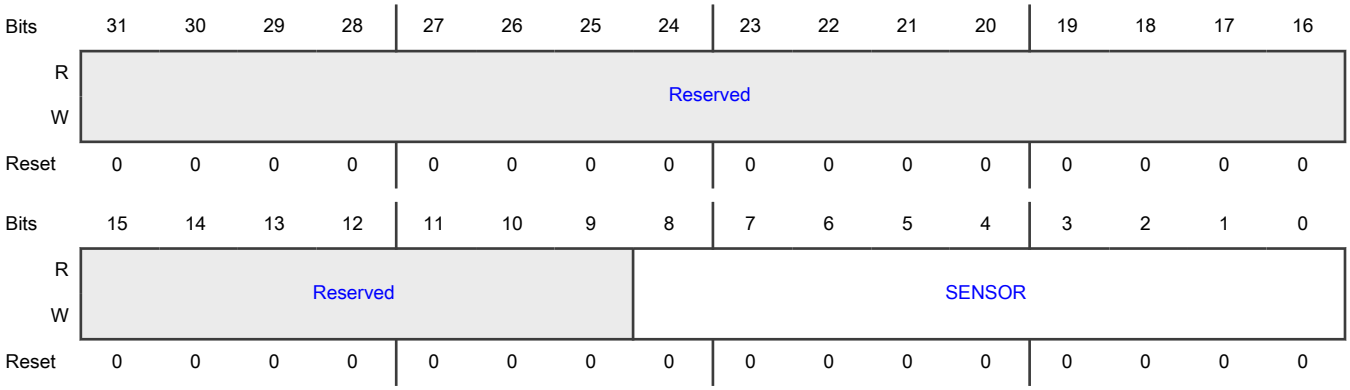
73.6.24 Sensor Configuration (TSCFGR)

This register, in conjunction with Temperature Configuration (TTCFGR), is used to initialize the internal sensor translation table used during monitoring. The register pair defines indirect access to the table. Reading Sensor Configuration (TSCFGR) returns the data from the translation table as defined in Temperature Configuration (TTCFGR).

Offset

Register	Offset
TSCFGR	84h

Diagram



Fields

Field	Description
31-9 —	Reserved
8-0 SENSOR	Sensor Value

73.6.25 Report Immediate Temperature at Site (TRITSR0 - TRITSR15)

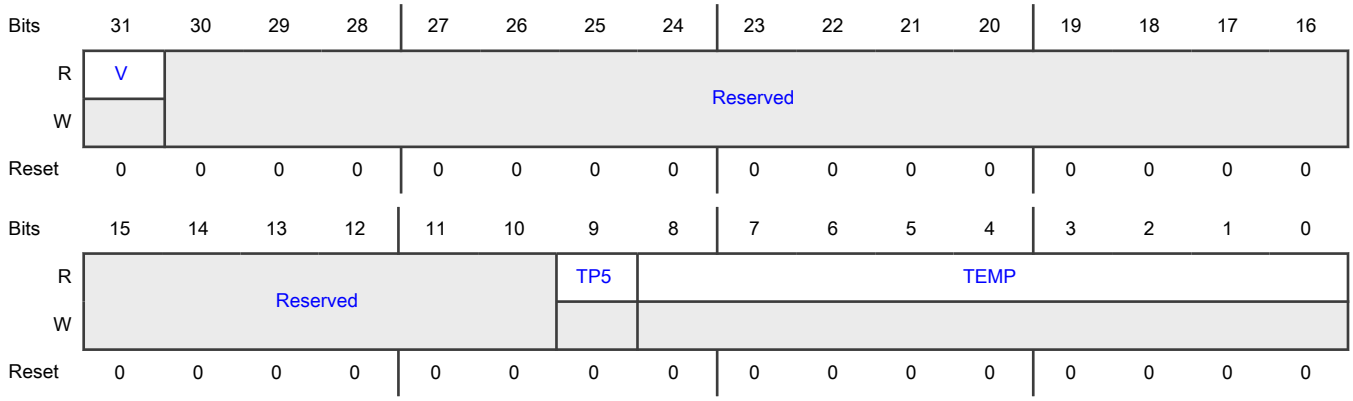
This register returns the last measured temperature at the site, which must be a part of the list of enabled monitored sites as defined by TMSR[SITE].

Offset

For a = 0 to 15:

Register	Offset
TRITSRa	100h + (a × 10h)

Diagram



Fields

Field	Description
31 V	Valid Measured Temperature 0 - Not valid; temperature is out of sensor calibration range or the first measurement is still pending 1 - Valid
30-10 —	Reserved
9 TP5	Last Temperature Reading in Kelvin at the Site Valid when V=1. When this field is 1, you must add 0.5 K to the temperature that TEMP reports. For example, if TEMP is 300 K and TP5=1, then the final temperature is 300.5 K.
8-0 TEMP	Last Temperature Reading in Kelvin at the Site Valid when V=1.

73.6.26 Report Average Temperature at Site (TRATSR0 - TRATSR15)

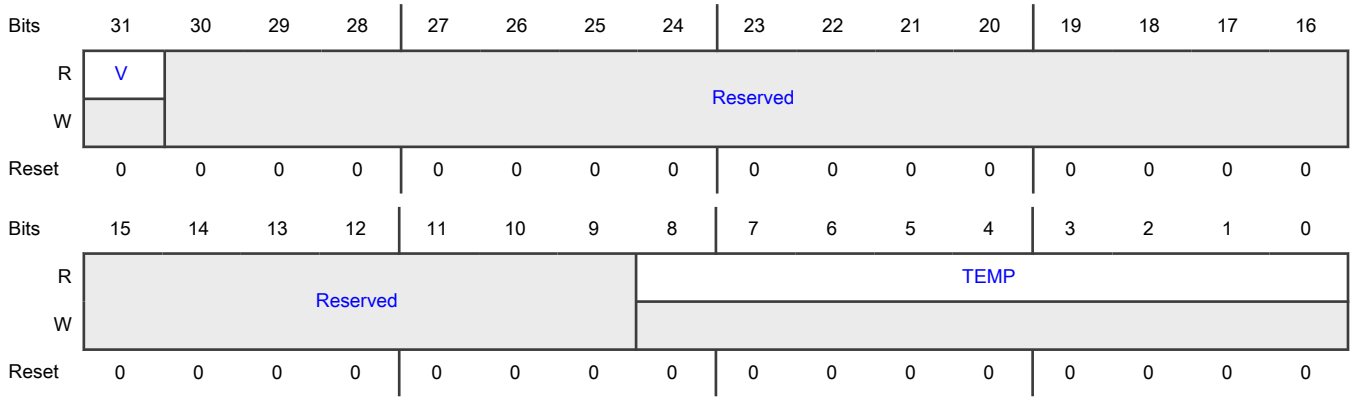
This register returns the average measured temperature at the site, which must be a part of the list of enabled monitored sites as defined by TMSR[SITE].

Offset

For a = 0 to 15:

Register	Offset
TRATSRa	104h + (a × 10h)

Diagram



Fields

Field	Description
31 V	Valid Measured Temperature 0 - Not valid; temperature is out of sensor range or the first measurement is still pending 1 - Valid
30-9 —	Reserved
8-0 TEMP	Average Temperature Reading in Kelvin at the Site Valid when V=1.

73.6.27 Central Module Configuration (TCMCFG)

This register controls the Successive Approximation Register (SAR) algorithm and clock settings. For proper operation, updates to the register must only occur when monitoring is disabled.

Writing to this register triggers an automatic clock speed change based on the new CLK_DIV and DFD settings. After writing to this register you must poll SAR_RDY to determine when it is safe to continue the monitoring operation or configuration procedure.

NOTE

The values for OCS, OCM, DEMA, DFDRCTC, and CLK_DIV are chip-specific. See the "Chip-specific TMU section" for the required settings.

Offset

Register	Offset
TCMCFG	F00h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DPM	OCM	OCS	DEMA	Reserved	RCTC			Reserved							SAR_RDY
W																
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CLK_DIV				DFD		CMET		Reserved	DAC_OFFSET						
W																
Reset	1	0	1	0	1	1	0	0	0	u[1]	u	u	u	u	u	u

1. The DAC_OFFSET value comes from SoC configuration and the values depend on the DAC calibration (See the chip specific information).

Fields

Field	Description
31 DPM	Dynamic Power Management See the chip-specific TMU information for configuration.
30 OCM	Offset Cancellation Mode See the chip-specific TMU information for configuration. **NOTE** This field must be set to 1 for the normal TMU monitor operation. For details on initialization, see Initialization.
29 OCS	Offset Cancellation Manual Setting See the chip-specific TMU information for configuration.
28 DEMA	Dynamic Element Match Averaging Mode See the chip-specific TMU information for configuration. **NOTE** This field must be set to 1 for the normal TMU monitor operation. For details on initialization, see Initialization.
27 —	Reserved
26-24 RCTC	RC Time Constant Setting See the chip-specific TMU information for configuration.
23-17 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
16 SAR_RDY	SAR Ready Indicates that the SAR module is ready to receive user commands. During normal functional mode, this field must be set for calibration or monitoring to work properly. During a clock switch, which occurs as a result of writing to this register, the SAR module may be temporarily unavailable. 0 - SAR not ready to receive command 1 - SAR ready to receive command
15-12 CLK_DIV	Clock Divider See the chip-specific TMU information for configuration.
11-10 DFD	Digital Filter Depth See the chip-specific TMU information for configuration.
9-8 CMET	Central Module Enable Time See the chip-specific TMU information for configuration.
7 —	Reserved
6-0 DAC_OFFSET	DAC Offset See the chip-specific TMU information for configuration.

73.6.28 Temperature Range Control a (TTRCR0 - TTRCR15)

These registers allow you to define the calibration points used to cover the temperature sensor range. TTRCR0 defines the start of the temperature range which, along with TTRCR1, must always be valid to account for at least two calibration points. Provide additional calibration points to account for any non-linear behavior in the sensor as a function of temperature. A perfectly linear sensor reading requires only two calibration points to find the temperature based on linear interpolation. You must enable the calibration points sequentially, starting with TTRCR0 through TTRCR15.

NOTE

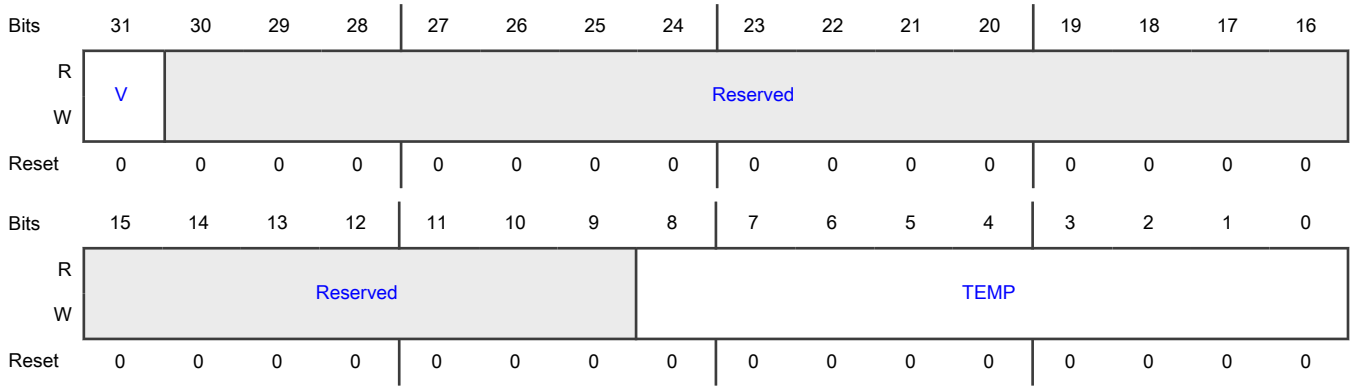
The circuit has been calibrated for measuring temperatures for a range defined in the provided configuration table. Without prior circuit knowledge, it is not recommended to go beyond this range when programming up the calibration points. The margin of error outside these limits cannot be guaranteed.

Offset

For a = 0 to 15:

Register	Offset
TTRCRa	F10h + (a × 4h)

Diagram



Fields

Field	Description
31 V	Calibration Point is Valid 0 - Not valid 1 - Valid
30-9 —	Reserved
8-0 TEMP	Temperature in Kelvin for the Calibration Point In the case of TTRCR0, this is the starting temperature for the range.

73.7 Glossary

Op amp	Operational amplifier
PTAT	Proportional to absolute temperature

Chapter 74
PLLs

74.1 Overview

PLL (as a fractional-N synthesizer) supports a 30-bit numerator and denominator. Numerator is a signed number that you can use to adjust the fractional portion of the feedback divider dynamically.

The PLL digital control unit provides a user interface and control of the PLL system, which consists of the following:

- A single PLL analog block
- A digital interface
- DFS blocks with a dedicated output divided by two dividers for fractional division with different output clocks and status signals

74.1.1 Block diagram

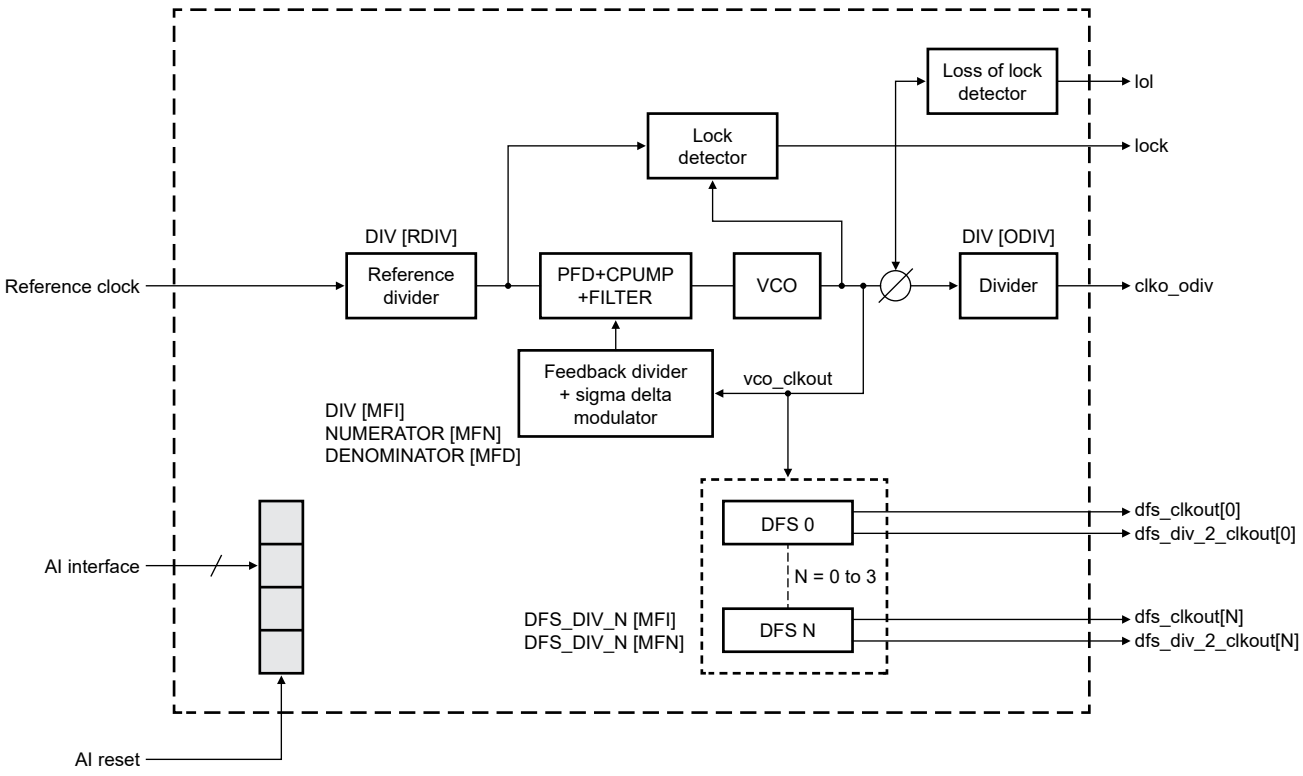


Figure 439. Block diagram

74.1.2 Features

- Programmable frequency modulation
- Programmable output dividers
- Capability to change runtime programmable fractional portion of feedback divider
- Programmable clock bypass feature
- Integer and fractional loop dividers

- Spread spectrum feature
- Support for DFS blocks
- DFS block has dedicated output divided by 2 dividers with different output clocks and status signals

74.2 Functional description

This section describes the operation and configuration of PLL.

74.2.1 PLL settings for operation in different modes

CTRL[POWERUP] and SPREAD_SPECTRUM[ENABLE] work together to enable or disable PLL in different modes. The following table lists the value settings of these fields.

Table 640. PLL settings for operation in different modes

CTRL[POWERUP]	SPREAD_SPECTRUM[ENABLE]	Functionality
0	0/1	PLL is disabled.
1	0	PLL is operational in both Integer and Fractional modes.
1	1	PLL is operational in SSCG mode.

74.2.2 Frequency range

PLL supports a reference clock frequency (F_{ref} ÷ DIV[RDIV]) in the range of 20−40 MHz, and VCO clock frequency (F_{vco_clk}) in the range of 2.5−5 GHz.

74.2.3 Clock configuration

74.2.3.1 PLL output clock frequency

$$F_{clko_odiv} = \frac{F_{vco_clk}}{DIV[ODIV]}$$

Equation 50. PLL output clock frequency

74.2.3.2 DFS output clock frequency

$$F_{dfs_clkout_N} = \frac{F_{vco_clk}}{DFS_DIV_N[MFI] + \left(\frac{DFS_DIV_N[MFN]}{5}\right)} \quad N = 0 \text{ to } 3$$

Equation 51. DFS output clock frequency

74.2.3.3 PLL VCO clock frequency in Integer mode

$$F_{vco_clk} = \frac{F_{ref}}{DIV[RDIV]} \times DIV[MFI]$$

Equation 52. PLL VCO clock frequency in Integer mode

74.2.3.4 PLL VCO clock frequency in Fractional mode

$$F_{vco_clk} = \frac{F_{ref}}{DIV[RDIV]} \times \left(DIV[MFI] + \frac{NUMERATOR[MFN]}{DENOMINATOR[MFD]} \right)$$

$$where \ \frac{NUMERATOR[MFN]}{DENOMINATOR[MFD]}$$

must be in the range of −2 to +2 when spread spectrum is disabled
must be in the range of 0 to 1 when spread spectrum is enabled

Equation 53. PLL VCO clock frequency in Fractional mode

74.2.4 Spread spectrum

This section describes the types of spread depending on the spread spectrum frequency, and the spread spectrum configuration along with a sample calculation.

74.2.4.1 Types of spreads

In Spread Spectrum mode, you must modulate the frequency of the output clock to lower the Electromagnetic interference (EMI) (for some specific applications). The frequency varies with time in a triangular fashion as shown in Figure 440 and Figure 441.

The time taken by the peak frequency to repeat itself is the modulation time and its inverse is called modulation frequency. The difference between high and low peak frequency is called spread frequency or modulation depth.

The spreads can be of the following two types:

- Center spread: The high peak frequency is above the base frequency by (spread_frequency ÷ 2) and low peak frequency is below the base frequency by (spread_frequency ÷ 2).
- Down spread: The high peak frequency is the base frequency itself, and low peak frequency is (base_frequency - spread_frequency).

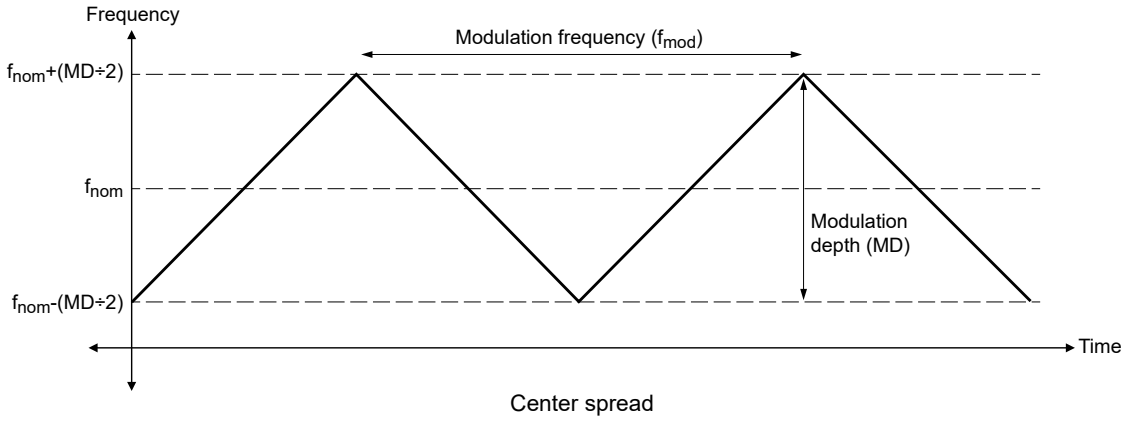


Figure 440. Spread spectrum frequency change vs time plot (center spread)

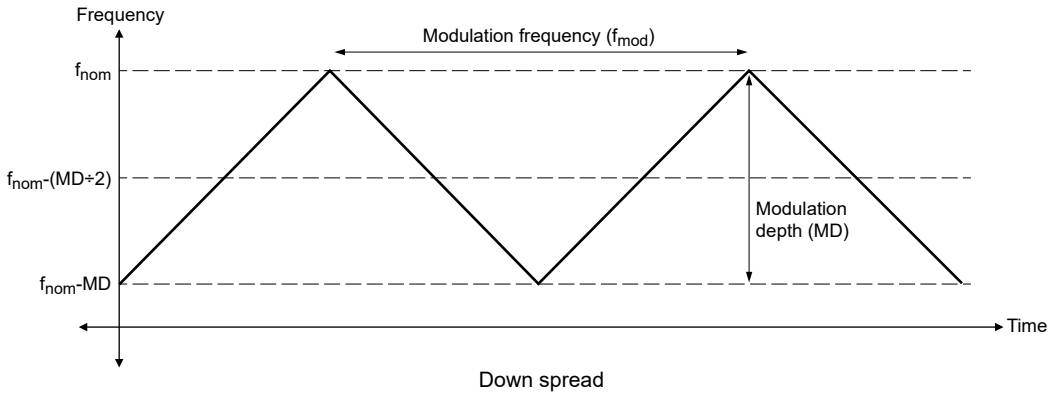


Figure 441. Spread spectrum frequency change vs time plot (down spread)

74.2.4.2 Spread spectrum configuration

You can configure Spread Spectrum mode using the following register fields:

- SPREAD_SPECTRUM[ENABLE]: Enables or disables spread spectrum.
- CTRL[SPREADCTL]: Selects between center spread and down spread.
- SPREAD_SPECTRUM[STEP] and SPREAD_SPECTRUM[STOP]: Helps program the modulation and spread frequency.

Prior to enabling the spread spectrum, you must configure the base frequency in either Integer mode or Fractional mode.

Use Equation 54 to calculate spread frequency.

$$spread_{freq} = \frac{SPREAD_SPECTRUM[STOP]}{DENOMINATOR[MFD]} \times \frac{F_{ref}}{DIV[RDIV]}$$

Equation 54. Spread frequency

Ensure that the following conditions are fulfilled for spread spectrum configuration:

- The value of SPREAD_SPECTRUM[STOP] must be less than four times the value of DENOMINATOR[MFD].
- The calculated spread frequency must not exceed the value of 4 × (F_{ref} ÷ DIV[RDIV]). For example, if (F_{ref} ÷ DIV[RDIV]) is 20 MHz, then spread frequency must be within 4 × 20 MHz, which means 80 MHz.

Use Equation 55 to calculate modulation frequency.

$$Modulation_{freq} = \frac{F_{ref}}{DIV[RDIV] \times 2} \times \frac{SPREAD_SPECTRUM[STEP]}{SPREAD_SPECTRUM[STOP]}$$

Equation 55. Modulation frequency

If (F_{ref} ÷ DIV[RDIV]) = 40 MHz, then modulation frequency must lie between 30 KHz and 128 KHz, and if (F_{ref} ÷ DIV[RDIV])) is less than 40 MHz, then modulation frequency must lie between 30 KHz and 64 KHz.

Use Equation 56 to calculate the step by which the frequency increases or decreases.

$$Step_{freq}change = \frac{SPREAD_SPECTRUM[STEP]}{DENOMINATOR[MFD]} \times \frac{F_{ref}}{DIV[RDIV]}$$

Equation 56. Step frequency change

The range in which the frequency varies during spread spectrum depends on the type of spread you program:

- For center spread, the range is from (F_{vco_clk} - (spread_frequency ÷ 2)) to (F_{vco_clk} + (spread_frequency ÷ 2)).
- For down spread, the range is from (F_{vco_clk} - spread_frequency) to F_{vco_clk}.

74.2.4.3 Sample calculations

Table 641. Calculations for SPREAD_SPECTRUM[STOP] and SPREAD_SPECTRUM[STEP]

Assumed parameters	SPREAD_SPECTRUM[STOP]	SPREAD_SPECTRUM[STEP]
• F_{ref} = 40 MHz • DIV[RDIV] = 1 • DENOMINATOR[MFD] = 10000 • NUMERATOR[MFN] = 9000 • Spread frequency = 32 MHz • Modulation frequency = 50 KHz	Formula: SPREAD_SPECTRUM[STOP] = (spread frequency × DENOMINATOR[MFD]) ÷ (F_{ref} ÷ DIV[RDIV])	Formula: SPREAD_SPECTRUM[STEP] = (modulation frequency × SPREAD_SPECTRUM[STOP]) ÷ (F_{ref} ÷ (DIV[RDIV] × 2))
	Calculation: SPREAD_SPECTRUM[STOP] = (32 MHz × 10000) ÷ (40 MHz ÷ 1) Therefore, SPREAD_SPECTRUM[STOP] = (32 × 10000) ÷ (40) = 8000	Calculation: SPREAD_SPECTRUM[STEP] = (50 KHz × 8000) ÷ (40 MHz ÷ (1 × 2)) Therefore,SPREAD_SPECTRUM[STEP] = (50 × 8) ÷ (40 × 2) = 20

74.2.5 Dynamic MFN update in PLL

In PLL, NUMERATOR[MFN] can be updated dynamically.

NOTE

If NUMERATOR[MFN] is to be updated dynamically, it is essential to ensure that the resulting frequency change does not exceed 1%. Otherwise, the PLL may lose its lock, leading to a loss of signal assertion.

74.2.6 Dynamic MFN update in DFS

In DFS, DFS_DIV_N[MFN] and DFS_DIV_N[MFI] can be updated dynamically.

74.2.7 Interrupts

This module has no interrupts.

74.3 External signals

This module has no external signals.

74.4 Initialization

74.4.1 Integer mode

Table 642. Initialization in Integer mode

Step	Purpose	Programming	Notes
1	Specify the divider value according to Integer mode.	Configure the following fields: • DIV[ODIV] • DIV[RDIV] • DIV[MFI]	—
2	—	Wait for 5 µs.	—
3	Enable the PLL block.	Configure CTRL[POWERUP].	—
4	Wait for lock.	Poll PLL_STATUS[PLL_LOCK] and wait until it becomes 1 and the pll_lock output signal is asserted.	The maximum wait time is 100 µs.
5	Enable the PLL output clock.	Write 1 to CTRL[CLKMUX_EN].	—
6	Specify the DFS division integer and fraction according to Integer mode.	Write appropriate values to DFS_DIV_N[MFN] and DFS_DIV_N[MFI].	N ranges from 0–3 and is calculated as NO_OF_DFS - 1.
7	Enable DFS.	Write 1 to DFS_CTRL_N[ENABLE].	N ranges from 0–3 and is calculated as NO_OF_DFS - 1.
8	Enable the PLL DFS clock.	Write 1 to DFS_CTRL_N[CLKOUT_EN] and/or DFS_CTRL_N[CLKOUT_DIVBY2_EN].	—
9	Wait for the DFS output clock to become valid.	Poll DFS_STATUS[DFS_OK] and wait until it becomes 1 and the dfs_ok[N-1] output signal is asserted.	—

NOTE

To change the value of DFS_DIV_N[MFI] or DFS_DIV_N[MFN] after DFS_STATUS[DFS_OK] becomes 1, configure DFS_DIV_N[MFI] or DFS_DIV_N[MFN] and check the DFS output clock.

74.4.2 Fractional mode

Table 643. Initialization in Fractional mode

Steps	Purpose	Programming	Notes
1	Specify the divider values according to Fractional mode.	Configure the following fields: • DIV[ODIV] • DIV[RDIV] • DIV[MFI]	—
2	Disable Spread-Spectrum mode.	Write 0 to SPREAD_SPECTRUM[ENABLE].	—

Table continues on the next page...

Table 643. Initialization in Fractional mode (continued)

Steps	Purpose	Programming	Notes
3	Specify the numerator and denominator of the fraction according to Fractional mode.	Configure NUMERATOR[MFN] and DENOMINATOR[MFD].	—
4	—	Wait for 5 µs.	—
5	Enable the PLL block.	Configure CTRL[POWERUP].	—
6	Wait for lock.	Poll PLL_STATUS[PLL_LOCK] and wait until it becomes 1 and the pll_lock output signal is asserted.	The maximum wait time is 100 µs.
7	Enable the PLL output clock.	Write 1 to CTRL[CLKMUX_EN].	—
8	Verify that the value of PLL_STATUS[ANA_MFN] matches the configured value of NUMERATOR[MFN].	Read PLL_STATUS[ANA_MFN].	—
9	To change the value of NUMERATOR[MFN] after PLL_STATUS[PLL_LOCK] becomes 1 (if needed).	Configure NUMERATOR[MFN] and check the PLL output clock.	—
10	Verify that the value of PLL_STATUS[ANA_MFN] matches the configured value of NUMERATOR[MFN].	Read PLL_STATUS[ANA_MFN].	—
11	Specify the DFS division integer and fraction according to Integer mode.	Write appropriate values to DFS_DIV_N[MFN] and DFS_DIV_N[MFI].	• N ranges from 0–3 and is calculated as NO_OF_DFS - 1. • The range for DFS_DIV_N[MFI] is 2–255 and for DFS_DIV_N[MFN] is 0–4.
12	Enable DFS.	Write 1 to DFS_CTRL_N[ENABLE].	N ranges from 0–3 and is calculated as NO_OF_DFS - 1.
13	Enable the PLL DFS clock.	Write 1 to DFS_CTRL_N[CLKOUT_EN] and/or DFS_CTRL_N[CLKOUT_DIVBY2_EN].	
14	Wait for the DFS output clock to become valid.	Poll DFS_STATUS[DFS_OK] and wait until it becomes 1 and the dfs_ok[N-1] output signal is asserted.	—

NOTE

To change the value of DFS_DIV_N[MFI] or DFS_DIV_N[MFN] after DFS_STATUS[DFS_OK] becomes 1, configure DFS_DIV_N[MFI] or DFS_DIV_N[MFN] and check the DFS output clock.

74.4.3 Spread-Spectrum mode

Table 644. Initialization in Spread-Spectrum mode

Steps	Purpose	Programming	Notes
1	Specify the divider values according to Fractional mode.	Configure the following fields: • DIV[ODIV] • DIV[RDIV] • DIV[MFI]	—
2	Enable Spread-Spectrum mode.	Write 1 to SPREAD_SPECTRUM[ENABLE].	—
3	Specify the step and stop values according to Spread spectrum.	Configure SPREAD_SPECTRUM[STEP] and SPREAD_SPECTRUM[STOP].	—
4	Specify the numerator and denominator of the fraction according to Fractional mode.	Configure NUMERATOR[MFN] and DENOMINATOR[MFD].	Ratio of MFN and MFD should be between 0 and 1.
5	—	Wait for 5 µs.	—
6	Enable the PLL block.	Configure CTRL[POWERUP].	—
7	Wait for lock.	Poll PLL_STATUS[PLL_LOCK] and wait until it becomes 1 and the pll_lock output signal is asserted.	The maximum wait time is 100 µs.
8	Enable the PLL output clock.	Write 1 to CTRL[CLKMUX_EN].	—
9	Verify that the value of PLL_STATUS[ANA_MFN] matches the configured value of NUMERATOR[MFN].	Read PLL_STATUS[ANA_MFN].	—
10	To change the value of NUMERATOR[MFN] after PLL_STATUS[PLL_LOCK] becomes 1 (if needed).	Configure NUMERATOR[MFN] and check the PLL output clock.	—
11	Verify that the value of PLL_STATUS[ANA_MFN] matches the configured value of NUMERATOR[MFN].	Read PLL_STATUS[ANA_MFN].	—
12	Specify the DFS division integer and fraction according to Integer mode.	Write appropriate values to DFS_DIV_N[MFN] and DFS_DIV_N[MFI].	• N ranges from 0–3 and is calculated as NO_OF_DFS - 1. • The range for DFS_DIV_N[MFI] is 2–255 and for DFS_DIV_N[MFN] is 0–4.
13	Enable DFS.	Write 1 to DFS_CTRL_N[ENABLE].	N ranges from 0–3 and is calculated as NO_OF_DFS - 1.
14	Enable the PLL DFS clock.	Write 1 to DFS_CTRL_N[CLKOUT_EN] and/or	—

Table continues on the next page...

Table 644. Initialization in Spread-Spectrum mode (continued)

Steps	Purpose	Programming	Notes
		DFS_CTRL_N[CLKOUT_DIVBY2_EN].	
15	Wait for the DFS output clock to become valid.	Poll DFS_STATUS[DFS_OK] and wait until it becomes 1 and the dfs_ok[N-1] output signal is asserted.	—

NOTE

To change the value of DFS_DIV_N[MFI] or DFS_DIV_N[MFN] after DFS_STATUS[DFS_OK] becomes 1, configure DFS_DIV_N[MFI] or DFS_DIV_N[MFN] and check the DFS output clock.

74.4.4 Powering down the module

Table 645. Settings for powering down the module

Steps	Purpose	Programming	Notes
1	Gate the DFS clock output.	Write 0 to DFS_CTRL_N[CLKOUT_EN].	N = 0–3 and is calculated as NO_OF_DFS - 1.
2	—	Wait for one AI clock cycle.	—
3	Disable DFS.	Write 0 to DFS_CTRL_N[ENABLE].	N = 0–3 and is calculated as NO_OF_DFS - 1.
4	Verify whether DFS is disabled.	Poll DFS_STATUS[DFS_OK].	—
5	—	Wait for three AI clock cycles.	—
6	Gate the PLL clock output.	Write 0 to CTRL[CLKMUX_EN].	—
7	—	Wait for one AI clock cycle.	—
8	Disable PLL.	Write 0 to CTRL[POWERUP].	—
9	Verify whether PLL is disabled.	Poll PLL_STATUS[PLL_LOCK].	—
10	Reset the module properly.	Prior to enabling PLL again, wait for 5 µs.	—

74.5 Memory map and register definitions

This section includes the PLL module memory map and detailed descriptions of all registers.

74.5.1 PLL register descriptions

74.5.1.1 PLL memory map

SYSPLL base address: 4448_1100h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	PLL Control (CTRL)	32	RW	0000_0000
4	PLL Control (CTRL_SET)	32	RW	0000_0000
8	PLL Control (CTRL_CLR)	32	RW	0000_0000
C	PLL Control (CTRL_TOG)	32	RW	0000_0000
30	Spread Spectrum (SPREAD_SPECTRUM)	32	RW	0000_0000
34	Spread Spectrum (SPREAD_SPECTRUM_SET)	32	RW	0000_0000
38	Spread Spectrum (SPREAD_SPECTRUM_CLR)	32	RW	0000_0000
3C	Spread Spectrum (SPREAD_SPECTRUM_TOG)	32	RW	0000_0000
40	Numerator (NUMERATOR)	32	RW	0000_0000
44	Numerator (NUMERATOR_SET)	32	RW	0000_0000
48	Numerator (NUMERATOR_CLR)	32	RW	0000_0000
4C	Numerator (NUMERATOR_TOG)	32	RW	0000_0000
50	Denominator (DENOMINATOR)	32	RW	0000_0001
54	Denominator (DENOMINATOR_SET)	32	RW	0000_0001
58	Denominator (DENOMINATOR_CLR)	32	RW	0000_0001
5C	Denominator (DENOMINATOR_TOG)	32	RW	0000_0001
60	PLL Dividers (DIV)	32	RW	00C8_0000
64	PLL Dividers (DIV_SET)	32	RW	00C8_0000
68	PLL Dividers (DIV_CLR)	32	RW	00C8_0000
6C	PLL Dividers (DIV_TOG)	32	RW	00C8_0000
70	DFS Control (DFS_CTRL_0)	32	RW	0000_0000
74	DFS Control (DFS_CTRL_0_SET)	32	RW	0000_0000
78	DFS Control (DFS_CTRL_0_CLR)	32	RW	0000_0000
7C	DFS Control (DFS_CTRL_0_TOG)	32	RW	0000_0000
80	DFS Division_N (DFS_DIV_0)	32	RW	0000_0000
84	DFS Division_N (DFS_DIV_0_SET)	32	RW	0000_0000
88	DFS Division_N (DFS_DIV_0_CLR)	32	RW	0000_0000
8C	DFS Division_N (DFS_DIV_0_TOG)	32	RW	0000_0000
90	DFS Control (DFS_CTRL_1)	32	RW	0000_0000
94	DFS Control (DFS_CTRL_1_SET)	32	RW	0000_0000
98	DFS Control (DFS_CTRL_1_CLR)	32	RW	0000_0000
9C	DFS Control (DFS_CTRL_1_TOG)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
A0	DFS Division_N (DFS_DIV_1)	32	RW	0000_0000
A4	DFS Division_N (DFS_DIV_1_SET)	32	RW	0000_0000
A8	DFS Division_N (DFS_DIV_1_CLR)	32	RW	0000_0000
AC	DFS Division_N (DFS_DIV_1_TOG)	32	RW	0000_0000
B0	DFS Control (DFS_CTRL_2)	32	RW	0000_0000
B4	DFS Control (DFS_CTRL_2_SET)	32	RW	0000_0000
B8	DFS Control (DFS_CTRL_2_CLR)	32	RW	0000_0000
BC	DFS Control (DFS_CTRL_2_TOG)	32	RW	0000_0000
C0	DFS Division_N (DFS_DIV_2)	32	RW	0000_0000
C4	DFS Division_N (DFS_DIV_2_SET)	32	RW	0000_0000
C8	DFS Division_N (DFS_DIV_2_CLR)	32	RW	0000_0000
CC	DFS Division_N (DFS_DIV_2_TOG)	32	RW	0000_0000
F0	PLL Status (PLL_STATUS)	32	R	0000_0000
F4	DFS Status (DFS_STATUS)	32	R	0000_0000

74.5.1.2 PLL Control (CTRL)

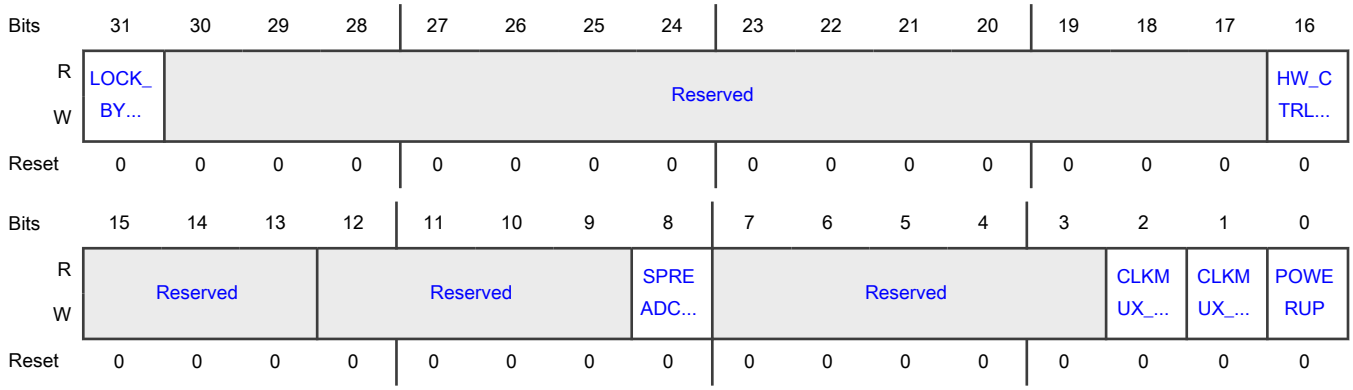
Enables PLL control and contains fractional PLL control registers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
CTRL	0h	PLL Control
CTRL_SET	4h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 1
CTRL_CLR	8h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 0
CTRL_TOG	Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CTRL

Diagram



Fields

Field	Description
31 LOCK_BYPASS	Lock Bypass Enables bypass for the lock detector. 0 - Disable 1 - Enable
30-17 —	Reserved
16 HW_CTRL_SEL	Hardware Control Select Controls PLL through hardware. If this field is disabled, PLL is controlled by register. If this field is enabled, PLL is controlled by hardware inputs and you cannot change NUMERATOR[MFN] dynamically. 0 - Disable 1 - Enable
15-13 —	Reserved
12-9 —	Reserved
8 SPREADCTL	Modulation Type Select Selects whether modulation is centered around, or spread below the nominal frequency. 0 - Centered around nominal frequency 1 - Spread below nominal frequency
7-3 —	Reserved
2	CLKMUX_Bypass

Table continues on the next page...

Table continued from the previous page...

Field	Description
CLKMUX_BYPASS	Bypasses PLL. The output clock is the PLL reference clock source. 0 - Normal mode 1 - PLL Bypass mode
1 CLKMUX_EN	CLKMUX Enable Enables output clock mux. 0 - Disable 1 - Enable
0 POWERUP	Power Up Enables PLL. 0 - Disable 1 - Enable

74.5.1.3 Spread Spectrum (SPREAD_SPECTRUM)

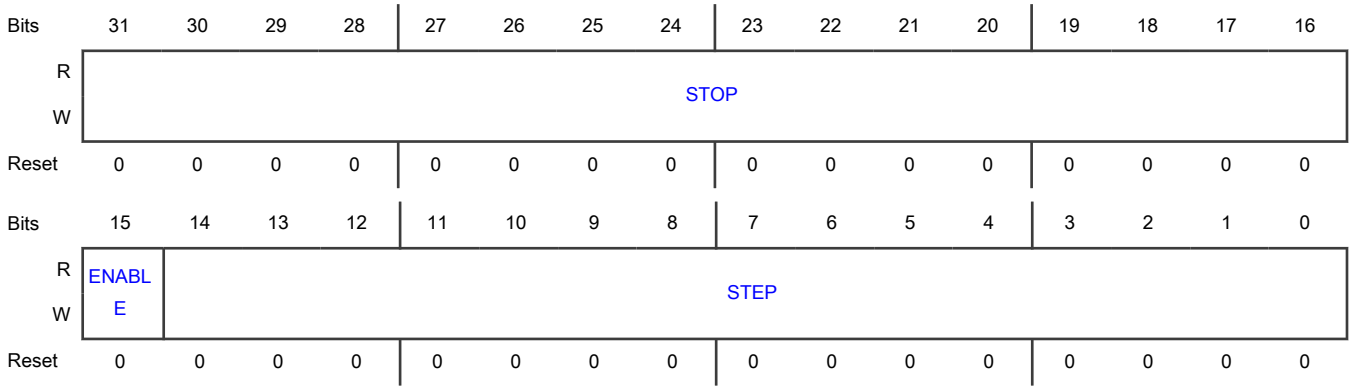
Sets spread spectrum control.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
SPREAD_SPECTRUM	30h	Spread Spectrum
SPREAD_SPECTRUM_SET	34h	Writing 1 to a bit in this register ensures that the corresponding bit in SPREAD_SPECTRUM is 1
SPREAD_SPECTRUM_CLR	38h	Writing 1 to a bit in this register ensures that the corresponding bit in SPREAD_SPECTRUM is 0
SPREAD_SPECTRUM_TOG	3Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in SPREAD_SPECTRUM

Diagram



Fields

Field	Description
31-16 STOP	Stop Sets spread spectrum stop.
15 ENABLE	Enable Enables or disables the spread spectrum modulation. 0 - Disable 1 - Enable
14-0 STEP	Step Sets spread spectrum step.

74.5.1.4 Numerator (NUMERATOR)

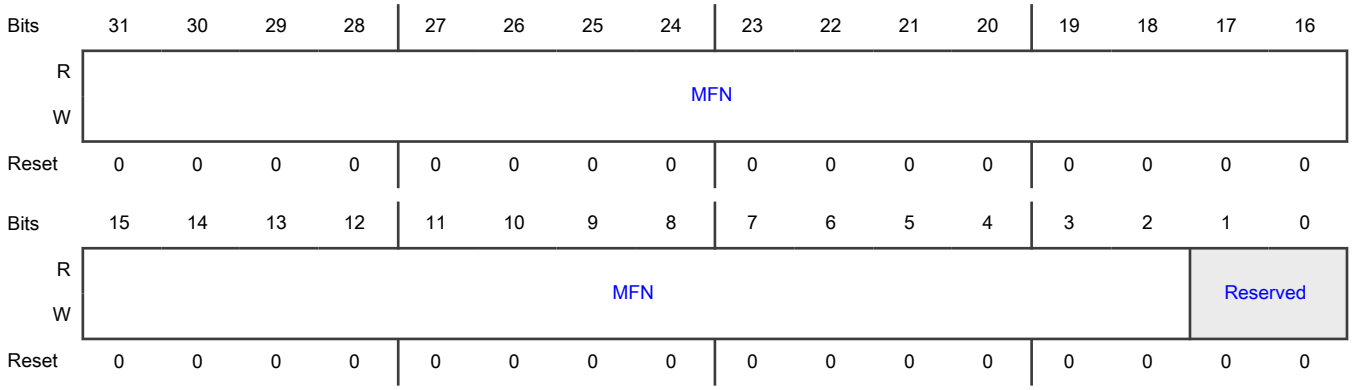
Sets numerator control.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
NUMERATOR	40h	Numerator
NUMERATOR_SET	44h	Writing 1 to a bit in this register ensures that the corresponding bit in NUMERATOR is 1
NUMERATOR_CLR	48h	Writing 1 to a bit in this register ensures that the corresponding bit in NUMERATOR is 0
NUMERATOR_TOG	4Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in NUMERATOR

Diagram



Fields

Field	Description
31-2 MFN	Numerator Sets the numerator of the fractional PLL loop divider (signed number). You can configure this field dynamically when PLL is enabled, except when CTRL[HW_CTRL_SEL] is 1.
1-0 —	Reserved

74.5.1.5 Denominator (DENOMINATOR)

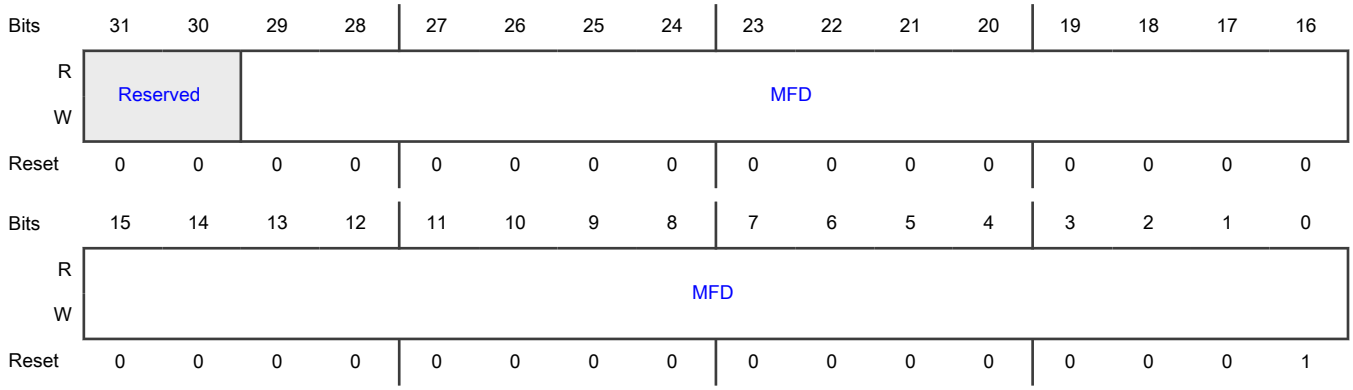
Sets denominator control.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DENOMINATOR	50h	Denominator
DENOMINATOR_SET	54h	Writing 1 to a bit in this register ensures that the corresponding bit in DENOMINATOR is 1
DENOMINATOR_CLR	58h	Writing 1 to a bit in this register ensures that the corresponding bit in DENOMINATOR is 0
DENOMINATOR_TOG	5Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in DENOMINATOR

Diagram



Fields

Field	Description
31-30 —	Reserved
29-0 MFD	Denominator Sets the denominator of the fractional loop divider (unsigned number). **NOTE** You are not allowed to write 0 to this field.

74.5.1.6 PLL Dividers (DIV)

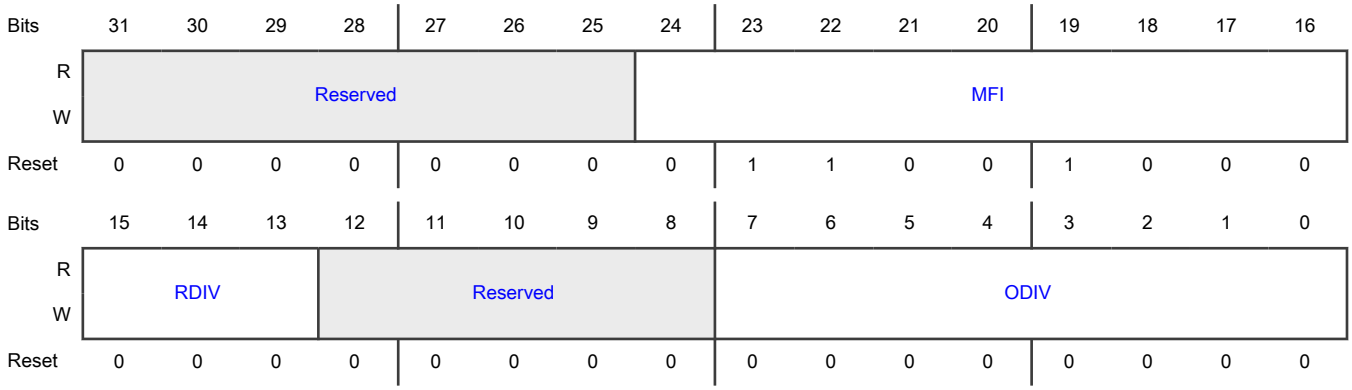
Controls PLL's digital and post dividers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DIV	60h	PLL Dividers
DIV_SET	64h	Writing 1 to a bit in this register ensures that the corresponding bit in DIV is 1
DIV_CLR	68h	Writing 1 to a bit in this register ensures that the corresponding bit in DIV is 0
DIV_TOG	6Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in DIV

Diagram



Fields

Field	Description
31-25 —	Reserved
24-16 MFI	Integer Portion of Loop Divider Sets the value of the divider in the PLL feedback loop. The value specified establishes the multiplication factor applied to the reference frequency. Divider value = MFI, where the chosen MFI does not violate VCO frequency specifications.
15-13 RDIV	Input Clock Predivider Sets the input clock divider. The output of the predivider circuit generates the PLL analog loop reference clock. 000 - Divide by 1 001 - Divide by 1 010 - Divide by 2 011 - Divide by 3 100 - Divide by 4 101 - Divide by 5 110 - Divide by 6 111 - Divide by 7
12-8 —	Reserved
7-0 ODIV	Output Frequency Divider for Clock Output Determines the VCO clock post divider for driving the final output clock. The value can be configured in the range of 0 to 255. If value is programmed as 0 or 1, output frequency will be F_{vco_clk}÷2. For all other values, output frequency will be as described in PLL output clock frequency.

74.5.1.7 DFS Control (DFS_CTRL_0 - DFS_CTRL_2)

Configures the DFS block.

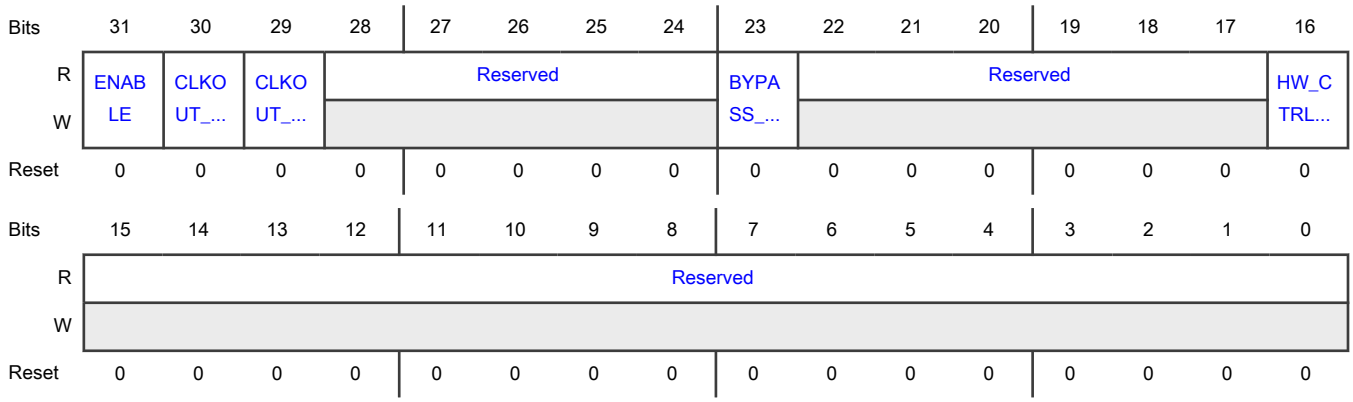
Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

For a = 0 to 2:

Register	Offset	Description
DFS_CTRL_a	70h + (a × 20h)	DFS Control
DFS_CTRL_a_SET	74h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in DFS_CTRL_a is 1
DFS_CTRL_a_CLR	78h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in DFS_CTRL_a is 0
DFS_CTRL_a_TOG	7Ch + (a × 20h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in DFS_CTRL_a

Diagram



Fields

Field	Description
31 ENABLE	DFS Block Enable Enables the DFS block. 0 - Disable 1 - Enable
30 CLKOUT_EN	DFS Clock Output Enable Enables the DFS clock output. 0 - Disable 1 - Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
29 CLKOUT_DIVBY2_EN	DFS Clock Output Divide by 2 Enable Enables DFS divide by 2 clock output. 0 - Disable 1 - Enable
28-24 —	Reserved
23 BYPASS_EN	Bypass Enable Enables bypass for DFS. The DFS clock output is the bypass clock source. 0 - Disable 1 - Enable
22-17 —	Reserved
16 HW_CTRL_SEL	Hardware Control Select Controls DFS through hardware. 0 - Controlled by register 1 - Controlled by hardware inputs
15-0 —	Reserved

74.5.1.8 DFS Division_N (DFS_DIV_0 - DFS_DIV_2)

Specifies the values for MFN and MFI in the DFS block.

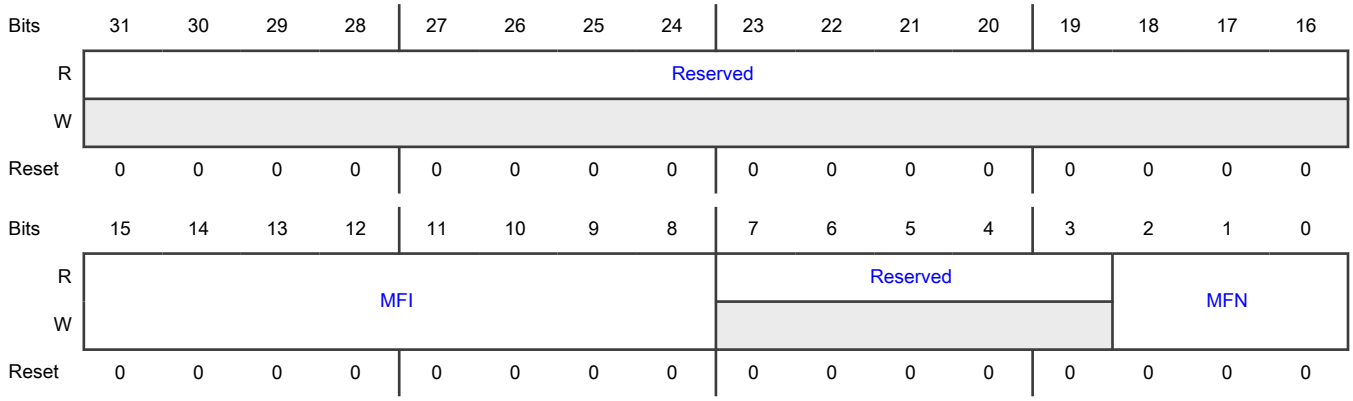
Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

For a = 0 to 2:

Register	Offset	Description
DFS_DIV_a	80h + (a × 20h)	DFS Division_N
DFS_DIV_a_SET	84h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in DFS_DIV_a is 1
DFS_DIV_a_CLR	88h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in DFS_DIV_a is 0
DFS_DIV_a_TOG	8Ch + (a × 20h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in DFS_DIV_a

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 MFI	MFI Sets the integer part of the division factor for port *n*.
7-3 —	Reserved
2-0 MFN	MFN Sets the numerator of the fractional part of the division factor for port *n*.

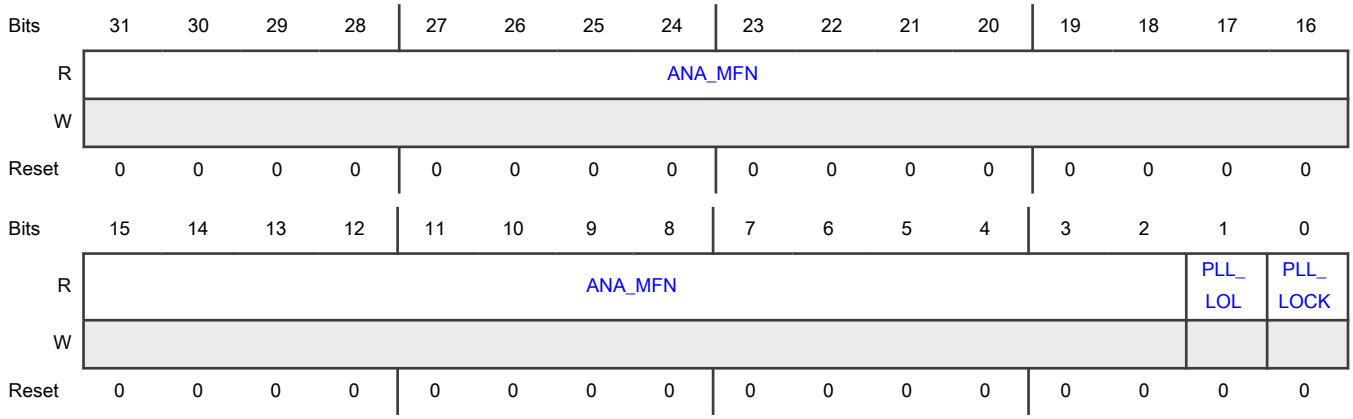
74.5.1.9 PLL Status (PLL_STATUS)

Indicates analog capture MNN values for debug purposes and PLL_LOCK status.

Offset

Register	Offset
PLL_STATUS	F0h

Diagram



Fields

Field	Description
31-2 ANA_MFN	ANA_MFN Stores analog block capture MFN value for debug purposes.
1 PLL_LOL	PLL_LOL Captures the loss of lock (LoL) signal when you write 0 to PLL_LOCK.
0 PLL_LOCK	PLL_LOCK Captures the lock signal to confirm whether the PLL clock is stable after you write 1 to PLL_LOCK.

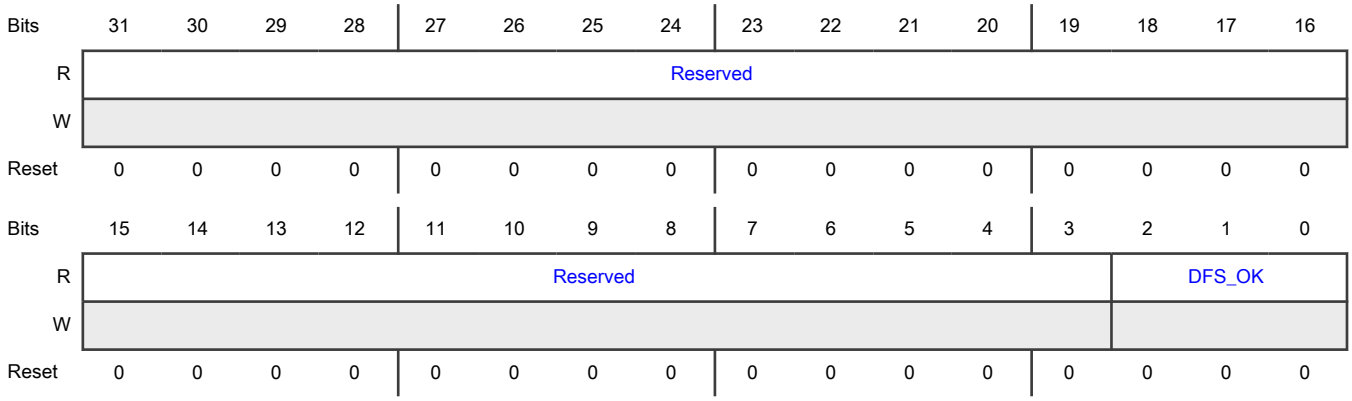
74.5.1.10 DFS Status (DFS_STATUS)

Indicates the status of clock out OK signals for DFS blocks.

Offset

Register	Offset
DFS_STATUS	F4h

Diagram



Fields

Field	Description
31-3 —	Reserved
2-0 DFS_OK	DFS OK Indicates the status of the DFS output clock. 000 - Invalid 001 - Valid

74.5.2 PLL register descriptions

74.5.2.1 PLL memory map

ARMPLL base address: 4448_1000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	PLL Control (CTRL)	32	RW	0000_0000
4	PLL Control (CTRL_SET)	32	RW	0000_0000
8	PLL Control (CTRL_CLR)	32	RW	0000_0000
C	PLL Control (CTRL_TOG)	32	RW	0000_0000
60	PLL Dividers (DIV)	32	RW	00C8_0000
64	PLL Dividers (DIV_SET)	32	RW	00C8_0000
68	PLL Dividers (DIV_CLR)	32	RW	00C8_0000
6C	PLL Dividers (DIV_TOG)	32	RW	00C8_0000
F0	PLL Status (PLL_STATUS)	32	R	0000_0000

74.5.2.2 PLL Control (CTRL)

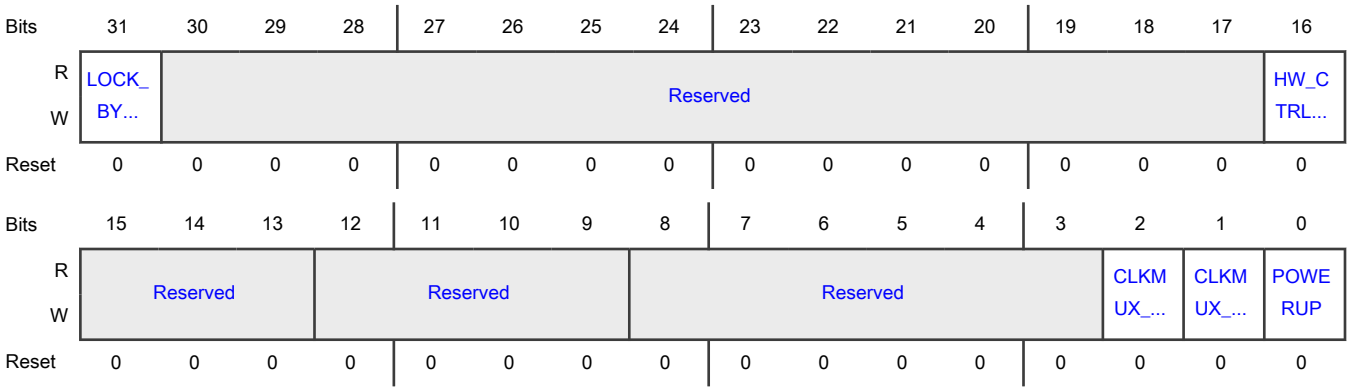
Enables PLL control and contains fractional PLL control registers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
CTRL	0h	PLL Control
CTRL_SET	4h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 1
CTRL_CLR	8h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 0
CTRL_TOG	Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CTRL

Diagram



Fields

Field	Description
31 LOCK_BYPASS	Lock Bypass Enables bypass for the lock detector. 0 - Disable 1 - Enable
30-17 —	Reserved
16 HW_CTRL_SEL	Hardware Control Select Controls PLL through hardware. If this field is disabled, PLL is controlled by register. If this field is enabled, PLL is controlled by hardware inputs and you cannot change dynamically. 0 - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable
15-13 —	Reserved
12-9 —	Reserved
8-3 —	Reserved
2 CLKMUX_BYPASS	CLKMUX_Bypass Bypasses PLL. The output clock is the PLL reference clock source. 0 - Normal mode 1 - PLL Bypass mode
1 CLKMUX_EN	CLKMUX Enable Enables output clock mux. 0 - Disable 1 - Enable
0 POWERUP	Power Up Enables PLL. 0 - Disable 1 - Enable

74.5.2.3 PLL Dividers (DIV)

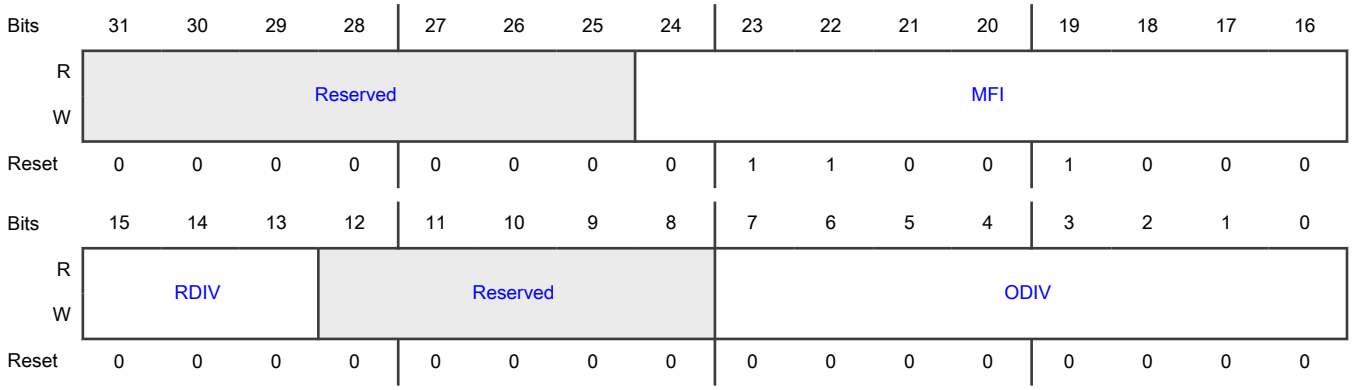
Controls PLL's digital and post dividers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DIV	60h	PLL Dividers
DIV_SET	64h	Writing 1 to a bit in this register ensures that the corresponding bit in DIV is 1
DIV_CLR	68h	Writing 1 to a bit in this register ensures that the corresponding bit in DIV is 0
DIV_TOG	6Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in DIV

Diagram



Fields

Field	Description
31-25 —	Reserved
24-16 MFI	Integer Portion of Loop Divider Sets the value of the divider in the PLL feedback loop. The value specified establishes the multiplication factor applied to the reference frequency. Divider value = MFI, where the chosen MFI does not violate VCO frequency specifications.
15-13 RDIV	Input Clock Predivider Sets the input clock divider. The output of the predivider circuit generates the PLL analog loop reference clock. 000 - Divide by 1 001 - Divide by 1 010 - Divide by 2 011 - Divide by 3 100 - Divide by 4 101 - Divide by 5 110 - Divide by 6 111 - Divide by 7
12-8 —	Reserved
7-0 ODIV	Output Frequency Divider for Clock Output Determines the VCO clock post divider for driving the final output clock. The value can be configured in the range of 0 to 255. If value is programmed as 0 or 1, output frequency will be F_{vco_clk}÷2. For all other values, output frequency will be as described in PLL output clock frequency.

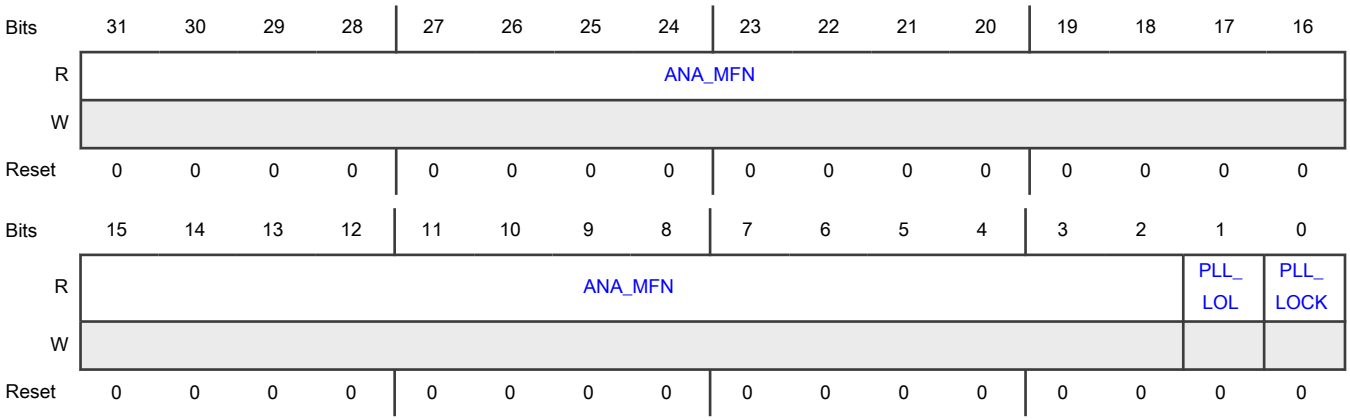
74.5.2.4 PLL Status (PLL_STATUS)

Indicates analog capture MNN values for debug purposes and PLL_LOCK status.

Offset

Register	Offset
PLL_STATUS	F0h

Diagram



Fields

Field	Description
31-2 ANA_MFN	ANA_MFN Stores analog block capture MFN value for debug purposes.
1 PLL_LOL	PLL_LOL Captures the loss of lock (LoL) signal when you write 0 to PLL_LOCK.
0 PLL_LOCK	PLL_LOCK Captures the lock signal to confirm whether the PLL clock is stable after you write 1 to PLL_LOCK.

74.5.3 PLL register descriptions

74.5.3.1 PLL memory map

AUDIOPLL base address: 4448_1200h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	PLL Control (CTRL)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	PLL Control (CTRL_SET)	32	RW	0000_0000
8	PLL Control (CTRL_CLR)	32	RW	0000_0000
C	PLL Control (CTRL_TOG)	32	RW	0000_0000
30	Spread Spectrum (SPREAD_SPECTRUM)	32	RW	0000_0000
34	Spread Spectrum (SPREAD_SPECTRUM_SET)	32	RW	0000_0000
38	Spread Spectrum (SPREAD_SPECTRUM_CLR)	32	RW	0000_0000
3C	Spread Spectrum (SPREAD_SPECTRUM_TOG)	32	RW	0000_0000
40	Numerator (NUMERATOR)	32	RW	0000_0000
44	Numerator (NUMERATOR_SET)	32	RW	0000_0000
48	Numerator (NUMERATOR_CLR)	32	RW	0000_0000
4C	Numerator (NUMERATOR_TOG)	32	RW	0000_0000
50	Denominator (DENOMINATOR)	32	RW	0000_0001
54	Denominator (DENOMINATOR_SET)	32	RW	0000_0001
58	Denominator (DENOMINATOR_CLR)	32	RW	0000_0001
5C	Denominator (DENOMINATOR_TOG)	32	RW	0000_0001
60	PLL Dividers (DIV)	32	RW	00C8_0000
64	PLL Dividers (DIV_SET)	32	RW	00C8_0000
68	PLL Dividers (DIV_CLR)	32	RW	00C8_0000
6C	PLL Dividers (DIV_TOG)	32	RW	00C8_0000
F0	PLL Status (PLL_STATUS)	32	R	0000_0000

74.5.3.2 PLL Control (CTRL)

Enables PLL control and contains fractional PLL control registers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

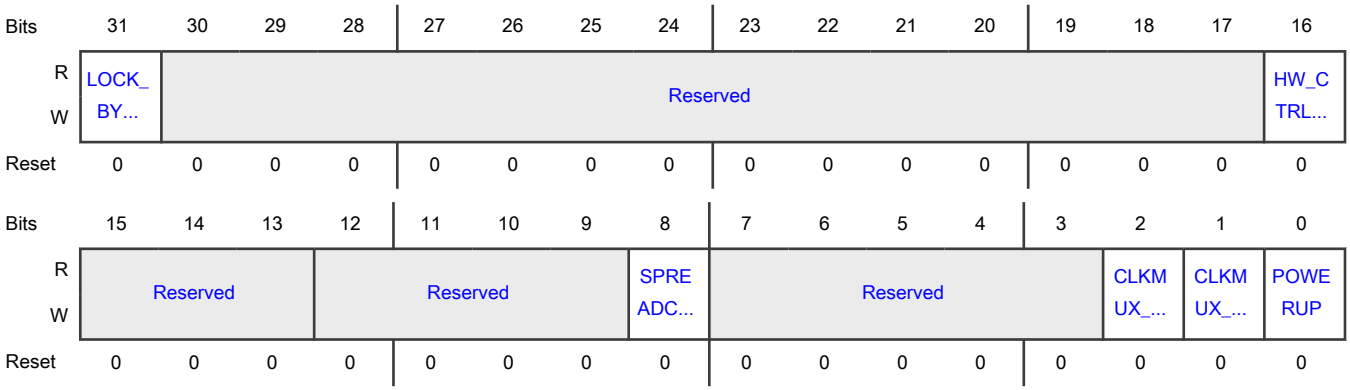
Register	Offset	Description
CTRL	0h	PLL Control
CTRL_SET	4h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 1

Table continues on the next page...

Table continued from the previous page...

Register	Offset	Description
CTRL_CLR	8h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 0
CTRL_TOG	Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CTRL

Diagram



Fields

Field	Description
31 LOCK_BYPASS	Lock Bypass Enables bypass for the lock detector. 0 - Disable 1 - Enable
30-17 —	Reserved
16 HW_CTRL_SEL	Hardware Control Select Controls PLL through hardware. If this field is disabled, PLL is controlled by register. If this field is enabled, PLL is controlled by hardware inputs and you cannot change NUMERATOR[MFN] dynamically. 0 - Disable 1 - Enable
15-13 —	Reserved
12-9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 SPREADCTL	Modulation Type Select Selects whether modulation is centered around, or spread below the nominal frequency. 0 - Centered around nominal frequency 1 - Spread below nominal frequency
7-3 —	Reserved
2 CLKMUX_BYPASS	CLKMUX_Bypass Bypasses PLL. The output clock is the PLL reference clock source. 0 - Normal mode 1 - PLL Bypass mode
1 CLKMUX_EN	CLKMUX Enable Enables output clock mux. 0 - Disable 1 - Enable
0 POWERUP	Power Up Enables PLL. 0 - Disable 1 - Enable

74.5.3.3 Spread Spectrum (SPREAD_SPECTRUM)

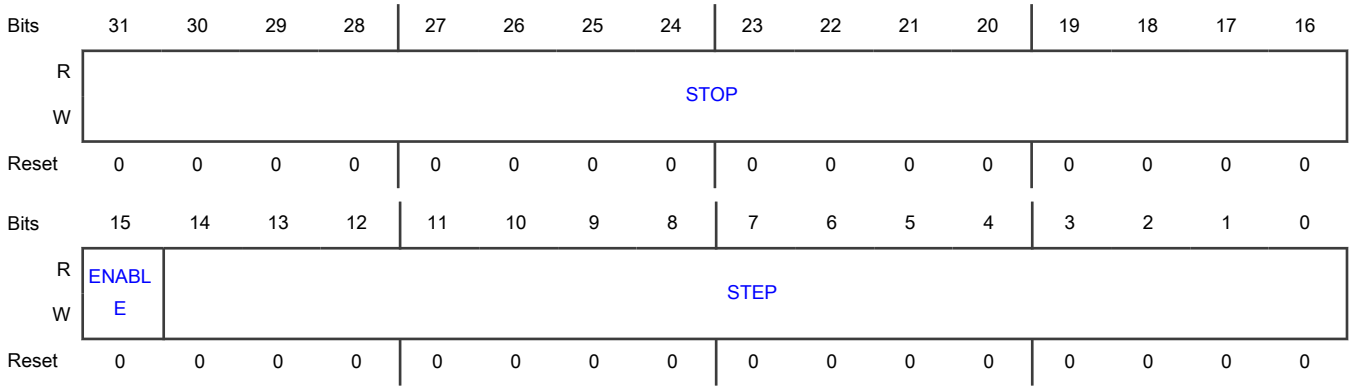
Sets spread spectrum control.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
SPREAD_SPECTRUM	30h	Spread Spectrum
SPREAD_SPECTRUM_SET	34h	Writing 1 to a bit in this register ensures that the corresponding bit in SPREAD_SPECTRUM is 1
SPREAD_SPECTRUM_CLR	38h	Writing 1 to a bit in this register ensures that the corresponding bit in SPREAD_SPECTRUM is 0
SPREAD_SPECTRUM_TOG	3Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in SPREAD_SPECTRUM

Diagram



Fields

Field	Description
31-16 STOP	Stop Sets spread spectrum stop.
15 ENABLE	Enable Enables or disables the spread spectrum modulation. 0 - Disable 1 - Enable
14-0 STEP	Step Sets spread spectrum step.

74.5.3.4 Numerator (NUMERATOR)

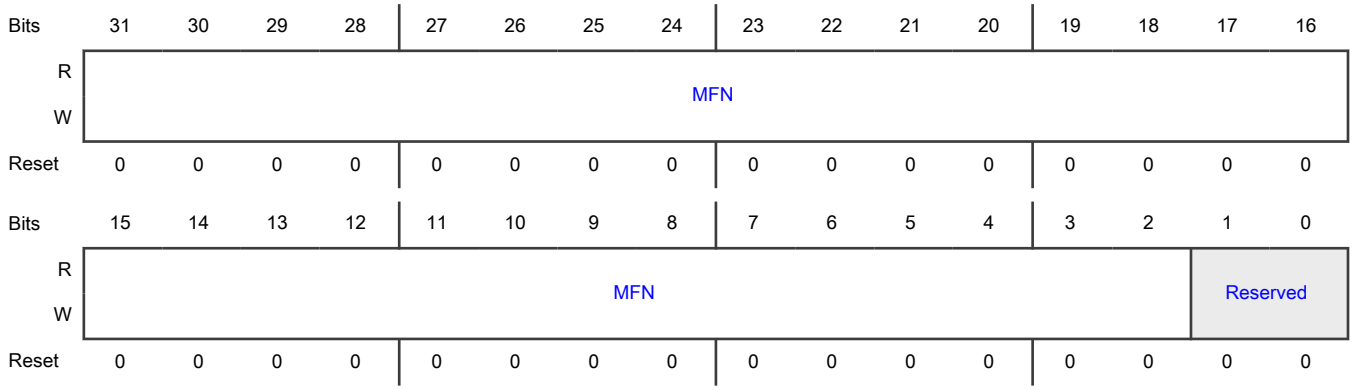
Sets numerator control.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
NUMERATOR	40h	Numerator
NUMERATOR_SET	44h	Writing 1 to a bit in this register ensures that the corresponding bit in NUMERATOR is 1
NUMERATOR_CLR	48h	Writing 1 to a bit in this register ensures that the corresponding bit in NUMERATOR is 0
NUMERATOR_TOG	4Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in NUMERATOR

Diagram



Fields

Field	Description
31-2 MFN	Numerator Sets the numerator of the fractional PLL loop divider (signed number). You can configure this field dynamically when PLL is enabled, except when CTRL[HW_CTRL_SEL] is 1.
1-0 —	Reserved

74.5.3.5 Denominator (DENOMINATOR)

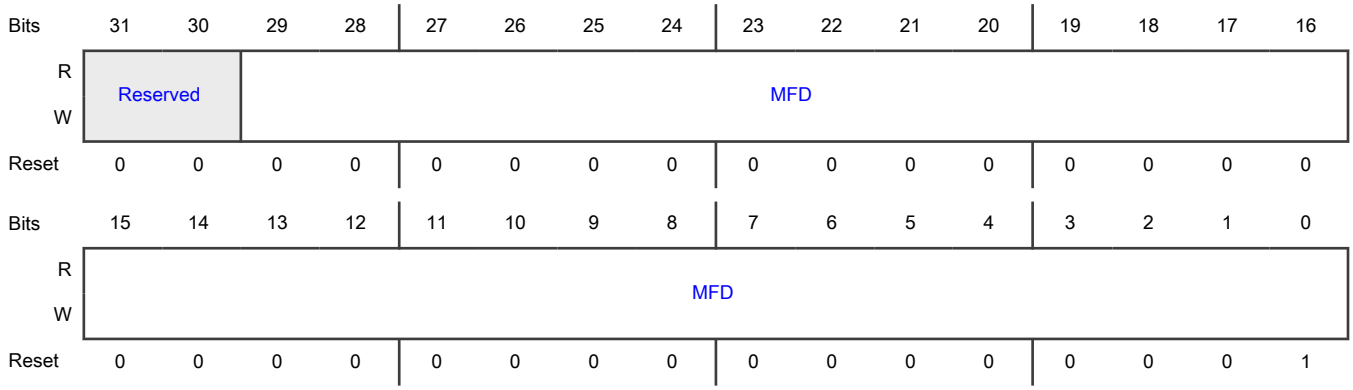
Sets denominator control.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DENOMINATOR	50h	Denominator
DENOMINATOR_SET	54h	Writing 1 to a bit in this register ensures that the corresponding bit in DENOMINATOR is 1
DENOMINATOR_CLR	58h	Writing 1 to a bit in this register ensures that the corresponding bit in DENOMINATOR is 0
DENOMINATOR_TOG	5Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in DENOMINATOR

Diagram



Fields

Field	Description
31-30 —	Reserved
29-0 MFD	Denominator Sets the denominator of the fractional loop divider (unsigned number). **NOTE** You are not allowed to write 0 to this field.

74.5.3.6 PLL Dividers (DIV)

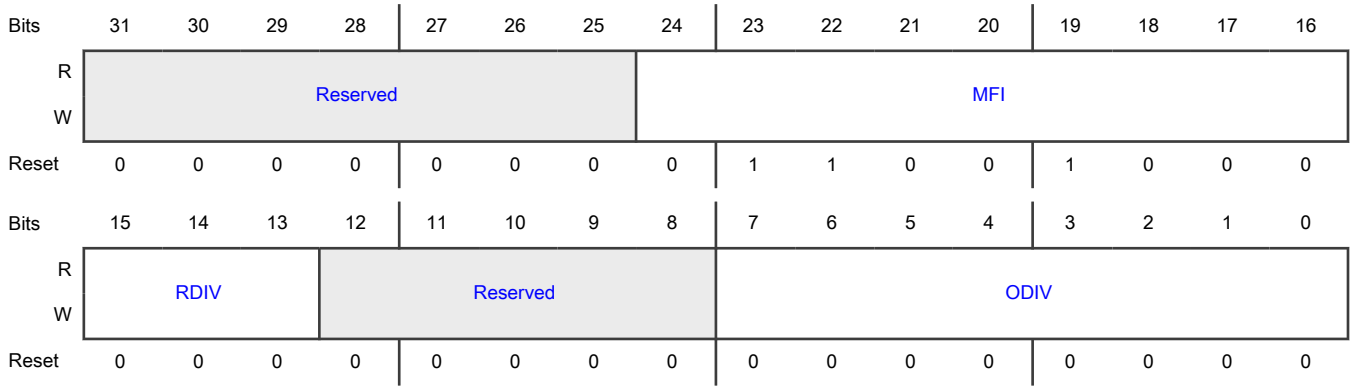
Controls PLL's digital and post dividers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DIV	60h	PLL Dividers
DIV_SET	64h	Writing 1 to a bit in this register ensures that the corresponding bit in DIV is 1
DIV_CLR	68h	Writing 1 to a bit in this register ensures that the corresponding bit in DIV is 0
DIV_TOG	6Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in DIV

Diagram



Fields

Field	Description
31-25 —	Reserved
24-16 MFI	Integer Portion of Loop Divider Sets the value of the divider in the PLL feedback loop. The value specified establishes the multiplication factor applied to the reference frequency. Divider value = MFI, where the chosen MFI does not violate VCO frequency specifications.
15-13 RDIV	Input Clock Predivider Sets the input clock divider. The output of the predivider circuit generates the PLL analog loop reference clock. 000 - Divide by 1 001 - Divide by 1 010 - Divide by 2 011 - Divide by 3 100 - Divide by 4 101 - Divide by 5 110 - Divide by 6 111 - Divide by 7
12-8 —	Reserved
7-0 ODIV	Output Frequency Divider for Clock Output Determines the VCO clock post divider for driving the final output clock. The value can be configured in the range of 0 to 255. If value is programmed as 0 or 1, output frequency will be F_{vco_clk}÷2. For all other values, output frequency will be as described in PLL output clock frequency.

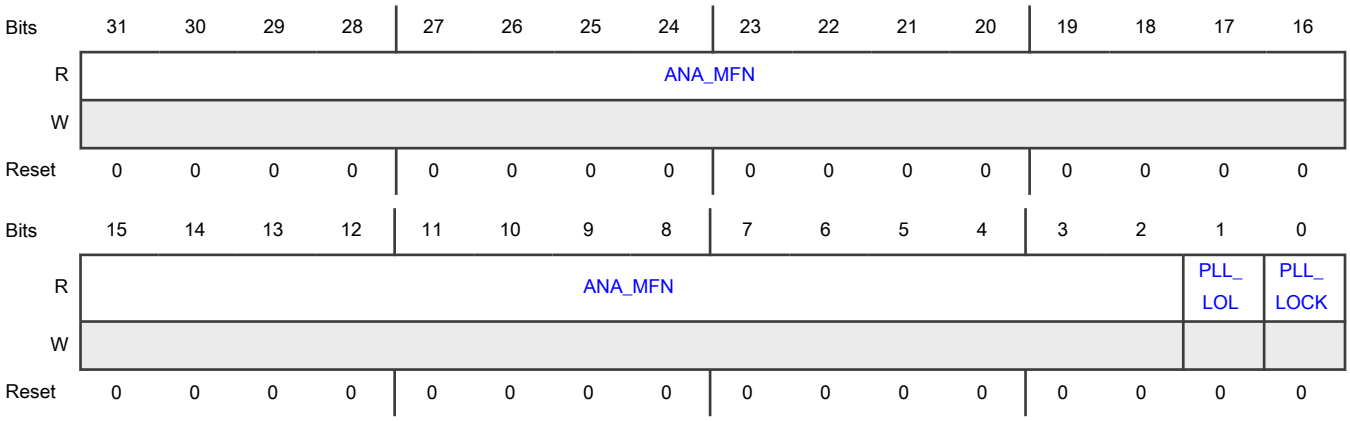
74.5.3.7 PLL Status (PLL_STATUS)

Indicates analog capture MNN values for debug purposes and PLL_LOCK status.

Offset

Register	Offset
PLL_STATUS	F0h

Diagram



Fields

Field	Description
31-2 ANA_MFN	ANA_MFN Stores analog block capture MFN value for debug purposes.
1 PLL_LOL	PLL_LOL Captures the loss of lock (LoL) signal when you write 0 to PLL_LOCK.
0 PLL_LOCK	PLL_LOCK Captures the lock signal to confirm whether the PLL clock is stable after you write 1 to PLL_LOCK.

74.5.4 PLL register descriptions

74.5.4.1 PLL memory map

DRAMPLL base address: 4448_1300h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	PLL Control (CTRL)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	PLL Control (CTRL_SET)	32	RW	0000_0000
8	PLL Control (CTRL_CLR)	32	RW	0000_0000
C	PLL Control (CTRL_TOG)	32	RW	0000_0000
30	Spread Spectrum (SPREAD_SPECTRUM)	32	RW	0000_0000
34	Spread Spectrum (SPREAD_SPECTRUM_SET)	32	RW	0000_0000
38	Spread Spectrum (SPREAD_SPECTRUM_CLR)	32	RW	0000_0000
3C	Spread Spectrum (SPREAD_SPECTRUM_TOG)	32	RW	0000_0000
40	Numerator (NUMERATOR)	32	RW	0000_0000
44	Numerator (NUMERATOR_SET)	32	RW	0000_0000
48	Numerator (NUMERATOR_CLR)	32	RW	0000_0000
4C	Numerator (NUMERATOR_TOG)	32	RW	0000_0000
50	Denominator (DENOMINATOR)	32	RW	0000_0001
54	Denominator (DENOMINATOR_SET)	32	RW	0000_0001
58	Denominator (DENOMINATOR_CLR)	32	RW	0000_0001
5C	Denominator (DENOMINATOR_TOG)	32	RW	0000_0001
60	PLL Dividers (DIV)	32	RW	00C8_0000
64	PLL Dividers (DIV_SET)	32	RW	00C8_0000
68	PLL Dividers (DIV_CLR)	32	RW	00C8_0000
6C	PLL Dividers (DIV_TOG)	32	RW	00C8_0000
F0	PLL Status (PLL_STATUS)	32	R	0000_0000

74.5.4.2 PLL Control (CTRL)

Enables PLL control and contains fractional PLL control registers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

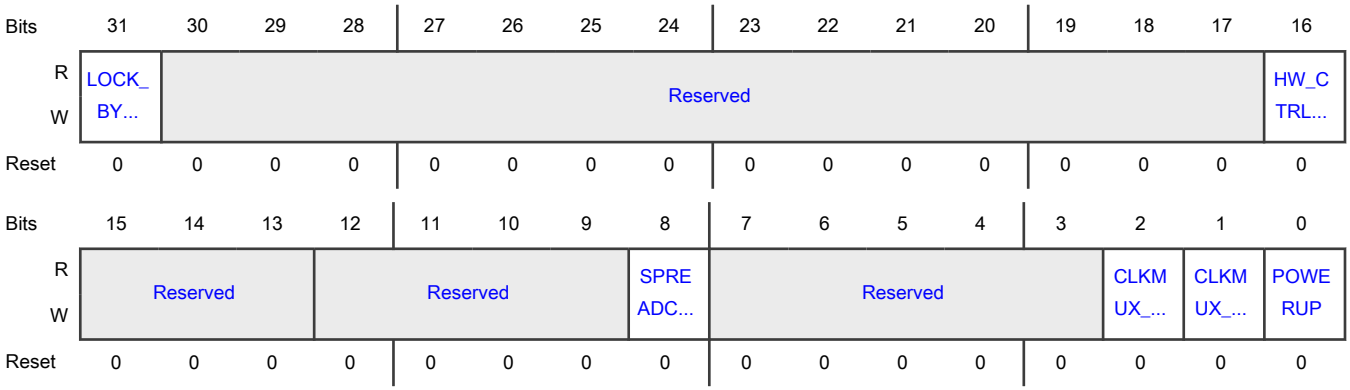
Register	Offset	Description
CTRL	0h	PLL Control
CTRL_SET	4h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 1

Table continues on the next page...

Table continued from the previous page...

Register	Offset	Description
CTRL_CLR	8h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 0
CTRL_TOG	Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CTRL

Diagram



Fields

Field	Description
31 LOCK_BYPASS	Lock Bypass Enables bypass for the lock detector. 0 - Disable 1 - Enable
30-17 —	Reserved
16 HW_CTRL_SEL	Hardware Control Select Controls PLL through hardware. If this field is disabled, PLL is controlled by register. If this field is enabled, PLL is controlled by hardware inputs and you cannot change NUMERATOR[MFN] dynamically. 0 - Disable 1 - Enable
15-13 —	Reserved
12-9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 SPREADCTL	Modulation Type Select Selects whether modulation is centered around, or spread below the nominal frequency. 0 - Centered around nominal frequency 1 - Spread below nominal frequency
7-3 —	Reserved
2 CLKMUX_BYPASS	CLKMUX_Bypass Bypasses PLL. The output clock is the PLL reference clock source. 0 - Normal mode 1 - PLL Bypass mode
1 CLKMUX_EN	CLKMUX Enable Enables output clock mux. 0 - Disable 1 - Enable
0 POWERUP	Power Up Enables PLL. 0 - Disable 1 - Enable

74.5.4.3 Spread Spectrum (SPREAD_SPECTRUM)

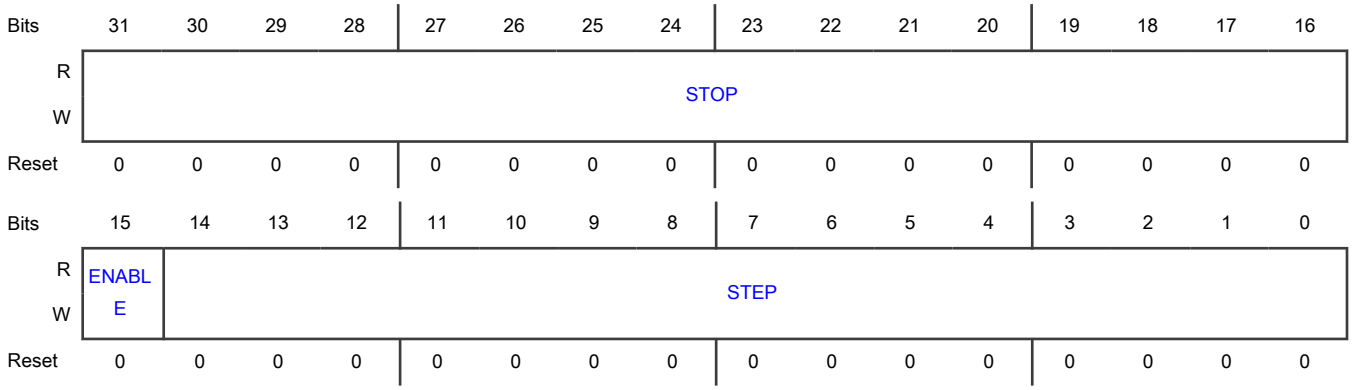
Sets spread spectrum control.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
SPREAD_SPECTRUM	30h	Spread Spectrum
SPREAD_SPECTRUM_SET	34h	Writing 1 to a bit in this register ensures that the corresponding bit in SPREAD_SPECTRUM is 1
SPREAD_SPECTRUM_CLR	38h	Writing 1 to a bit in this register ensures that the corresponding bit in SPREAD_SPECTRUM is 0
SPREAD_SPECTRUM_TOG	3Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in SPREAD_SPECTRUM

Diagram



Fields

Field	Description
31-16 STOP	Stop Sets spread spectrum stop.
15 ENABLE	Enable Enables or disables the spread spectrum modulation. 0 - Disable 1 - Enable
14-0 STEP	Step Sets spread spectrum step.

74.5.4.4 Numerator (NUMERATOR)

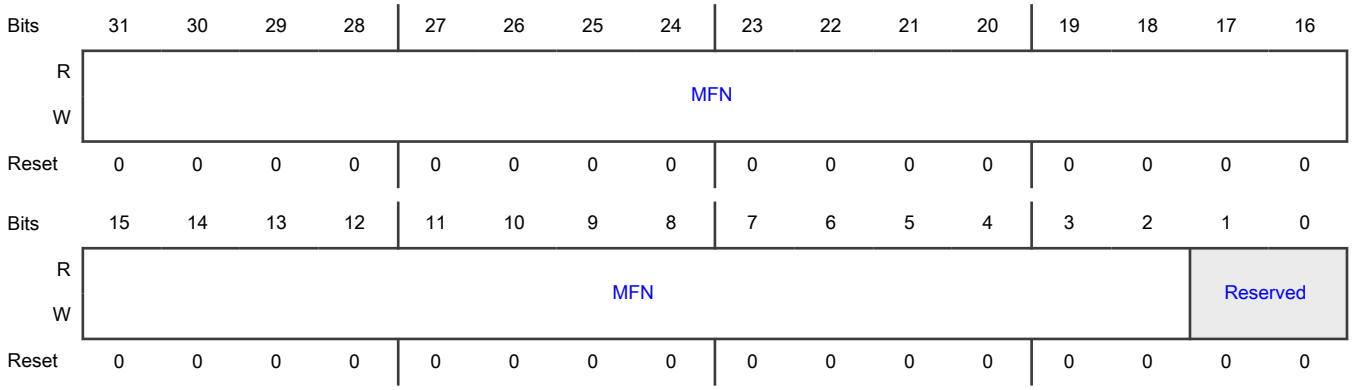
Sets numerator control.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
NUMERATOR	40h	Numerator
NUMERATOR_SET	44h	Writing 1 to a bit in this register ensures that the corresponding bit in NUMERATOR is 1
NUMERATOR_CLR	48h	Writing 1 to a bit in this register ensures that the corresponding bit in NUMERATOR is 0
NUMERATOR_TOG	4Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in NUMERATOR

Diagram



Fields

Field	Description
31-2 MFN	Numerator Sets the numerator of the fractional PLL loop divider (signed number). You can configure this field dynamically when PLL is enabled, except when CTRL[HW_CTRL_SEL] is 1.
1-0 —	Reserved

74.5.4.5 Denominator (DENOMINATOR)

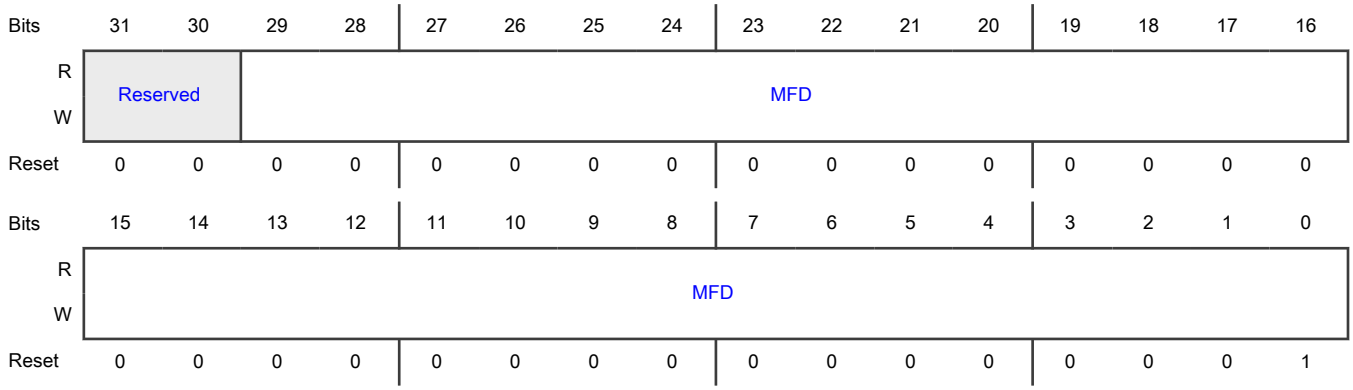
Sets denominator control.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DENOMINATOR	50h	Denominator
DENOMINATOR_SET	54h	Writing 1 to a bit in this register ensures that the corresponding bit in DENOMINATOR is 1
DENOMINATOR_CLR	58h	Writing 1 to a bit in this register ensures that the corresponding bit in DENOMINATOR is 0
DENOMINATOR_TOG	5Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in DENOMINATOR

Diagram



Fields

Field	Description
31-30 —	Reserved
29-0 MFD	Denominator Sets the denominator of the fractional loop divider (unsigned number). **NOTE** You are not allowed to write 0 to this field.

74.5.4.6 PLL Dividers (DIV)

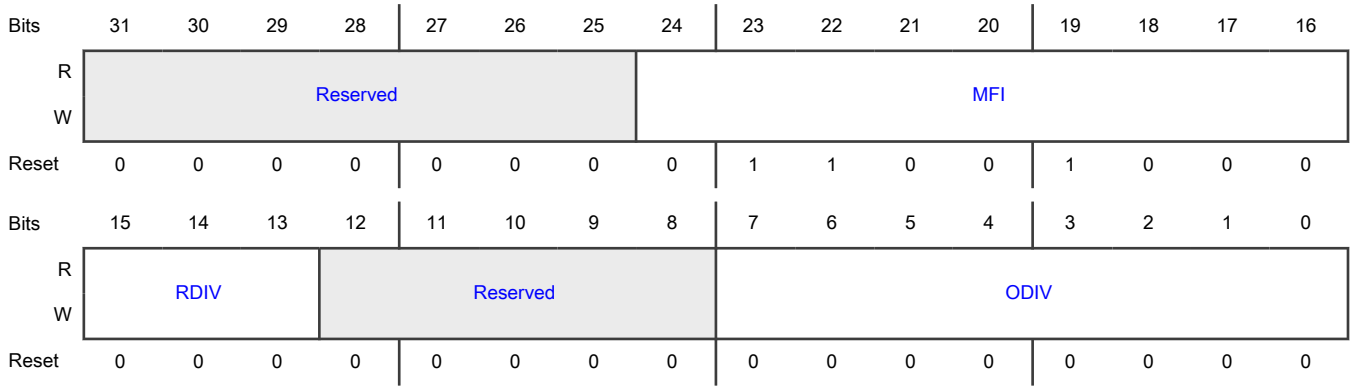
Controls PLL's digital and post dividers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DIV	60h	PLL Dividers
DIV_SET	64h	Writing 1 to a bit in this register ensures that the corresponding bit in DIV is 1
DIV_CLR	68h	Writing 1 to a bit in this register ensures that the corresponding bit in DIV is 0
DIV_TOG	6Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in DIV

Diagram



Fields

Field	Description
31-25 —	Reserved
24-16 MFI	Integer Portion of Loop Divider Sets the value of the divider in the PLL feedback loop. The value specified establishes the multiplication factor applied to the reference frequency. Divider value = MFI, where the chosen MFI does not violate VCO frequency specifications.
15-13 RDIV	Input Clock Predivider Sets the input clock divider. The output of the predivider circuit generates the PLL analog loop reference clock. 000 - Divide by 1 001 - Divide by 1 010 - Divide by 2 011 - Divide by 3 100 - Divide by 4 101 - Divide by 5 110 - Divide by 6 111 - Divide by 7
12-8 —	Reserved
7-0 ODIV	Output Frequency Divider for Clock Output Determines the VCO clock post divider for driving the final output clock. The value can be configured in the range of 0 to 255. If value is programmed as 0 or 1, output frequency will be F_{vco_clk}÷2. For all other values, output frequency will be as described in PLL output clock frequency.

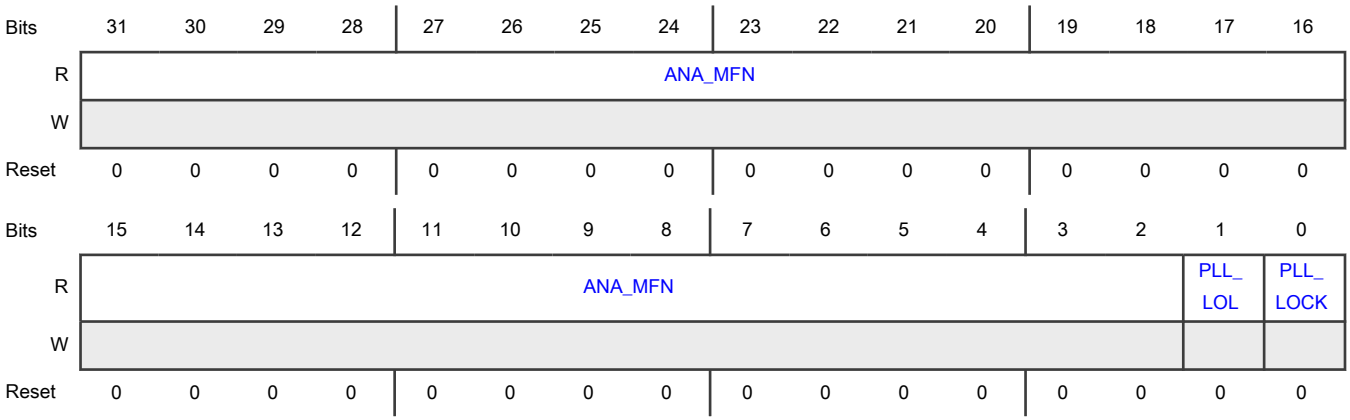
74.5.4.7 PLL Status (PLL_STATUS)

Indicates analog capture MNN values for debug purposes and PLL_LOCK status.

Offset

Register	Offset
PLL_STATUS	F0h

Diagram



Fields

Field	Description
31-2 ANA_MFN	ANA_MFN Stores analog block capture MFN value for debug purposes.
1 PLL_LOL	PLL_LOL Captures the loss of lock (LoL) signal when you write 0 to PLL_LOCK.
0 PLL_LOCK	PLL_LOCK Captures the lock signal to confirm whether the PLL clock is stable after you write 1 to PLL_LOCK.

74.5.5 PLL register descriptions

74.5.5.1 PLL memory map

VIDEOPLL base address: 4448_1400h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	PLL Control (CTRL)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	PLL Control (CTRL_SET)	32	RW	0000_0000
8	PLL Control (CTRL_CLR)	32	RW	0000_0000
C	PLL Control (CTRL_TOG)	32	RW	0000_0000
30	Spread Spectrum (SPREAD_SPECTRUM)	32	RW	0000_0000
34	Spread Spectrum (SPREAD_SPECTRUM_SET)	32	RW	0000_0000
38	Spread Spectrum (SPREAD_SPECTRUM_CLR)	32	RW	0000_0000
3C	Spread Spectrum (SPREAD_SPECTRUM_TOG)	32	RW	0000_0000
40	Numerator (NUMERATOR)	32	RW	0000_0000
44	Numerator (NUMERATOR_SET)	32	RW	0000_0000
48	Numerator (NUMERATOR_CLR)	32	RW	0000_0000
4C	Numerator (NUMERATOR_TOG)	32	RW	0000_0000
50	Denominator (DENOMINATOR)	32	RW	0000_0001
54	Denominator (DENOMINATOR_SET)	32	RW	0000_0001
58	Denominator (DENOMINATOR_CLR)	32	RW	0000_0001
5C	Denominator (DENOMINATOR_TOG)	32	RW	0000_0001
60	PLL Dividers (DIV)	32	RW	00C8_0000
64	PLL Dividers (DIV_SET)	32	RW	00C8_0000
68	PLL Dividers (DIV_CLR)	32	RW	00C8_0000
6C	PLL Dividers (DIV_TOG)	32	RW	00C8_0000
F0	PLL Status (PLL_STATUS)	32	R	0000_0000

74.5.5.2 PLL Control (CTRL)

Enables PLL control and contains fractional PLL control registers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

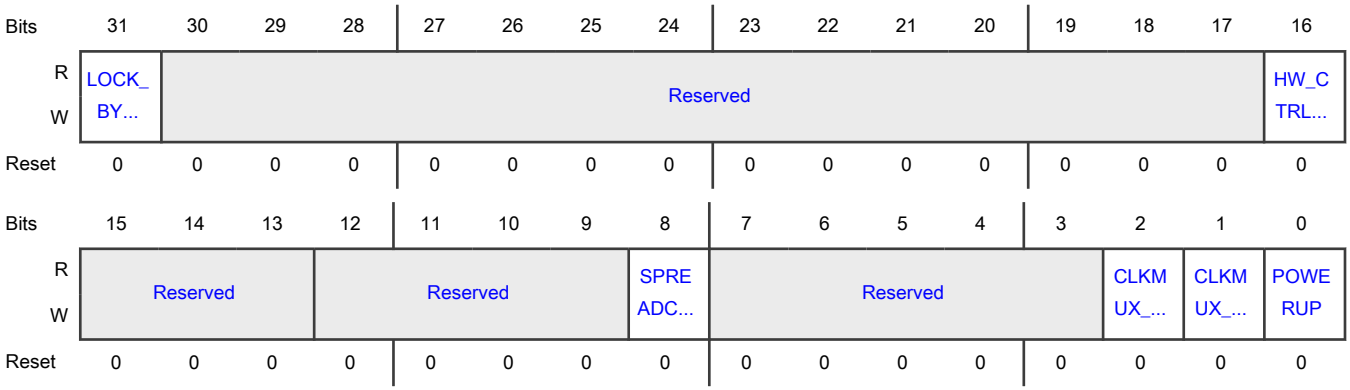
Register	Offset	Description
CTRL	0h	PLL Control
CTRL_SET	4h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 1

Table continues on the next page...

Table continued from the previous page...

Register	Offset	Description
CTRL_CLR	8h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 0
CTRL_TOG	Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CTRL

Diagram



Fields

Field	Description
31 LOCK_BYPASS	Lock Bypass Enables bypass for the lock detector. 0 - Disable 1 - Enable
30-17 —	Reserved
16 HW_CTRL_SEL	Hardware Control Select Controls PLL through hardware. If this field is disabled, PLL is controlled by register. If this field is enabled, PLL is controlled by hardware inputs and you cannot change NUMERATOR[MFN] dynamically. 0 - Disable 1 - Enable
15-13 —	Reserved
12-9 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
8 SPREADCTL	Modulation Type Select Selects whether modulation is centered around, or spread below the nominal frequency. 0 - Centered around nominal frequency 1 - Spread below nominal frequency
7-3 —	Reserved
2 CLKMUX_BYPASS	CLKMUX_Bypass Bypasses PLL. The output clock is the PLL reference clock source. 0 - Normal mode 1 - PLL Bypass mode
1 CLKMUX_EN	CLKMUX Enable Enables output clock mux. 0 - Disable 1 - Enable
0 POWERUP	Power Up Enables PLL. 0 - Disable 1 - Enable

74.5.5.3 Spread Spectrum (SPREAD_SPECTRUM)

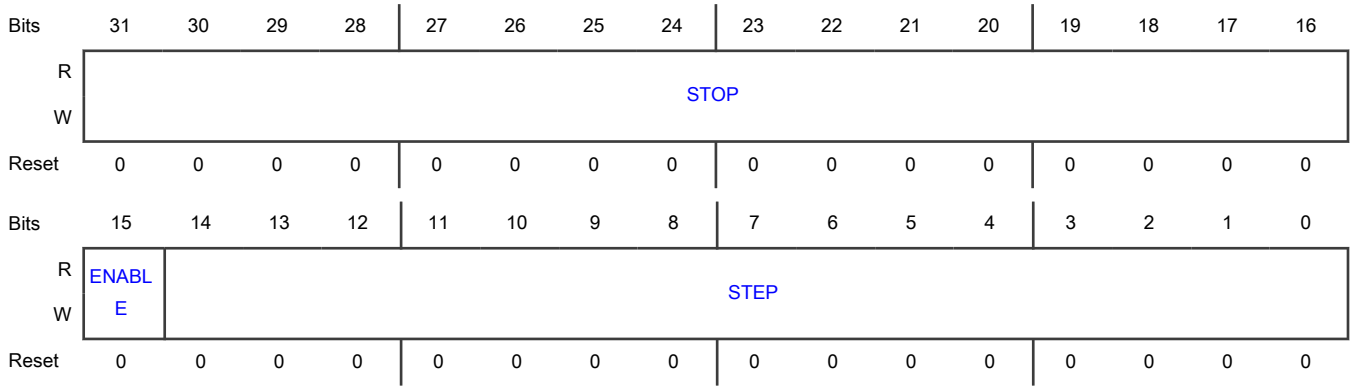
Sets spread spectrum control.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
SPREAD_SPECTRUM	30h	Spread Spectrum
SPREAD_SPECTRUM_SET	34h	Writing 1 to a bit in this register ensures that the corresponding bit in SPREAD_SPECTRUM is 1
SPREAD_SPECTRUM_CLR	38h	Writing 1 to a bit in this register ensures that the corresponding bit in SPREAD_SPECTRUM is 0
SPREAD_SPECTRUM_TOG	3Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in SPREAD_SPECTRUM

Diagram



Fields

Field	Description
31-16 STOP	Stop Sets spread spectrum stop.
15 ENABLE	Enable Enables or disables the spread spectrum modulation. 0 - Disable 1 - Enable
14-0 STEP	Step Sets spread spectrum step.

74.5.5.4 Numerator (NUMERATOR)

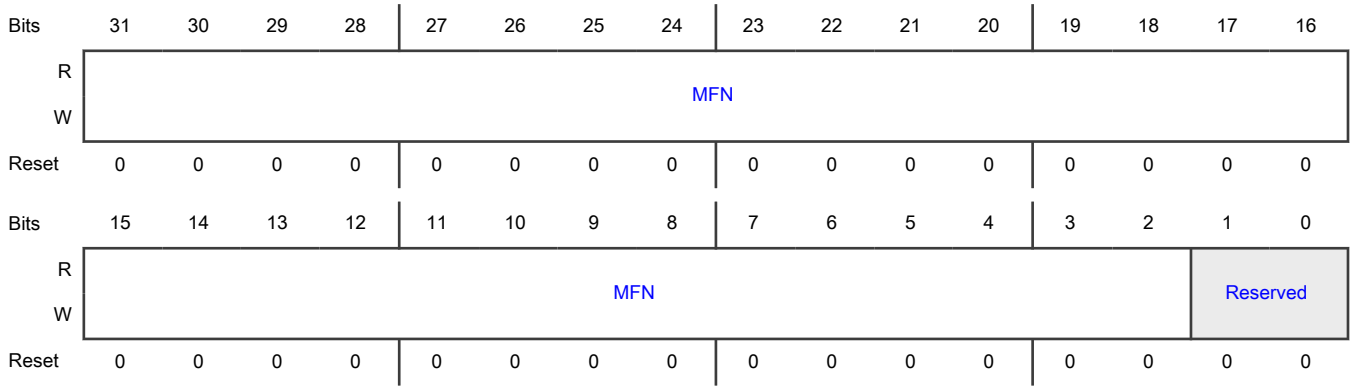
Sets numerator control.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
NUMERATOR	40h	Numerator
NUMERATOR_SET	44h	Writing 1 to a bit in this register ensures that the corresponding bit in NUMERATOR is 1
NUMERATOR_CLR	48h	Writing 1 to a bit in this register ensures that the corresponding bit in NUMERATOR is 0
NUMERATOR_TOG	4Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in NUMERATOR

Diagram



Fields

Field	Description
31-2 MFN	Numerator Sets the numerator of the fractional PLL loop divider (signed number). You can configure this field dynamically when PLL is enabled, except when CTRL[HW_CTRL_SEL] is 1.
1-0 —	Reserved

74.5.5.5 Denominator (DENOMINATOR)

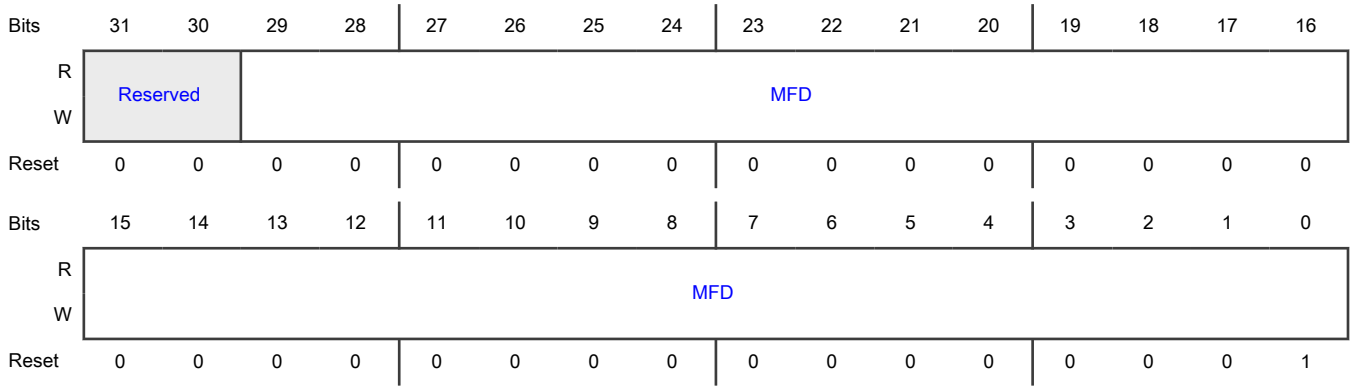
Sets denominator control.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DENOMINATOR	50h	Denominator
DENOMINATOR_SET	54h	Writing 1 to a bit in this register ensures that the corresponding bit in DENOMINATOR is 1
DENOMINATOR_CLR	58h	Writing 1 to a bit in this register ensures that the corresponding bit in DENOMINATOR is 0
DENOMINATOR_TOG	5Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in DENOMINATOR

Diagram



Fields

Field	Description
31-30 —	Reserved
29-0 MFD	Denominator Sets the denominator of the fractional loop divider (unsigned number). **NOTE** You are not allowed to write 0 to this field.

74.5.5.6 PLL Dividers (DIV)

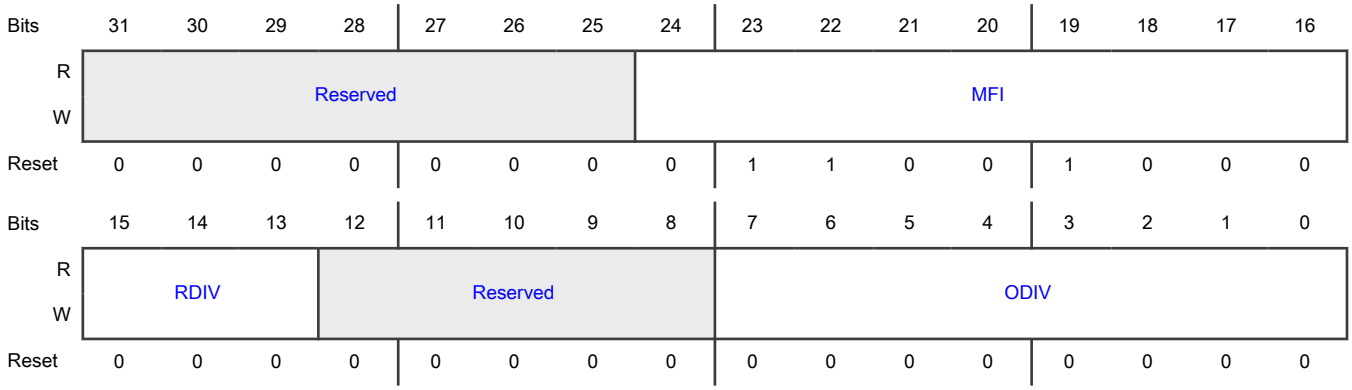
Controls PLL's digital and post dividers.

Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
DIV	60h	PLL Dividers
DIV_SET	64h	Writing 1 to a bit in this register ensures that the corresponding bit in DIV is 1
DIV_CLR	68h	Writing 1 to a bit in this register ensures that the corresponding bit in DIV is 0
DIV_TOG	6Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in DIV

Diagram



Fields

Field	Description
31-25 —	Reserved
24-16 MFI	Integer Portion of Loop Divider Sets the value of the divider in the PLL feedback loop. The value specified establishes the multiplication factor applied to the reference frequency. Divider value = MFI, where the chosen MFI does not violate VCO frequency specifications.
15-13 RDIV	Input Clock Predivider Sets the input clock divider. The output of the predivider circuit generates the PLL analog loop reference clock. 000 - Divide by 1 001 - Divide by 1 010 - Divide by 2 011 - Divide by 3 100 - Divide by 4 101 - Divide by 5 110 - Divide by 6 111 - Divide by 7
12-8 —	Reserved
7-0 ODIV	Output Frequency Divider for Clock Output Determines the VCO clock post divider for driving the final output clock. The value can be configured in the range of 0 to 255. If value is programmed as 0 or 1, output frequency will be F_{vco_clk}÷2. For all other values, output frequency will be as described in PLL output clock frequency.

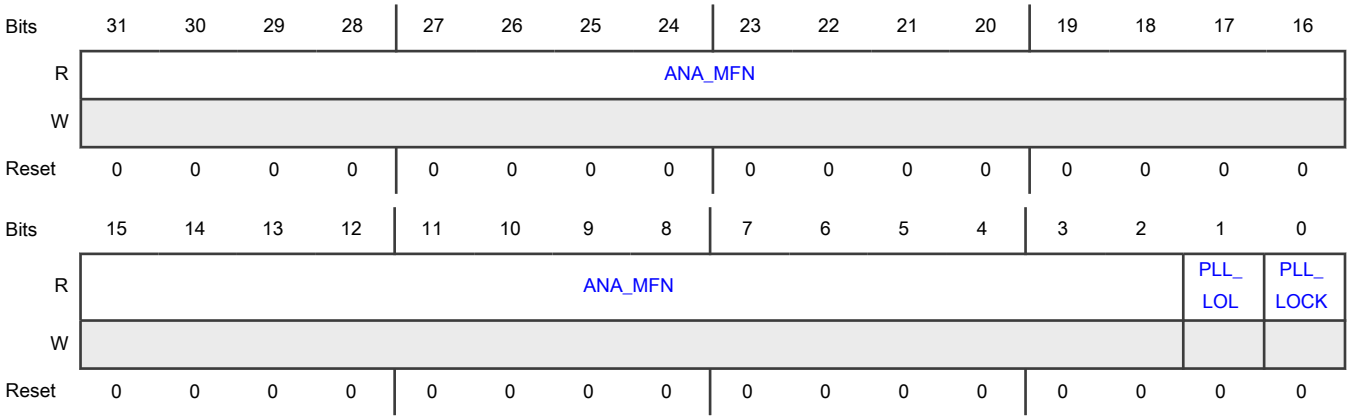
74.5.5.7 PLL Status (PLL_STATUS)

Indicates analog capture MNN values for debug purposes and PLL_LOCK status.

Offset

Register	Offset
PLL_STATUS	F0h

Diagram



Fields

Field	Description
31-2 ANA_MFN	ANA_MFN Stores analog block capture MFN value for debug purposes.
1 PLL_LOL	PLL_LOL Captures the loss of lock (LoL) signal when you write 0 to PLL_LOCK.
0 PLL_LOCK	PLL_LOCK Captures the lock signal to confirm whether the PLL clock is stable after you write 1 to PLL_LOCK.

Chapter 75
OSC

75.1 Overview

The OSC module includes the DIGPROG Device ID register for the chip.

75.2 Memory map and register definition

The OSC Memory map includes the DIGPROG Device ID register for the chip. This register allows software to identify the silicon version it is running on.

75.2.1 OSC register descriptions

75.2.1.1 OSC_REGS memory map

ANA_OSC base address: 4448_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
800	Device ID (DIGPROG_DEVICE_ID)	32	R	0090_2010

75.2.1.2 Device ID (DIGPROG_DEVICE_ID)

This allows software to identify the silicon version.

Offset

Register	Offset
DIGPROG_DEVICE_ID	800h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								DIGPROG_MAJOR_UPPER							
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DIGPROG_MAJOR_LOWER								DIGPROG_MINOR							
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0

Fields

Field	Description
31-24 —	Reserved
23-16 DIGPROG_MAJOR_UPPER	DIGPROG_MAJOR_UPPER Bit[7:4] is 0x9. Bit[3:0] is 0x0, Reserved.
15-8 DIGPROG_MAJOR_LOWER	DIGPROG_MAJOR_LOWER Bit[7:4] is the number of Cortex-A cores physically implemented on the chip. Bit[3:0] is 0x20.
7-0 DIGPROG_MINOR	Bit[3:0] is the metal layer revision. Bit[7:4] is the base layer revision. DIGPROG_MINOR[3:0] is programmable by VIA0 change.

Chapter 76
Always-on Domain, Non-secure Block Control (blk_ctrl_ns_aonmix)

76.1 Introduction

The blk_ctrl_ns_aonmix contains control and status registers in the AONMIX domain, that can be accessed by both secure and non-secure processors. These registers include core sleep control, ipg debug mask, upper address extention, OCOTP fuse data and control, and some control bits for modules in the AONMIX domain.

76.1.1 Features

Block Control Non-Secure AON Domain has the following features:

- Sleeping or deep sleeping selection for low power mode entry
- Debug halt mask bits for peripherals in AON Domain
- AON Domain SSI master block low power bits
- SAI1 MCLK direction control
- MQS settings
- OCOTP fuse register
- OCOTP Status register
- PDM clock select
- I3C1 asynchronous interrupt enable
- Fastboot enable bits
- I3C1 IO on chip strong pull enable bit
- QCHANNEL disable bits for modules.

76.1.2 Block diagram

A block diagram of the NS_AONMIX implementation is shown below.

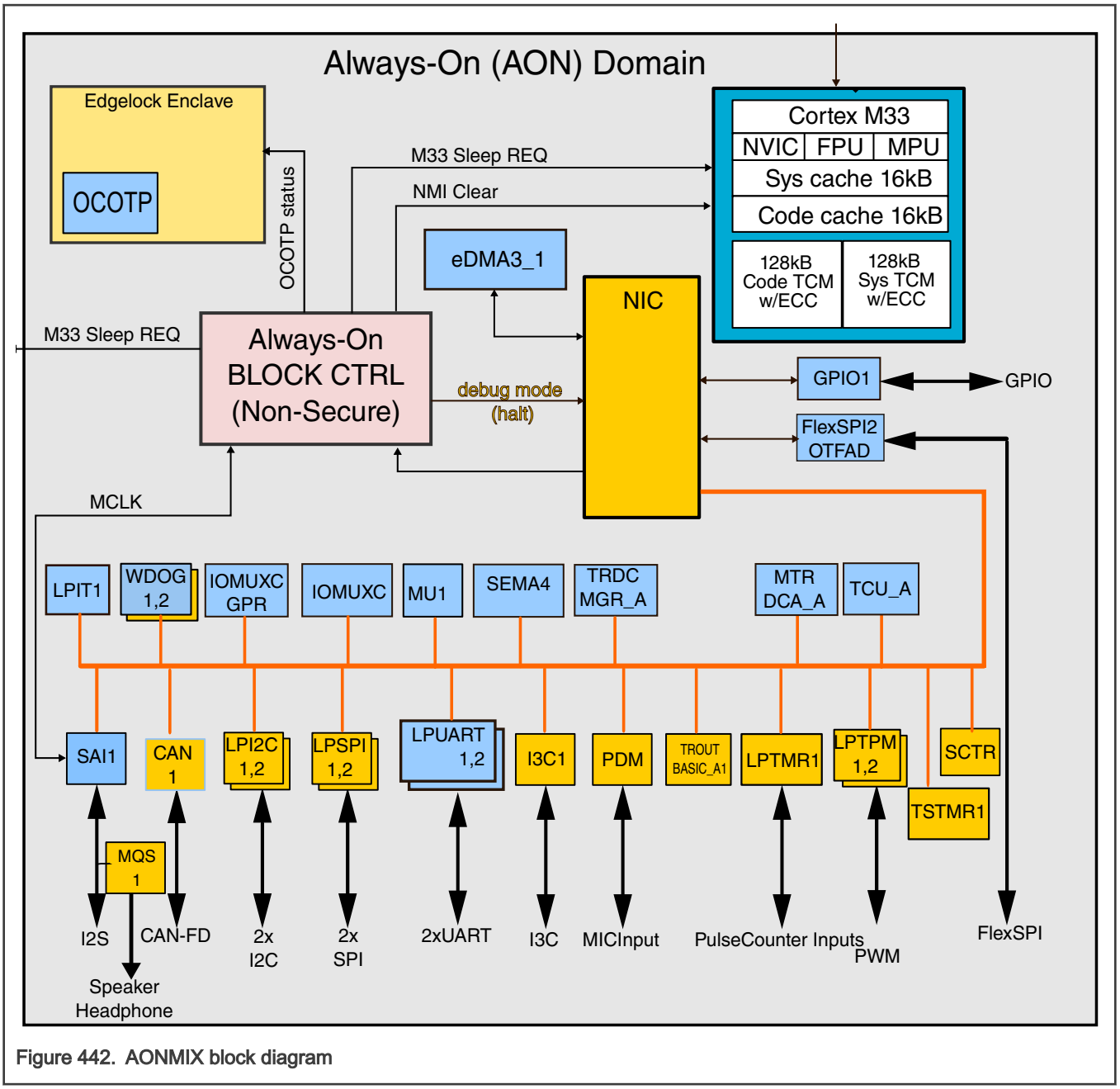


Figure 442. AONMIX block diagram

76.2 Functional description

This block is a register container and doesn't have any function itself. The regisers it contained cover:

- Sleeping or deep sleeping selection for low power mode entry
- Debug halt mask bits for peripherals in AON Domain
- AON Domain SSI master block low power bits
- SAI1 MCLK direction control
- DCDC status bit and exception bits read and clear
- OCOTP miscellaneous status bits
- CM33 NMI interrupt clear bit

- I3C1 asynchronous interrupt clear bit
- I3C1 IO on chip strong pull enable bit
- GPIO_AON bank voltage range control

76.2.1 Clocks

The table found here describes the clock sources for BLK_CTRL_NS_AON. Please see the Clock Control chapter for clock setting, configuration, and gating information.

Table 646. Clocks

Clock Name	Description
bus_aon_clk	AON bus clock for register access.

76.2.2 Reset

The AON NS BLK_CTRL resets with the AON Domain.

76.3 Initialization

There are no initialization steps required.

76.4 Memory Map and register definition

This section includes the blk_ctrl_ns_aonmix module memory map and detailed descriptions of all registers.

76.4.1 blk_ctrl_ns_aonmix register descriptions

76.4.1.1 AONMIX Non_Secure Block Control memory map

BLK_CTRL_NS_AONMIX base address: 4421_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
C	IPG DEBUG MASK BIT CA55 CORE0 (IPG_DEBUG_CA55C0)	32	RW	0000_0000
14	QCHANNEL DISABLE (QCH_DIS)	32	RW	0000_003F
20	MQS settings. (MQS_SETTINGS)	32	RW	0000_0000
28	Read-only version of the OCOTP fuse-access-disable bit (FUSE_ACC_DIS)	32	R	0000_0000
30	Read-only version of OCOTP fusedata_mtr_cfg_0 (OCOTP_FUSE_DATA0)	32	R	0000_0000
34	Read-only version of OCOTP fusedata_mtr_cfg_1 (OCOTP_FUSE_DATA1)	32	R	0000_0000
38	Read-only version of OCOTP fusedata_mtr_cfg_2 (OCOTP_FUSE_DATA2)	32	R	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
3C	Read-only version of OCOTP fusedata_mtr_cfg_3 (OCOTP_FUSE_DATA3)	32	R	0000_0000
40	Read-only version of OCOTP fusedata_mtr_cfg_4 (OCOTP_FUSE_DATA4)	32	R	0000_0000
44	Read-only version of OCOTP fusedata_mtr_cfg_5 (OCOTP_FUSE_DATA5)	32	R	0000_0000
48	Read-only version of OCOTP fusedata_mtr_cfg_6 (OCOTP_FUSE_DATA6)	32	R	0000_0000
4C	Read-only version of OCOTP fusedata_mtr_cfg_7 (OCOTP_FUSE_DATA7)	32	R	0000_0000
50	Read-only version of OCOTP fusedata_mem_trim_cfg0 (OCOTP_FUSE_DATA8)	32	R	0000_0000
54	Read-only version of OCOTP fusedata_mem_trim_cfg1 (OCOTP_FUSE_DATA9)	32	R	0000_0000
58	Read-only version of OCOTP fusedata_mem_trim_cfg2 (OCOTP_FUSE_DATA10)	32	R	0000_0000
5C	Read-only version of OCOTP fusedata_mem_trim_cfg3 (OCOTP_FUSE_DATA11)	32	R	0000_0000
60	Read-only version of OCOTP fusedata_mem_trim_cfg4 (OCOTP_FUSE_DATA12)	32	R	0000_0000
64	Read-only version of OCOTP fusedata_mem_trim_cfg5 (OCOTP_FUSE_DATA13)	32	R	0000_0000
68	Read-only version of OCOTP fusedata_mem_trim_cfg6 (OCOTP_FUSE_DATA14)	32	R	0000_0000
6C	Read-only version of OCOTP fusedata_mem_trim_cfg7 (OCOTP_FUSE_DATA15)	32	R	0000_0000
70	I3C1 WAKEUPX CLR (I3C1_WAKEUP)	32	RW	0000_0000
74	OCOTP status (OCOTP_STATUS)	32	R	0000_0000
78	PDM clock selection register (PDM_CLK_SEL)	32	RW	0000_0000
7C	I3C1 SDA IRQ CONTROL (I3C1_SDA_IRQ)	32	RW	0000_0000
80	Fastboot enable (FASTBOOT_ENABLE)	32	R	0000_000C
84	Read only Edgelock fuse (EDGELOCK_FW_PRESENT)	32	R	0000_0000

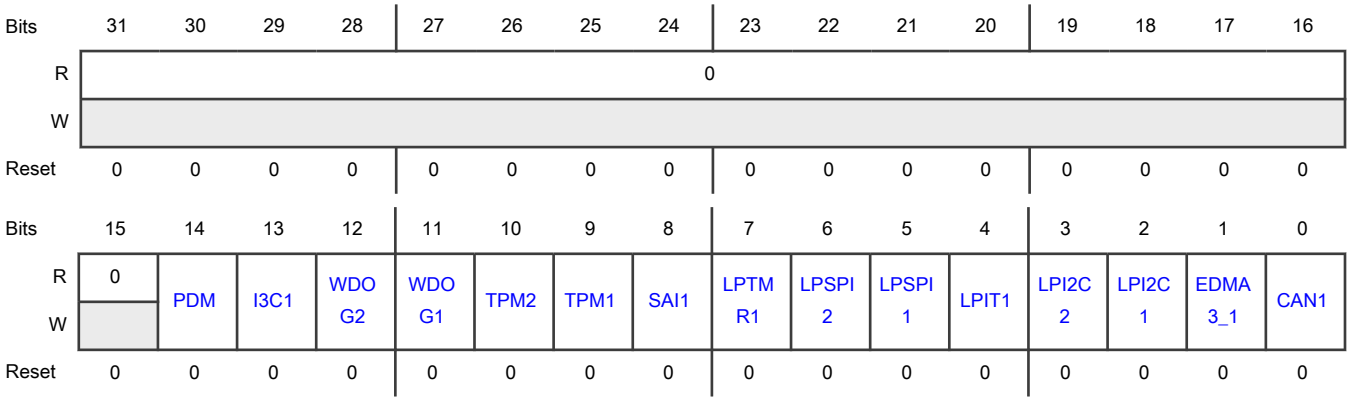
76.4.1.2 IPG DEBUG MASK BIT CA55 CORE0 (IPG_DEBUG_CA55C0)

mask bit for HALTED from CA55 core 0 to peripheral ipg_debug 0: mask to 0 1: unmask

Offset

Register	Offset
IPG_DEBUG_CA55C0	Ch

Diagram



Fields

Field	Description
31-15 —	Reserved
14 PDM	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
13 I3C1	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
12 WDOG2	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
11 WDOG1	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
10 TPM2	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted

Table continues on the next page...

Table continued from the previous page...

Field	Description
9 TPM1	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
8 SAI1	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
7 LPTMR1	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
6 LPSPI2	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
5 LPSPI1	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
4 LPIT1	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
3 LPI2C2	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
2 LPI2C1	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
1 EDMA3_1	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted
0 CAN1	Mask bit for debug halted mode 0 - Block does not enter debug halted mode with CA55 1 - Block enters debug halted mode when CA55 is debug halted

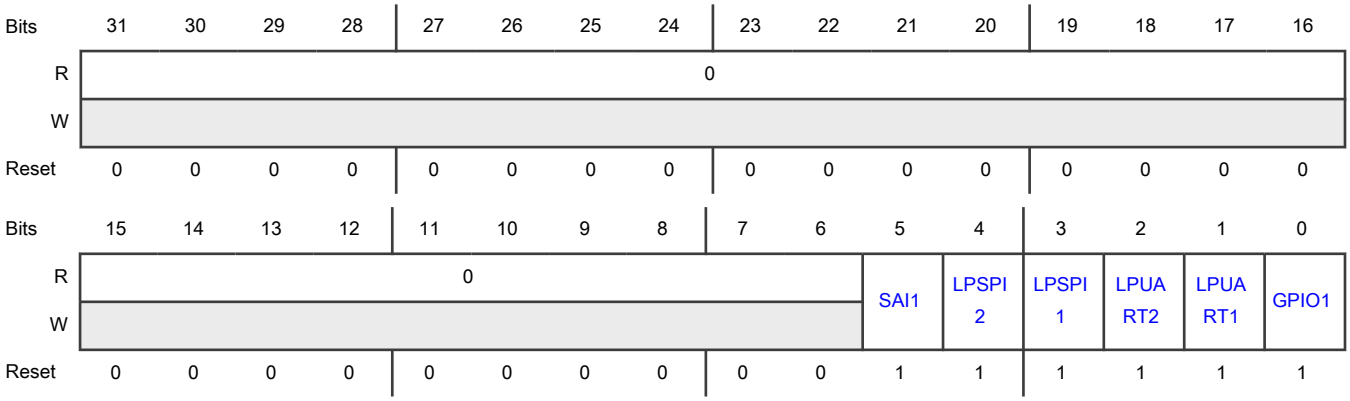
76.4.1.3 QCHANNEL DISABLE (QCH_DIS)

This register includes bits to disable QCHANNEL for modules on the chip.

Offset

Register	Offset
QCH_DIS	14h

Diagram



Fields

Field	Description
31-6 —	Reserved
5 SAI1	Module QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled
4 LPSPI2	Module QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled
3 LPSPI1	Module QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled
2 LPUART2	Module QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled
1 LPUART1	Module QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 GPIO1	Module QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled

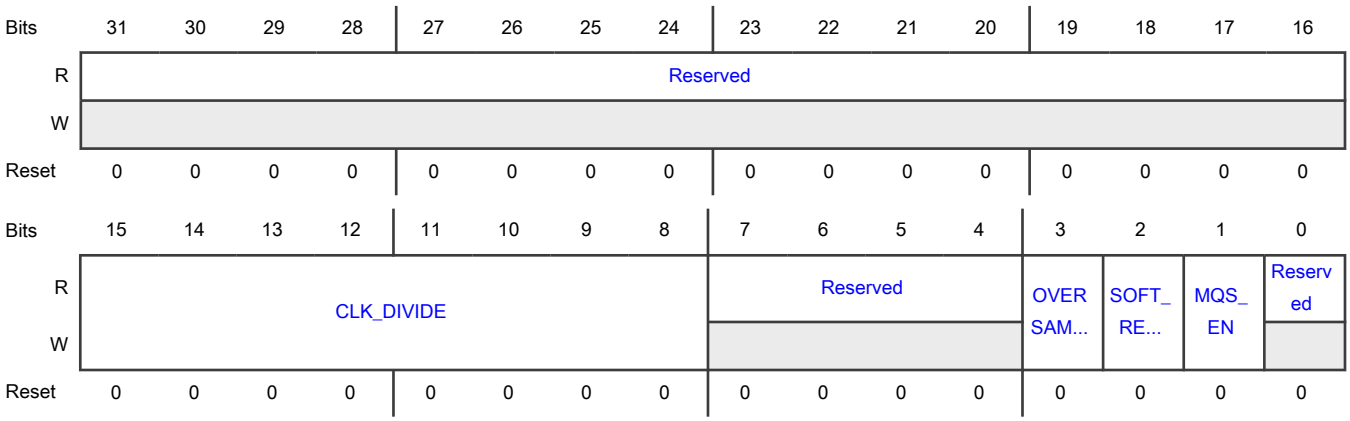
76.4.1.4 MQS settings. (MQS_SETTINGS)

This register includes MQS settings for SAI selection, Enable, Software Reset, clock divide factor and Oversample.

Offset

Register	Offset
MQS_SETTINGS	20h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 CLK_DIVIDE	Clock divide factor configuration
7-4 —	Reserved
3 OVERSAMPLE	Oversample enable

Table continues on the next page...

Table continued from the previous page...

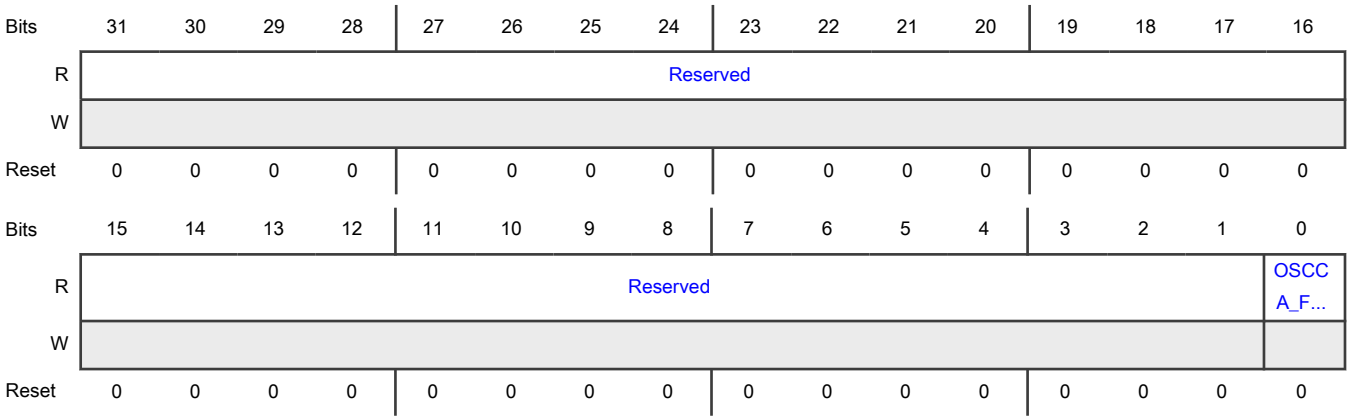
Field	Description
2 SOFT_RESET	Software Reset
1 MQS_EN	MQS Enable
0 —	Reserved

76.4.1.5 Read-only version of the OCOTP fuse-access-disable bit (FUSE_ACC_DIS)

Offset

Register	Offset
FUSE_ACC_DIS	28h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 OSCCA_FUSE_READ_DIS	OSCCA fuse read disable This bit enables the chip to access OCOTP registers. 0 - The chip is allowed to access the OCOTP registers 1 - The chip is not allowed to access the OCOTP registers

76.4.1.6 Read-only version of OCOTP fusedata_mtr_cfg_0 (OCOTP_FUSE_DATA0)

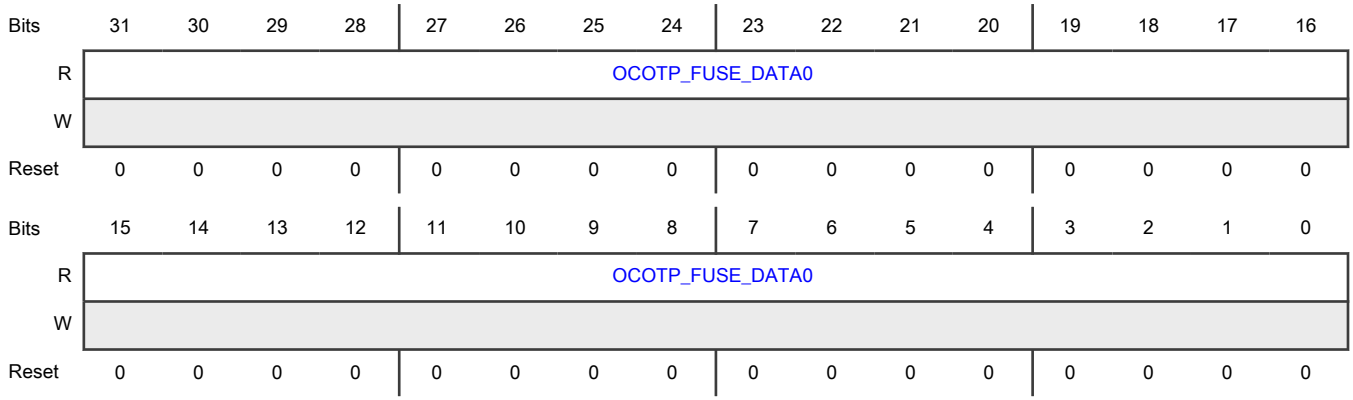
NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA0	30h

Diagram



Fields

Field	Description
31-0 OCOTP_FUSE_DATA0	OCOTP_FUSE_DATA0

76.4.1.7 Read-only version of OCOTP fusedata_mtr_cfg_1 (OCOTP_FUSE_DATA1)

NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA1	34h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCOTP_FUSE_DATA1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCOTP_FUSE_DATA1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OCOTP_FUSE_DATA1	OCOTP_FUSE_DATA1

76.4.1.8 Read-only version of OCOTP fusedata_mtr_cfg_2 (OCOTP_FUSE_DATA2)

NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA2	38h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCOTP_FUSE_DATA2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCOTP_FUSE_DATA2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OCOTP_FUSE_DATA2	OCOTP_FUSE_DATA2

76.4.1.9 Read-only version of OCOTP fusedata_mtr_cfg_3 (OCOTP_FUSE_DATA3)

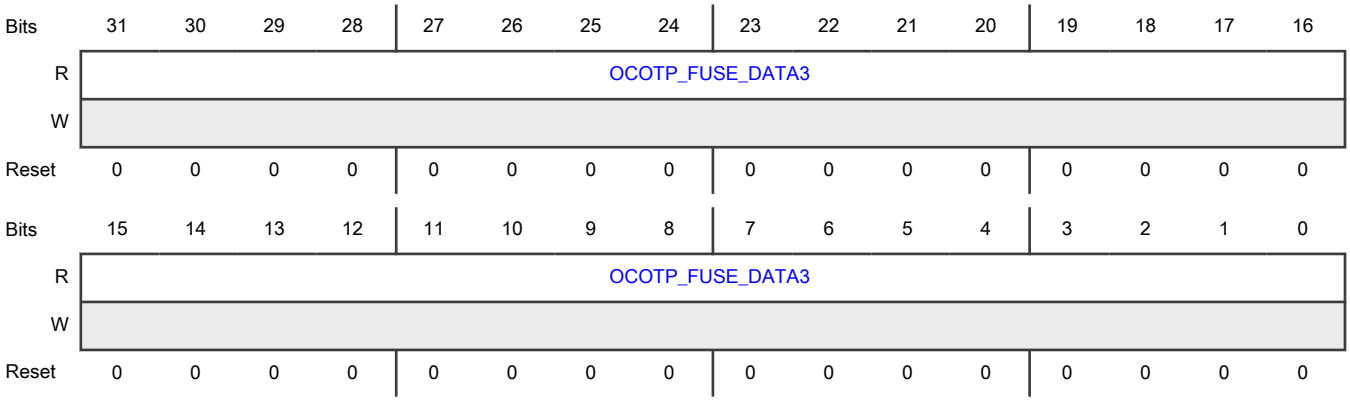
NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA3	3Ch

Diagram



Fields

Field	Description
31-0 OCOTP_FUSE_DATA3	OCOTP_FUSE_DATA3

76.4.1.10 Read-only version of OCOTP fusedata_mtr_cfg_4 (OCOTP_FUSE_DATA4)

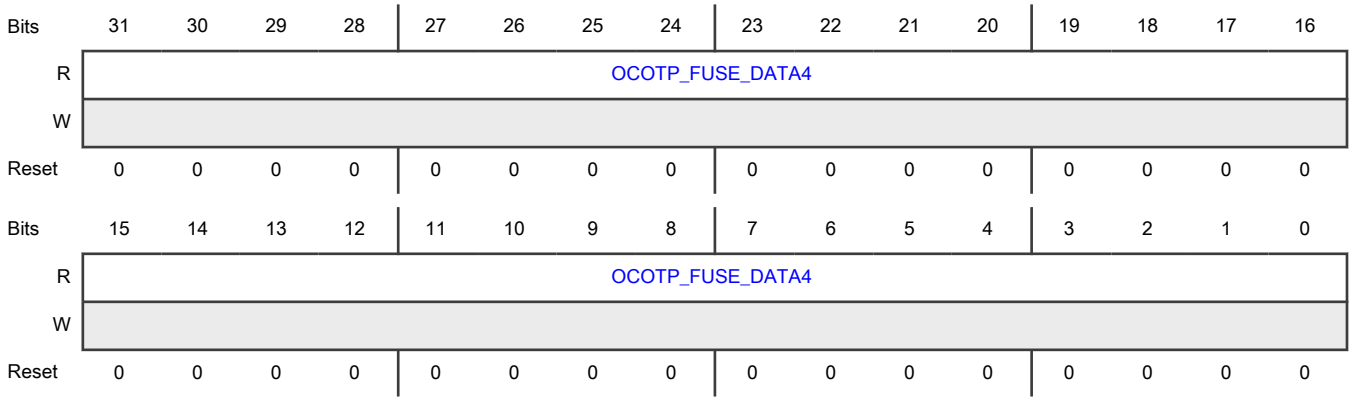
NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA4	40h

Diagram



Fields

Field	Description
31-0 OCOTP_FUSE_DATA4	OCOTP_FUSE_DATA4

76.4.1.11 Read-only version of OCOTP fusedata_mtr_cfg_5 (OCOTP_FUSE_DATA5)

NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA5	44h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCOTP_FUSE_DATA5															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCOTP_FUSE_DATA5															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OCOTP_FUSE_DATA5	OCOTP_FUSE_DATA5

76.4.1.12 Read-only version of OCOTP fusedata_mtr_cfg_6 (OCOTP_FUSE_DATA6)

NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA6	48h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCOTP_FUSE_DATA6															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCOTP_FUSE_DATA6															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OCOTP_FUSE_DATA6	OCOTP_FUSE_DATA6

76.4.1.13 Read-only version of OCOTP fusedata_mtr_cfg_7 (OCOTP_FUSE_DATA7)

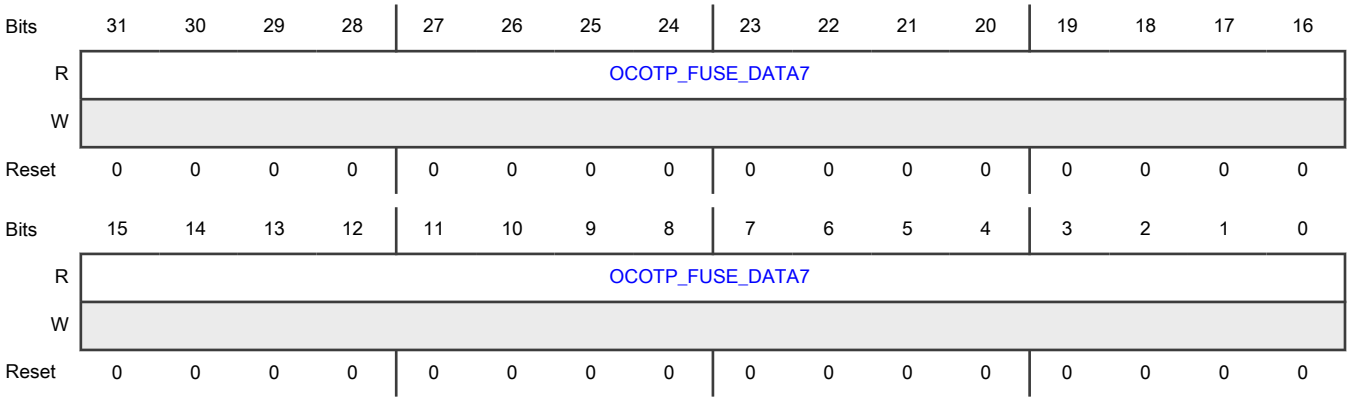
NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA7	4Ch

Diagram



Fields

Field	Description
31-0 OCOTP_FUSE_DATA7	OCOTP_FUSE_DATA7

76.4.1.14 Read-only version of OCOTP fusedata_mem_trim_cfg0 (OCOTP_FUSE_DATA8)

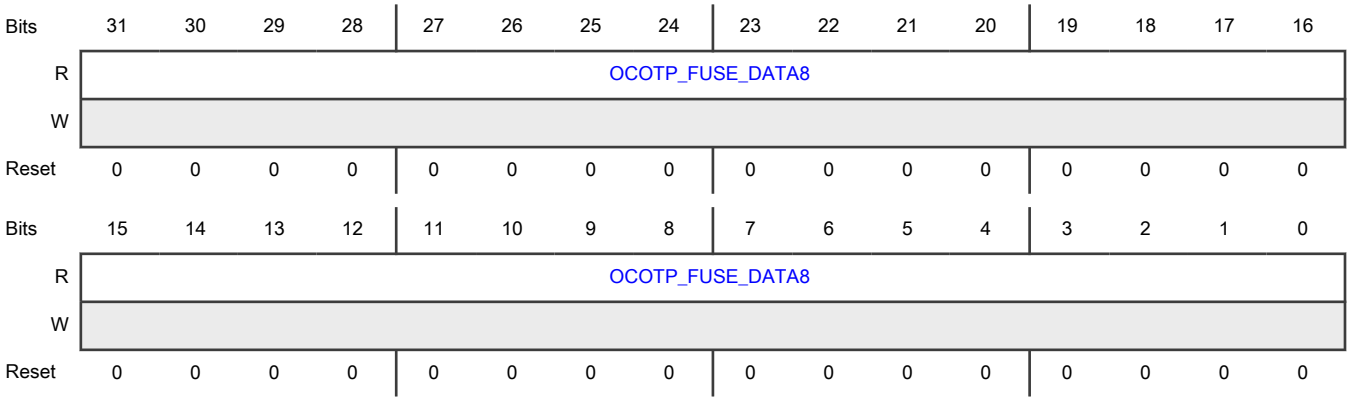
NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA8	50h

Diagram



Fields

Field	Description
31-0 OCOTP_FUSE_DATA8	OCOTP_FUSE_DATA8

76.4.1.15 Read-only version of OCOTP fusedata_mem_trim_cfg1 (OCOTP_FUSE_DATA9)

NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA9	54h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCOTP_FUSE_DATA9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCOTP_FUSE_DATA9															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OCOTP_FUSE_DATA9	OCOTP_FUSE_DATA9

76.4.1.16 Read-only version of OCOTP fusedata_mem_trim_cfg2 (OCOTP_FUSE_DATA10)

NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA10	58h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCOTP_FUSE_DATA10															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCOTP_FUSE_DATA10															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OCOTP_FUSE_DATA10	OCOTP_FUSE_DATA10

76.4.1.17 Read-only version of OCOTP fusedata_mem_trim_cfg3 (OCOTP_FUSE_DATA11)

NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA11	5Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCOTP_FUSE_DATA11															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCOTP_FUSE_DATA11															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OCOTP_FUSE_DATA11	OCOTP_FUSE_DATA12

76.4.1.18 Read-only version of OCOTP fusedata_mem_trim_cfg4 (OCOTP_FUSE_DATA12)

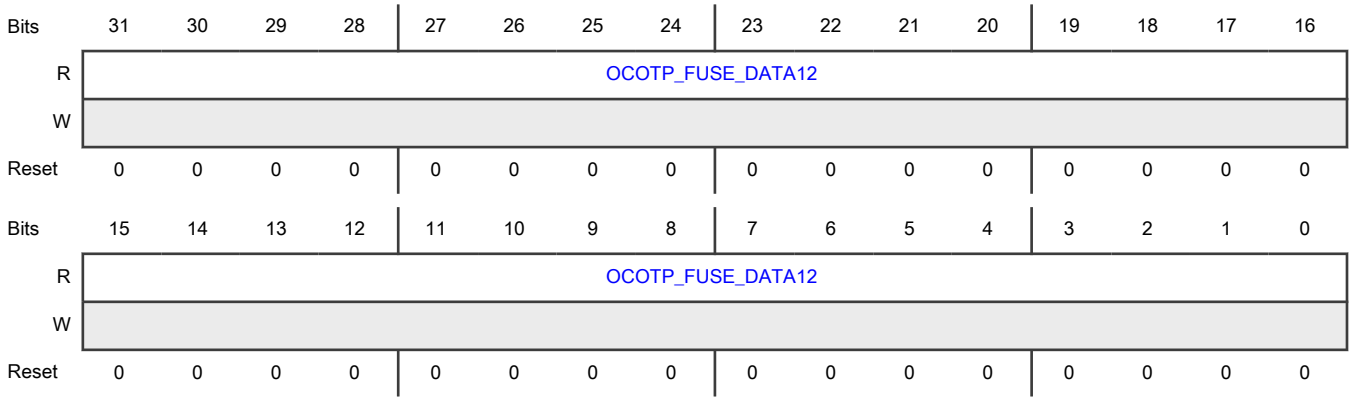
NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA12	60h

Diagram



Fields

Field	Description
31-0 OCOTP_FUSE_DATA12	OCOTP_FUSE_DATA13

76.4.1.19 Read-only version of OCOTP fusedata_mem_trim_cfg5 (OCOTP_FUSE_DATA13)

NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA13	64h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCOTP_FUSE_DATA13															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCOTP_FUSE_DATA13															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OCOTP_FUSE_DATA13	OCOTP_FUSE_DATA13

76.4.1.20 Read-only version of OCOTP fusedata_mem_trim_cfg6 (OCOTP_FUSE_DATA14)

NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA14	68h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCOTP_FUSE_DATA14															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCOTP_FUSE_DATA14															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OCOTP_FUSE_DATA14	OCOTP_FUSE_DATA14

76.4.1.21 Read-only version of OCOTP fusedata_mem_trim_cfg7 (OCOTP_FUSE_DATA15)

NOTE

Reset value of shadow registers will change depending on the fuses programmed

Offset

Register	Offset
OCOTP_FUSE_DATA15	6Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCOTP_FUSE_DATA15															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCOTP_FUSE_DATA15															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 OCOTP_FUSE_DATA15	OCOTP_FUSE_DATA15

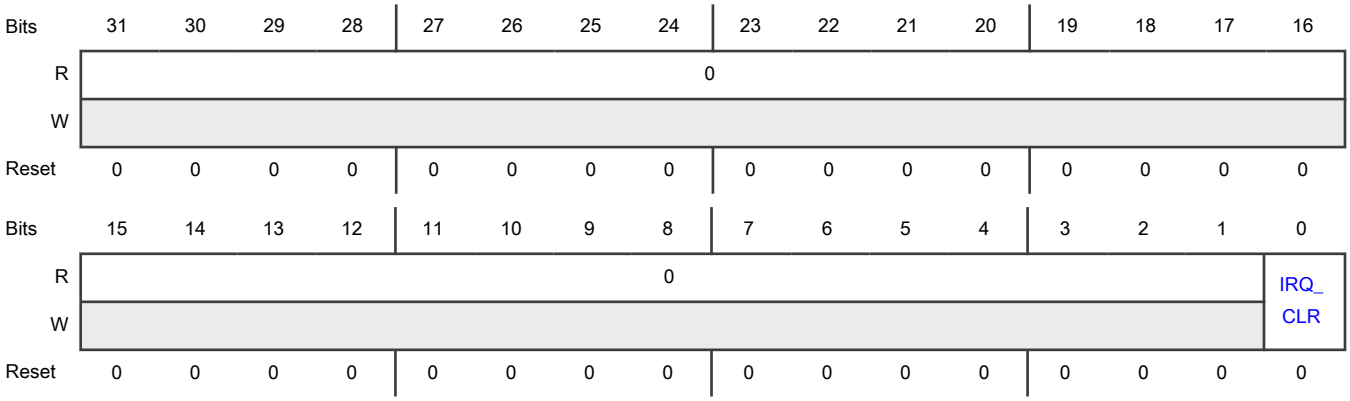
76.4.1.22 I3C1 WAKEUPX CLR (I3C1_WAKEUP)

Clears the interrupt of the I3C1 wakeup interrupt

Offset

Register	Offset
I3C1_WAKEUP	70h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 IRQ_CLR	I3C1 Interrupt request clear

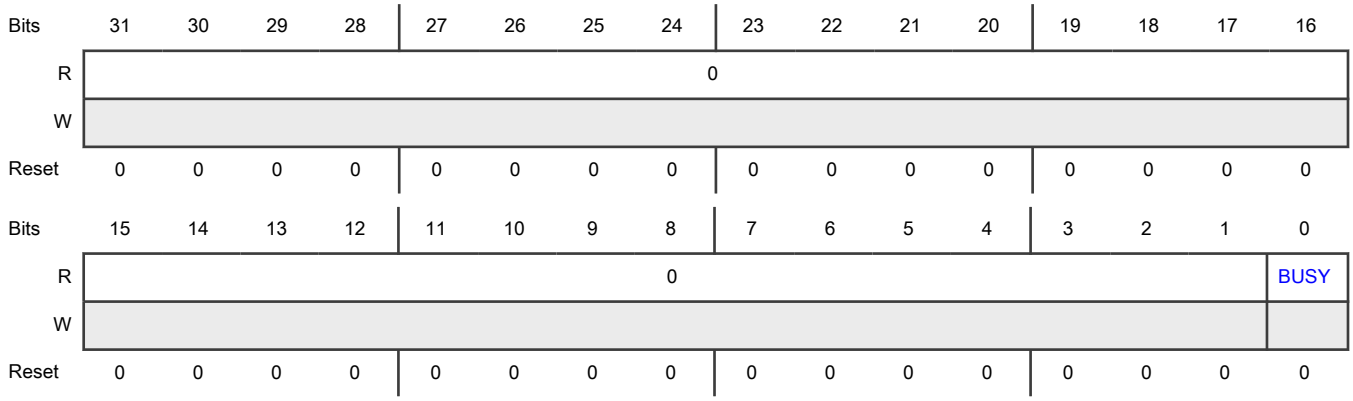
76.4.1.23 OCOTP status (OCOTP_STATUS)

This register shows the status of the OCOTP.

Offset

Register	Offset
OCOTP_STATUS	74h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 BUSY	OCOTP controller busy bit Indicates OTP controller is busy 0 - Idle 1 - OCOTP is Busy

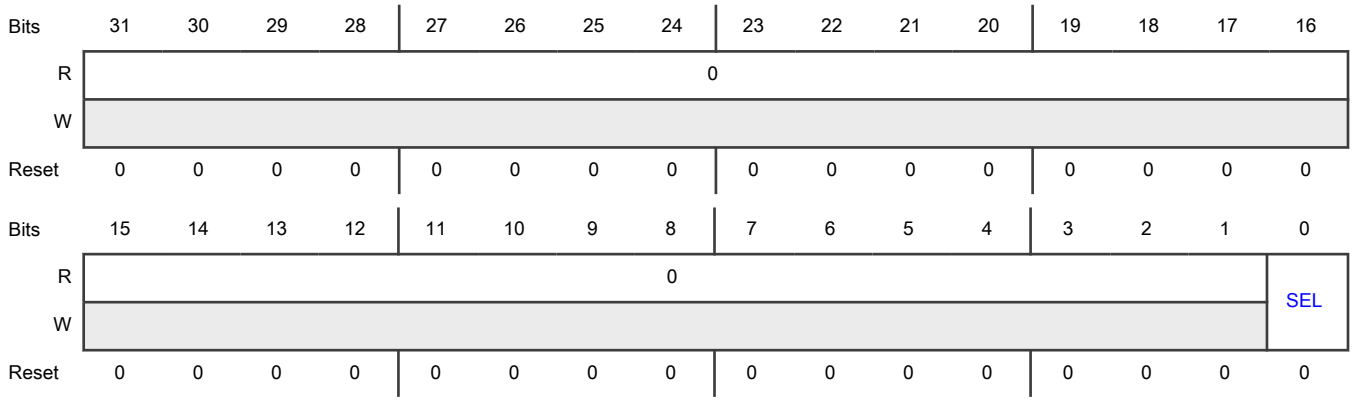
76.4.1.24 PDM clock selection register (PDM_CLK_SEL)

select source of PDM clock

Offset

Register	Offset
PDM_CLK_SEL	78h

Diagram



Fields

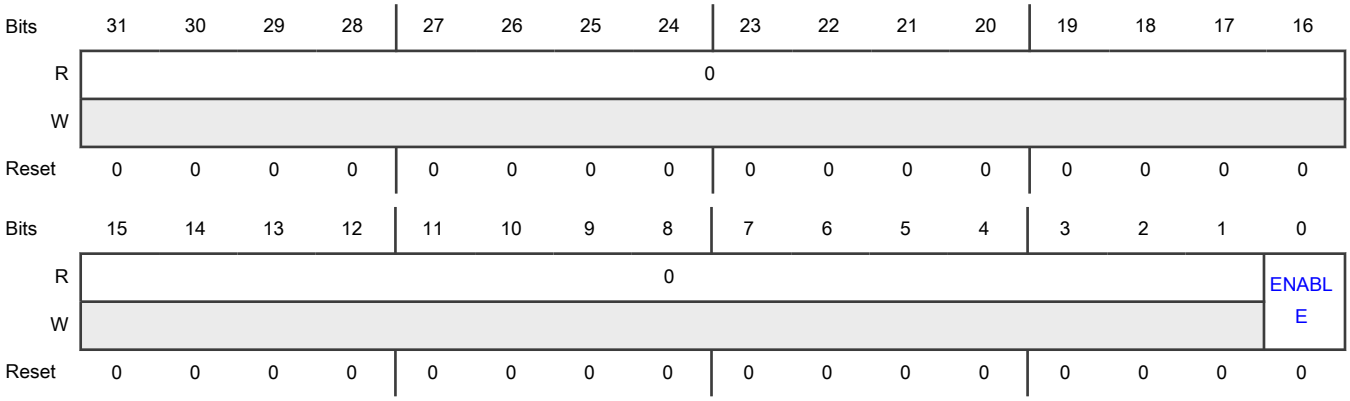
Field	Description
31-1 —	Reserved
0 SEL	Select source for PDM clock This bit enables source clock selection for PDM clock. 0 - PDM root clock 1 - SAI1_MCLK

76.4.1.25 I3C1 SDA IRQ CONTROL (I3C1_SDA_IRQ)

Offset

Register	Offset
I3C1_SDA_IRQ	7Ch

Diagram



Fields

Field	Description
31-1 —	Reserved
0 ENABLE	IRQ enable bit 0 - I3C1 SDA IRQ disable 1 - I3C1 SDA IRQ enable

76.4.1.26 Fastboot enable (FASTBOOT_ENABLE)

This register includes read-only fastboot enable bits.

Offset

Register	Offset
FASTBOOT_ENABLE	80h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												BP_FASTBOOT_ENA...		FASTBOOT_ENABLE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Fields

Field	Description
31-4 —	Reserved
3-2 BP_FASTBOOT_ENABLE	BP_FASTBOOT_ENABLE
1-0 FASTBOOT_ENABLE	FASTBOOT_ENABLE bits Read value will be inverted of FASTBOOT_ENABLE & BP_FASTBOOT_ENABLE , As if FASTBOOT_ENABLE will "00" then BP_FASTBOOT_ENABLE will be "11"

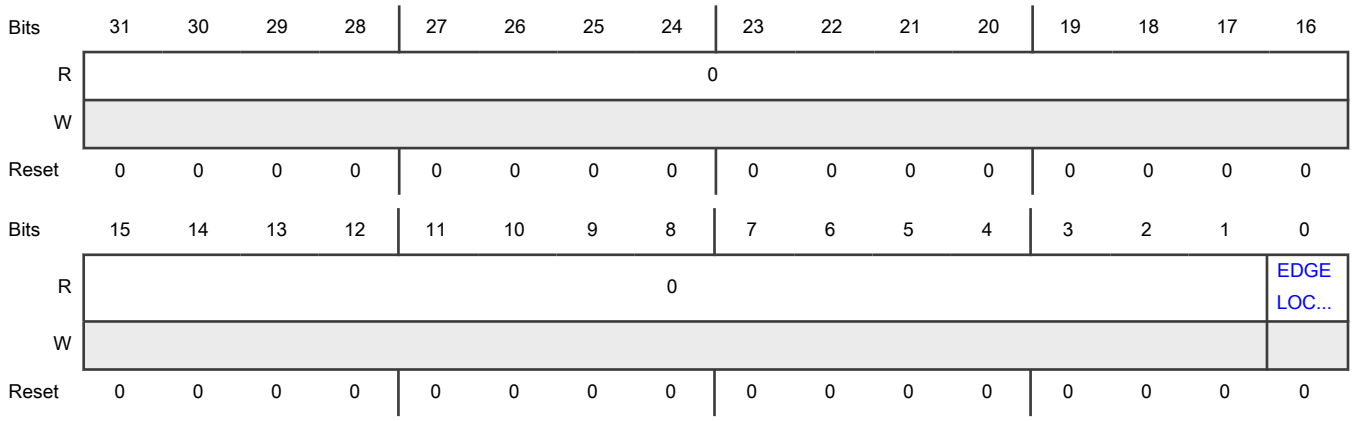
76.4.1.27 Read only Edgelock fuse (EDGELOCK_FW_PRESENT)

If/when programmed, this bit will indicate to Edgelock ROM that is *must* load Edgelock FW (NXP signed/encrypted), *before* any other system image. It is there to allow enforcing Edgelock ROM patching (for example if we hit ROM issues that might represent system level vulnerabilities). This bit can be set in NXP manufacturing and/or in field.

Offset

Register	Offset
EDGELOCK_FW_PRESENT	84h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 EDGELOCK_FW_PRESENT	Read only bit for Edgelock fuse

Chapter 77
Always-on Domain, Secure Block Control (blk_ctrl_s_aonmix)

77.1 Introduction

The BLK_CTRL_S_AONMIX contains control and status registers, that can be accessed only by secure processors. These registers include IRQ mask bits, low power control bits, WDOG output mask bits, CA55 and CM33 start address and some other core-related control bits.

77.1.1 Block diagram

A block diagram of the AONMIX implementation is shown below.

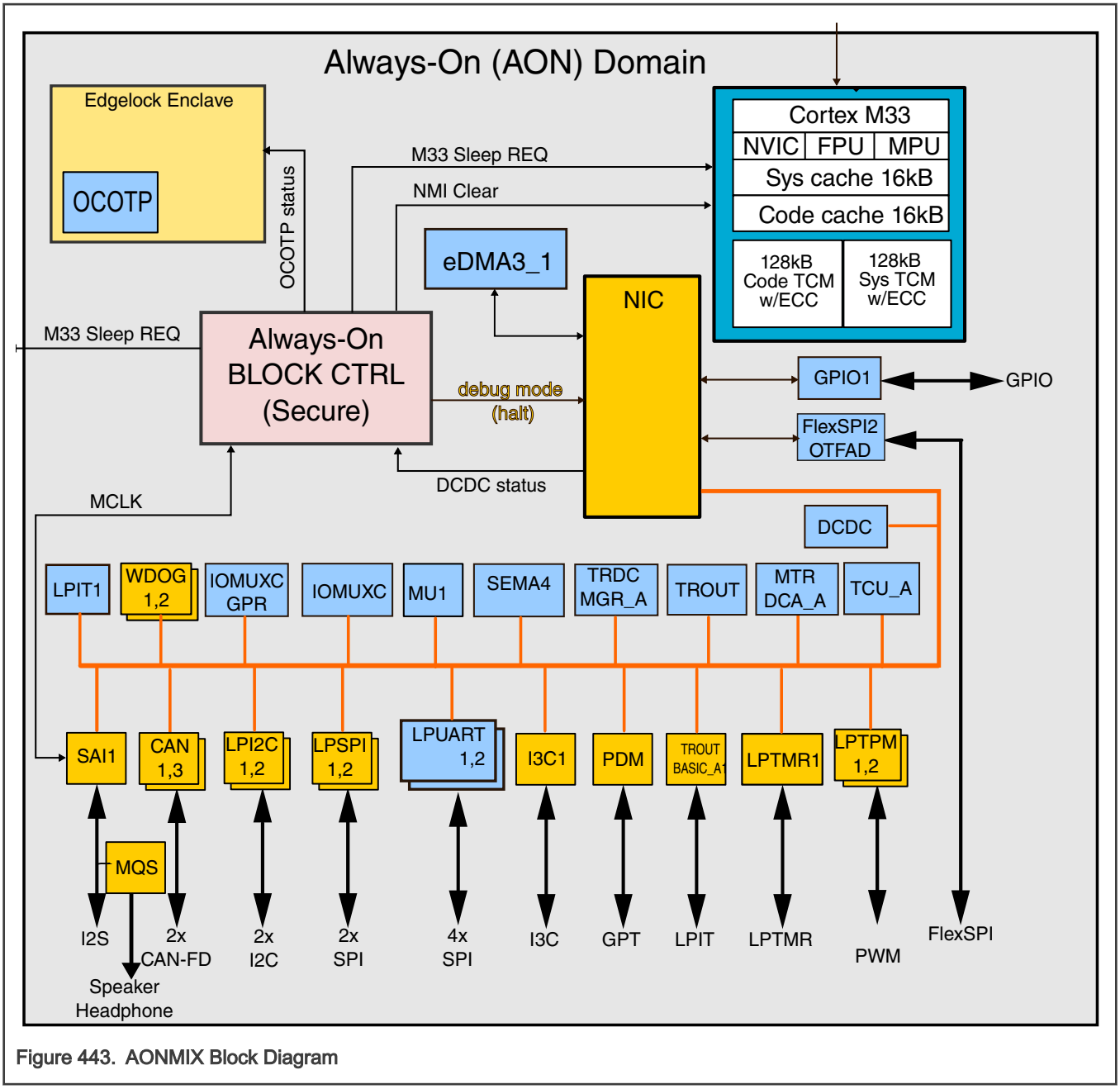


Figure 443. AONMIX Block Diagram

77.2 Functional description

Please refer to RM for detailed function description.

77.2.1 Clocks

The table found here describes the clock sources for BLK_CTRL_S_AON. Please see the Clock Control chapter for clock setting, configuration, and gating information.

Clock Name	Description

Table continues on the next page...

Table continued from the previous page...

edgelock_clk	Edgelock Trust

77.2.2 Reset

BLK CTRL S AONMIX resets with the AONMIX domain.

77.3 Initialization

There are no initialization steps required.

77.4 Memory Map and register definition

This section includes the S_AONMIX module memory map and detailed descriptions of all registers.

77.4.1 blk_ctrl_s_aonmix register descriptions

77.4.1.1 AONMIX Block Control For Secure World memory map

BLK_CTRL_S_AONMIX base address: 444F_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	CM33_IRQ_MASK0 (CM33_IRQ_MASK0)	32	RW	FFFF_FFFF
4	CM33_IRQ_MASK1 (CM33_IRQ_MASK1)	32	RW	FFFF_FFFF
8	CM33_IRQ_MASK2 (CM33_IRQ_MASK2)	32	RW	FFFF_FFFF
C	CM33_IRQ_MASK3 (CM33_IRQ_MASK3)	32	RW	FFFF_FFFF
10	CM33_IRQ_MASK4 (CM33_IRQ_MASK4)	32	RW	FFFF_FFFF
14	CM33_IRQ_MASK5 (CM33_IRQ_MASK5)	32	RW	FFFF_FFFF
18	CM33_IRQ_MASK6 (CM33_IRQ_MASK6)	32	RW	FFFF_FFFF
1C	M33 restart non-secure address (INITNSVTOR)	32	RW	FFFF_FFFF
40	CA55_IRQ_MASK0 (CA55_IRQ_MASK0)	32	RW	FFFF_FFFF
44	CA55_IRQ_MASK1 (CA55_IRQ_MASK1)	32	RW	FFFF_FFFF
48	CA55_IRQ_MASK2 (CA55_IRQ_MASK2)	32	RW	FFFF_FFFF
4C	CA55_IRQ_MASK3 (CA55_IRQ_MASK3)	32	RW	FFFF_FFFF
50	CA55_IRQ_MASK4 (CA55_IRQ_MASK4)	32	RW	FFFF_FFFF
54	CA55_IRQ_MASK5 (CA55_IRQ_MASK5)	32	RW	FFFF_FFFF
58	CA55_IRQ_MASK6 (CA55_IRQ_MASK6)	32	RW	FFFF_FFFF
5C	M33 restart secure address (INITSVTOR)	32	RW	FFFF_FFFF

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
60	M33 Configure (M33_CFG)	32	RW	0000_0004
100	DAP Access Sticky (DAP_ACCESS_STKYBIT)	32	RW	0000_0000
110	Low power handshake enable (LP_HANDSHAKE)	32	RW	0000_0000
114	Low power handshake enable (LP_HANDSHAKE2)	32	RW	0000_0000
118	CPUWAIT settings for CA55 CPU (CA55_CPUWAIT)	32	RW	0000_0003
11C	CA55 core0 lower boot address bits (CA55_RVBARADDR0_L)	32	RW	0000_0400
120	CA55 core0 higher boot address bits (CA55_RVBARADDR0_H)	32	RW	0000_0000
124	CA55 core1 lower boot address bits (CA55_RVBARADDR1_L)	32	RW	0000_0400
128	CA55 core1 higher boot address bits (CA55_RVBARADDR1_H)	32	RW	0000_0000
12C	Mask bits of Edgelock interrupt (ELE_IRQ_MASK)	32	RW	0000_01DF
130	Mask bits of ELE reset (ELE_RESET_REQ_MASK)	32	RW	0000_0093
134	ELE halt status (ELE_HALT_ST)	32	RW	0000_0000
138	Control the boot mode of two ca55 cores (CA55_MODE)	32	RW	0000_0003
13C	NMI MASK bits (NMI_MASK)	32	RW	0000_0000
140	NMI clear (NMI_CLR)	32	RW	0000_0000
144	WDOG any mask (WDOG_ANY_MASK)	32	RW	0000_001F
148	ELEV1_IPI_NOCLK_REF1 clear (ELEV1_IPI_NOCLK_REF1)	32	RW	0000_0000

77.4.1.2 CM33_IRQ_MASK0 (CM33_IRQ_MASK0)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CM33_IRQ_MASK0	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 M	CM33 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

77.4.1.3 CM33_IRQ_MASK1 (CM33_IRQ_MASK1)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CM33_IRQ_MASK1	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 M	CM33 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

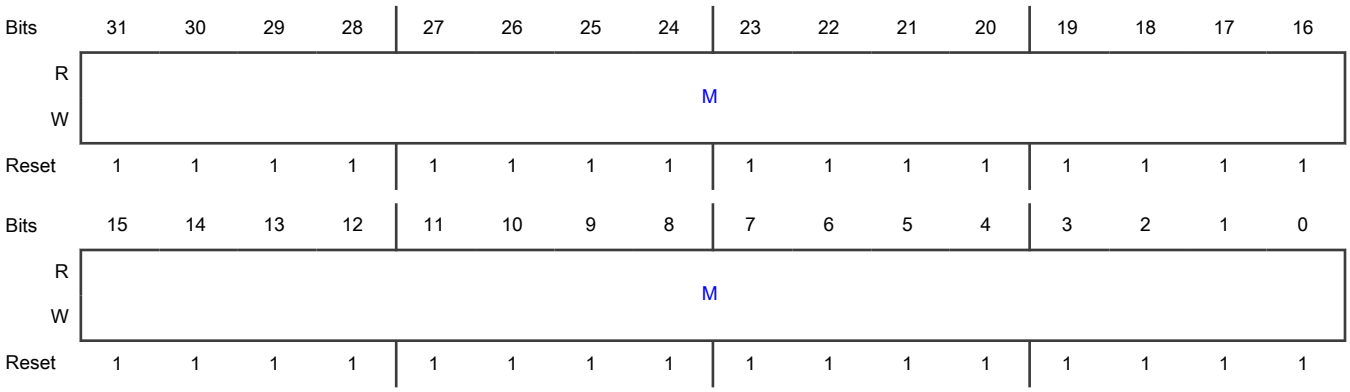
77.4.1.4 CM33_IRQ_MASK2 (CM33_IRQ_MASK2)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CM33_IRQ_MASK2	8h

Diagram



Fields

Field	Description
31-0 M	CM33 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

77.4.1.5 CM33_IRQ_MASK3 (CM33_IRQ_MASK3)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CM33_IRQ_MASK3	Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 M	CM33 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

77.4.1.6 CM33_IRQ_MASK4 (CM33_IRQ_MASK4)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CM33_IRQ_MASK4	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 M	CM33 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

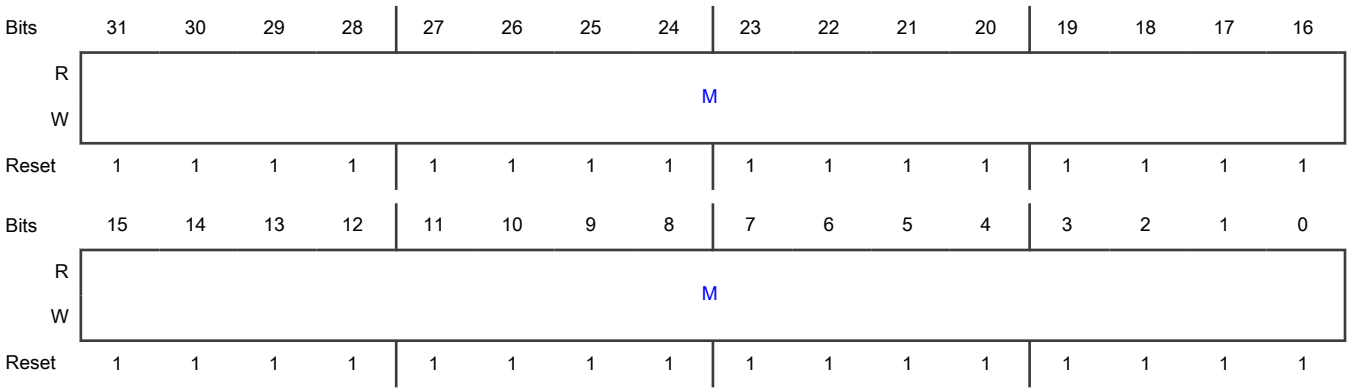
77.4.1.7 CM33_IRQ_MASK5 (CM33_IRQ_MASK5)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CM33_IRQ_MASK5	14h

Diagram



Fields

Field	Description
31-0 M	CM33 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

77.4.1.8 CM33_IRQ_MASK6 (CM33_IRQ_MASK6)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CM33_IRQ_MASK6	18h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 M	CM33 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

77.4.1.9 M33 restart non-secure address (INITNSVTOR)

After recovering from LP mode, M33 goes to this address when non-secure

Offset

Register	Offset
INITNSVTOR	1Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 M	INITSVTOR M33 restart secure address

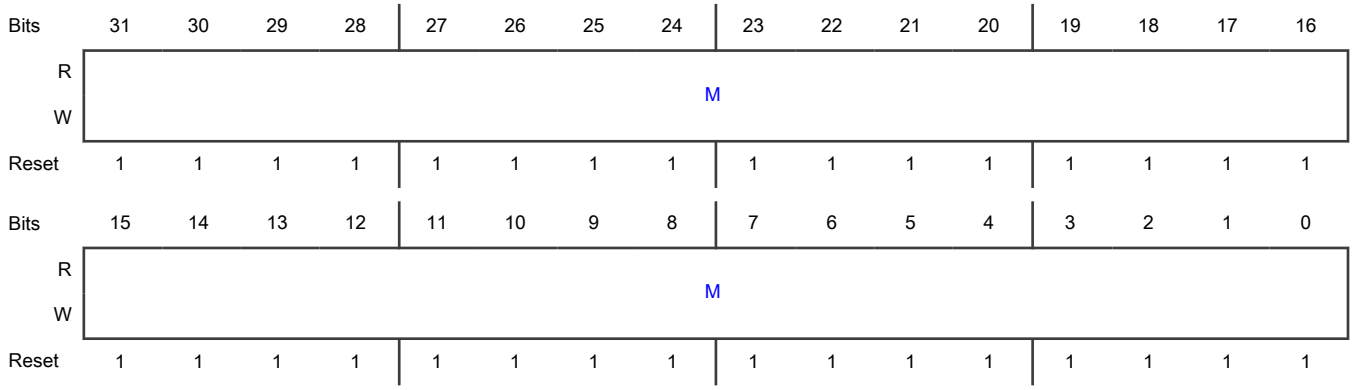
77.4.1.10 CA55_IRQ_MASK0 (CA55_IRQ_MASK0)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CA55_IRQ_MASK0	40h

Diagram



Fields

Field	Description
31-0 M	CA55 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

77.4.1.11 CA55_IRQ_MASK1 (CA55_IRQ_MASK1)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CA55_IRQ_MASK1	44h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 M	CA55 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

77.4.1.12 CA55_IRQ_MASK2 (CA55_IRQ_MASK2)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CA55_IRQ_MASK2	48h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 M	CA55 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

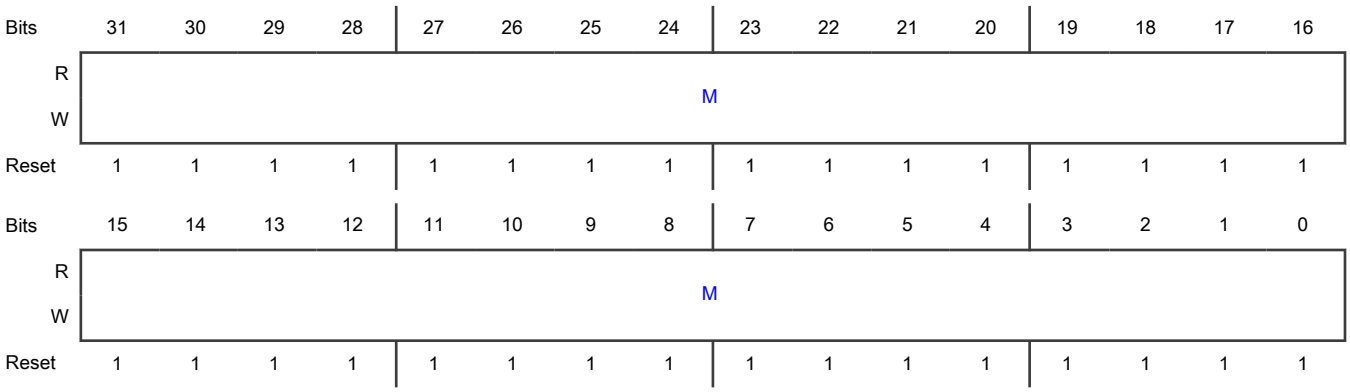
77.4.1.13 CA55_IRQ_MASK3 (CA55_IRQ_MASK3)

CA55 IRQ MASK3, refer to IRQ table for the usage of each bit

Offset

Register	Offset
CA55_IRQ_MASK3	4Ch

Diagram



Fields

Field	Description
31-0 M	CA55 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

77.4.1.14 CA55_IRQ_MASK4 (CA55_IRQ_MASK4)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CA55_IRQ_MASK4	50h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 M	CA55 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

77.4.1.15 CA55_IRQ_MASK5 (CA55_IRQ_MASK5)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CA55_IRQ_MASK5	54h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 M	CA55 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

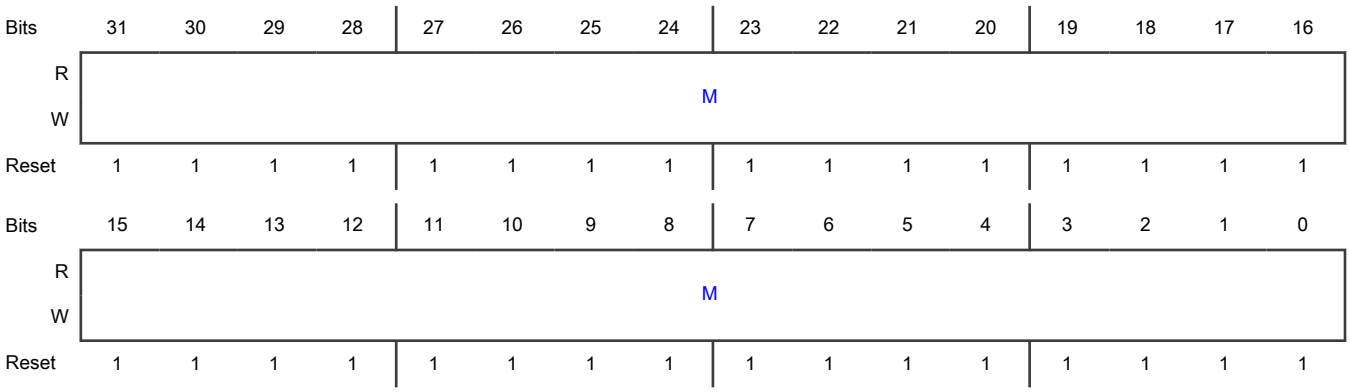
77.4.1.16 CA55_IRQ_MASK6 (CA55_IRQ_MASK6)

Please refer to System interrupt summary table for the usage of each bit.

Offset

Register	Offset
CA55_IRQ_MASK6	58h

Diagram



Fields

Field	Description
31-0 M	CA55 IRQ MASK 0000_0000_0000_0000_0000_0000_0000_0000 - IRQ masked 0000_0000_0000_0000_0000_0000_0000_0001 - IRQ not masked

77.4.1.17 M33 restart secure address (INITSVTOR)

After recover from LP mode, goes to this address when secure

Offset

Register	Offset
INITSVTOR	5Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	M															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	M															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-0 M	INITSVTOR M33 restart secure address

77.4.1.18 M33 Configure (M33_CFG)

This register is used for m33 core configuration

Offset

Register	Offset
M33_CFG	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											TCM_SIZE		WAIT	Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Fields

Field	Description
31-5 —	Reserved
4-3 TCM_SIZE	M33 TCM SIZE 00 - Regular TCM, 128KB Code TCM and 128KB Sys TCM 01 - Double Code TCM, 256KB Code TCM 10 - Double Sys TCM, 256KB Sys TCM 11 - Reserved
2 WAIT	M33 CPU WAIT
1-0 —	Reserved

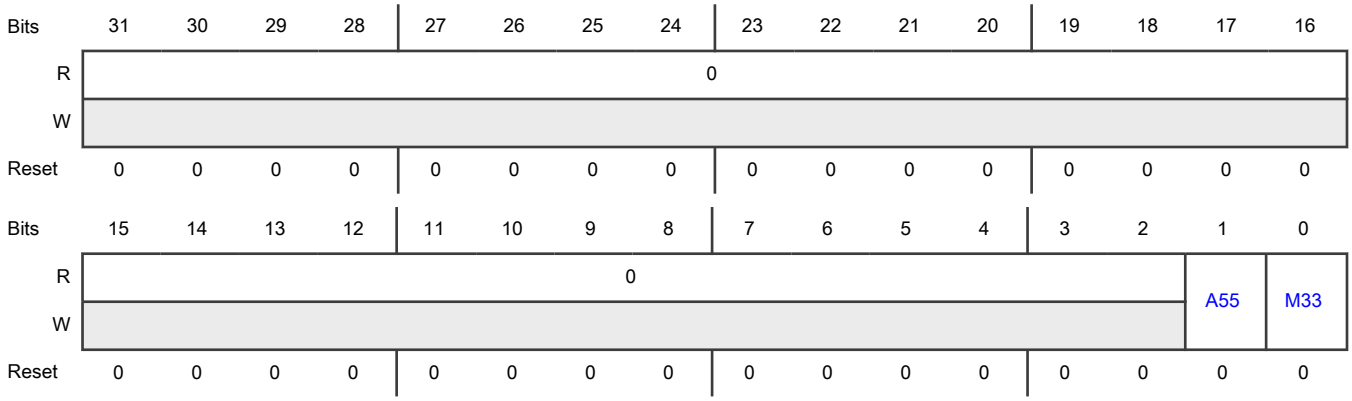
77.4.1.19 DAP Access Sticky (DAP_ACCESS_STKYBIT)

Write 1 Only

Offset

Register	Offset
DAP_ACCESS_STKYBIT	100h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 A55	A55 DAP_ACCESS_STKY 0 - A55 core can be accessed by DAP 1 - A55 core cannot be accessed by DAPCore0 works normally
0 M33	M33 DAP_ACCESS_STKY 0 - M33 core can be accessed by DAP 1 - M33 core cannot be accessed by DAPCore0 works normally

77.4.1.20 Low power handshake enable (LP_HANDSHAKE)

Offset

Register	Offset
LP_HANDSHAKE	110h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WAKEUPM...	FLEXSPI...	NIC_APB...	NIC_MED...	USB_CLK...	CM33_CL...	A55_CLK...	ELE_CLK...	ELE_CLK...	A55_CLK...	CM33_CL...	USB_CLK...	NIC_MED...	NIC_APB...	FLEXSPI...	WAKEUP_...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AON_TRD...	HSIOMIX...	MEDIAMI...	MEDIAMI...	NICMIX_...	NICMIX_...	WAKEUPM...	WAKEUPM...	AONMIX_...	CM33_PL...	CA55_PL...	CA55_PL...	CA55_CP...	CA55_CP...	CA55_CP...	CA55_CP...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 WAKEUPMIX_TRDC_CLK_ON_HS_EN	WAKEUPMIX_TRDC clock on handshake enable
30	FLEXSPI clock on handshake enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
FLEXSPI_CLK_ON_HS_EN	0 - Disable 1 - Enable
29 NIC_APB_CLK_ON_HS_EN	NIC_APB clock on handshake enable
28 NIC_MEDIA_CLK_ON_HS_EN	NIC_MEDIA clock on handshake enable 0 - Disable 1 - Enable
27 USB_CLK_ON_HS_EN	USB Controller clock on handshake enable
26 CM33_CLK_ON_HS_EN	CM33 clock on handshake enable 0 - Disable 1 - Enable
25 A55_CLK_ON_HS_EN	A55 clock on handshake enable
24 ELE_CLK_ON_HS_EN	Edgelock Enclave clock on handshake enable 0 - Disable 1 - Enable
23 ELE_CLK_OFF_HS_EN	Edgelock Enclave clock off handshake enable 0 - Disable 1 - Enable
22 A55_CLK_OFF_HS_EN	A55_CLK clock off handshake enable
21 CM33_CLK_OFF_HS_EN	CM33_CLK clock off handshake enable 0 - Disable 1 - Enable
20 USB_CLK_OFF_HS_EN	USB_CLK clock off handshake enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
19 NIC_MEDIA_CLK_OFF_HS_EN	NIC_MEDIA clock off handshake enable 0 - Disable 1 - Enable
18 NIC_APB_CLK_OFF_HS_EN	NIC_APB clock off handshake enable
17 FLEXSPI_CLK_OFF_HS_EN	FLEXSPI clock off handshake enable 0 - Disable 1 - Enable
16 WAKEUP_TRDC_CLK_OFF_HS_EN	WAKEUP TRDC clock off handshake enable Low power handshake gasket enable
15 AON_TRDC_CLK_OFF_HS_EN	AON TRDC clock off handshake enable Low power handshake gasket enable 0 - Disable 1 - Enable
14 HSIOMIX_COLD_RST_HS_EN	HSIOMIX cold reset handshake enable Low power handshake gasket enable
13 MEDIAMIX_LP_HS_EN	MEDIAMIX low power handshake enable Low power handshake gasket enable 0 - Disable 1 - Enable
12 MEDIAMIX_COLD_RST_HS_EN	MEDIAMIX cold reset handshake enable Low power handshake gasket enable
11 NICMIX_LP_HS_EN	NICMIX low power handshake enable Low power handshake gasket enable 0 - Disable 1 - Enable
10	NICMIX cold reset handshake enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
NICMIX_COLD_RST_HS_EN	Low power handshake gasket enable
9 WAKEUPMIX_LP_HS_EN	WAKEUPMIX low power handshake enable Low power handshake gasket enable 0 - Disable 1 - Enable
8 WAKEUPMIX_COLD_RST_HS_EN	WAKEUPMIX cold reset handshake enable Low power handshake gasket enable
7 AONMIX_COLD_RST_HS_EN	AONMIX cold reset handshake enable Low power handshake gasket enable 0 - Disable 1 - Enable
6 CM33_PLT_COLD_RST_HS_EN	CM33 platform cold reset handshake enable Low power handshake gasket enable
5 CA55_PLT_LP_HS_EN	CA55 platform cold reset handshake enable Low power handshake gasket enable 0 - Disable 1 - Enable
4 CA55_PLT_COLD_RST_HS_EN	CA55 platform cold reset handshake enable Low power handshake gasket enable
3 CA55_CPU1_LP_HS_EN	CA55_CPU1 low power handshake enable Low power handshake gasket enable 0 - Disable 1 - Enable
2 CA55_CPU1_COLD_RST_HS_EN	CA55_CPU1 cold reset handshake enable Low power handshake gasket enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
1 CA55_CPU0_LP_HS_EN	CA55_CPU0 low power handshake enable Low power handshake gasket enable 0 - Disable 1 - Enable
0 CA55_CPU0_COLD_RST_HS_EN	CA55_CPU0 cold reset handshake enable Low power handshake gasket enable 0 - Disable 1 - Enable

77.4.1.21 Low power handshake enable (LP_HANDSHAKE2)

Offset

Register	Offset
LP_HANDSHAKE2	114h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															ENABLE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

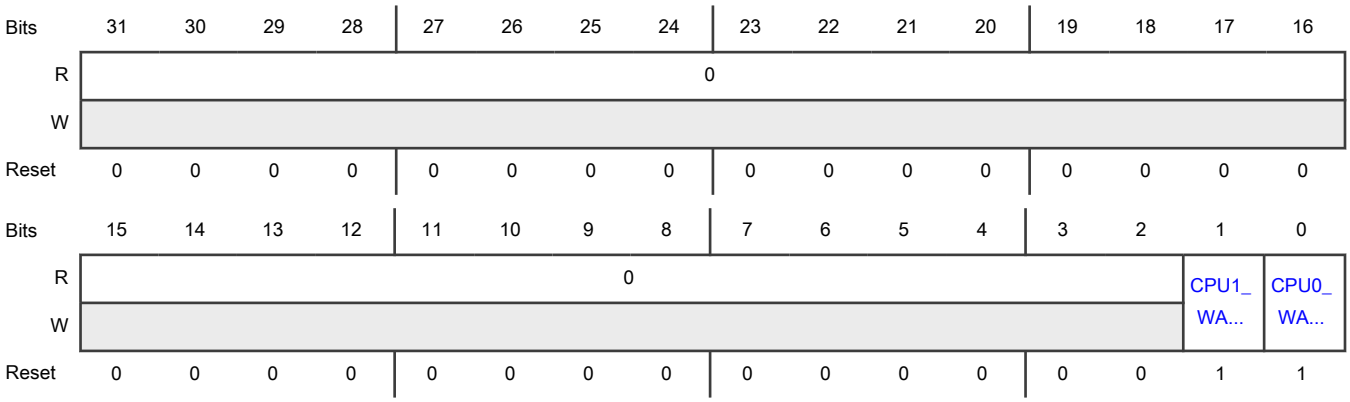
Field	Description
31-1 —	Reserved
0 ENABLE	AONMIX TRDC clock on handshake enable 0 - Disable 1 - Enable

77.4.1.22 CPUWAIT settings for CA55 CPU (CA55_CPUWAIT)

Offset

Register	Offset
CA55_CPUWAIT	118h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 CPU1_WAIT	CPU1_WAIT 0 - Core1 works normally 1 - Core1 stops working
0 CPU0_WAIT	CPU0_WAIT 0 - Core0 works normally 1 - Core0 stops working

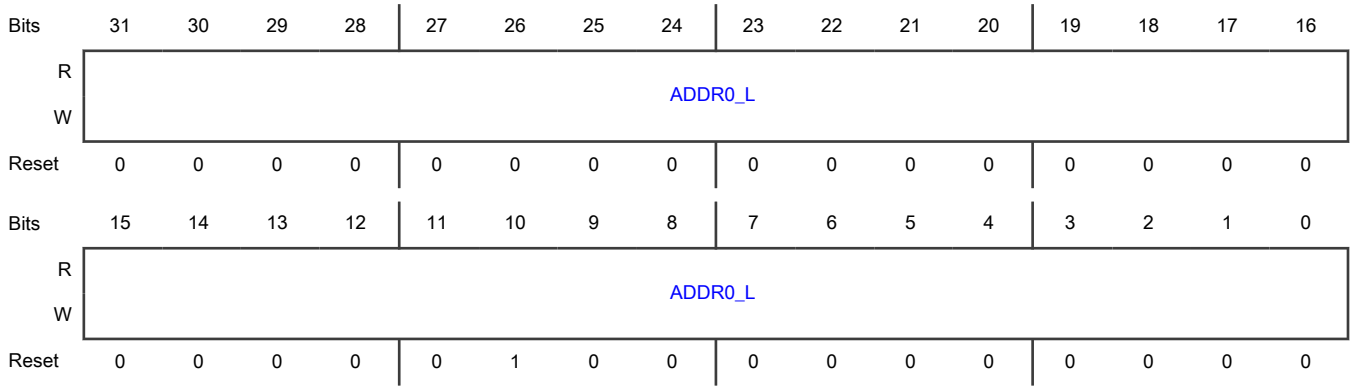
77.4.1.23 CA55 core0 lower boot address bits (CA55_RVBARADDR0_L)

Lower 32 bits of the boot address of CA55 core0 in 64bit mode

Offset

Register	Offset
CA55_RVBARADDR0_L	11Ch

Diagram



Fields

Field	Description
31-0 ADDR0_L	Lower 32 address bits

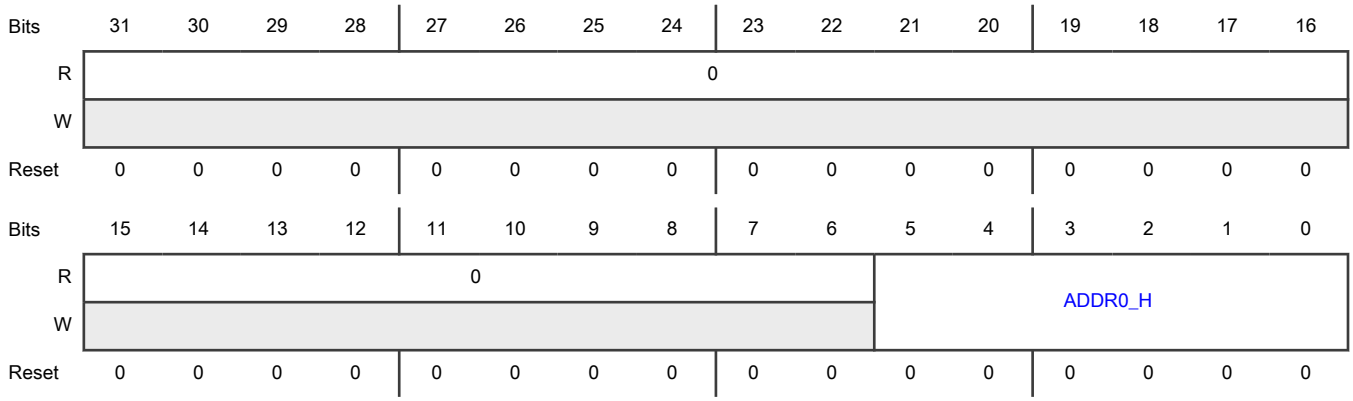
77.4.1.24 CA55 core0 higher boot address bits (CA55_RVBARADDR0_H)

Higher 6 bits of the boot address of CA55 core0 in 64bit mode

Offset

Register	Offset
CA55_RVBARADDR0_H	120h

Diagram



Fields

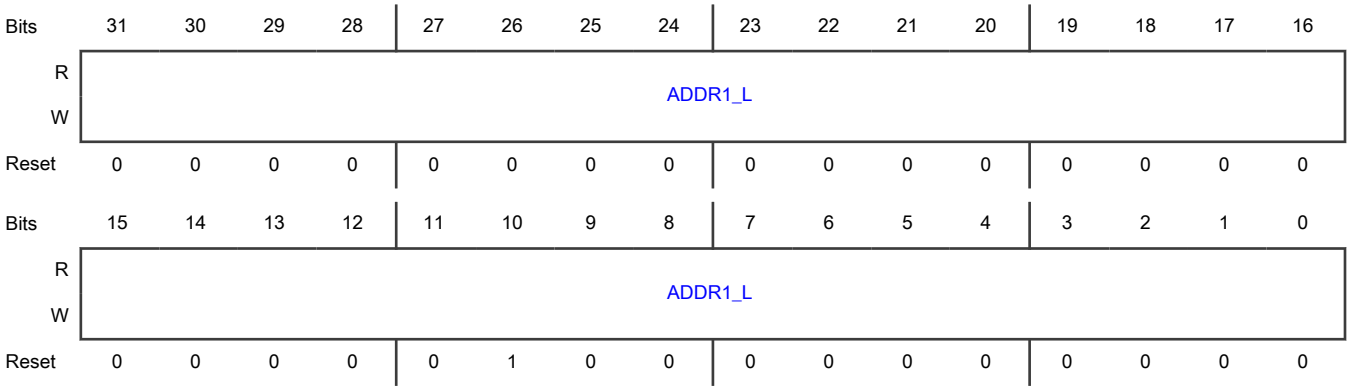
Field	Description
31-6 —	Reserved
5-0 ADDR0_H	Higher address bits Higher 6 bits of the boot address of CA55 core0 in 64bit mode

77.4.1.25 CA55 core1 lower boot address bits (CA55_RVBARADDR1_L)

Offset

Register	Offset
CA55_RVBARADDR1_L	124h

Diagram



Fields

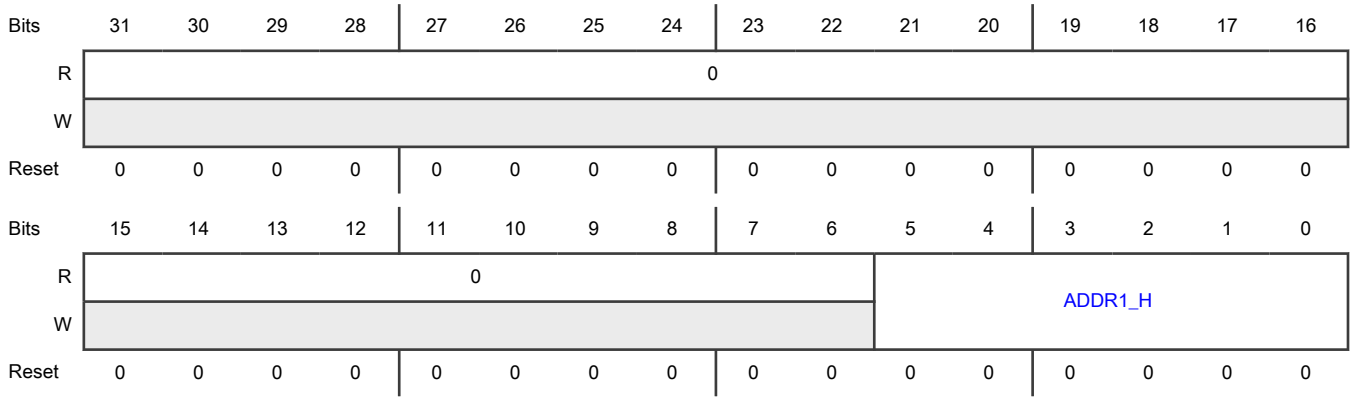
Field	Description
31-0 ADDR1_L	CA55_RVBARADDR1_L Lower 32 bits of the boot address of CA55 core1 in 64bit mode

77.4.1.26 CA55 core1 higher boot address bits (CA55_RVBARADDR1_H)

Offset

Register	Offset
CA55_RVBARADDR1_H	128h

Diagram



Fields

Field	Description
31-6 —	Reserved
5-0 ADDR1_H	CA55_RVBARADDR1_H Higher 6 bits of the boot address of CA55 core1 in 64bit mode

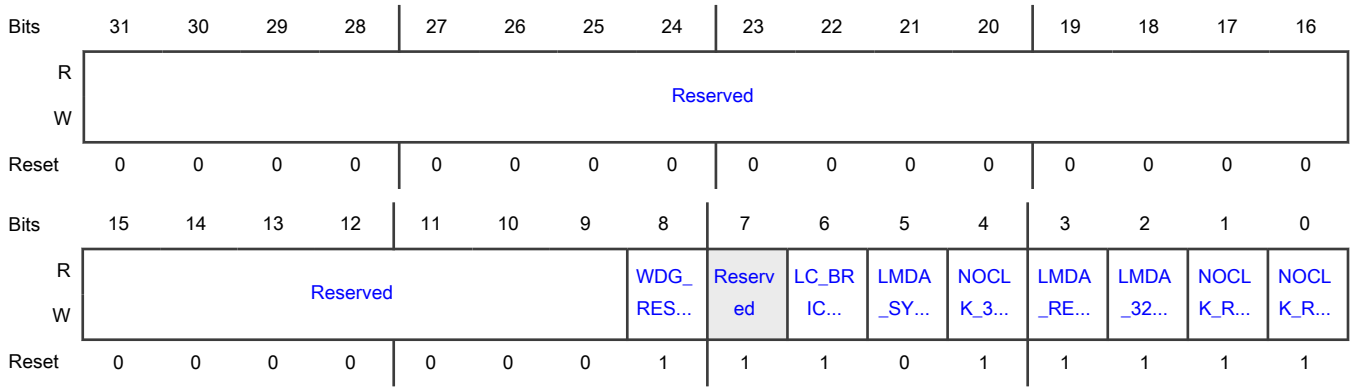
77.4.1.27 Mask bits of Edgelock interrupt (ELE_IRQ_MASK)

This register provides mask bits for the interrupt requests from EdgeLock to ARM core. When the mask bit is set 1 the corresponding interrupt will not be sent to ARM core.

Offset

Register	Offset
ELE_IRQ_MASK	12Ch

Diagram



Fields

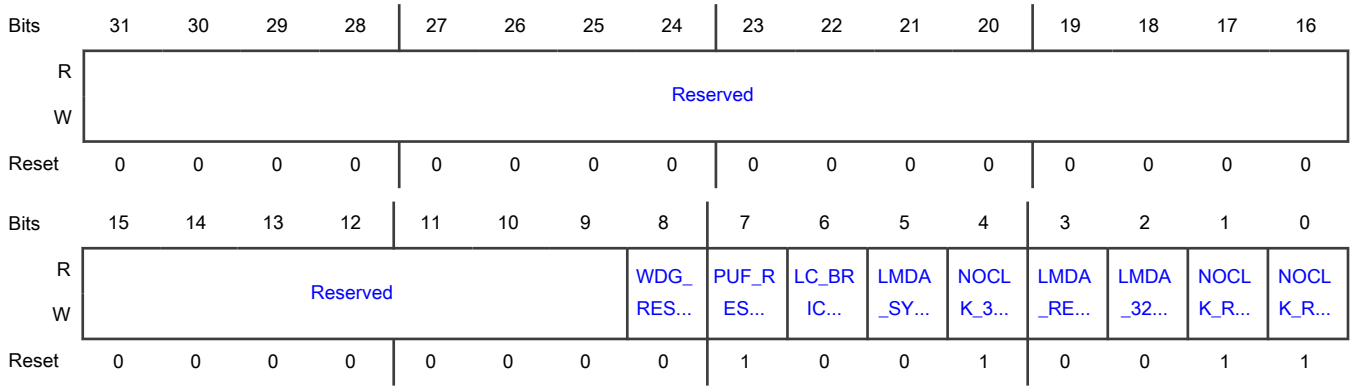
Field	Description
31-9 —	Reserved
8 WDG_RESET	Watchdog reset request
7 —	Reserved
6 LC_BRICKED	LMDA lifecycle is bricked state
5 LMDA_SYS_FAIL	System failure, reset chip or Edgelock 0 - Mask interrupt 1 - Unmask interrupt
4 NOCLK_32K	Edgelock FDET clock not detected interrupt mask
3 LMDA_RESET_REQ	Request Edgelock reset, active low, interrupt request 0 - Mask interrupt 1 - Unmask interrupt
2 LMDA_32K_RESET_REQ	Request Edgelock reset from 32 KHz clock domain, active low, interrupt request
1 NOCLK_REF1	No Clock Reference 1 0 - Mask interrupt 1 - Unmask interrupt
0 NOCLK_REF2	No Clock Reference 2 0 - Mask interrupt 1 - Unmask interrupt

77.4.1.28 Mask bits of ELE reset (ELE_RESET_REQ_MASK)

Offset

Register	Offset
ELE_RESET_REQ_MASK	130h

Diagram



Fields

Field	Description
31-9 —	Reserved
8 WDG_RESET	Watchdog reset request
7 PUF_RESET	PUF reset request 0 - Mask interrupt 1 - Unmask interrupt
6 LC_BRICKED	LMDA lifecycle is bricked state
5 LMDA_SYS_FAIL	System failure, reset chip or Edgelock 0 - Mask interrupt 1 - Unmask interrupt
4 NOCLK_32K	Edgelock FDET clock not detected interrupt mask
3 LMDA_RESET_REQ	Request Edgelock reset, active low, interrupt request 0 - Mask interrupt 1 - Unmask interrupt
2 LMDA_32K_RESET_REQ	Request Edgelock reset from 32 KHz clock domain, active low, interrupt request 0 - Mask interrupt 1 - Unmask interrupt
1	No Clock Reference 1

Table continues on the next page...

Table continued from the previous page...

Field	Description
NOCLK_REF1	
0 NOCLK_REF2	No Clock Reference 2 0 - Mask interrupt 1 - Unmask interrupt

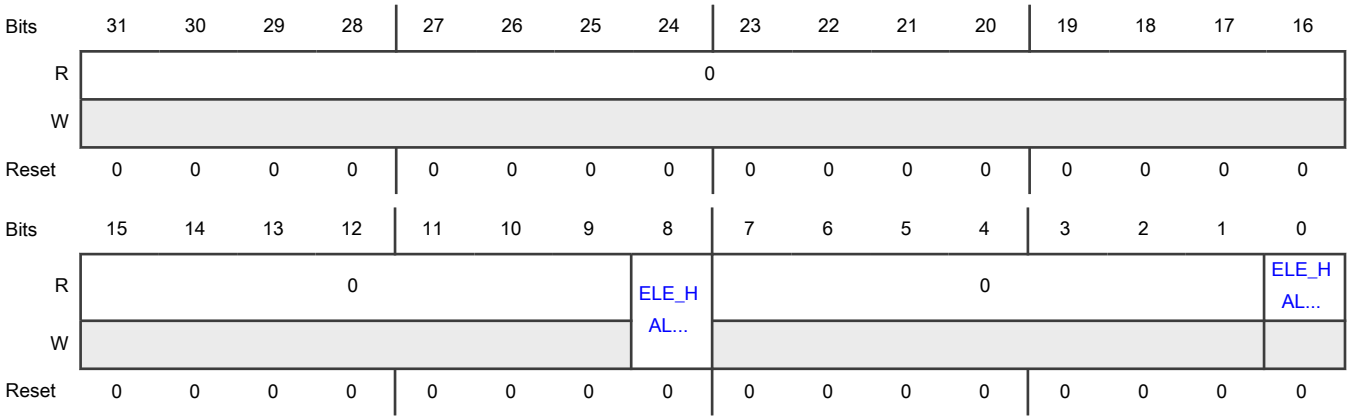
77.4.1.29 ELE halt status (ELE_HALT_ST)

To allow clock stopping for Edgelock, ELE_HALK_ACK bit need be checked high. When ELE need system to recover its clock, it will assert halt exit interrupt. ELE_HALT_EXIT_IRQ_CLR is used to clear this interrupt.

Offset

Register	Offset
ELE_HALT_ST	134h

Diagram



Fields

Field	Description
31-9 —	Reserved
8 ELE_HALT_EXIT_IRQ_CLR	EdgeLock halt exit interrupt clear 0 - Remove the clear signal. This bit is not self-clearing and need SW to clear. 1 - Clear EdgeLock halt exit interrupt
7-1	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
0 ELE_HALT_ACK	EdgeLock halt and clock status 0 - EdgeLock is not fully halted and its clocks must be enabled 1 - EdgeLock is fully halted indicating clocks may be removed

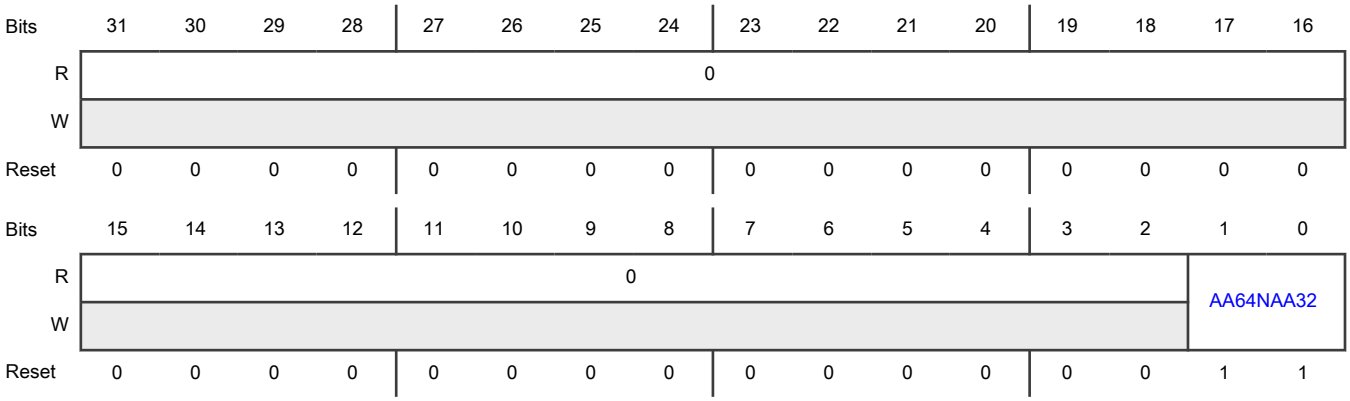
77.4.1.30 Control the boot mode of two ca55 cores (CA55_MODE)

Control whether the CA55 cores starts with 32bit mode or 64bit mode

Offset

Register	Offset
CA55_MODE	138h

Diagram



Fields

Field	Description
31-2 —	Reserved
1-0 AA64NAA32	Core0 initial mode control Control the core0 and core 1, 0: state after reset is aarch32 1: state after reset is aarch64 00 - Mask interrupt 01 - Unmask interrupt

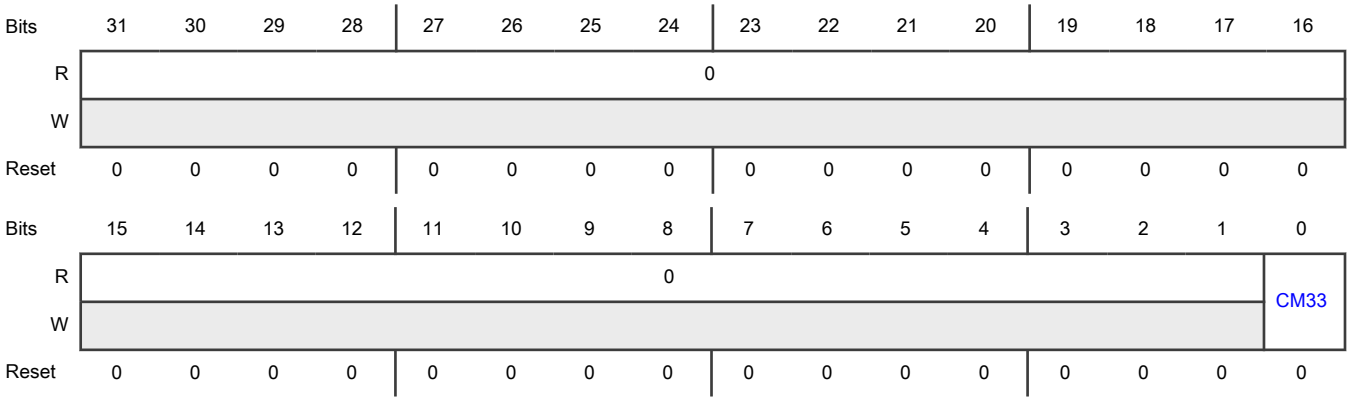
77.4.1.31 NMI MASK bits (NMI_MASK)

Control whether the NMI from pad is maksed

Offset

Register	Offset
NMI_MASK	13Ch

Diagram



Fields

Field	Description
31-1 —	Reserved
0 CM33	CM33 NMI mask 0 - NMI is usable 1 - NMI is masked

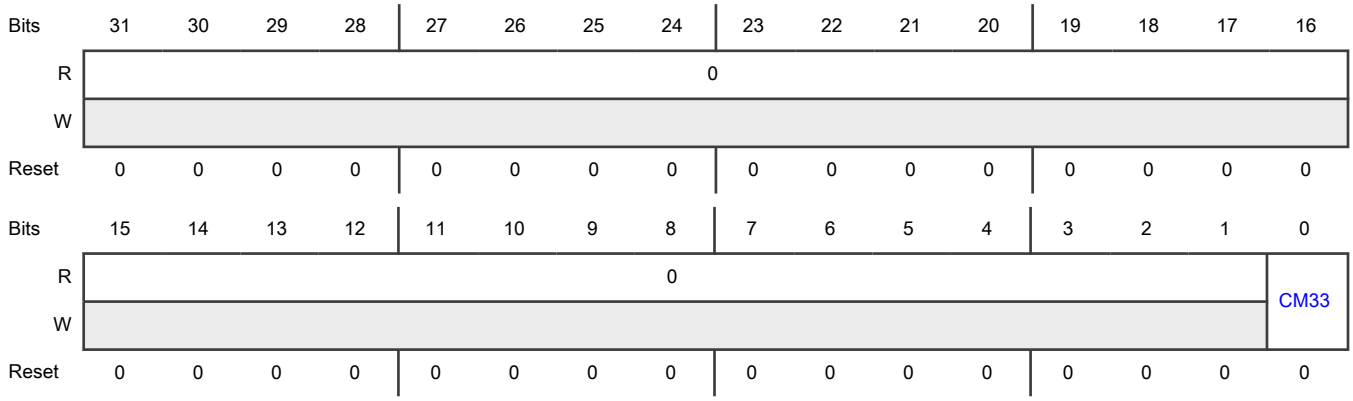
77.4.1.32 NMI clear (NMI_CLR)

Clear the internally registered NMI register

Offset

Register	Offset
NMI_CLR	140h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 CM33	NMI clear A high of this bit will clear the registered NMI bit

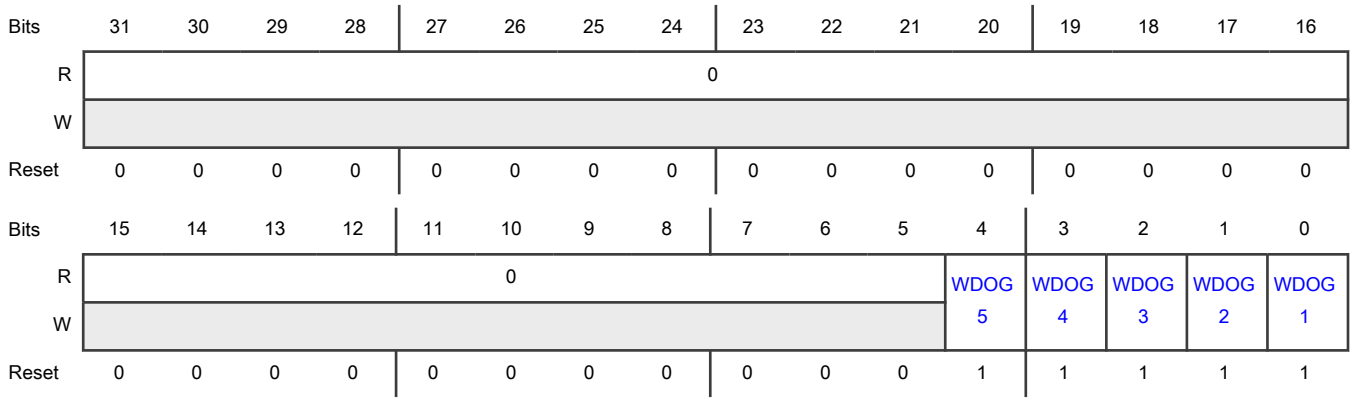
77.4.1.33 WDOG any mask (WDOG_ANY_MASK)

Mask the Watchdog timeout signals to the outside of the chip

Offset

Register	Offset
WDOG_ANY_MASK	144h

Diagram



Fields

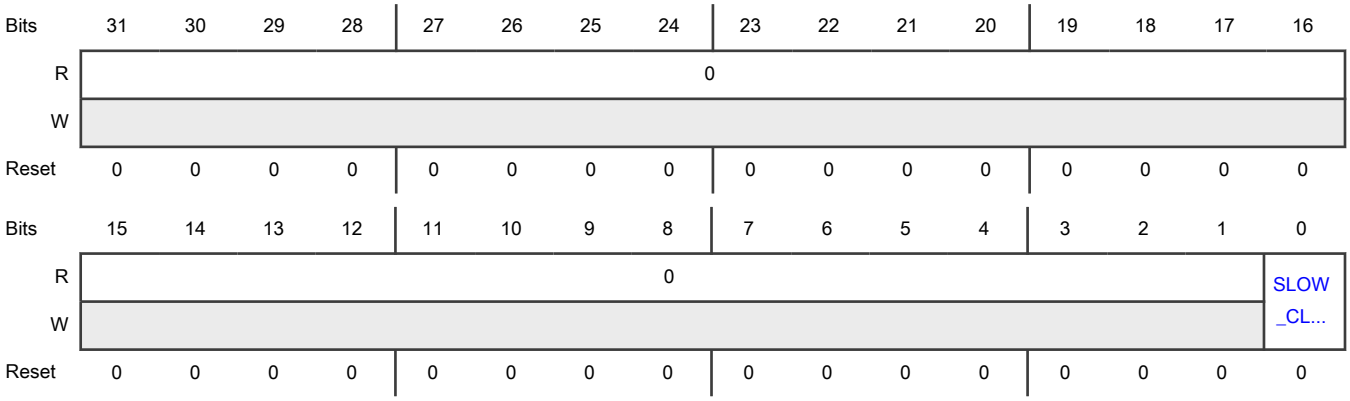
Field	Description
31-5 —	Reserved
4 WDOG5	WDOG5 to WDOG_ANY mask 0 - DISABLE 1 - ENABLE
3 WDOG4	WDOG4 to WDOG_ANY mask 0 - DISABLE 1 - ENABLE
2 WDOG3	WDOG3 to WDOG_ANY mask 0 - DISABLE 1 - ENABLE
1 WDOG2	WDOG2 to WDOG_ANY mask 0 - DISABLE 1 - ENABLE
0 WDOG1	WDOG1 to WDOG_ANY mask 0 - DISABLE 1 - ENABLE

77.4.1.34 ELEV1_IPI_NOCLK_REF1 clear (ELEV1_IPI_NOCLK_REF1)

Offset

Register	Offset
ELEV1_IPI_NOCLK_REF1	148h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 SLOW_CLEAR	Interrupt clear 0 - Interrupt not cleared 1 - Interrupt cleared

Chapter 78
Wake-up Domain Block Control (blk_ctrl_wakeupmix)

78.1 Introduction

The WAKEUPMIX domain contains control and status registers. These registers include upper address extension, debug mask bits, pad voltage status bits, and control registers for the modules in the WAKEUPMIX domain. These registers provide chip-level control outside the peripherals.

78.1.1 Block diagram

A block diagram of the Wakeup BLK_CTRL implementation is shown below.

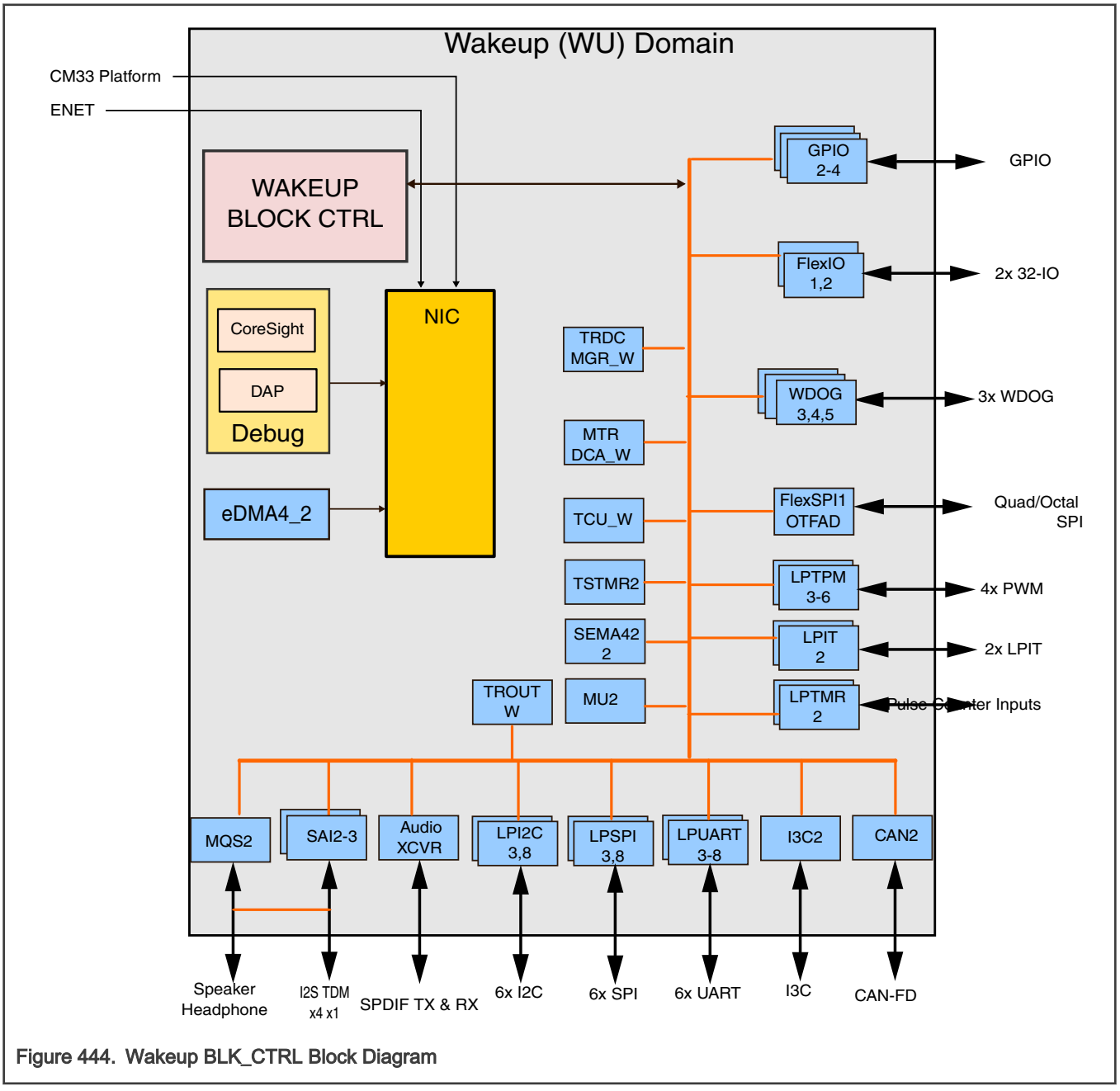


Figure 444. Wakeup BLK_CTRL Block Diagram

78.1.2 Features

The WAKEUP BLK_CTRL has the following features:

- Debug halt mask bits for peripherals in WAKEUP Domain
- WAKEUP SSI master block low power bits
- SAI2/3 MCLK direction control and MCLK source selection
- USBPHY asynchronous interrupt clear bit
- I3C2 wakeup clear bit
- I3C2 IO on-chip strong pull enable bit
- MQS settings
- Voltage detector output

78.2 Functional description

This block is a register container and doesn't have any function itself. The registers included are as below:

- Debug halt mask bits for peripherals in WAKEUP Domain
- WAKEUP SSI master block low power bits
- SAI2/3 MCLK direction control and MCLK source selection
- USBPHY asynchronous interrupt clear bit
- I3C2 wakeup clear bit
- I3C2 IO on-chip strong pull enable bit
- MQS settings
- Voltage detector output

78.2.1 Clocks

The table below describes the clock sources for Wakeup BLK_CTRL. Please see the Clock Control chapter for clock setting, configuration, and gating information.

Table 647. WAKEUPMIX clocks

Clock name	Description
wakeup_axi_clk_root	uSDHC1,2,3 NIC_mega NIC_main EDMA4_2 (AXI) FlexSPI1 (both its AHB and APB buses) SAI2, SAI3
bus_wakeup_clk_root	AIPS2 and its peripherals AIPS3 and its peripherals GPIO2, 3

78.2.2 Reset

The Wakeup BLK_CTRL resets with the Wakeup Domain.

78.2.3 Interrupts

Please refer to the individual peripheral chapters for detailed interrupt information.

78.3 Initialization

There are no initialization steps required.

78.4 Memory Map and register definition

This section includes the blk_ctrl_wakeupmix module memory map and detailed descriptions of all registers.

78.4.1 blk_ctrl_wakeupmix register descriptions

78.4.1.1 WAKEUPMIX Block Control memory map

BLK_CTRL_WAKEUPMIX base address: 4242_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	IPG DEBUG mask bit (IPG_DEBUG_CM33)	32	RW	0000_0000
10	QCHANNEL DISABLE (QCH_DIS)	32	RW	0000_03FF
1C	DEXSC error response configuration (DEXSC_ERR)	32	RW	0000_0000
20	MQS Settings for MQS2 (MQS_SETTING)	32	RW	0000_0000
24	SAI2 and SAI3 MCLK1~3 CLK root mux settings (SAI_CLK_SEL)	32	RW	0000_0000
28	ENET QOS control signals (GPR)	32	RW	0000_0000
2C	ENET CLK direction selection (ENET_CLK_SEL)	32	RW	0000_0003
34	Voltage detectors output (VOLT_DETECT)	32	R	0000_0000
38	I3C2 WAKEUPX CLR (I3C2_WAKEUP)	32	RW	0000_0000
3C	IPG DEBUG mask bit for CA55 core0 (IPG_DEBUG_CA55C0)	32	RW	0000_0000
40	IPG DEBUG mask bit for CA55 core1 (IPG_DEBUG_CA55C1)	32	RW	0000_0000
44	AXI CACHE OVERRITE BIT (AXI_ATTR_CFG)	32	RW	0000_0000
48	I3C SDA IRQ Control (I3C2_SDA_IRQ)	32	RW	0000_0000

78.4.1.2 IPG DEBUG mask bit (IPG_DEBUG_CM33)

Mask bit for HALTED from CM33 to peripheral ipg_debug

Offset

Register	Offset
IPG_DEBUG_CM33	4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			ENET1	SAI3	SAI2	I3C2	WDOG5	WDOG4	WDOG3	TPM6	TPM5	TPM4	TPM3	LPTMR2	LPSPI8
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LPSPI7	LPSPI6	LPSPI5	LPSPI4	LPSPI3	LPIT2	LPI2C8	LPI2C7	LPI2C6	LPI2C5	LPI2C4	LPI2C3	FLEXIO2	FLEXIO1	EDMA4_2	CAN2
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 ENET1	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
27 SAI3	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
26 SAI2	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
25 I3C2	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
24 WDOG5	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
23 WDOG4	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
22 WDOG3	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted

Table continues on the next page...

Table continued from the previous page...

Field	Description
21 TPM6	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
20 TPM5	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
19 TPM4	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
18 TPM3	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
17 LPTMR2	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
16 LPSPI8	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
15 LPSPI7	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
14 LPSPI6	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
13 LPSPI5	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
12 LPSPI4	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
11 LPSPI3	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
10 LPIT2	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
9 LPI2C8	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
8 LPI2C7	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
7	0 - Module does not enter debug halted mode with CM33

Table continues on the next page...

Table continued from the previous page...

Field	Description
LPI2C6	1 - Module enters debug halted mode when CM33 is debug halted
6 LPI2C5	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
5 LPI2C4	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
4 LPI2C3	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
3 FLEXIO2	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
2 FLEXIO1	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
1 EDMA4_2	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted
0 CAN2	0 - Module does not enter debug halted mode with CM33 1 - Module enters debug halted mode when CM33 is debug halted

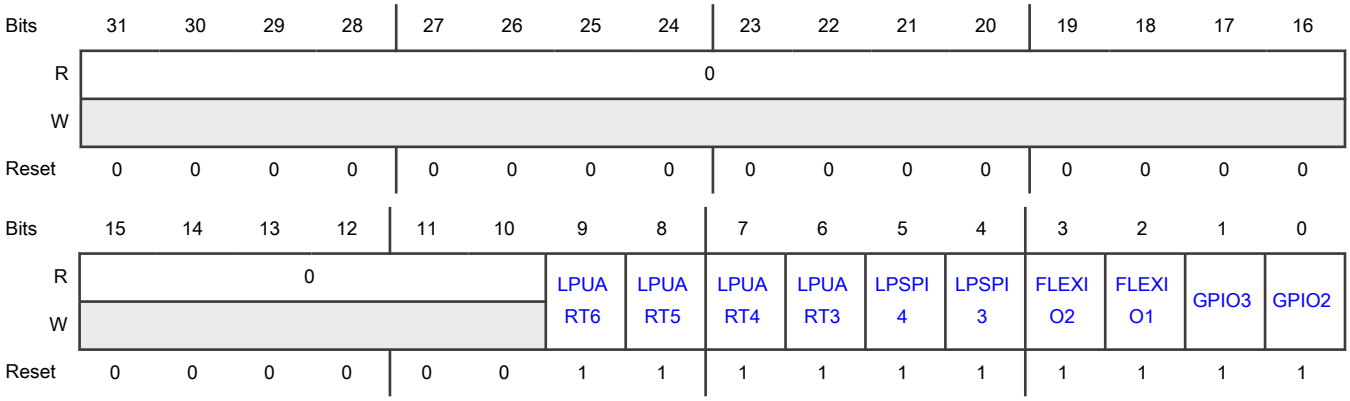
78.4.1.3 QCHANNEL DISABLE (QCH_DIS)

This register allows disabling QCHANNEL for modules.

Offset

Register	Offset
QCH_DIS	10h

Diagram



Fields

Field	Description
31-10 —	Reserved
9 LPUART6	LPUART6 QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled
8 LPUART5	LPUART5 QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled
7 LPUART4	LPUART4 QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled
6 LPUART3	LPUART3 QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled
5 LPSPI4	LPSPI4 QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled
4 LPSPI3	LPSPI3 QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled
3 FLEXIO2	FLEXIO2 QCHANNEL disable 0 - QCHANNEL is enabled

Table continues on the next page...

Table continued from the previous page...

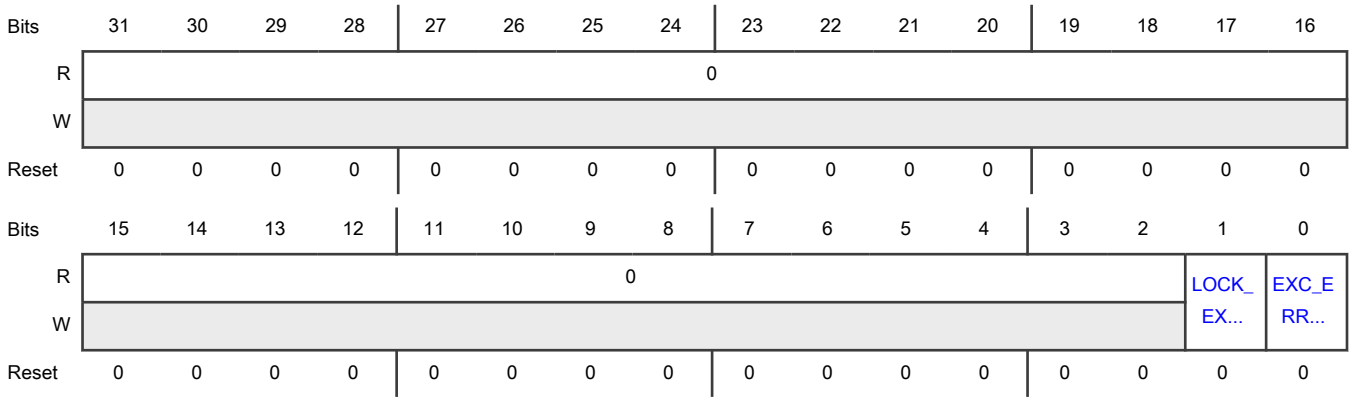
Field	Description
	1 - QCHANNEL is disabled
2 FLEXIO1	FLEXIO1 QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled
1 GPIO3	0 - QCHANNEL is enabled 1 - QCHANNEL is disabled
0 GPIO2	GPIO2 QCHANNEL disable 0 - QCHANNEL is enabled 1 - QCHANNEL is disabled

78.4.1.4 DEXSC error response configuration (DEXSC_ERR)

Offset

Register	Offset
DEXSC_ERR	1Ch

Diagram



Fields

Field	Description
31-2 —	Reserved
1	Lock bit of EXC_ERR_RESP_EN

Table continues on the next page...

Table continued from the previous page...

Field	Description
LOCK_EXC_ERR_RESP_EN	Once is it set to 1, EXC_ERR_RESP_EN is locked
0 EXC_ERR_RESP_EN	Exclusive error response enable This bit sets the bus response value when CM33 or CM33 exclusive access to CM33 TCM, OCRAM1 or OCRAM2 fails. 0 - OKAY response 1 - SLVError response

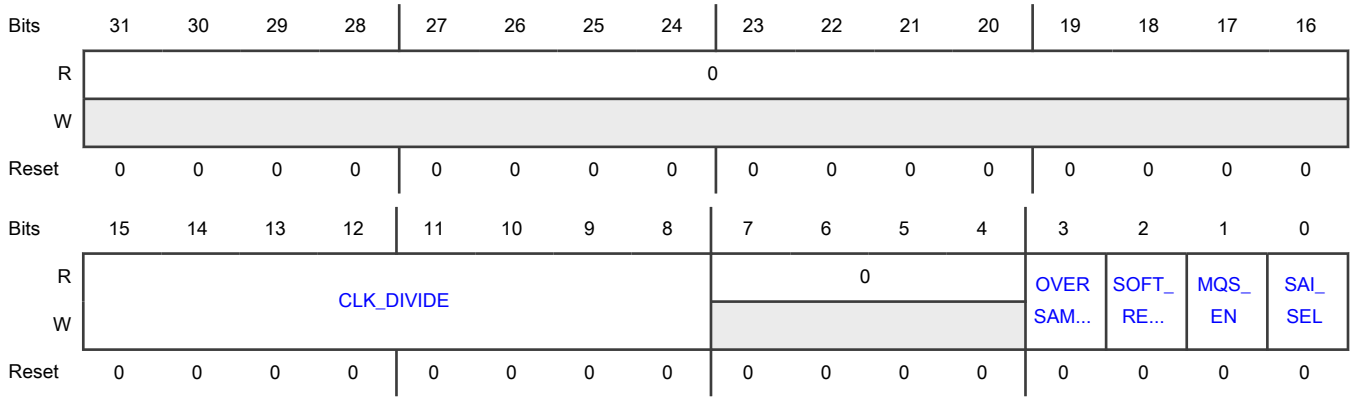
78.4.1.5 MQS Settings for MQS2 (MQS_SETTING)

This register includes several MQS settings: SAI selection, Enable, Software Reset, clock divide factor and Oversample.

Offset

Register	Offset
MQS_SETTING	20h

Diagram



Fields

Field	Description
31-16 —	Reserved
15-8 CLK_DIVIDE	Clock divide factor configuration MQS2 HMCLK switching between SAI2 (SAI_MCLK_OUT) and SAI3 (SAI_MCLK_OUT). 0000_0000 - hmclk frequency

Table continues on the next page...

Table continued from the previous page...

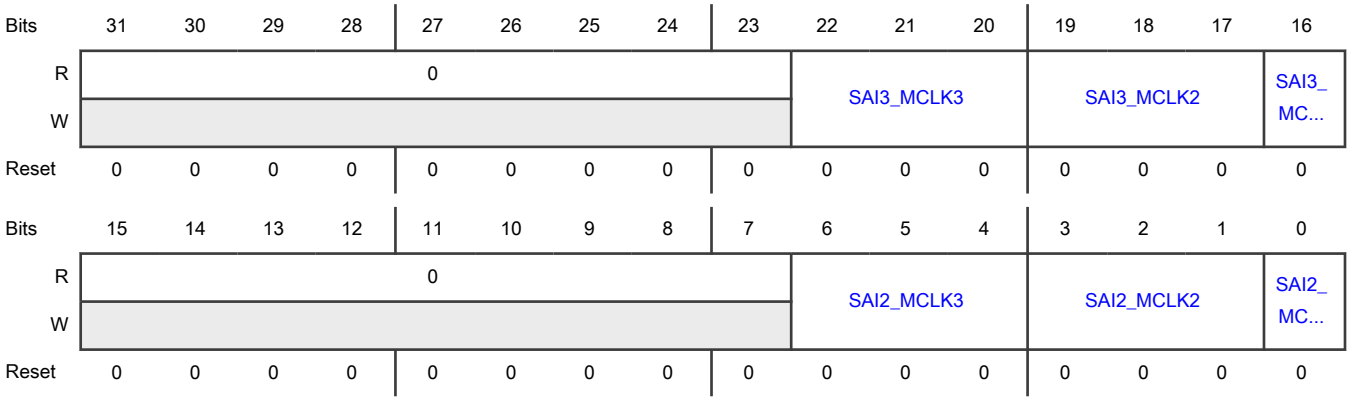
Field	Description
	0000_0001 - hmclk frequency x 1/2 0000_0010 - hmclk frequency x 1/3 0000_0011 - hmclk frequency x 1/4 0000_0100-1111_1111 - mclk frequency = hmclk frequency x 1/(n+1)
7-4 —	Reserved
3 OVERSAMPLE	Oversample enable 0 - 32 1 - 64
2 SOFT_RESET	Software Reset 0 - Reset is enabled 1 - Reset is disabled
1 MQS_EN	MQS Enable 0 - MQS2 is disabled 1 - MQS2 is enabled
0 SAI_SEL	SAI select MQS2 can be configured to come from lane 0 of either SAI2 or SAI3. 0 - SAI2 1 - SAI3

78.4.1.6 SAI2 and SAI3 MCLK1~3 CLK root mux settings (SAI_CLK_SEL)

Offset

Register	Offset
SAI_CLK_SEL	24h

Diagram



Fields

Field	Description
31-23 —	Reserved
22-20 SAI3_MCLK3	SAI3 MCLK3 clock source selection 000 - SAI1 CLK ROOT from CCM 001 - SAI2 CLK ROOT from CCM 010 - SAI3 CLK ROOT from CCM 011 - SAI1_MCLK from IOMUX 100 - SAI2_MCLK from IOMUX 101 - SAI3_MCLK from IOMUX 110 - SPDIF CLK ROOT from CCM 111 - SPDIF RX clock
19-17 SAI3_MCLK2	SAI2 MCLK2 clock source selection 000 - SAI1 CLK ROOT from CCM 001 - SAI2 CLK ROOT from CCM 010 - SAI3 CLK ROOT from CCM 011 - SAI1_MCLK from IOMUX 100 - SAI2_MCLK from IOMUX 101 - SAI3_MCLK from IOMUX 110 - SPDIF CLK ROOT from CCM 111 - SPDIF RX clock
16 SAI3_MCLK1	SAI3 MCLK1 clock source selection 0 - SAI3 CLK_ROOT from CCM 1 - SAI3 MCLK

Table continues on the next page...

Table continued from the previous page...

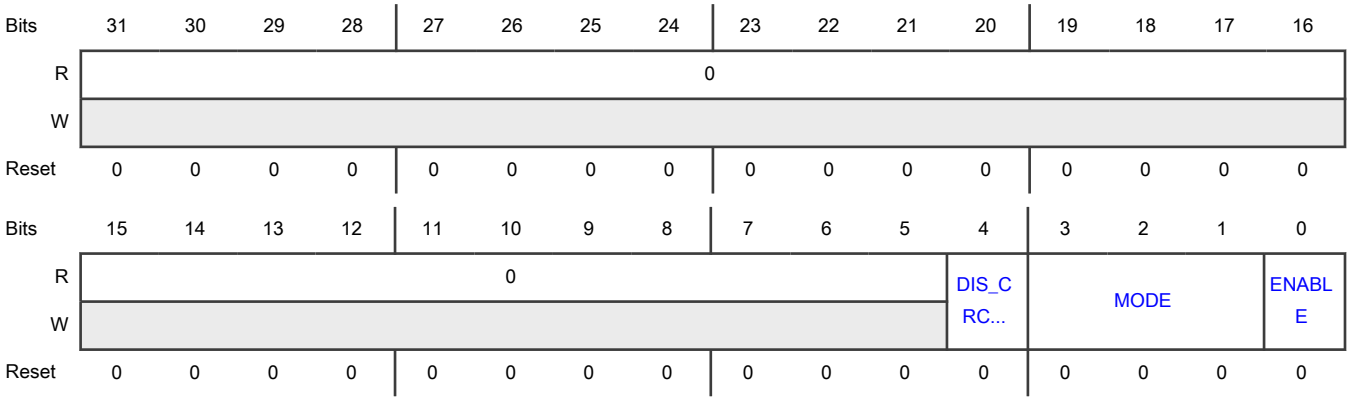
Field	Description
15-7 —	Reserved
6-4 SAI2_MCLK3	SAI2 MCLK3 clock source selection 000 - SAI1 CLK ROOT from CCM 001 - SAI2 CLK ROOT from CCM 010 - SAI3 CLK ROOT from CCM 011 - SAI1_MCLK from IOMUX 100 - SAI2_MCLK from IOMUX 101 - SAI3_MCLK from IOMUX 110 - SPDIF CLK ROOT from CCM 111 - SPDIF RX clock
3-1 SAI2_MCLK2	SAI2 MCLK2 clock source selection 000 - SAI1 CLK ROOT from CCM 001 - SAI2 CLK ROOT from CCM 010 - SAI3 CLK ROOT from CCM 011 - SAI1_MCLK from IOMUX 100 - SAI2_MCLK from IOMUX 101 - SAI3_MCLK from IOMUX 110 - SPDIF CLK ROOT from CCM 111 - SPDIF RX clock
0 SAI2_MCLK1	SAI2 MCLK1 clock source selection 0 - SAI2 CLK_ROOT from CCM 1 - SAI2 MCLK

78.4.1.7 ENET QOS control signals (GPR)

Offset

Register	Offset
GPR	28h

Diagram



Fields

Field	Description
31-5 —	Reserved
4 DIS_CRC_CHK	Disable ENET QoS CRC check 0 - CRC check is disabled 1 - CRC check is enabled
3-1 MODE	ENET QOS mode selection 000 - MII mode is selected 001 - RGMII mode is selected 100 - RMII mode is selected
0 ENABLE	ENET QOS enable 0 - ENET QoS is disabled 1 - ENET QoS is enabled

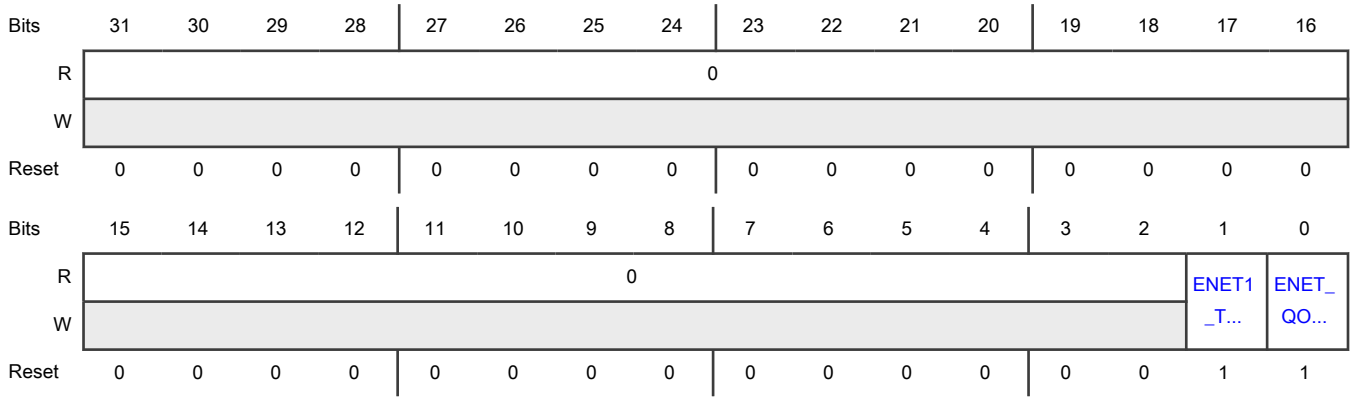
78.4.1.8 ENET CLK direction selection (ENET_CLK_SEL)

Control the direction of TX CLK of ENET and ENET_QOS

Offset

Register	Offset
ENET_CLK_SEL	2Ch

Diagram



Fields

Field	Description
31-2 —	Reserved
1 ENET1_TX_CLK_SEL	Direction of TX_CLK of ENET 0 - ENET TX CLK is Input 1 - ENET TX CLK is Output
0 ENET_QOS_CLK_TX_CLK_SEL	Direction of TX_CLK of ENET QOS 0 - ENET QOS TX CLK is Input 1 - ENET QOS TX CLK is Output

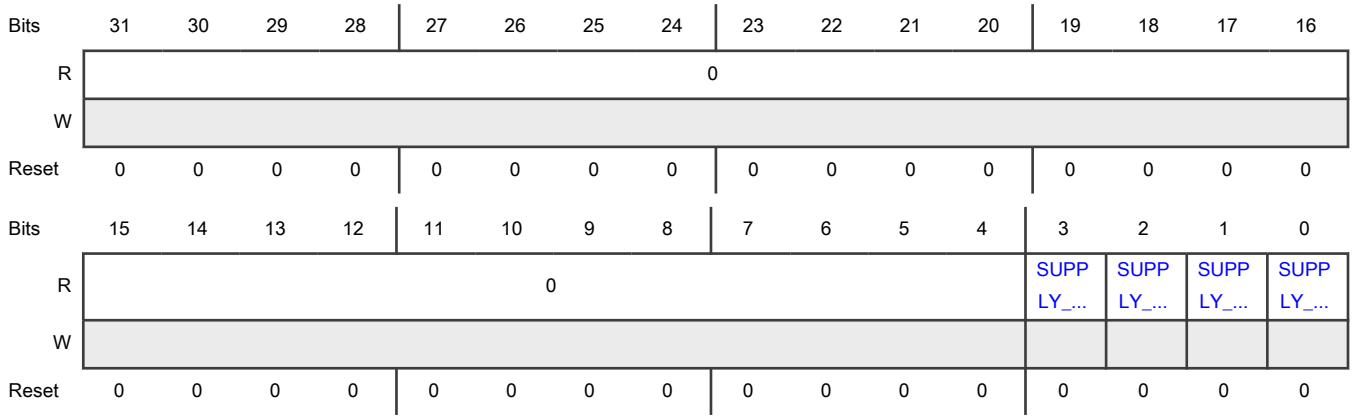
78.4.1.9 Voltage detectors output (VOLT_DETECT)

Read-only registers for the output of voltage detectors.

Offset

Register	Offset
VOLT_DETECT	34h

Diagram



Fields

Field	Description
31-4 —	Reserved
3 SUPPLY_DETECTOR_WAKEUP	Voltage detectors output of WAKEUP Read-only bit for the output of WAKEUP voltage detectors 0 - The input voltage of the domain is 1.8V 1 - The input voltage of the domain is 3.3V
2 SUPPLY_DETECTOR_SD2	Voltage detectors output of SD2 Read-only bit for the output of SD2 voltage detectors 0 - The input voltage of the domain is 1.8V 1 - The input voltage of the domain is 3.3V
1 SUPPLY_DETECTOR_GPIO	Voltage detectors output of GPIO Read-only bit for the output of GPIO voltage detectors 0 - The input voltage of the domain is 1.8V 1 - The input voltage of the domain is 3.3V
0 SUPPLY_DETECTOR_AON	Voltage detectors output of AON Read-only bit for the output of AON voltage detectors. 0 - The input voltage of the domain is 1.8V 1 - The input voltage of the domain is 3.3V

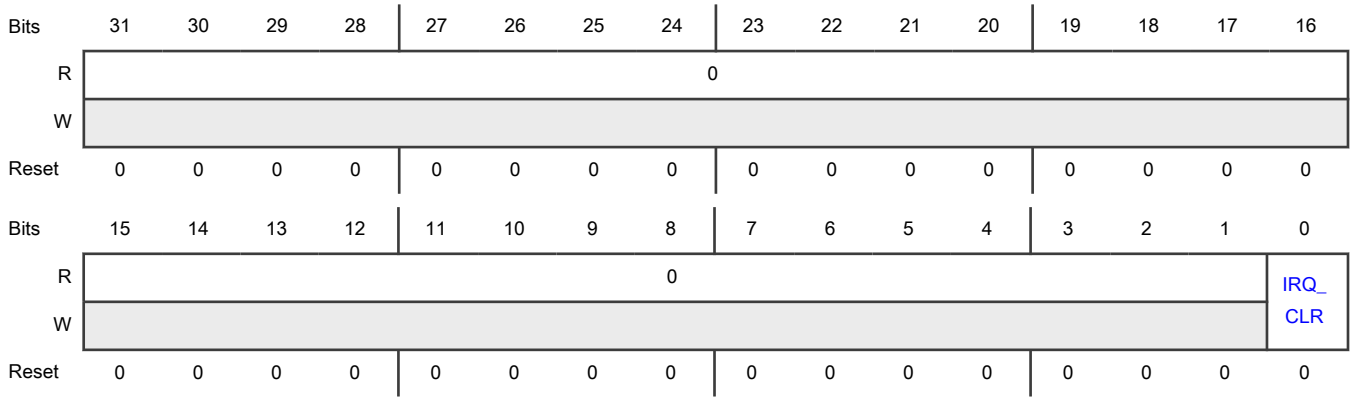
78.4.1.10 I3C2 WAKEUPX CLR (I3C2_WAKEUP)

Clear the interrupt of the I3C2 wakeup interrupt

Offset

Register	Offset
I3C2_WAKEUP	38h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 IRQ_CLR	

78.4.1.11 IPG DEBUG mask bit for CA55 core0 (IPG_DEBUG_CA55C0)

Mask bit for HALTED from CA55 core0 to peripheral ipg_debug 0: maske to 0 1: unmask

Offset

Register	Offset
IPG_DEBUG_CA55C0	3Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			ENET1	SAI3	SAI2	I3C2	WDOG5	WDOG4	WDOG3	TPM6	TPM5	TPM4	TPM3	LPTMR2	LPSPI8
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LPSPI7	LPSPI6	LPSPI5	LPSPI4	LPSPI3	LPIT2	LPI2C8	LPI2C7	LPI2C6	LPI2C5	LPI2C4	LPI2C3	FLEXIO2	FLEXIO1	EDMA4_2	CAN2
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 ENET1	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
27 SAI3	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
26 SAI2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
25 I3C2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
24 WDOG5	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
23 WDOG4	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
22 WDOG3	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
21 TPM6	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
20 TPM5	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted

Table continues on the next page...

Table continued from the previous page...

Field	Description
19 TPM4	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
18 TPM3	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
17 LPTMR2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
16 LPSPI8	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
15 LPSPI7	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
14 LPSPI6	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
13 LPSPI5	
12 LPSPI4	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
11 LPSPI3	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
10 LPIT2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
9 LPI2C8	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
8 LPI2C7	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
7 LPI2C6	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
6 LPI2C5	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
5	0 - Module does not enter debug halted mode with A55

Table continues on the next page...

Table continued from the previous page...

Field	Description
LPI2C4	1 - Module enters debug halted mode when A55 is debug halted
4 LPI2C3	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
3 FLEXIO2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
2 FLEXIO1	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
1 EDMA4_2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
0 CAN2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted

78.4.1.12 IPG DEBUG mask bit for CA55 core1 (IPG_DEBUG_CA55C1)

Mask bit for HALTED from CA55 core1 to peripheral

Offset

Register	Offset
IPG_DEBUG_CA55C1	40h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			ENET1	SAI3	SAI2	I3C2	WDOG5	WDOG4	WDOG3	TPM6	TPM5	TPM4	TPM3	LPTMR2	LPSPI8
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LPSPI7	LPSPI6	LPSPI5	LPSPI4	LPSPI3	LPIT2	LPI2C8	LPI2C7	LPI2C6	LPI2C5	LPI2C4	LPI2C3	FLEXIO2	FLEXIO1	EDMA4_2	CAN2
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-29 —	Reserved
28 ENET1	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
27 SAI3	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
26 SAI2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
25 I3C2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
24 WDOG5	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
23 WDOG4	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
22 WDOG3	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
21 TPM6	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
20 TPM5	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
19 TPM4	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
18 TPM3	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
17 LPTMR2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
16 LPSPI8	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
15	0 - Module does not enter debug halted mode with A55

Table continues on the next page...

Table continued from the previous page...

Field	Description
LPSPI7	1 - Module enters debug halted mode when A55 is debug halted
14 LPSPI6	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
13 LPSPI5	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
12 LPSPI4	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
11 LPSPI3	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
10 LPIT2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
9 LPI2C8	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
8 LPI2C7	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
7 LPI2C6	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
6 LPI2C5	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
5 LPI2C4	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
4 LPI2C3	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
3 FLEXIO2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
2 FLEXIO1	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted
1 EDMA4_2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 CAN2	0 - Module does not enter debug halted mode with A55 1 - Module enters debug halted mode when A55 is debug halted

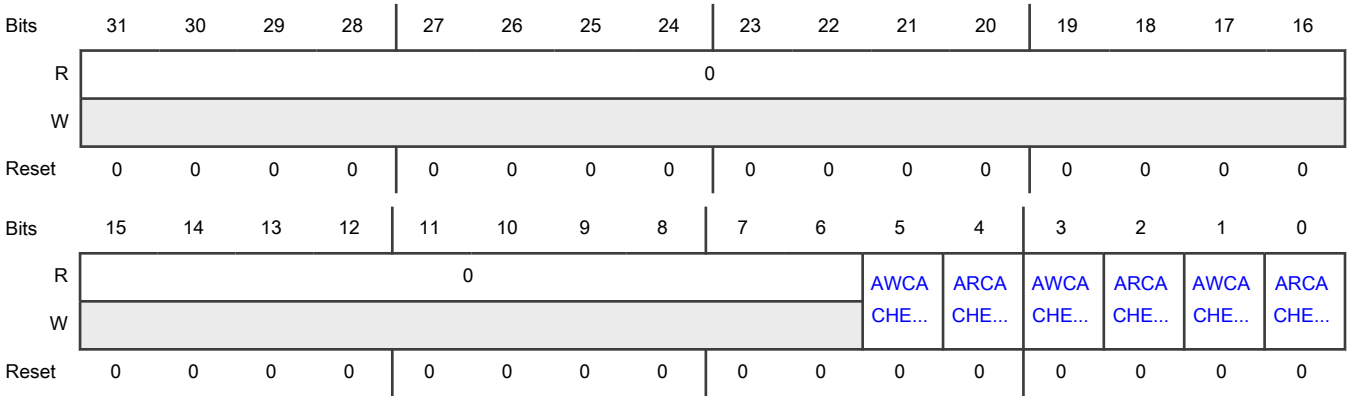
78.4.1.13 AXI CACHE OVERRITE BIT (AXI_ATTR_CFG)

Overwrite the cache[1] bit of the AXI protocol

Offset

Register	Offset
AXI_ATTR_CFG	44h

Diagram



Fields

Field	Description
31-6 —	Reserved
5 AWCACHE_USDHC3	Overwrite awcache of USDHC3
4 ARCACHE_USDHC3	Overwrite arcache of USDHC3
3	Overwrite awcache of USDHC2

Table continues on the next page...

Table continued from the previous page...

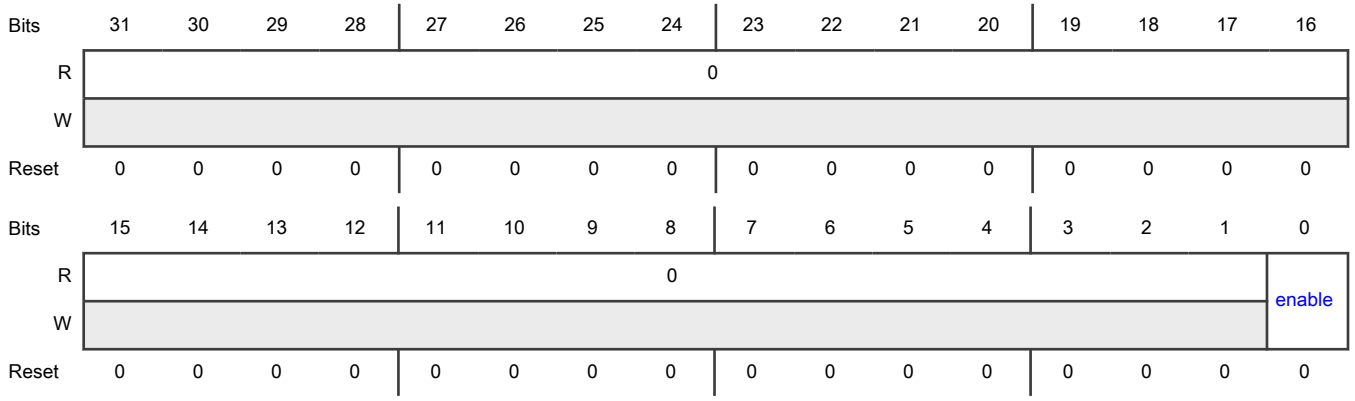
Field	Description
AWCACHE_USDHC2	
2 ARCACHE_USDHC2	Overwrite arcache of USDHC2
1 AWCACHE_USDHC1	Overwrite awcache of USDHC1
0 ARCACHE_USDHC1	Overwrite arcache of USDHC1

78.4.1.14 I3C SDA IRQ Control (I3C2_SDA_IRQ)

Offset

Register	Offset
I3C2_SDA_IRQ	48h

Diagram



Fields

Field	Description
31-1 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
0 enable	IRQ enable bit 0 - I3C2 SDA IRQ disable 1 - I3C2 SDA IRQ enable

Chapter 79
NIC Mix Block Control (blk_ctrl_nicmmix)

79.1 Introduction

The NICMIX domain includes configuration registers for DEXSC error response, QoS value from A55, MLMIX, AXI limit and HSIOMIX.

79.1.1 Block diagram

The figure below shows NICMIX.

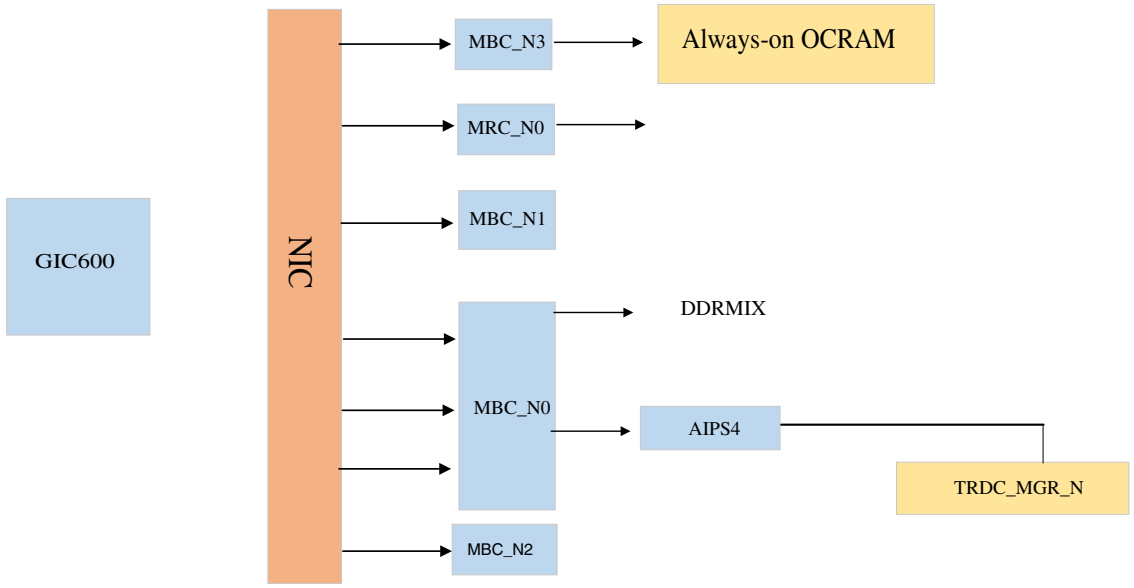


Figure 445. NICMIX block diagram

79.1.2 Features

- Configuration of DEXSC error response
- OCRAM RMW configuration
- WAKEUPMIX AXI limit configuration
- QoS Configuration from A55, MLMIX0, MLMIX1, HSIOMIX
- AxCACHE and AXI limit configuration

79.2 Functional description

The following sections describe functional details of the BLK CTRL NICMIX.

79.2.1 Clocks

The table found here describes the clock sources for BLK_CTRL_NS_AON. Please see the Clock Control chapter for clock setting, configuration, and gating information.

Clock Name	Description
nic_clk	NIC_Central GIC600 OCRAM
nic_apb_clk	AIPS4 Peripherals

79.2.2 Reset

BLK CTRL NICMIX resets with the NICMIX domain.

79.3 Initialization

There are no initialization steps required.

79.4 Memory Map and register definition

This section includes the BLK CTRL NICMIX module memory map and detailed descriptions of all registers.

NOTE

Write/Read to the Reserved locations will not generate a transfer error.

79.4.1 BLK CTRL NICMIX register descriptions

79.4.1.1 NICMIX Block Control memory map

These registers are GPRs(General Purpose Registers) of NICMIX. Please note that xfr_err cannot be generated for the module's illegal/Reserved accesses.

BLK_CTRL_NIC_WRAPPER1 base address: 4900_0000h

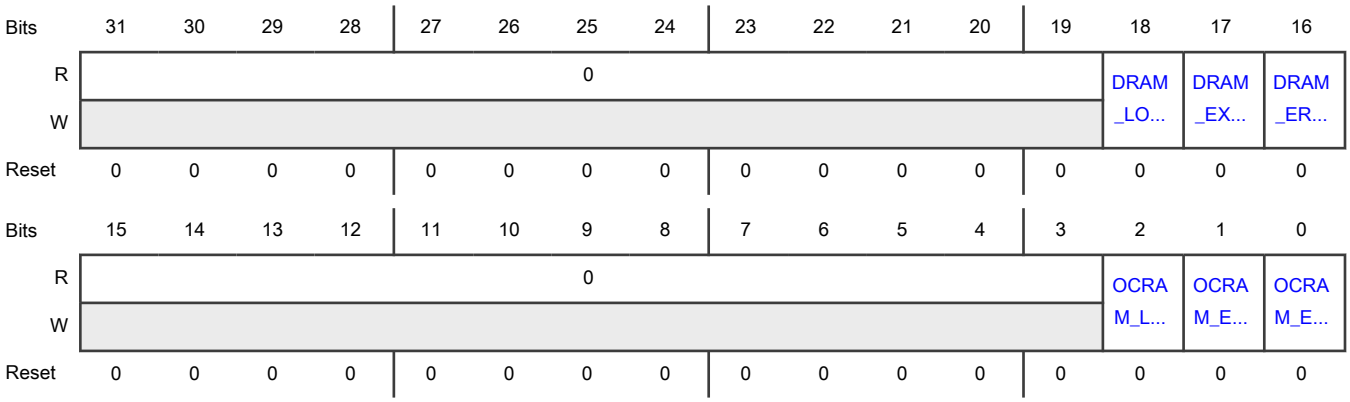
Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	DEXSC error response configuration (DEXSC_ERR)	32	RW	0000_0000
4	RMW configuration (OCRAM_RMW)	32	RW	0000_0000
8	Configuration register for axi_limit_wakeupmix (AXI_LIMIT_WAKEUPMIX)	32	RW	0000_0000
C	Configuration register for AxCACHE[1] override (CACHE_ATTR)	32	RW	003F_003F
10	Configuration register for QoS value from wakeupmix (WAKEUPMIX_QOS)	32	RW	0000_0000
14	Configuration register for QoS value from A55 (CACHE_QOS)	32	RW	0000_0000
18	Configuration register for QoS value from mlmix m0 (MLMIX0_QOS)	32	RW	0000_0000
1C	Configuration register for QoS value from mlmix m1 (MLMIX1_QOS)	32	RW	0000_0000
20	Configuration register for QoS value from hsiomix (HSIOMIX_QOS)	32	RW	0000_0000
24	GPR for uncertain tie0 or tie1 (TIE_VALUE)	32	RW	0000_000F

79.4.1.2 DEXSC error response configuration (DEXSC_ERR)

Offset

Register	Offset
DEXSC_ERR	0h

Diagram



Fields

Field	Description
31-19 —	Reserved
18 DRAM_LOCK_ERR_RESP_EN	Lock bit of DRAM_ERR_RESP_EN and DRAM_EXC_ERR_RESP_EN Once is it set to 1, DRAM_ERR_RESP_EN and DRAM_EXC_ERR_RESP_EN are locked
17 DRAM_EXC_ERR_RESP_EN	DRAM DEXSC exclusive error response enable 0 - DEXSC will not respond error if there is exclusive error on DEXSC 1 - DEXSC will respond slave error if there is exclusive error on DEXSC
16 DRAM_ERR_RESP_EN	DRAM DEXSC error response enable 0 - DEXSC will not respond error if there is read error on DEXSC 1 - DEXSC will respond slave error if there is read error on DEXSC
15-3 —	Reserved
2	Lock bit of OCRAM_ERR_RESP_EN and OCRAM_EXC_ERR_RESP_EN Once is it set to 1, OCRAM_ERR_RESP_EN and OCRAM_EXC_ERR_RESP_EN are locked

Table continues on the next page...

Table continued from the previous page...

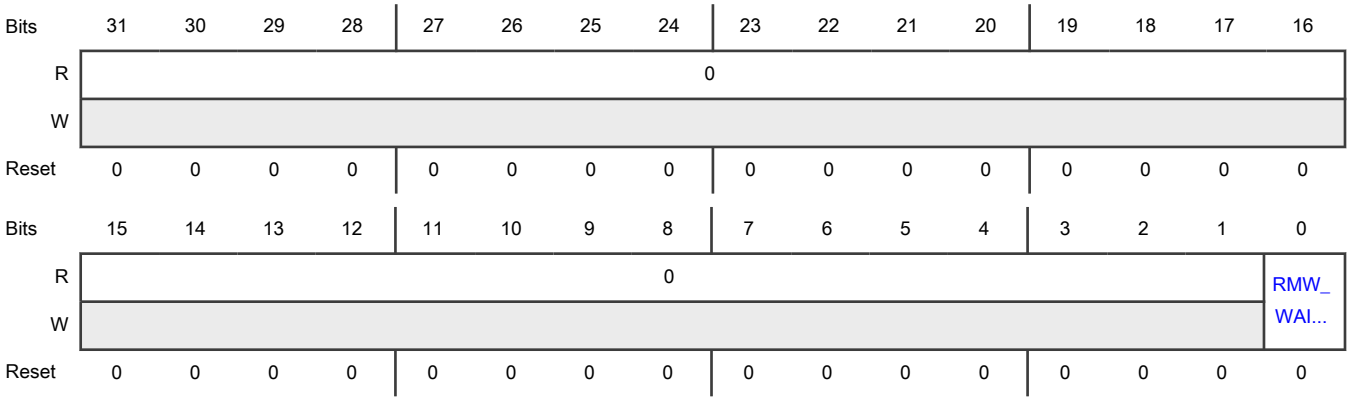
Field	Description
OCRAM_LOCK_ERR_RESP_EN	
1 OCRAM_EXC_ERR_RESP_EN	OCRAM DEXSC exclusive error response enable 0 - DEXSC will not respond error if there is exclusive error on DEXSC 1 - DEXSC will respond slave error if there is exclusive error on DEXSC
0 OCRAM_ERR_RESP_EN	OCRAM DEXSC error response enable 0 - DEXSC will not respond error if there is read error on DEXSC 1 - DEXSC will respond slave error if there is read error on DEXSC

79.4.1.3 RMW configuration (OCRAM_RMW)

Offset

Register	Offset
OCRAM_RMW	4h

Diagram



Fields

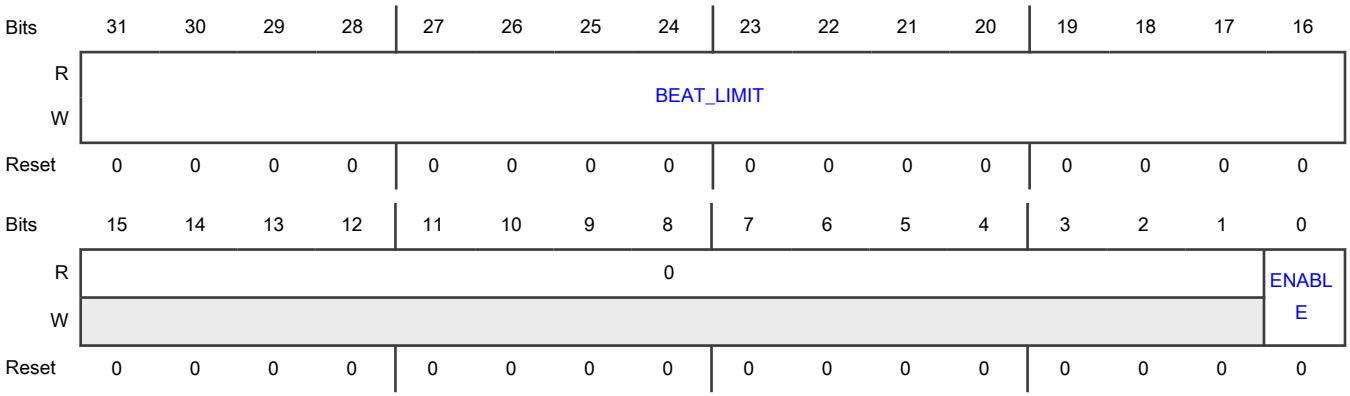
Field	Description
31-1 —	Reserved
0 RMW_WAIT_BVALID	Configure OCRAM RMW to wait for write response 0 - RMW will not wait for the write response and begin to work for another access 1 - RMW will not work for another access until it receives the write response of last access

79.4.1.4 Configuration register for axi_limit_wakeupmix (AXI_LIMIT_WAKEUPMIX)

Offset

Register	Offset
AXI_LIMIT_WAKEUPMIX	8h

Diagram



Fields

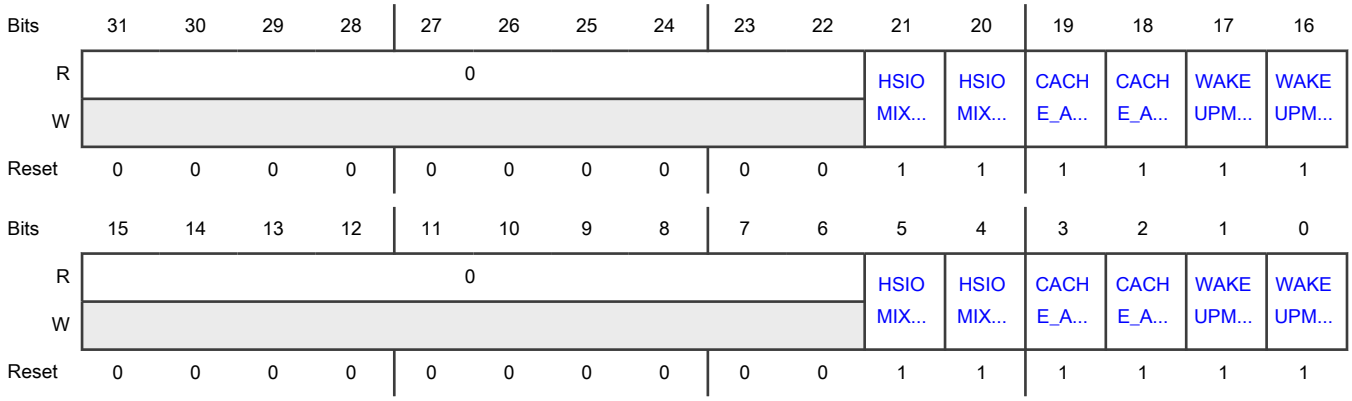
Field	Description
31-16 BEAT_LIMIT	Beat limit number
15-1 —	Reserved
0 ENABLE	Enable the beat limit on the access from wakeupmix 0 - Disable the beat limit 1 - Enable the beat limit

79.4.1.5 Configuration register for AxCACHE[1] override (CACHE_ATTR)

Offset

Register	Offset
CACHE_ATTR	Ch

Diagram



Fields

Field	Description
31-22 —	Reserved
21 HSIOMIX_AWCACHE	Override value 0 - Override AWCACHE[1] from ssi_slv_hsiomix to 0 1 - Override AWCACHE[1] from ssi_slv_hsiomix to 1
20 HSIOMIX_ARCACHE	Override value 0 - Override ARCACHE[1] from ssi_slv_hsiomix to 0 1 - Override ARCACHE[1] from ssi_slv_hsiomix to 1
19 CACHE_AWCACHE	Override value 0 - Override AWCACHE[1] from ssi_slv_cache to 0 1 - Override AWCACHE[1] from ssi_slv_cache to 1
18 CACHE_ARCACHE	Override value 0 - Override ARCACHE[1] from ssi_slv_cache to 0 1 - Override ARCACHE[1] from ssi_slv_cache to 1
17 WAKEUPMIX_AWCACHE	Override value 0 - Override AWCACHE[1] from ssi_slv_wakeupmix to 0 1 - Override AWCACHE[1] from ssi_slv_wakeupmix to 1
16 WAKEUPMIX_ARCACHE	Override value 0 - Override ARCACHE[1] from ssi_slv_wakeupmix to 0 1 - Override ARCACHE[1] from ssi_slv_wakeupmix to 1
15-6 —	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
5 HSIOMIX_AWCACHE_EN	Enable AWCACHE[1] override 0 - Disable override AWCACHE[1] from ssi_slv_hsiomix 1 - enable override AWCACHE[1] from ssi_slv_hsiomix
4 HSIOMIX_ARCACHE_EN	Enable ARCACHE[1] override 0 - Disable override ARCACHE[1] from ssi_slv_hsiomix 1 - enable override ARCACHE[1] from ssi_slv_hsiomix
3 CACHE_AWCACHE_EN	Enable AWCACHE[1] override 0 - Disable override AWCACHE[1] from ssi_slv_cache 1 - enable override AWCACHE[1] from ssi_slv_cache
2 CACHE_ARCACHE_EN	Enable ARCACHE[1] override 0 - Disable override ARCACHE[1] from ssi_slv_cache 1 - enable override ARCACHE[1] from ssi_slv_cache
1 WAKEUPMIX_AWCACHE_EN	Enable AWCACHE[1] override 0 - Disable override AWCACHE[1] from ssi_slv_wakeupmix 1 - enable override AWCACHE[1] from ssi_slv_wakeupmix
0 WAKEUPMIX_ARCACHE_EN	Enable ARCACHE[1] override 0 - Disable override ARCACHE[1] from ssi_slv_wakeupmix 1 - enable override ARCACHE[1] from ssi_slv_wakeupmix

79.4.1.6 Configuration register for QoS value from wakeupmix (WAKEUPMIX_QOS)

Offset

Register	Offset
WAKEUPMIX_QOS	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PANIC_AR				PANIC_AW			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-8 —	Reserved
7-4 PANIC_AR	Value of ar_qos_paNIC from wakeupmix
3-0 PANIC_AW	Value of aw_qos_paNIC from wakeupmix

79.4.1.7 Configuration register for QoS value from A55 (CACHE_QOS)

Offset

Register	Offset
CACHE_QOS	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								DEFAULT_AR				DEFAULT_AW			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PANIC_AR				PANIC_AW			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

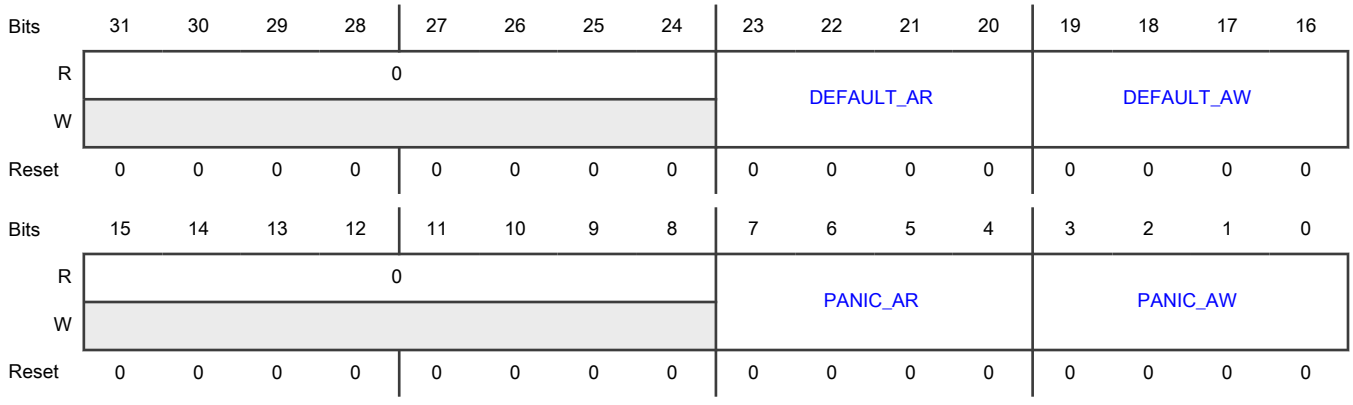
Field	Description
31-24 —	Reserved
23-20 DEFAULT_AR	Value of ar_qos_default from A55
19-16 DEFAULT_AW	Value of aw_qos_default from A55
15-8 —	Reserved
7-4 PANIC_AR	Value of ar_qos_paNIC from A55
3-0 PANIC_AW	Value of aw_qos_paNIC from A55

79.4.1.8 Configuration register for QoS value from mlmix m0 (MLMIX0_QOS)

Offset

Register	Offset
MLMIX0_QOS	18h

Diagram



Fields

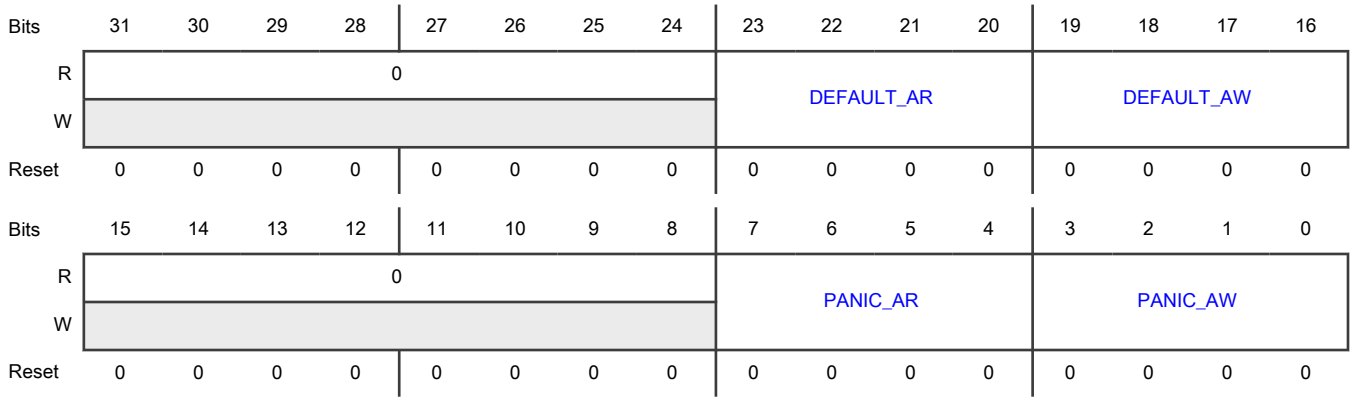
Field	Description
31-24 —	Reserved
23-20 DEFAULT_AR	Value of ar_qos_default from mlmix m0
19-16 DEFAULT_AW	Value of aw_qos_default from mlmix m0
15-8 —	Reserved
7-4 PANIC_AR	Value of ar_qos_paNIC from mlmix m0
3-0 PANIC_AW	Value of aw_qos_paNIC from mlmix m0

79.4.1.9 Configuration register for QoS value from mlmix m1 (MLMIX1_QOS)

Offset

Register	Offset
MLMIX1_QOS	1Ch

Diagram



Fields

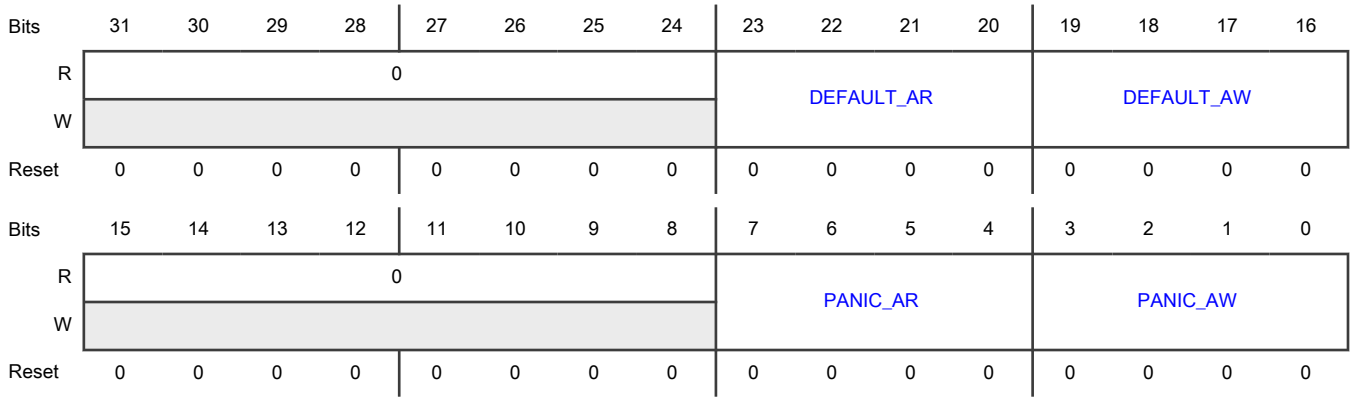
Field	Description
31-24 —	Reserved
23-20 DEFAULT_AR	Value of ar_qos_default from mlmix m1
19-16 DEFAULT_AW	Value of aw_qos_default from mlmix m1
15-8 —	Reserved
7-4 PANIC_AR	Value of ar_qos_paNIC from mlmix m1
3-0 PANIC_AW	Value of aw_qos_paNIC from mlmix m1

79.4.1.10 Configuration register for QoS value from hsiomix (HSIOMIX_QOS)

Offset

Register	Offset
HSIOMIX_QOS	20h

Diagram



Fields

Field	Description
31-24 —	Reserved
23-20 DEFAULT_AR	Value of ar_qos_default from hsiomix
19-16 DEFAULT_AW	Value of aw_qos_default from hsiomix
15-8 —	Reserved
7-4 PANIC_AR	Value of ar_qos_paNIC from hsiomix
3-0 PANIC_AW	Value of aw_qos_paNIC from hsiomix

79.4.1.11 GPR for uncertain tie0 or tie1 (TIE_VALUE)

Offset

Register	Offset
TIE_VALUE	24h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		DFTRAMH...		HPROT_GPV_CENTRAL				AWSNOOP_S				ARSNOOP_S			AWUSER_S
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AWUSER_S		ARUSER_S		AWBAR_S		ARBAR_S		GICP_AL...	GICT_AL...	XCPT_RTN		AIPS4_HBSTRB			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Fields

Field	Description
31-30 —	Reserved
29 DFTRAMHOLD	GIC600.dftramhold
28-25 HPROT_GPV_CENTRAL	NIC400_central.HPROT_gpv_central
24-22 AWSNOOP_S	GIC600.awsnoop_s
21-18 ARSNOOP_S	GIC600.arsnoop_s
17-15 AWUSER_S	GIC600.AWUSER_S
14-12 ARUSER_S	GIC600.ARUSER_S
11-10 AWBAR_S	GIC600.AWBAR_S
9-8 ARBAR_S	GIC600.ARBAR_S
7 GICP_ALLOW_NS	GIC600.gicp_allow_ns
6 GICT_ALLOW_NS	GIC600.gict_allow_ns
5-4 XCPT_RTN	Dac_cache_r.xcpt_rtn and dac_cache_w.xcpt_rtn
3-0 AIPS4_HBSTRB	Value of aips4.hbstrb

Chapter 80
ML Mix Domain Block Control (blk_ctrl_mlmix)

80.1 Overview

The MLMIX domain contains control and status registers known as the MLMIX Block Control (MLMIX BLK_CTRL). The register set has top-level controls for peripherals within the MLMIX domain. Please see the clock control chapter for more information regarding the clock input sources to the MLMIX.

80.1.1 Block diagram

The MLMIX BLK_CTRL connects to the MLMIX domain peripherals. It provides clock gating and power management control of the NPU.

The figure below shows the block diagram of the MLMIX BLK_CTRL.

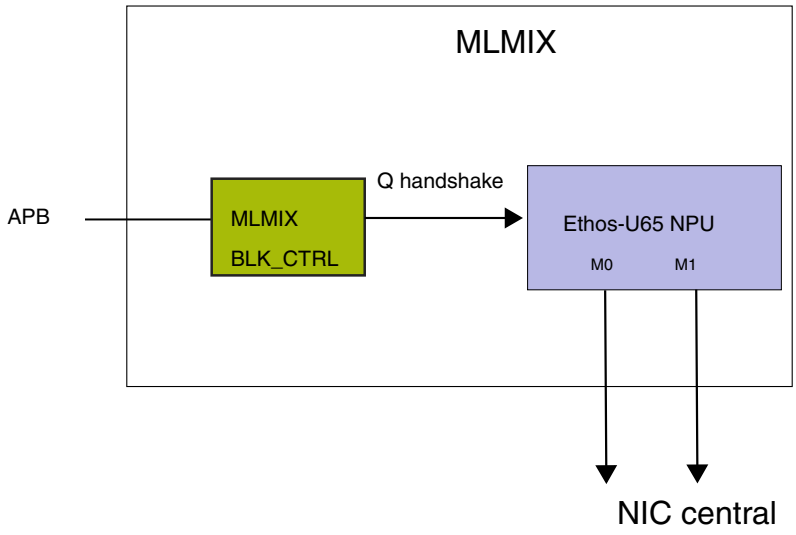


Figure 446. MLMIX BLK_CTRL block diagram

80.1.2 Features

The MLMIX BLK_CTRL includes a register to enable/disable the NPU's clock Q channel in the MLMIX domain.

80.2 Functional description

The following sections describe functional details of the MLMIX BLK_CTRL module.

80.2.1 Clocks

Refer to the NPU chapter for additional information on the clock Q-channel.

80.3 External signals

There are no external signals pinned out for the MLMIX BLK_CTRL.

80.4 Memory map and register definition

This section includes the blk_ctrl_mlmix module memory map and detailed descriptions of all registers.

80.4.1 BLK_CTRL_MLMIX register descriptions

80.4.1.1 MLMIX Block Control memory map

blk_ctrl_mlmix base address: 4A81_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Enable Q handshake module (HS_EN)	32	RW	0000_0000

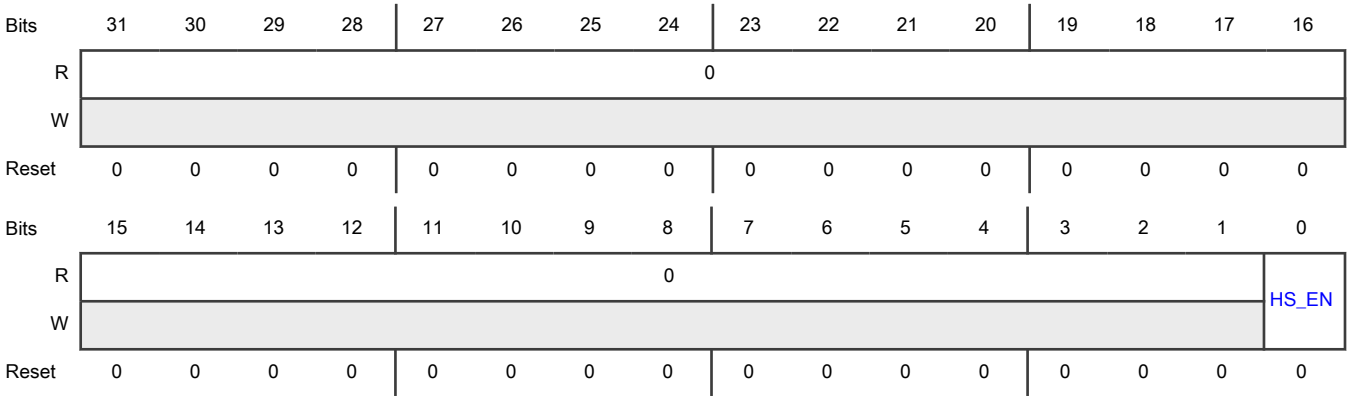
80.4.1.2 Enable Q handshake module (HS_EN)

Controls the NPU clock Q channel.

Offset

Register	Offset
HS_EN	0h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 HS_EN	Enable Q handshake module. The Q handshake module handles Low power Interface (Q-channel) signals from NPU to gate the clock, and therefore realize power management.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	0 - Q handshake module is disabled. Q channel signals from NPU will not be handled and the clock gate will not be controlled by NPU. 1 - Q handshake module is enabled. Clock is able to be gated by NPU through Q channel signals

Chapter 81
DDR Mix Domain Block Control (blk_ctrl_ddrmix)

81.1 Overview

The DDRMIX contains DDR Controller, LPDDR4/4x PHY and DRAM PLL.

It also contains control and status registers, that can be accessed by both secure and non-secure processors, known as blk_ctrl_ddrmix. These registers include Hardware Fast Frequency Change(HWFFC) control and DDRC stop signal control. There is also a register for PLL test observability. Please refer to the register description below for more information.

81.1.1 Block Diagram

This section includes the DDRMIX block diagram.

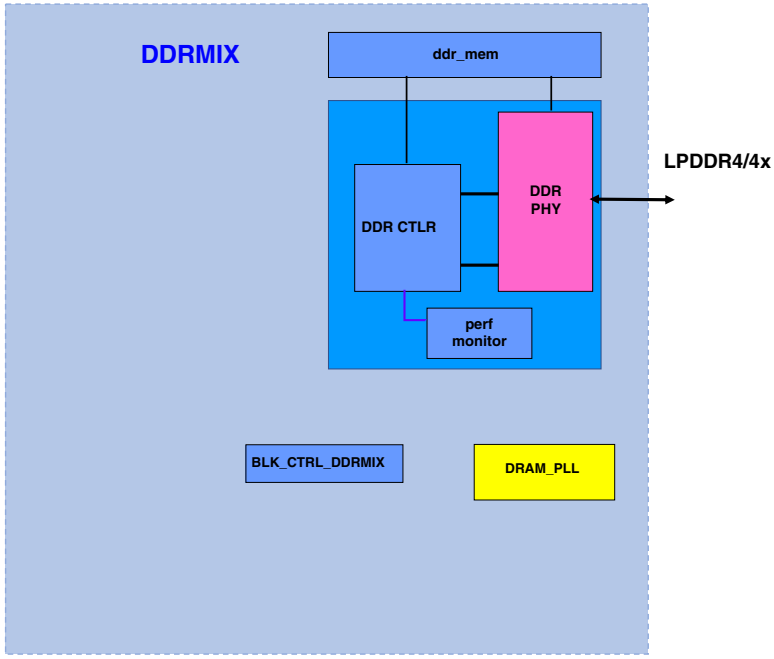


Figure 447. Block diagram

81.1.2 Features

Please refer to DDR Memory Controller Features chapter.

81.2 Functional Description

This section describes the functional details of DDRMIX.

81.2.1 Clocks

DRAM CLK is the source for DDRMIX.

81.2.2 Interrupts

Please refer to Interrupts and DMA Assignments Chapter.

81.3 HWFCC and SWFFC Procedure

The DDR Mix Domain and DDR Memory Controller (DDRC) allow for dynamic frequency scaling of the DDR frequency to save power during use cases where higher DDR performance is not required. There are two methods which can be utilized: Hardware Fast Frequency Change (HWFFC) and Software Fast Frequency Change (SWFFC).

HWFFC is utilized when switching from the maximum DDR frequency setpoint (full-speed mode) to half of the maximum frequency setpoint (half-speed mode). As the name implies, it uses the DDR Mix hardware mechanism to change DDR frequencies (between full-speed and half-speed mode) in a relatively short amount of time.

SWFFC is utilized when switching between frequencies that do not meet the half-frequency guideline used for HWFFC. SWFFC is especially useful when switching to very low DDR frequencies where minimum DDR performance is needed (for example, in an audio playback use case).

Both HWFFC and SWFFC can be used together when configuring the system for up to three frequency setpoints (also known as Frequency states). In this case, it is assumed that Frequency State 0 is configured for full-speed mode, Frequency State 1 for half-speed mode, and Frequency State 2 configured for a frequency much less than half the full-speed mode, for example 100 MT/s (50 MHz). In this scenario, HWFFC is used for switching between Frequency State 0 and Frequency State 1 and SWFCC used for switching between Frequency State 0 and Frequency State 2.

The following sections describe the flow for HWFCC and SWFFC. It is the responsibility of the user to decide which method is used and when to employ each method. In addition, the code location for implementing these procedures should reside in a memory other than the DRAM and in this case it is recommended to use the On Chip SRAM (OCRAM).

81.3.1 HWFFC Procedure

The following section describes the HWFFC flow. Before performing HWFFC, an initial setup is required during DRAM initialization and is outlined first. This is followed by the actual implementation of the HWFFC flow.

The following figure illustrates the HWFFC flow:

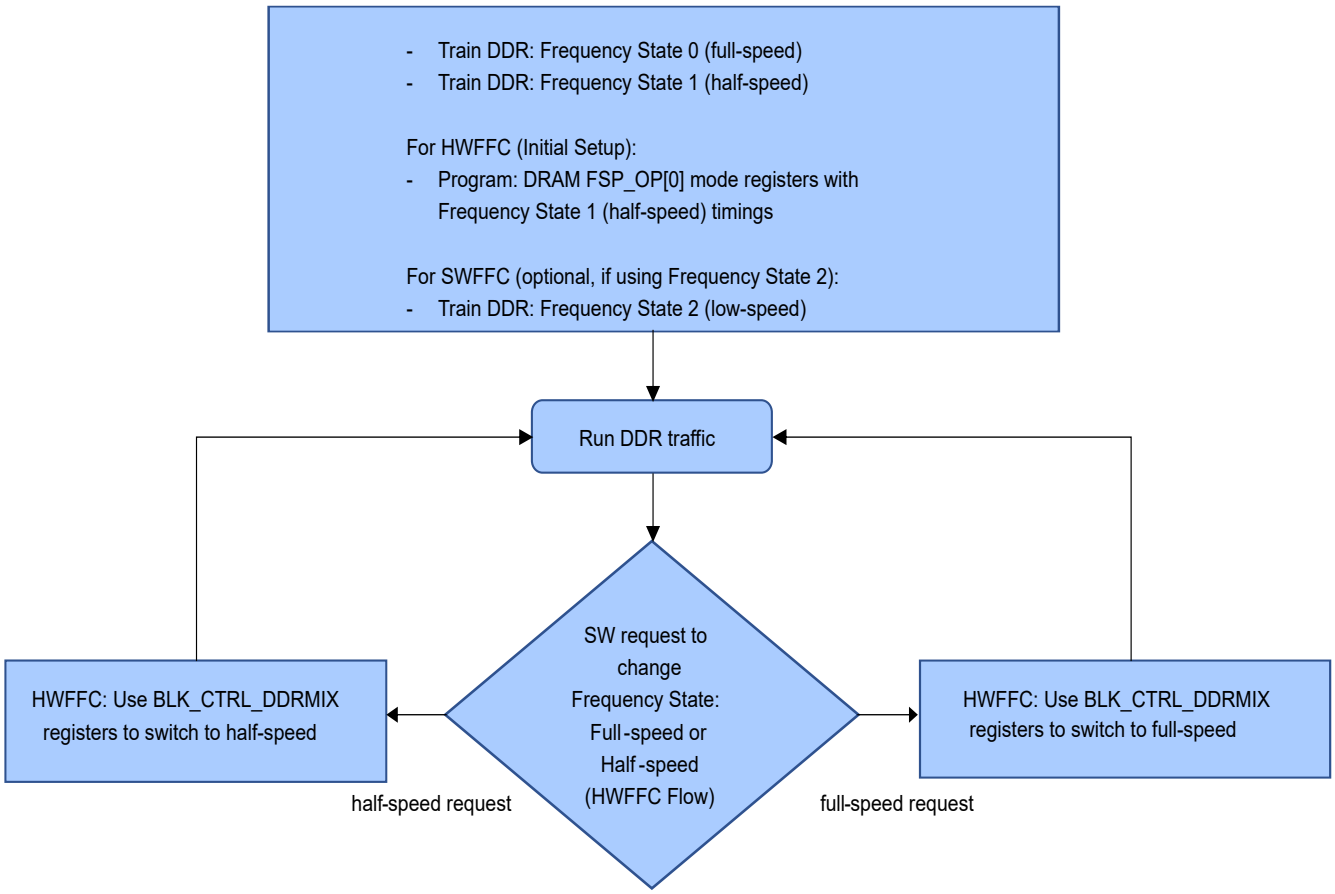


Figure 448. HWFFC Procedure

HWFFC Initial Setup:

1. HWFFC is supported between the full frequency (full-speed mode) and half frequency (half-speed mode) only. Note, not all the DDRC TIMING_CFG_12/13/14 registers for half-speed configurations can be programmed to the half-speed values. Instead, these should be programmed based on the guidelines below:

    a. Match the corresponding full-speed values for the TIMING_CFG_12[PRETOACT_HS], TIMING_CFG_12[ACTTOPRE_HS], TIMING_CFG_12[ACTTORW_HS], TIMING_CFG_13[WRREC_HS], TIMING_CFG_13[ACTTOACT_HS], TIMING_CFG_14[RD_TO_PRE_HS] register bits in DDRC. See DDR Memory Controller (DDRC) for more information.

    NOTE

    These bits are automatically configured by the NXP's DDR Config tool.

    b. Program per the desired half-speed values for the DDRC TIMING_CFG_12[CASLAT_HS], TIMING_CFG_13[REFREC_HS], TIMING_CFG_14[WRLAT_HS], TIMING_CFG_14[REFINT_HS] register bits in DDRC. See DDR Memory Controller (DDRC) for more information.

    NOTE

    These bits are automatically configured by the NXP's DDR Config tool.

    c. Note that the DDRC registers should not be accessed while operating at half-speed. A switch to full-speed is first required (for example, if an error register needs to be read).

2. When HWFFC is to be used, then train Frequency State 0 of the DDR PHY to full frequency (full-speed mode) and then train Frequency State 1 of the DDR PHY to half frequency (half-speed mode).
3. After 2D training is run, the DDR PHY will leave the DRAM in FSP-OP[1], and the PHY will be in Frequency State 0 (full frequency). At this point, the DDRC DDR_SDRAM_MD_CNTL can be used to perform mode register writes to program the FSP-OP[0] registers of the LPDDR4 DRAM to match the half-speed requirements (the DDR PHY does not do this automatically).
4. The following LPDDR4 registers have shadow copies for different frequency setpoints (FSPs): MR1, MR2, MR3, MR11, MR12, MR14, MR22.
5. After programming the DRAM's FSP-OP[0] registers, the DDRC is now ready for operation.

HWFFC Implementation (used when switching between full-speed and half-speed mode):

1. AUTO_CG_CTRL[AUTO_CG_ENA] bit in BLK_CTRL_DDRMIX should be cleared to disable DDR Controller automatic clock gating.
2. Set AUTO_CG_CTRL[HWFFC_ACG_FORCE_B] bit in BLK_CTRL_DDRMIX to 1. This bit will allow HWFFC sequence to launch ipg_stop to DDR Controller, which is necessary for HWFFC.
3. DDR_SDRAM_CFG[SREN] in DDRC should be set to 1 to enable self-refresh during sleep.
    a. This step should be performed in the DDR initialization by setting DDR_SDRAM_CFG[MEM_EN]
4. Set HWFFC_CTRL[HWFFC_EN] in BLK_CTRL_DDRMIX to 1 to enable DDR HWFFC. This step can be done together with Step 5.
5. Configure HWFFC_CTRL[HWFFC_SEL] to select the option of switching to full-speed or half-speed. HWFFC_CTRL[HWFFC_SEL]=0 configures switching to full speed and HWFFC_CTRL[HWFFC_SEL]=1 for half speed. This can be done together with Step 4.
6. Poll DDRC_STOP_CTRL[DDRC_STOP_ACK] bit in BLK_CTRL_DDRMIX until is returns 0.
7. Set DDRC_STOP_CTRL[DDRC_STOP] to 1.
8. Poll DDRC_STOP_CTRL[DDRC_STOP_ACK] until is returns 1.
9. Clear DDRC_STOP_CTRL[DDRC_STOP] to 0.
10. Additional steps when switching from half-speed to full-speed:
    a. If the DDRC DDR_SDRAM_CFG_3[DYN_REF_RATE_EN] = 1, then save this state into a variable and clear this bit.
    b. Perform a DDR mode register write to MR13 to 0xC0 to set the FSP-WR and FSP-OP to frequency setpoint 1.
    c. If the DDR_SDRAM_CFG_3[DYN_REF_RATE_EN] was previously set above, restore the state of this bit.
11. The AUTO_CG_CTRL[AUTO_CG_ENA] in BLK_CTRL_DDRMIX can be set to re-enable DDR Controller automatic clock gating.

The following steps needed to take the full system from a high performance mode to a low performance mode :

- SW detects event to switch from high to low perf/power mode (UD Mode).
- Enter low perf/power use case.
- Clock frequency reduction sequence
    1. Reduce all system clock (GPU, USB, and so on) frequencies except for DRAM, NoC, CPU clock frequency. This should take these components into low performance mode while DDR is still in high performance run.
    2. Reduce CPU clock frequency only IF needed. This is after #1, so CPU can still function at high performance while other components are reduced to UDM first.
    3. Reduce DRAM clock frequency to /2 using HWFFC. This is last so that DDR is still running at high performance during the switch to avoid BW issues.
    4. Reduce NOC clock frequency. Final step to complete LD mode entry, again to avoid any internal BW limitation that might cause an issue.

- Reduce voltage level to UD mode voltage specification for power savings.
- Commence with UD mode operation.

The following steps needed to take the full system from a low performance mode to a high performance mode :

- SW detects event to switch from low to high perf/power mode.
- Increase voltage level to high OD performance voltage level.
- Clock frequency increase sequence back to OD mode.
    1. Increase NOC clock frequency.
    2. Increase DRAM clock frequency using PLL post divider /1 and HWFFC.
    3. Increase the CPU clock frequency.
    4. Increase all remaining system clock frequencies (GPU, etc). Enter high perf/power use case. Commence with high performance use case.

81.3.2 SWFFC Procedure

The following section describes the SWFFC flow. The following figure illustrates the SWFFC flow:

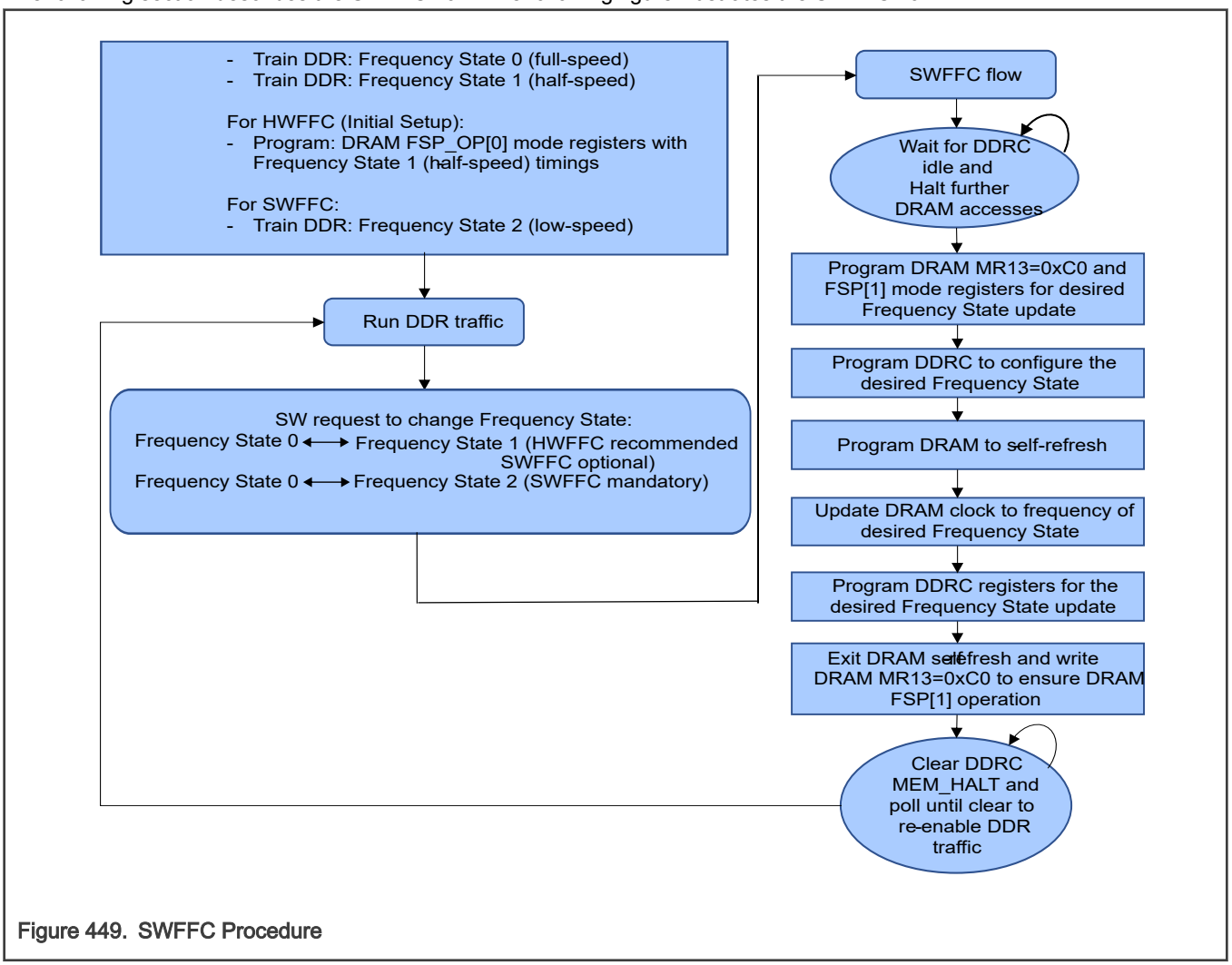


Figure 449. SWFFC Procedure

The following are the detailed steps for the SWFFC flow:

1. Prior to implementing the SWFFC procedure, if both HWFFC and SWFFC are used, then train Frequency State 2 (the third and lowest frequency setpoint) of the DDR PHY to the SWFFC Frequency State 2 frequency (Frequency State 1 can also be used for SWFFC for half-frequency switching, even though it meets the requirements for HWFFC).
2. In the DDRC, the register bit settings for DDR_SDRAM_CFG_3[SR_FAST_WK_EN] and [DYN_REF_RATE_EN] should be cleared. In BLK_CTRL_DDRMIX, AUTO_CG_CTRL[AUTO_CG_ENA] should be cleared to disable auto clock gating (this step can be moved to step 1 if desired).
    a. If inline ECC is enabled (global or per region), then ensure that the DDRC DDR_TX_CFG_1 bit[3:0] is cleared to 0000.
3. Halt further DDRC transfers by setting DDR_SDRAM_CFG[MEM_HALT] and poll for [MEM_HALT] to be set.
4. In the DDRC, poll for the DDRDSR_2[IDLE] to be set. If inline ECC is enabled, also poll for DDRDSR_2[NML] to be set.
5. Use the DDRC DDR_SDRAM_MD_CNTL register to issue a DRAM mode register write to MR13 in order to set FSP-WP and FSP-OP to 1 (MR13 value of 0xC0) for frequency setpoint1 operation. As the DRAM and its mode registers are now configured for frequency setpoint1 operation, issue a series of mode register writes to configure the DRAM mode registers to operate at the newly desired Frequency State ’s DDR frequency (for example, update MR2 for the desired frequency’s WL/RL setting, and so on). However it is recommended to leave VRef(CA) as it should be for the higher of the all frequencies.
    a. These series of mode registers are configured for each Frequency State (frequency setpoint) using NXP’s DDR Config tool.
6. Write the DDRC DDR_SDRAM_CFG_4[FRQCH_RET] to the desired Frequency State value (0x0 for Frequency State 0, 0x1 for Frequency State 1, 0x2 for Frequency State 2) then poll DDR_SDRAM_CFG_4[FRQCH_RET] to ensure desired Frequency State value configuration has completed.
7. If the desired Frequency State data rate <=533MT/s, then the following sequence should be followed:
    a. Wait 2 micro-seconds.
    b. Reset the DDRC by setting DDR_SDRAM_CFG_3[DDRC_RST]=1 and then poll this bit to ensure it is set.
    c. Wait 5 micro-seconds before proceeding to the next step.
8. Set the DDRC DDR_SDRAM_CFG_2 [FRC_SR] to place the DRAM in self-refresh and wait sometime to ensure DRAM in self-refresh. The following wait times are recommended based on the DRAM data rate of the current Frequency State :

Data rate of current Frequency State	Wait time after self-refresh entry
Greater than or equal to 3733 MT/s	3 micro-seconds
Greater than or equal to 625MT/s	7 micro-seconds
Less than 625MT/s	70 micro-seconds

9. Switch the DDR clock (frequency) by programming the DRAM PLL to match the new frequency of the desired Frequency State or switch to the CCM DRAM root clock in bypass mode.
10. Update the following DDRC registers to align with the timings for the desired Frequency State : TIMING_CFG_{0, 1, 2, 3, 4, 7, 8, 9, 10, 11}, DDR_SDRAM_INTERVAL, and DDR_ZQ_CNTL.
    a. These registers are configured for each Frequency State (frequency setpoint) using NXP’s DDR Config Tool.
11. Clear the DDRC DDR_SDRAM_CFG_2 [FRC_SR] to force the DRAM to exit self-refresh.
    a. If the desired Frequency State data rate <=533MT/s, then wait 5 micro-seconds before proceeding to the next step.
12. Use the DDRC DDR_SDRAM_MD_CNTL register to issue a DRAM mode register write to MR13 in order to set FSP-WP and FSP-OP to 1 (MR13 value of 0xC0) for frequency setpoint1 operation. This ensures that the DRAM is configured for frequency setpoint1 operation in case this may have been inadvertently re-configured.

13. Restore the previous state of the DDRC DDR_SDRAM_CFG_3[SR_FAST_WK_EN] and [DYN_REF_RATE_EN] which were previously disabled in Step 2. Also, re-enable auto clock gating be setting the DDR Mix AUTO_CG_CTRL[AUTO_CG_ENA] = 1.
14. Clear the DDRC DDR_SDRAM_CFG [MEM_HALT] and poll for [MEM_HALT] to be cleared.

81.4 DDR IO Retention Enter/Exit Procedure

The DDR IO Retention standby power state is a non-operational low power state. It is the lowest possible power state for the DDR Controller and PHY. In this state, the DRAM is in self refresh to maintain its data contents, while the DDR Controller and PHY are powered down.

The prerequisite prior to entering and exiting DDR retention is to save the DDR controller and DDR PHY configuration data (address and data pairs) and store this data to a non-DRAM location; in this case it is recommended to use the On-Chip SRAM (OCRAM). This step is necessary as this configuration data must be restored when exiting DDR retention. It is recommended to save the DDR controller and DDR PHY configuration data during the system initialization process after DRAM initialization and data training have completed.

The DDR controller and DDR PHY address/data pair configuration structures can be found in the file lpddr4x_timing.c. This file, generically named henceforth as timing.c, is generated using NXP's DDR Config Tool (available at nxp.com). Refer to the NXP DDR Config Tool's User Guide for more information on the tool and its usage.

The DDR controller and DDR PHY configuration data can be found in the timing.c file structure called:

```
struct dram_timing_info dram_timing
```

Specifically, the user will need to save the address/data pairs of the following dram_timing members to a designated area in the OCRAM:

```
.ddrc_cfg – DDR Controller (DDRC) configuration data
.ddrc_cfg_num – size of the ddrc_cfg data structure
.ddrphy_cfg – DDR PHY configuration data
.ddrphy_cfg_num – size of the ddrphy_cfg data structure
.ddrphy_trained_csr – DDR PHY trained values configuration data (initialize after data
      training)
.ddrphy_trained_csr_num – size of the ddrphy_trained_csr data structure
.ddrphy_pie – DDR PHY configuration data
.ddrphy_pie_num – size of the ddrphy_pie data structure
.fsp_cfg – DDRC timing registers and DRAM mode registers for each Frequency State
.fsp_cfg_num – size of the fsp_cfg data structure
```

For the data structure ddr_ddrphy_trained_csr, the data values are initialized to zero. The data for the structure will need to be obtained and saved after performing DRAM initialization, specifically after data training. This data consists of trained values specific for the DDR PHY and hence why they can only be saved after performing DRAM data training. The following section outlines how to save this trained data into the data structure ddr_ddrphy_trained_csr.

81.4.1 Procedure on how to save the DDR PHY trained data

The DDR PHY configuration data structure is defined in the timing.c file and consists of the following structures:

```
.ddrphy_cfg – DDR PHY configuration data
.ddrphy_trained_csr – DDR PHY trained values configuration data (initialize after data
      training)
.ddrphy_pie – DDR PHY configuration data
```

Most of these can simply be saved using the data provided in the timing.c file. However, in the case of ddrphy_trained_csr, this consists of trained data values which can only be obtained after data training has been completed.

The following describes the steps necessary to save the DDR PHY trained data.

- First, ensure DDR data training has completed, then proceed with the following
- Write 0x0 to address 0x4E2B8000
- Write 0x3 to address 0x4E2B4200
- In a for-loop, perform the following:
    1. Read from the address in the structure
        a. You will need to covert this address to the AXI address equivalent – refer Procedure on translating the DDR PHY address to the equivalent AXI address
    2. Store the read value into the associated data of the address/data pair of the structure
    3. Increment to the next structure member address
    4. Continue these steps until the end of the data structure size
- Write 0x2 to address 0x4E2B4200
- Write 0x1 to address 0x4E2B8000

81.4.2 DDR IO Retention Enter Procedure

The figure below illustrates the flow for entering DDR IO Retention:

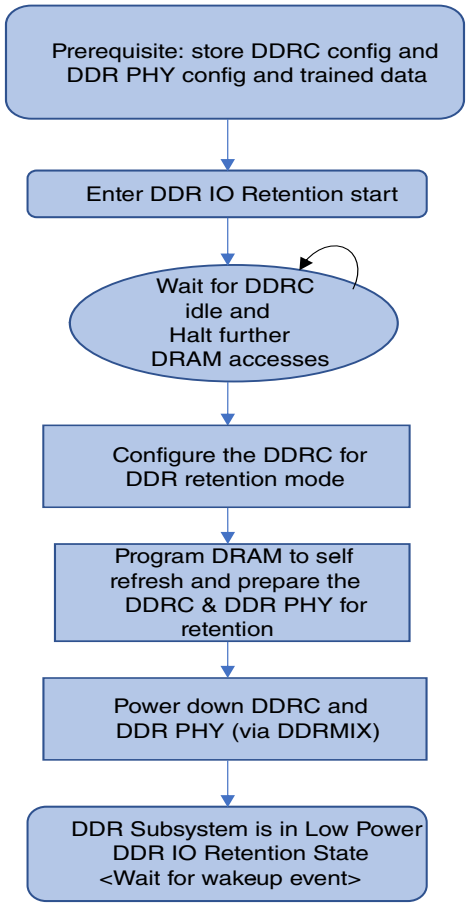


Figure 450. DDR IO Retention Enter Procedure

The following are the detailed steps for entering DDR IO Retention:

1. Poll for the DDRC DDRDSR_2[IDLE] to be set to 1 and then halt further DDRC transfers by setting DDRC DDR_SDRAM_CFG[MEM_HALT] to 1
2. Set DDRC DDR_SDRAM_CFG_3[SR_FAST_WK_EN] to 1 then poll to ensure this is set to 1
3. Program DDR_SDRAM_CFG_4[FRQCH_RET] to 1_1111 (0x1F) then poll to ensure this is configured correctly to 0x1F
4. Clear the DDRC DDR_ZQ_CNTL register to 0
5. Force the DDRC to enter self-refresh by setting DDRC DDR_SDRAM_CFG_2[FRC_SR] to 1 then perform the following tasks:
    a. Set BLK_CTRL_DDRMIX AUTO_CG_CTRL[HWFFC_ACG_FORCE_B] to 1 to allow the auto clock gate to send ipg_stop
    b. Set DDRC DDR_SDRAM_CFG[SREN] to 1 to enable self-refresh during sleep
    c. Set DDRC DDR_SDRAM_CFG_3[DRAIN_FOR_SR] to 1 to force the DDRC to drain its main command queue before servicing a self-refresh request
6. Set the BLK_CTRL_DDRMIX DDR Controller ipg_stop signal and wait for DDR Controller ipg_stop_ack signal acknowledge as follows:
    a. Poll BLK_CTRL_DDRMIX DDRC_STOP_CTRL[DDRC_STOP_ACK] bit until it returns 0
    b. Set DDRC_STOP_CTRL[DDRC_STOP] to 1
    c. Poll DDRC_STOP_CTRL[DDRC_STOP_ACK] bit until it returns 1
7. Disable auto refresh by clearing DDRC DDR_INTERVAL to 0
8. Set DDRC DDR_SDRAM_MD_CNTL to 0x00100000 which forces CKE to remain low
9. Clear DDRC_STOP_CTRL[DDRC_STOP] to 0, then perform the following tasks:
    a. Poll DDRC DDRDSR_2[PHY_INIT_CMPLT] until it returns 1
    b. Now set DDRDSR_2[PHY_INIT_CMPLT] to 1 to clear this bit
10. Clear DDRC DDR_SDRAM_CFG[SREN] to 0
11. Clear DDRC DDR_SDRAM_CFG_3[SR_FAST_WK_EN] to 0
12. Set the BLK_CTRL_DDRMIX DDR Controller ipg_stop signal and wait for DDR Controller ipg_stop_ack signal acknowledge as follows:
    a. Poll BLK_CTRL_DDRMIX DDRC_STOP_CTRL[DDRC_STOP_ACK] bit until it returns 0
    b. Set DDRC_STOP_CTRL[DDRC_STOP] to 1
    c. Poll DDRC_STOP_CTRL[DDRC_STOP_ACK] bit until it returns 1
13. Clear DDRC_STOP_CTRL[DDRC_STOP] to 0
14. Check the status of the internal DDR PHY to ensure it has entered low power (DDR Retention) mode
    a. Write 0x0 to address 0x4E2B8000
    b. Read from address 0x4E2A40A0 and ensure it returns 1
        - If it doesn’t return 1, re-visit the steps above to ensure they’ve been implemented correctly
    c. Write 0x1 to address 0x4E2B8000
15. Clear DDR PHY PwrOkIn via System Reset Controller (SRC) src_dphy_slice SINGLE_RESET_SW_CTRL [RST_CTRL_SOFT_0] (address 0x44461424, bit[0] = 0)
16. Power off the DDR subsystem by setting SRC src_ddr_slice SLICE_SW_CTRL[PDN_SOFT] = 1 (address 0x44461020, bit [31] = 1)

81.4.3 DDR IO Retention Exit Procedure

The figure below illustrates the flow for exiting DDR IO Retention:

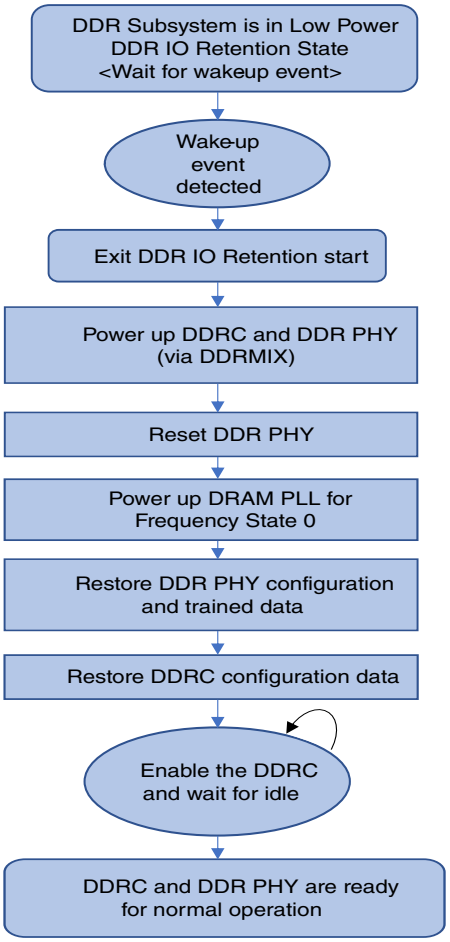


Figure 451. DDR IO Retention Exit Procedure

The following are the detailed steps for exiting DDR IO Retention:

1. First, power up and reset the DDRMIX subsystem by
    a. Power up the DDRMIX: Clear System Reset Controller (SRC) src_ddr_slice SLICE_SW_CTRL[PDN_SOFT] = 0 (address 0x44461020, bit [31] = 0)
    b. Reset to the DDRMIX subsystem as follows:
        i. Set SRC src_ddr_slice SLICE_SW_CTRL[RST_CTRL_SOFT_0] = 1 (address 0x44461020, bit [0] = 1)
        ii. Wait for a minimum of 10 milli-seconds
        iii. Clear SRC src_ddr_slice SLICE_SW_CTRL[RST_CTRL_SOFT_0] = 0 (address 0x44461020, bit [0] = 0)
        iv. Wait for a minimum of 10 milli-seconds
2. Reset the DDR PHY as follows:
    a. Clear SRC src_dphy_slice SLICE_SW_CTRL[RST_CTRL_SOFT] = 0 (address 0x44461420, bit[0] = 0)
    b. Clear SRC src_dphy_slice SINGLE_RESET_SW_CTRL[RST_CTRL_SOFT_1] = 0 (address 0x44461424, bit[2] = 0)

c. Set SRC src_dphy_slice SINGLE_RESET_SW_CTRL [RST_CTRL_SOFT_0] = 1 (address 0x44461424, bit[0] = 1)

d. Wait a minimum of 10 micro-seconds

e. Set SRC src_dphy_slice SLICE_SW_CTRL[RST_CTRL_SOFT] = 1 (address 0x44461420, bit[0] = 1)

f. Set SRC src_dphy_slice SINGLE_RESET_SW_CTRL[RST_CTRL_SOFT_1] = 1 (address 0x44461424, bit[2] = 1)

g. Wait a minimum of 10 micro-seconds

h. Clear SRC src_dphy_slice SINGLE_RESET_SW_CTRL [RST_CTRL_SOFT_0] = 0 (address 0x44461424, bit[0] = 0)

i. Wait a minimum of 10 micro-seconds

j. Clear SRC src_dphy_slice SLICE_SW_CTRL[RST_CTRL_SOFT] = 0 (address 0x44461420, bit[0] = 0)

k. Clear SRC src_dphy_slice SINGLE_RESET_SW_CTRL[RST_CTRL_SOFT_1] = 0 (address 0x44461424, bit[2] = 0)

3. Configure the DRAM PLL for Frequency State 0 as follows:

   a. Set DRAM PLL Control (CTRL_SET) to 1 (address 0x44481304 = 0x1) to power up and enable the DRAM PLL

   b. Wait for PLL lock by polling DRAM PLL PLL_STATUS[PLL_LOCK] until it becomes set to 1 (poll address 0x444813F0, bit[0] until it becomes a 1)

   c. Write 0x4 to DRAM PLL CTRL_CLR (write 0x4 to address 0x44481308)

   d. Write 0x2 to DRAM PLL CTRL_SET (write 0x2 to address 0x44481304)

   e. Perform the following to disable the DDR clock bypass

      i. Configure the DRAM APB to 133MHz: Write 0x202 to bits[9:0] of the DRAM_APB_CLK (slice 77, dram_apb_clk_root, CLOCK_ROOT77_CONTROL) register (address 0x44452680, bits[9:0] = 0x202).

      ii. Wait (poll) to ensure the setting was updated: Read bit[28] of address 0x444526A0 until it returns 0 (slice 77, dram_apb_clk_root, Clock root working status register (CLOCK_ROOT77_STATUS0)).

      iii. Switch from DRAM clock root from CCM to PLL: Write 1 to DRAM_PLL_BYPASS bit[0] of the CCM_GPR2 (GPR_SHARED2) register (address 0x44454840, bit[0] = 1).

4. Restore the DDR PHY configuration as follows. Note, when writing back the address/data pairs, the address portion will need to be re-mapped to the AXI address. Refer Procedure on how to save the DDR PHY trained data .

   a. Restore the ddrphy_cfg data (previously saved from the timing.c file). As noted above, when writing the values back to the DDR PHY registers, these register addresses will need to be translated to the equivalent AXI address.

   b. Restore the ddr_ddrphy_trained_csr data (mentioned previously, where this data should have been saved following DRAM initialization). The restore process for ddr_ddrphy_trained_csr data is as follows:

      i. Write 0x0 to address 0x4E2B8000

      ii. Write 0x3 to address 0x4E2B4200

      iii. In a for-loop, perform the following:

         i. Restore (write) the saved data of the associated address/data pair to the (translated AXI) address

         ii. Increment to the next structure member address

         iii. Continue these steps until the end of the data structure size

      iv. Write 0x2 to address 0x4E2B4200

      v. Write 0x1 to address 0x4E2B8000

c. Restore the ddr_phy_pie data (previously saved from the timing.c file). As noted above, when writing the values back to the DDR PHY registers, these register addresses will need to be translated to the equivalent AXI address.

5. Restore the DDR Controller configuration data ddr_ddrc_cfg (previously saved from the timing.c file) as follows:
   a. Restore the ddrc_cfg data
      - When writing back address 0x4e300114 (DDR_SDRAM_CFG_2), clear bit[4] (D_INIT)
   b. Next, restore the Frequency Setpoint 0 (FSP[0]) DDRC timing registers. This is data structure fsp_cfg[0] found in timing.c (and should've been previously saved). Only restore the DDRC registers in this structure's FSP[0] data and not the DRAM mode registers.
   c. Poll DDRC DDRDSR_2[PHY_INIT_CMPLT] until it returns 1
   d. Now set DDRDSR_2[PHY_INIT_CMPLT] to 1 to clear this bit
   e. Enable the DDRC by writing MEM_EN (bit[31] = 1) in DDR_SDRAM_CFG
   f. Wait until the DDRC is idle: Read IDLE bit[31] in DDRDSR_2 until it is set to 1

The DDR Controller and PHY should now have exited DDR IO Retention mode and is ready for normal operation.

81.4.4 Procedure on translating the DDR PHY address to the equivalent AXI address

The various DDR PHY data structures contain configuration data for the DDR PHY in address/data pairs and are defined in the timing.c file. However, these addresses are defined from the perspective of the internal PHY module.

Whenever reading configuration data from or writing configuration data to the DDR PHY, the user will need to convert the PHY-based address to the equivalent AXI address. The following describes how to perform this address translation. It is recommended to implement the following steps into a function where the user passes in the DDR PHY address (DDR_PHY_ADDR) and returns the equivalent AXI address:

1. Function parameter input: DDR_PHY_ADDR
2. Create variable "X1" and configure as follows:
   - X1 = (DDR_PHY_ADDR >> 12) & 0x3FF
3. Create variable "X2" and configure as follows:
   - X2= DDR_PHY_ADDR & 0xFFF
4. Create variable "Y1" and configure this based "X1" in the lookup table below
5. Create variable "Y2" and configure as follows:
   - Y2=((Y1<<13) | (X2<<1)) << 1
6. Finally add "Y2" with 0x4E100000 and return the resultant value – this is the AXI address

Table 648. DDR PHY Address to AXI address lookup table

"X1"	"Y1"	"X1"	"Y1"	"X1"	"Y1"	"X1"	"Y1"
0x000	0x00	0x204	0x1c	0x018	0x38	0x316	0x54
0x001	0x01	0x205	0x1d	0x019	0x39	0x317	0x55
0x002	0x02	0x206	0x1e	0x110	0x3a	0x318	0x56
0x003	0x03	0x207	0x1f	0x111	0x3b	0x319	0x57
0x004	0x04	0x208	0x20	0x112	0x3c	0x020	0x58
0x005	0x05	0x209	0x21	0x113	0x3d	0x120	0x59

Table continues on the next page...

Table 648. DDR PHY Address to AXI address lookup table (continued)

0x006	0x06	0x20a	0x22	0x114	0x3e	0x220	0x5a
0x007	0x07	0x20b	0x23	0x115	0x3f	0x320	0x5b
0x008	0x08	0x300	0x24	0x116	0x40	0x040	0x5c
0x009	0x09	0x301	0x25	0x117	0x41	0x140	0x5d
0x00a	0x0a	0x302	0x26	0x118	0x42	0x240	0x5e
0x00b	0x0b	0x303	0x27	0x119	0x43	0x340	0x5f
0x100	0x0c	0x304	0x28	0x210	0x44	0x050	0x60
0x101	0x0d	0x305	0x29	0x211	0x45	0x051	0x61
0x102	0x0e	0x306	0x2a	0x212	0x46	0x052	0x62
0x103	0x0f	0x307	0x2b	0x213	0x47	0x053	0x63
0x104	0x10	0x308	0x2c	0x214	0x48	0x054	0x64
0x105	0x11	0x309	0x2d	0x215	0x49	0x055	0x65
0x106	0x12	0x30a	0x2e	0x216	0x4a	0x056	0x66
0x107	0x13	0x30b	0x2f	0x217	0x4b	0x057	0x67
0x108	0x14	0x010	0x30	0x218	0x4c	0x070	0x68
0x109	0x15	0x011	0x31	0x219	0x4d	0x090	0x69
0x10a	0x16	0x012	0x32	0x310	0x4e	0x190	0x6a
0x10b	0x17	0x013	0x33	0x311	0x4f	0x290	0x6b
0x200	0x18	0x014	0x34	0x312	0x50	0x390	0x6c
0x201	0x19	0x015	0x35	0x313	0x51	0x0c0	0x6d
0x202	0x1a	0x016	0x36	0x314	0x52	0x0d0	0x6e
0x203	0x1b	0x017	0x37	0x315	0x53		

81.5 Memory Map and register definition

This section includes the blk_ctrl_ddrmix module memory map and detailed descriptions of all registers.

81.5.1 blk_ctrl_ddrmix register descriptions

81.5.1.1 DDRMIX Block Control memory map

BLK_CTRL_DDRMIX base address: 4E01_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	DDRPHY DfiClk, DfiCtlClk HWFFC Control (HWFFC_CTRL)	32	RW	0000_0000
4	CA55 Arm PLL Observe Select (CA55_SEL_CTRL)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
8	DRAM_VREF power switch (VREF_PSW_CTRL)	32	RW	0000_0000
C	DDR Controller ipg_stop SW control (DDRC_STOP_CTRL)	32	RW	0000_0000
10	DDR Controller automatic clock gating (AUTO_CG_CTRL)	32	RW	0000_0000

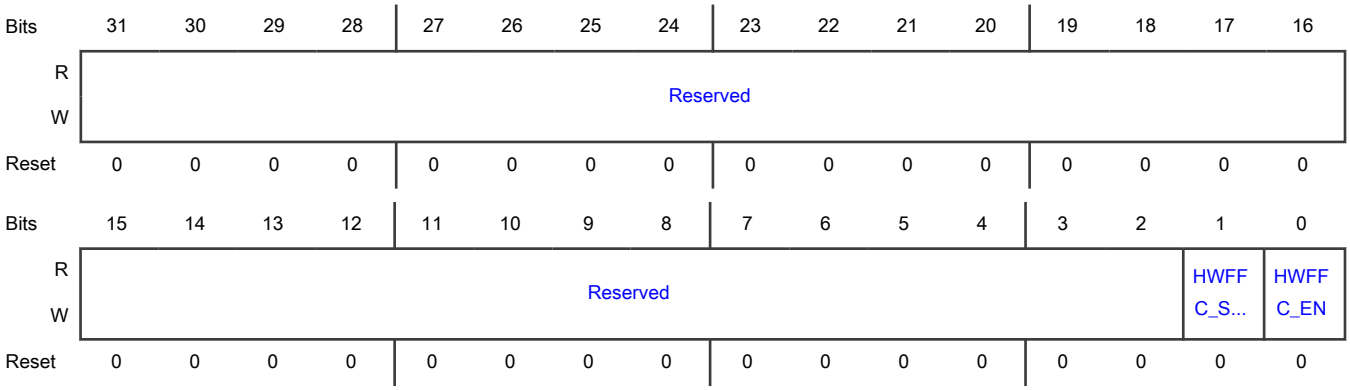
81.5.1.2 DDRPHY DfiClk, DfiCtlClk HWFFC Control (HWFFC_CTRL)

DDRPHY DfiClk, DfiCtlClk HWFFC functionality, selects normal clock or div2 frequency clock.

Offset

Register	Offset
HWFFC_CTRL	0h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 HWFFC_SEL	DDRPHY DfiClk, DfiCtlClk HWFFC Select This field select normal clock or div2 clock. 0 - Normal clock is selected 1 - Div2 frequency clock is selected
0	DDRPHY DfiClk, DfiCtlClk HWFFC Enable

Table continues on the next page...

Table continued from the previous page...

Field	Description
HWFFC_EN	This field enables DDRPHY HWFFC functionality. 0 - DDRPHY HWFFC is disabled 1 - DDRPHY HWFFC is enabled

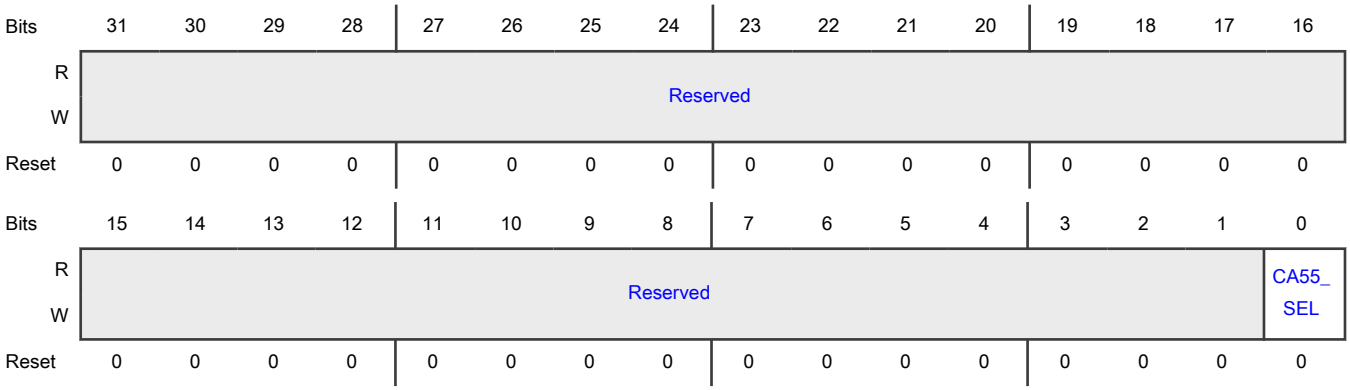
81.5.1.3 CA55 Arm PLL Observe Select (CA55_SEL_CTRL)

CA55 mix Arm PLL output clock needs observability support from DDRPHY.

Offset

Register	Offset
CA55_SEL_CTRL	4h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 CA55_SEL	CA55 Arm PLL select into DDRPHY This field select CA55mix Arm PLL output clock into DDRPHY DfiClk. This shows the Arm PLL output clock from DDRPHY digital observe bump DRAM_MALERT_B. 0 - Normal DfiClk from DRAM PLL is selected 1 - CA55mix Arm PLL is selected

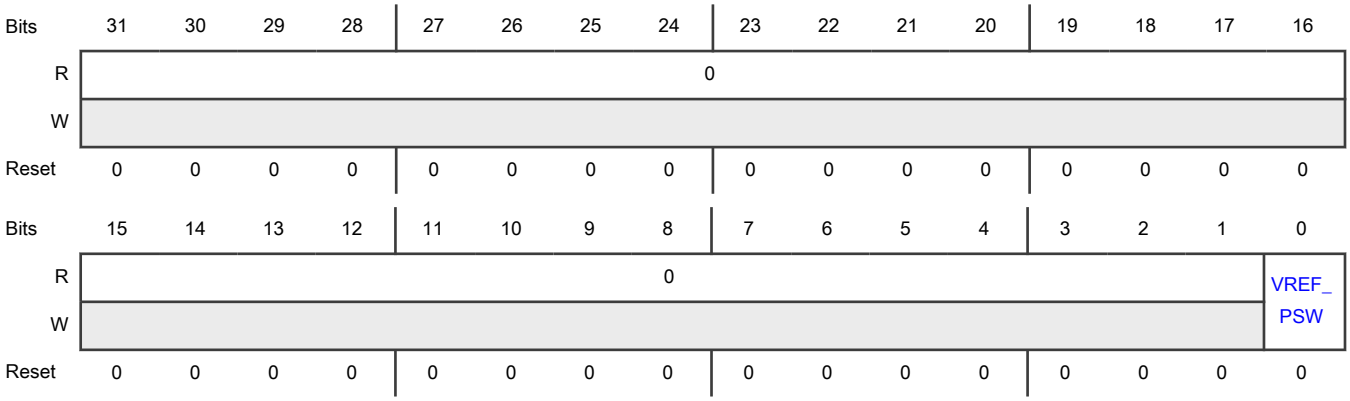
81.5.1.4 DRAM_VREF power switch (VREF_PSW_CTRL)

DDRPHY DRAM_VREF connected to ADC 1.8V rather than a bump, need power switch to prevent leakage.

Offset

Register	Offset
VREF_PSW_CTRL	8h

Diagram



Fields

Field	Description
31-1 —	Reserved
0 VREF_PSW	DDRPHY DRAM_VREF Power Switch This field shows 1.1V to ADC 1.8V power switch. DDRPHY DRAM_VREF is connected to ADC rather than BUMP. ADC supply is 1.8V versus DDRPHY supply 1.1V. A power switch is needed to prevent leakage when DRAM_VREF is not used for analog observability. 0 - Power switch is closed to prevent leakage 1 - Power switch is opened and is set together with DDRPHY[VrefInGlobal]

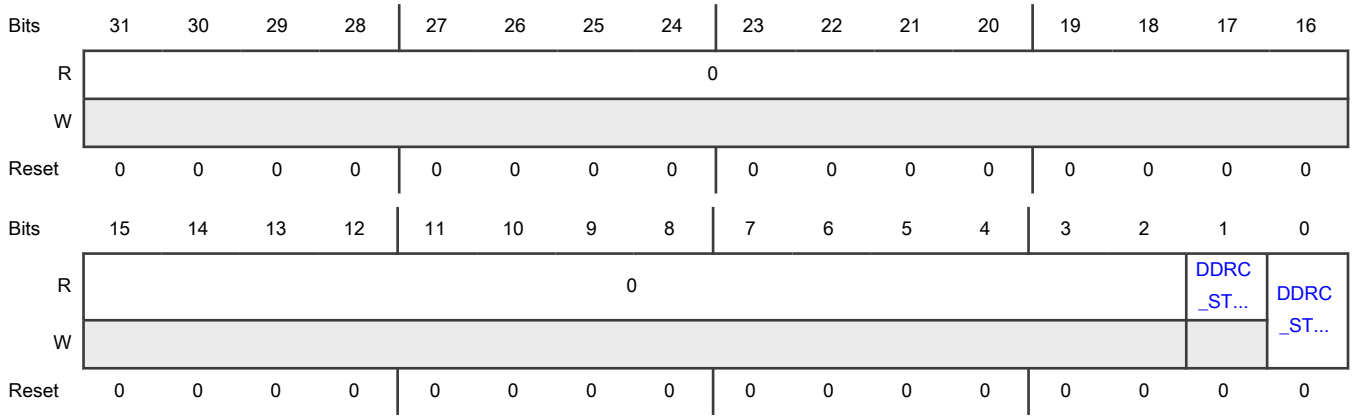
81.5.1.5 DDR Controller ipg_stop SW control (DDRC_STOP_CTRL)

Allow software control on DDR Controller ipg_stop signal, mainly used for HWFFC.

Offset

Register	Offset
DDRC_STOP_CTRL	Ch

Diagram



Fields

Field	Description
31-2 —	Reserved
1 DDRC_STOP_ACK	DDR Controller ipg_stop_ack This field is DDR Controller ipg_stop_ack signal. DDR Controller ipg_stop_ack is read only. This field should be polled before setting or clearing DDRC_STOP field. 0 - DDR Controller ipg_stop_ack is a 0 1 - DDR Controller ipg_stop_ack is a 1
0 DDRC_STOP	DDR Controller ipg_stop DDR Controller ipg_stop signal can be launched by this bit field. This bit is mainly used for HWFFC. The DDRC_STOP_ACK field should be 0 before setting this bit, and it should be a 1 before clearing this bit. 0 - Clear DDR Controller ipg_stop signal 1 - Set DDR Controller ipg_stop signal

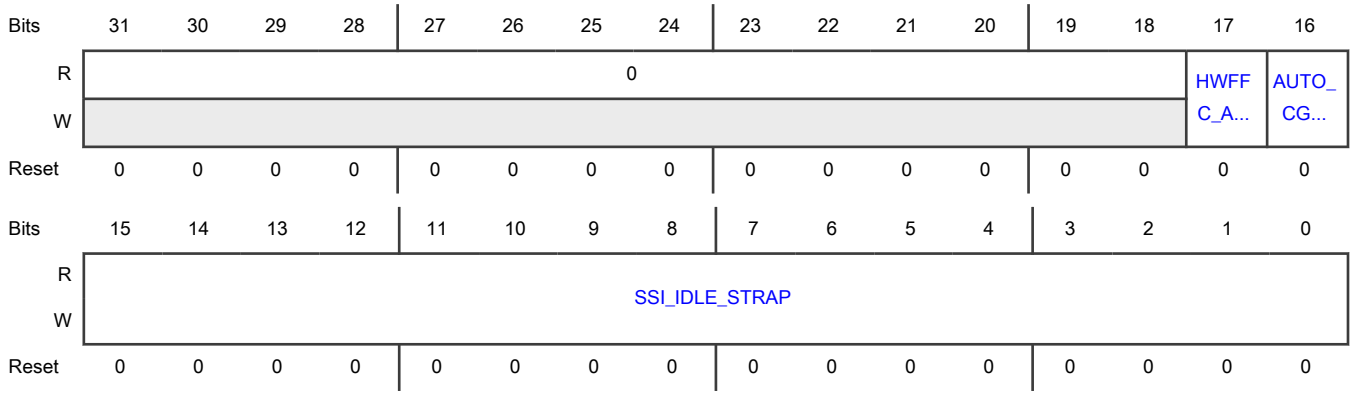
81.5.1.6 DDR Controller automatic clock gating (AUTO_CG_CTRL)

DDR Controller automatic clock gating when no AXI transmit(SSI idle and DDRC idle).

Offset

Register	Offset
AUTO_CG_CTRL	10h

Diagram



Fields

Field	Description
31-18 —	Reserved
17 HWFFC_ACG_FORCE_B	DDR Controller hwffc and auto CG send ipg_stop allow This field can be only programmed once. Please note that once programmed to 1, this bit cannot be cleared until next reset. 0 - DDR Controller hwffc and auto CG cannot send ipg_stop 1 - DDR Controller hwffc and auto CG can send ipg_stop
16 AUTO_CG_ENA	DDR Controller automatic clock gating enable 0 - DDR Controller automatic clock gating is disabled 1 - DDR Controller automatic clock gating is enabled
15-0 SSI_IDLE_STRAP	SSI Idle Strap This field is the number of cycles for SSI being idle before DDRC automaic clock gating. SSI slave module clk_wake_glitch signal needs to be 0 during this period. SSI prim channel needs to be idle, ddrc ipg_idle_ddrc need to be 1. All values: Clock cycles before DDRC automatic clock gating.

Chapter 82
Media Mix Domain Block Control (blk_ctrl_mediamix)

82.1 Overview

The MEDIAMIX domain contains control and status registers known as MEDIAMIX Block Control (MEDIAMIX BLK_CTRL). These registers include clocking, reset, and miscellaneous top-level controls for peripherals within the MEDIAMIX domain. Please see the clock control chapter and reset chapters for additional information.

82.1.1 Block diagram

The MEDIA BLK_CTRL interface connects to the MEDIAMIX domain peripherals. A block diagram illustrating the MEDIAMIX domain is shown below.

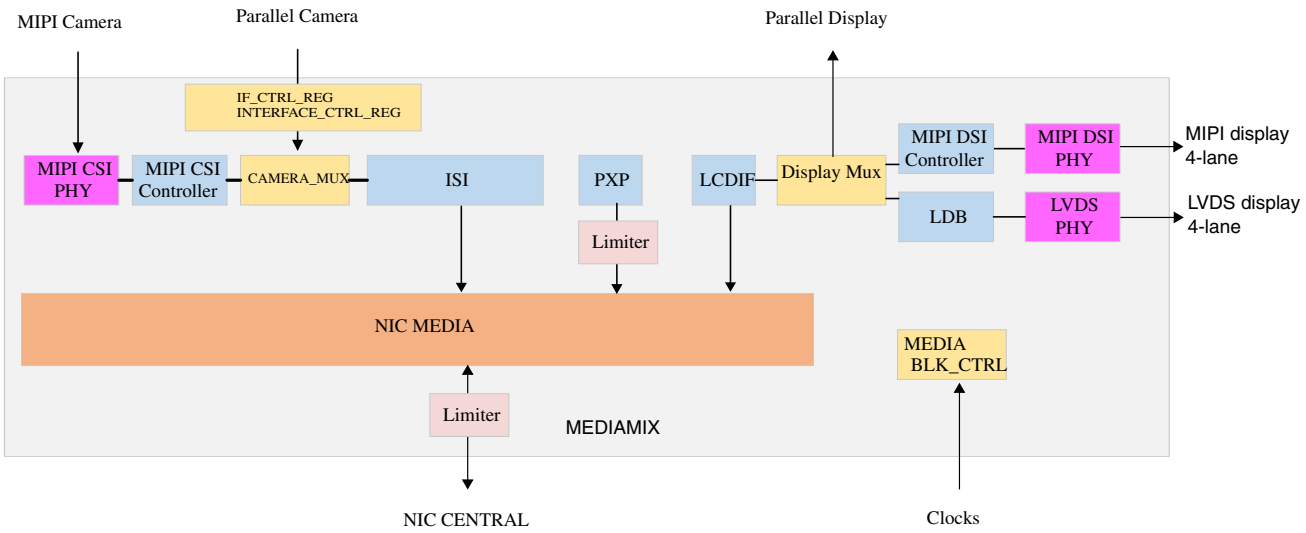


Figure 452. MEDIAMIX block diagram

82.1.2 Features

The MEDIA BLK_CTRL features include:

- Resets for each MEDIAMIX peripheral's clock related logic
- Clock gating of each MEDIAMIX peripheral's clock
- Top-level control and status registers for the MEDIAMIX peripherals
  - LCDIF
  - LVDS Display Bridge (LDB LVDS Tx Controller) and LVDS PHY
  - Image Sensor Interface (ISI)
  - Pixel Pipeline (PX)
  - MIPI CSI2 controller
  - MIPI DSI controller and MIPI DPHY

82.2 Functional description

The following sections describe functional details of the MEDIAMIX BLK_CTRL module.

82.2.1 Clocks and resets

The MEDIA BLK_CTRL contains clock distribution and gating controls, as well as reset handling to several MEDIAMIX peripherals. The RESET Control Register and the CLK Control Register provide reset and clock gating of the MEDIAMIX peripherals as shown below. Please see related clock and reset controls in the BLK_CTRL register description.

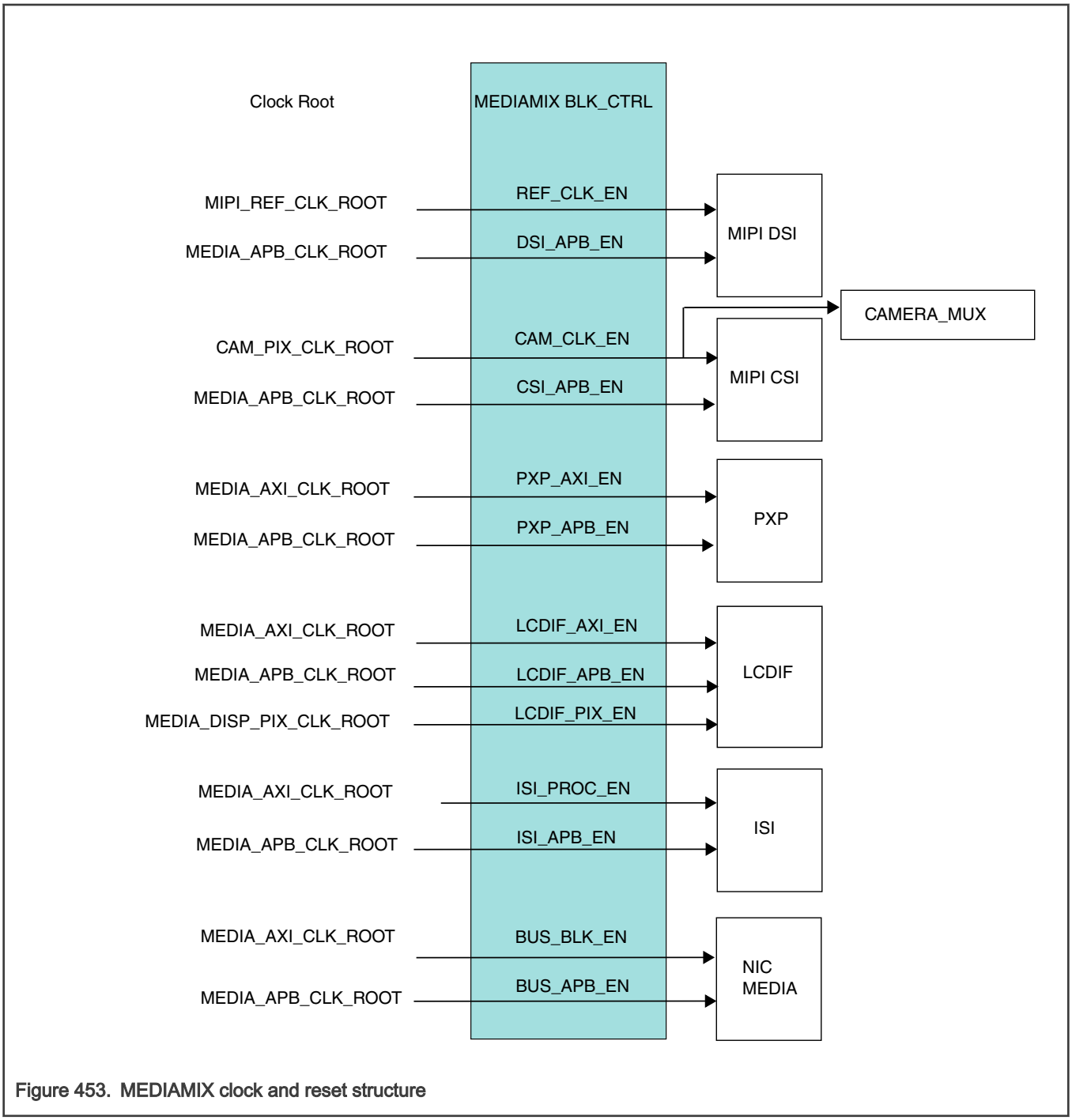


Figure 453. MEDIAMIX clock and reset structure

82.2.2 Interrupts

Refer to the individual display, imaging, and camera peripherals chapters for interrupt signal details.

82.3 External signals

There are no external signals pinned out for the MEDIAMIX Blk_Ctrl.

82.4 Memory map and register definition

This section includes the MEDIAMIX_GPR_CTRL module memory map and detailed descriptions of all registers.

NOTE

Disabling the MEDIAMIX reset does not generate a transfer error. Disabling power down will generate a transfer error.

82.4.1 BLK CTRL MEDIAMIX register descriptions

The BLK CTRL MEDIAMIX provides the following registers for controlling MEDIAMIX operation.

82.4.1.1 mediamix_GPR_ctrl memory map

BLK_CTRL_MEDIAMIX base address: 4AC1_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	RESET Control Register (RESET)	32	RW	0000_1FFF
4	CLK Control Register (CLK)	32	RW	0000_0000
C	QOS and cache of LCDIF (LCDIF)	32	RW	0000_0012
10	QOS and cache of PXP (PXP)	32	RW	0000_1212
14	Cache of ISI (ISI0)	32	RW	000F_2222
1C	QoS of ISI (ISI1)	32	RW	0000_0000
20	LDB Control Register (LDB_CTRL)	32	RW	0000_0001
24	LVDS Control Register (LVDS)	32	RW	0000_0000
28	Limiter Enable Control Register (LIMITER0)	32	RW	0000_0000
2C	Limiter Threshold Control Register (LIMITER1)	32	RW	0000_0000
30	Camera Mux Control Register (CAMERA_MUX)	32	RW	0001_0000
3C	Read Pixel Control Register (PIXEL_CTRL)	32	R	0000_0007
40	Read Line Count Register (LINE_CNT)	32	R	0000_0000
44	Read Pixel Count Register (PIXEL_CNT)	32	R	0000_0000
48	CSI Register (CSI)	32	RW	0000_2C1C
4C	DSI Register (DSI)	32	RW	0000_2C1C
50	DSI Write Register 0 (DSI_W0)	32	RW	0000_0000

Table continues on the next page...

Table continued from the previous page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
54	DSI Write Register 1 (DSI_W1)	32	RW	0000_0000
58	DSI Read Register 0 (DSI_R0)	32	R	0000_0000
5C	DSI Read Register 1 (DSI_R1)	32	R	0000_0000
60	Display Mux Control Register (DISPLAY_MUX)	32	RW	0000_0000
70	Parallel Camera Interface Register (IF_CTRL_REG)	32	RW	0000_0000
74	Interface Status Register (INTERFACE_STATUS)	32	R	0000_0000
78	Interface Control Register (INTERFACE_CTRL_REG)	32	RW	8000_0000
7C	Interface Control Register 1 (INTERFACE_CTRL_REG1)	32	RW	0000_0000

82.4.1.2 RESET Control Register (RESET)

Provides software reset control to the MEDIAMIX peripherals.

Offset

Register	Offset
RESET	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			ref_clk...	dsi_apb...	cam_clk...	csi_apb...	pxp_axi...	pxp_apb...	lcdif_p...	lcdif_a...	lcdif_a...	isi_pro...	isi_apb...	bus_blk...	bus_apb...
Reset	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Description
31-13 —	Reserved
12	DSI ref_clk reset

Table continues on the next page...

Table continued from the previous page...

Field	Description
ref_clk_en	Reset DSI ref_clk related logic 0 - Reset DSI reference clock related logic 1 - Do not reset DSI reference clock related logic
11 dsi_apb_en	DSI apb_clk reset Reset DSI apb_clk related logic 0 - Reset DSI apb_clk related logic 1 - Do not reset DSI apb_clk related logic
10 cam_clk_en	Camera clock reset Reset camera clock related logic 0 - Reset camera related logic 1 - Do not reset camera clock related logic
9 csi_apb_en	CSI apb_clk reset Resest CSI apb_clk related logic 0 - Reset CSI apb_clk related logic 1 - Do not reest CSI apb_clk related logic
8 pxp_axi_en	PXP axi_clk reset Reset PXP axi_clk related logic 0 - Reset PXP axi_clk related logic 1 - Do not reset PXP axi_clk related logic
7 pxp_apb_en	PXP apb_clk reset Reset PXP apb_clk related logic 0 - Reset PXP apb_clk related logic 1 - Do not reset PXP apb_clk related logic
6 lcdif_pix_en	LCDIF pix_clk reset Reset LCDIF pix_clk related logic 0 - Reset LCDIF pix_clk related logic 1 - Do not reset LCDIF pix_clk related logic
5 lcdif_axi_en	LCDIF axi_clk reset Reset LCDIF axi_clk related logic 0 - Reset LCDIF axi_clk related logic 1 - Do not reset LCDIF axi_clk related logic

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 lcdif_apb_en	LCDIF apb_clk reset Reset LCDIF apb_clk related logic 0 - Reset LCDIF apb_clk related logic 1 - Do not reset LCDIF apb_clk related logic
3 isi_proc_en	ISI axi_clk reset Reset ISI axi_clk related logic 0 - Reset ISI axi_clk related logic 1 - Do not reset ISI axi_clk related logic
2 isi_apb_en	ISI apb_clk reset Reset ISI apb_clk related logic 0 - Reset ISI apb_clk related logic 1 - Do not reset ISI apb_clk related logic
1 bus_blk_en	Bus axi_clk reset Reset bus axi_clk related logic 0 - Reset bus axi_clk related logic 1 - Do not reset bus axi_clk related logic
0 bus_apb_en	Bus apb_clk reset Reset bus apb_clk related logic 0 - Reset bus apb_clk related logic 1 - Do not reset bus apb_clk related logic

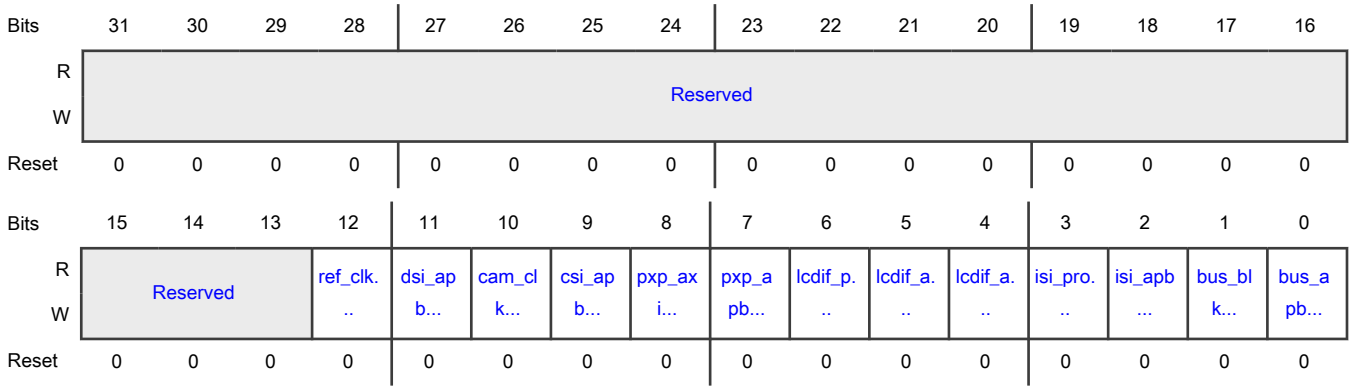
82.4.1.3 CLK Control Register (CLK)

Provides software clock gating control to the MEDIAMIX peripherals.

Offset

Register	Offset
CLK	4h

Diagram



Fields

Field	Description
31-13 —	Reserved
12 ref_clk_en	DSI ref_clk gate enable Gate DSI ref_clk root 0 - Do not gate DSI ref_clk root 1 - Gate DSI ref_clk root
11 dsi_apb_en	DSI apb_clk gate enable Gate DSI apb_clk root 0 - Do not gate DSI apb_clk root 1 - Gate DSI apb_clk root
10 cam_clk_en	cam_clk gate enable Gate cam_clk root 0 - Do not gate cam_clk root 1 - Gate cam_clk root
9 csi_apb_en	CSI apb_clk gate enable Gate CSI apb_clk root 0 - Do not gate CSI apb_clk root 1 - Gate CSI apb_clk root
8 pxp_axi_en	PXP axi_clk gate enable Gate PXP axi_clk gate root 0 - Do not gate PXP axi_clk root 1 - Gate PXP axi_clk root

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 pxp_apb_en	PXP apb_clk gate enable Gate PXP apb_clk root 0 - Do not gate PXP apb_clk root 1 - Gate PXP apb_clk gate root
6 lcdif_pix_en	LCDIF pix_clk gate enable Gate LCDIF pix_clk root 0 - Do not gate LCDIF pix_clk root 1 - Gate LCDIF pix_clk root
5 lcdif_axi_en	LCDIF axi_clk gate enable Gate LCDIF axi_clk root 0 - Do not gate LCDIF axi_clk root 1 - Gate LCDIF axi_clk root
4 lcdif_apb_en	LCDIF apb_clk gate enable Gate LCDIF apb_clk root 0 - Do not gate LCDIF apb_clk root 1 - Gate LCDIF apb_clk root
3 isi_proc_en	ISI axi_clk gate enable Gate ISI axi_clk root 0 - Do not gate ISI axi_clk root 1 - Gate ISI axi_clk root
2 isi_apb_en	ISI apb_clk gate enable Gate ISI apb_clk root 0 - Do not gate ISI apb_clk root 1 - Gate ISI apb_clk root
1 bus_blk_en	Bus axi_clk gate enable Gate bus axi_clk root 0 - Do not gate bus axi_clk root 1 - Gate bus axi_clk root
0 bus_apb_en	Bus apb_clk gate enable Gate bus apb_clk root 0 - Do not gate bus apb_clk root

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Gate bus apb_clk root

82.4.1.4 QOS and cache of LCDIF (LCDIF)

Provides controls for the LCDIF QoS and cache

Offset

Register	Offset
LCDIF	Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DEFAULT_QOS				CFG_QOS				Reserved			ARCACHE...	ARCACHE			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Fields

Field	Description
31-16 —	Reserved
15-12 DEFAULT_QOS	Default QoS value of LCDIF
11-8 CFG_QOS	cfg_qos value of LCDIF
7-5 —	Reserved
4	ARCACHE enable of LCDIF

Table continues on the next page...

Table continued from the previous page...

Field	Description
ARCACHE_EN	0 - Do not enable ARCACHE 1 - Enable ARCACHE
3-0 ARCACHE	ARCACHE value of LCDIF

82.4.1.5 QOS and cache of PXP (PXP)

Provides controls for the PXP QoS and cache

Offset

Register	Offset
PXP	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DEFAULT_QOS_OTHER_R				CFG_QOS_OTHER_R				DEFAULT_QOS_OTHER_W				CFG_QOS_OTHER_W			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			AWCACHE...	AWCACHE				Reserved			ARCACHE...	ARCACHE			
Reset	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0

Fields

Field	Description
31-28 DEFAULT_QOS_OTHER_R	Default QoS R value of PXP A low priority MEDIAMIX request, either LCDIF or ISI, may be blocked behind medium QoS transactions from other masters. However, when panic goes high, the LCDIF/ISI request QOS can go high, but these medium QoS transactions are blocked behind previous low priority LCDIF/ISI requests. In this case they can be blocked for a long time. This is the reason for panic to also go to the other master to control their default/panic QoS value. When panic goes high, the other masters will be set to a lower priority, lower than LCDIF/ISI (default), and thus the system will flush and allow the LCDIF/ISI requests to go through. In this case, the PXP CFG_QOS R should be lower than the DEFAULT_QOS.
27-24	cfg_qos R value of PXP

Table continues on the next page...

Table continued from the previous page...

Field	Description
CFG_QOS_OTHER_R	A low priority MEDIAMIX request, either LCDIF or ISI, may be blocked behind medium QoS transactions from other masters. However, when panic goes high, the LCDIF/ISI request QOS can go high, but these medium QoS transactions are blocked behind previous low priority LCDIF/ISI requests. In this case they can be blocked for a long time. This is the reason for panic to also go to the other master to control their default/panic QoS value. When panic goes high, the other masters will be set to a lower priority, lower than LCDIF/ISI (default), and thus the system will flush and allow the LCDIF/ISI requests to go through. In this case, the PXP CFG_QOS R should be lower than the DEFAULT_QOS.
23-20 DEFAULT_QOS_OTHER_W	Default QoS W value of PXP A low priority MEDIAMIX request, either LCDIF or ISI, may be blocked behind medium QoS transactions from other masters. However, when panic goes high, the LCDIF/ISI request QOS can go high, but these medium QoS transactions are blocked behind previous low priority LCDIF/ISI requests. In this case they can be blocked for a long time. This is the reason for panic to also go to the other master to control their default/panic QoS value. When panic goes high, the other masters will be set to a lower priority, lower than LCDIF/ISI (default), and thus the system will flush and allow the LCDIF/ISI requests to go through. In this case other masters like PXP, the CFG_QOS W should be lower than the DEFAULT_QOS.
19-16 CFG_QOS_OTHER_W	cfg_qos W value of PXP A low priority MEDIAMIX request, either LCDIF or ISI, may be blocked behind medium QoS transactions from other masters. However, when panic goes high, the LCDIF/ISI request QOS can go high, but these medium QoS transactions are blocked behind previous low priority LCDIF/ISI requests. In this case they can be blocked for a long time. This is the reason for panic to also go to the other master to control their default/panic QoS value. When panic goes high, the other masters will be set to a lower priority, lower than LCDIF/ISI (default), and thus the system will flush and allow the LCDIF/ISI requests to go through. In this case other masters like PXP, the CFG_QOS W should be lower than the DEFAULT_QOS.
15-13 —	Reserved
12 AWCACHE_EN	AWCACHE enable of PXP 0 - Do not enable AWCACHE 1 - Enable AWCACHE
11-8 AWCACHE	AWCACHE value of PXP
7-5 —	Reserved
4 ARCACHE_EN	ARCACHE enable of PXP 0 - Do not enable ARCACHE 1 - Enable ARCACHE
3-0	ARCACHE value of PXP

Table continues on the next page...

Table continued from the previous page...

Field	Description
ARCACHE	

82.4.1.6 Cache of ISI (ISI0)

Provides controls for the ISI cache

Offset

Register	Offset
ISI0	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												AWCACHE...	AWCACHE...	AWCACHE...	ARCACHE...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AWCACHE_V				AWCACHE_U				AWCACHE_Y				ARCACHE_Y			
Reset	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0

Fields

Field	Description
31-20 —	Reserved
19 AWCACHE_V_EN	AWCACHE_V enable of ISI 0 - Do not enable AWCACHE_V 1 - Enable AWCACHE_V
18 AWCACHE_U_EN	AWCACHE_U enable of ISI 0 - Do not enable AWACHE_U 1 - Enable AWCACHE_U
17 AWCACHE_Y_EN	AWCACHE_Y enable of ISI 0 - Do not enable AWCACHE_Y 1 - Enable AWCACHE_Y

Table continues on the next page...

Table continued from the previous page...

Field	Description
16 ARCACHE_Y_EN	ARCACHE_Y enable of ISI 0 - Do not enable ARCACHE_Y 1 - Enable ARCACHE_Y
15-12 AWCACHE_V	AWCACHE_V value of ISI
11-8 AWCACHE_U	AWCACHE_U value of ISI
7-4 AWCACHE_Y	AWCACHE_Y value of ISI
3-0 ARCACHE_Y	ARCACHE_Y value of ISI

82.4.1.7 QoS of ISI (ISI1)

Provides controls for the ISI QoS

Offset

Register	Offset
ISI1	1Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DEFAULT_QOS_V				CFG_QOS_V				DEFAULT_QOS_U				CFG_QOS_U			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DEFAULT_QOS_Y_R				CFG_QOS_Y_R				DEFAULT_QOS_Y_W				CFG_QOS_Y_W			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 DEFAULT_QOS_V	Default QOS_V value of ISI
27-24 CFG_QOS_V	CFG_QOS_V value of ISI
23-20 DEFAULT_QOS_U	Default QoS value of ISI
19-16 CFG_QOS_U	cfg_qos_u value of ISI
15-12 DEFAULT_QOS_Y_R	Default QOS_Y_R value of ISI A low priority MEDIAMIX request, either LCDIF or ISI, may be blocked behind medium QoS transactions from other masters. However, when panic goes high, the LCDIF/ISI request QOS can go high, but these medium QoS transactions are blocked behind previous low priority LCDIF/ISI requests. In this case they can be blocked for a long time. This is the reason for panic to also go to the other master to control their default/panic QoS value. When panic goes high, the other masters will be set to a lower priority, lower than LCDIF/ISI (default), and thus the system will flush and allow the LCDIF/ISI requests to go through. In this case, the ISI CFG_QOS_Y_R should be lower than the DEFAULT_QOS.
11-8 CFG_QOS_Y_R	CFG_QOS_Y_R value of ISI A low priority MEDIAMIX request, either LCDIF or ISI, may be blocked behind medium QoS transactions from other masters. However, when panic goes high, the LCDIF/ISI request QOS can go high, but these medium QoS transactions are blocked behind previous low priority LCDIF/ISI requests. In this case they can be blocked for a long time. This is the reason for panic to also go to the other master to control their default/panic QoS value. When panic goes high, the other masters will be set to a lower priority, lower than LCDIF/ISI (default), and thus the system will flush and allow the LCDIF/ISI requests to go through. In this case, the ISI CFG_QOS_Y_R should be lower than the DEFAULT_QOS.
7-4 DEFAULT_QOS_Y_W	Default QOS_Y_W value of ISI
3-0 CFG_QOS_Y_W	cfg_qos_y_w value of ISI

82.4.1.8 LDB Control Register (LDB_CTRL)

Provides controls for the LDB

Offset

Register	Offset
LDB_CTRL	20h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	Reserved			Reserved				Reserved				Reserved			
W					ASYNC_FIFO_THRESHOLD			ASYNC_F...								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved			CH0_FIF...	Reserved	CH0_VSY...	Reserved	Reserved	CH0_BIT...	CH0_DAT...	Reserved	Reserved	Reserved	Reserved	CH0_ENA...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Description
31 —	Reserved Software must not change the value of this bit.
30-29 —	Reserved Software must not change the value of these bits.
28 —	Reserved Software must not change the value of this bit.
27-25 ASYNC_FIFO_THRESHOLD	LVDS channel 0 async FIFO threshold Controls the threshold of the async FIFO. The first data buffered by async FIFO will be popped after the number of data in the async fifo is greater tha async FIFO threshold. The remaining data will be popped when async FIFO is not empty.
24 ASYNC_FIFO_ENABLE	LVDS channel 0 async FIFO enable 0 - Disable 1 - Enable async FIFO to buffer RGB data
23 —	Reserved Software must not change the value of this bit.
22 —	Reserved Software must not change the value of this bit.

Table continues on the next page...

Table continued from the previous page...

Field	Description
21-20 —	Reserved Software must not change the value of these bits.
19 —	Reserved Software must not change the value of this bit.
18-16 —	Reserved Software must not change the value of these bits.
15 —	Reserved Software must not change the value of this bit.
14-13 —	Reserved Software must not change the value of these bits.
12 —	Reserved Software must not change the value of this bit.
11 CH0_FIFO_RESET	LVDS channel 0 async FIFO software reset 0 - No action 1 - Software reset
10 —	Reserved Software must not change the value of this bit.
9 CH0_VSYNC_POLARITY	VSYNC polarity select for LVDS channel 0 0 - VSYNC is active low 1 - VSYNC is active high
8 —	Reserved Software must not change the value of this bit.
7 —	Reserved Software must not change the value of this bit.
6 CH0_BIT_MAPPING	Data mapping for LVDS channel 0 0 - Use SPWG standard 1 - Use JEIDA standard
5 CH0_DATA_WIDTH	Data width for LVDS channel 0 0 - Reserved 1 - Data width is 24 bits wide.

Table continues on the next page...

Table continued from the previous page...

Field	Description
4 —	Reserved. Software must not change the value of this bit.
3 —	Reserved Software must not change the value of this bit.
2 —	Reserved Software must not change the value of this bit.
1 —	Reserved. Software must not change the value of this bit.
0 CH0_ENABLE	LVDS channel 0 enable 0 - Channel disabled 1 - Channel enabled

82.4.1.9 LVDS Control Register (LVDS)

Control parameters for LVDS

Offset

Register	Offset
LVDS	24h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				SPARE_IN			Reserved	Reserved		Reserved	Reserved	VBG_ADJ			SLEW_ADJ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SLEW_ADJ		CC_ADJ		CM_ADJ				PRE_EMPH_ADJ			PRE_EMP...	HS_EN	BG_EN	LVDS_EN	CH0_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-28 —	Reserved
27-25 SPARE_IN	Spare ports spare_in[2] is used for high speed mode for the random number generator (RNG). spare_in[2] = 0: RNG generates the 1.2 Gbps random sequence based on 600 Mhz clock. spare_in[2] = 1: RNG generates the 1.2 Gbps random sequence based on 1200 Mhz clock.
24 —	Reserved
23-22 —	Reserved
21 —	Reserved
20 —	Reserved
19-17 VBG_ADJ	Bandgap adjustment
16-14 SLEW_ADJ	Output transition time adjustment
13-11 CC_ADJ	Output current adjustment
10-8 CM_ADJ	Output common mode (Vos) adjustment
7-5 PRE_EMPH_ADJ	Pre-emphasis adjustment.
4 PRE_EMPH_EN	Enable pre-emphasis 0 - Disable 1 - Enable
3 HS_EN	Enable 100 ohm termination in the chip enable which also doubles power dissipation. 0 - Disable

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - Enable
2 BG_EN	Bandgap enable 0 - Disable 1 - Enable
1 LVDS_EN	LVDS PHY enable Before VDD_SOC_SW power is turned off, this bit must be set to 1'b1. 0 - Enable. LVDS function is normal. 1 - Disable
0 CH0_EN	Channel 0 enable 0 - Disabled 1 - Enabled

82.4.1.10 Limiter Enable Control Register (LIMITER0)

Controls limiter enable. Enabling the limiter allows the number of outstanding data beats that to be limited up to the threshold value defined in LIMITER1. The beat limiter prevents a low speed bus master from initiating read transactions faster than it may be possible to sink the returned data.

Offset

Register	Offset
LIMITER0	28h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														EN_PXP	EN_SF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-2 —	Reserved
1 EN_PXP	Limiter enable of PXP 0 - Do not enable limiter 1 - Enable limiter
0 EN_SF	Limiter enable of NIC_MEDIA 0 - Do not enable limiter 1 - Enable limiter

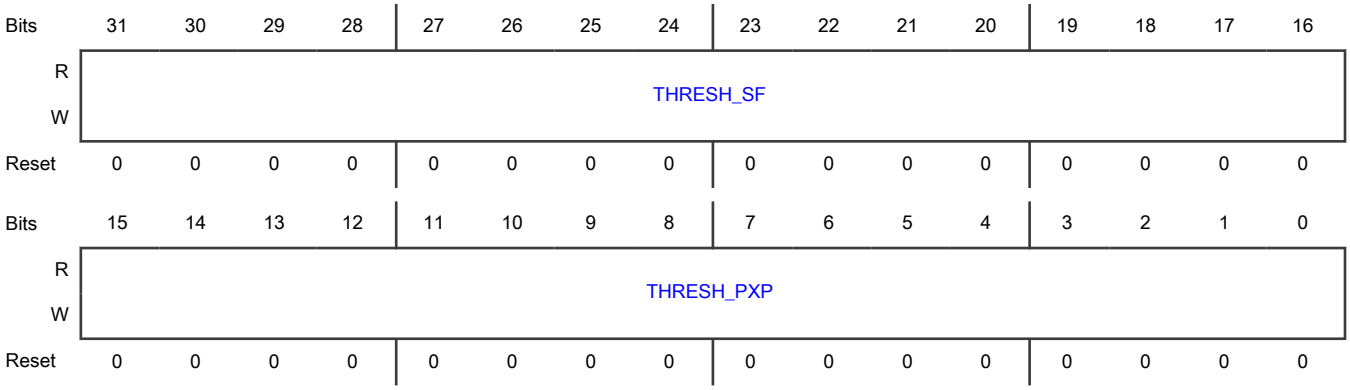
82.4.1.11 Limiter Threshold Control Register (LIMITER1)

Controls threshold

Offset

Register	Offset
LIMITER1	2Ch

Diagram



Fields

Field	Description
31-16 THRESH_SF	NIC_MEDIA read threshold of outstanding beats from NIC_CENTRAL
15-0 THRESH_PXP	PXP read threshold of outstanding beats from NIC_MEDIA.

82.4.1.12 Camera Mux Control Register (CAMERA_MUX)

MIPI CSI and Parallel camera to ISI gasket configration.

Align with MIPI CSI ipi_mode[8] configuration:

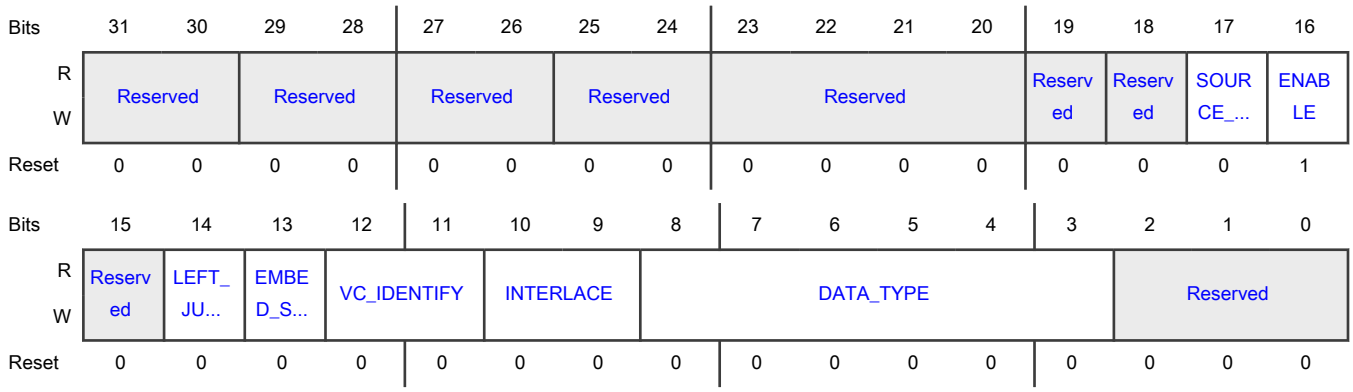
[0] 48bit : other (YUV422 8/10, YUV420 Legacy, RGB565/666/888, user define32)

[1] 16bit : 0x18(YUV420 8bit), 0x19(YUV420 10bit), RAW6/7/8/10/12/14, RAW32(embeded), user_define16

Offset

Register	Offset
CAMERA_MUX	30h

Diagram



Fields

Field	Description
31-30 —	Reserved
29-28 —	Reserved
27-26 —	Reserved
25-24 —	Reserved
23-20 —	Reserved
19	Reserved

Table continues on the next page...

Table continued from the previous page...

Field	Description
—	
18 —	Reserved
17 SOURCE_TYPE	Source type Selects source input to gasket. 0 - Data from MIPI CSI 1 - Data from parallel camera
16 ENABLE	Gasket enable 0 - Disable gasket 1 - Enable gasket
15 —	Reserved
14 LEFT_JUST_MODE	Left justified mode The left justified mode depends on the data type selected: RGB565/RGB666/RGB888 or RAW. 0 - For RGB565/RGB666/RGB888: Unused LSB equal most significant bit of valid data. For RAW: LSB aligned selection. 1 - For RGB565/RGB666/RGB888: Unused LSB equal least significant bit of valid data. For RAW: MSB aligned selection.
13 EMBED_SWAP	Swap for embedded data 0 - Swap for 16-bit embedded data 1 - No Swap for 16-bit embedded data
12-11 VC_IDENTIFY	Virtual channel identifier In non-interlaced mode, each ISI process channel has fixed input from MIPI CSI. In interlaced mode (line doubling deinterlaced mode), maybe different ISI process channel have same input from MIPI CSI.
10-9 INTERLACE	Interface mode 00 - Not interlaced 01 - Interfacled left (odd frame) 10 - Interlaced right (even frame) 11 - Reserved
8-3 DATA_TYPE	Data type Automatic override to 6'b010010 when embeded_en active.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	01_1000 - YUV420 8-bit 01_1001 - YUV420 10-bit 01_1010 - Legacy YUV420 (8-bit) 01_1100 - YUV420 8-bit (Chroma Shifted Pixel Sampling) 01_1101 - YUV420 10-bit (Chroma Shifted Pixel Sampling) 01_1110 - YUV422 8-bit 01_1111 - YUV422 10-bit 10_0010 - RGB565 10_0011 - RGB666 10_0100 - RGB888 10_1000 - RAW6 10_1001 - RAW7 10_1010 - RAW8 10_1011 - RAW10 10_1100 - RAW12 10_1101 - RAW14 11_0000 - User define32. Align with MIPI CSI ipi_mode[8] configure to 1'b0. 11_0001 - User define16. Align with MIPI CSI ipi_mode[8] configure to 1'b1.
2-0 —	Reserved

82.4.1.13 Read Pixel Control Register (PIXEL_CTRL)

Read pixel control information status.

Offset

Register	Offset
PIXEL_CTRL	3Ch

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				PIXEL_CTRL											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Fields

Field	Description
31-12 —	Reserved
11-0 PIXEL_CTRL	Read pixel control information status

82.4.1.14 Read Line Count Register (LINE_CNT)

Read line count status.

Offset

Register	Offset
LINE_CNT	40h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LINE_CNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LINE_CNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-0 LINE_CNT	Read line count status from MIPI to ISI wrapper. The actual line number is counter +1.

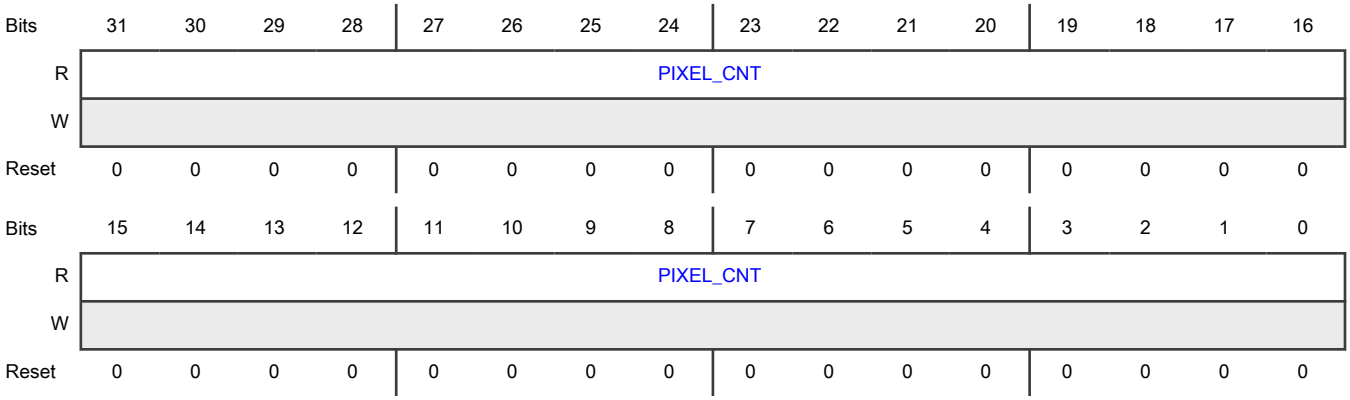
82.4.1.15 Read Pixel Count Register (PIXEL_CNT)

Read pixel count status.

Offset

Register	Offset
PIXEL_CNT	44h

Diagram



Fields

Field	Description
31-0 PIXEL_CNT	Read pixel count status from MIPI to ISI wrapper

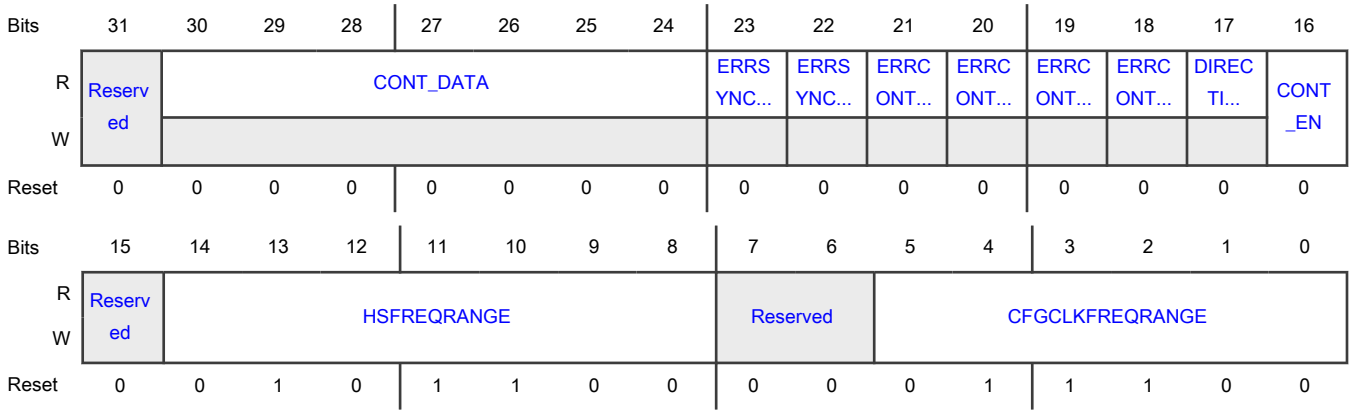
82.4.1.16 CSI Register (CSI)

MIPI D-PHY CSI control and status information.

Offset

Register	Offset
CSI	48h

Diagram



Fields

Field	Description
31 —	Reserved
30-24 CONT_DATA	Continuity data Each pin of this output bus is used to verify the continuity from the chips pin all the way to the PHYs analog ports cont_data[0]: rext cont_data[1]: clkn cont_data[2]: clkp cont_data[3]: datan0 cont_data[4]: datap0 cont_data[5]: datan1 cont_data[6]: datap1
23 ERRSYNCESC_1	Low power data transmission synchronization error lane 1 0 - No error detected. 1 - The number of bits received during low-power transmission is not a mulitple of 8 when the transmission ends. This bit remains high until the line returns to Stop state.
22 ERRSYNCESC_0	Low power data transmission synchronization error lane 0 0 - No error detected. 1 - The number of bits received during low-power transmission is not a mulitple of 8 when the transmission ends. This bit remains high until the line returns to Stop state.
21 ERRCONTROL_1	Control error lane 1 0 - No error detected.

Table continues on the next page...

Table continued from the previous page...

Field	Description
	1 - An incorrect line state sequence is detected. For example, if a turnaround request or Escape mode request is immediately followed by a Stop state instead of the required Bridge state, this signal is asserted and remains high until the line returns to Stop state.
20 ERRCONTROL_0	Control error lane 0 0 - No error detected. 1 - An incorrect line state sequence is detected. For example, if a turnaround request or Escape mode request is immediately followed by a Stop state instead of the required Bridge state, this signal is asserted and remains high until the line returns to Stop state.
19 ERRCONTENTIONLP1_0	LP1 contention error This signal is asserted when a lane module functioning as TX, while its base direction is RX detects a contention situation on a line while trying to drive the line low. 0 - No error detected. 1 - A lane module functioning as Tx, while its base direction is Rx detects a contention situation on a line while trying to drive the line low.
18 ERRCONTENTIONLP0_0	LP0 contention error This signal is asserted when a lane module functioning as TX, while its base direction is RX detects a contention situation on a line while trying to drive the line low. 0 - No error detected. 1 - A lane module functioning as Tx, while its base direction is Rx detects a contention situation on a line while trying to drive the line low.
17 DIRECTION_0	Transmit/receive direction 0 - The lane interconnect is in transmit mode. 1 - The lane interconnect is in receive mode.
16 CONT_EN	Continuity test mode enable 0 - Disble I/O continuity test mode. 1 - Enable I/O continuity test mode.
15 —	Reserved
14-8 HSFREQRANGE	High-speed frequency range default: 7'b0101100
7-6 —	Reserved
5-0	System clock frequency configuration

Table continues on the next page...

Table continued from the previous page...

Field	Description
CFGCLKFREQRANGE	cfgclkfreqrange[5:0] = round[(Fcfg_clk(MHz)-17)*4]; Fcfg_clk is 24 MHz, so cfgclkfreqrange[5:0] = 6'b011100

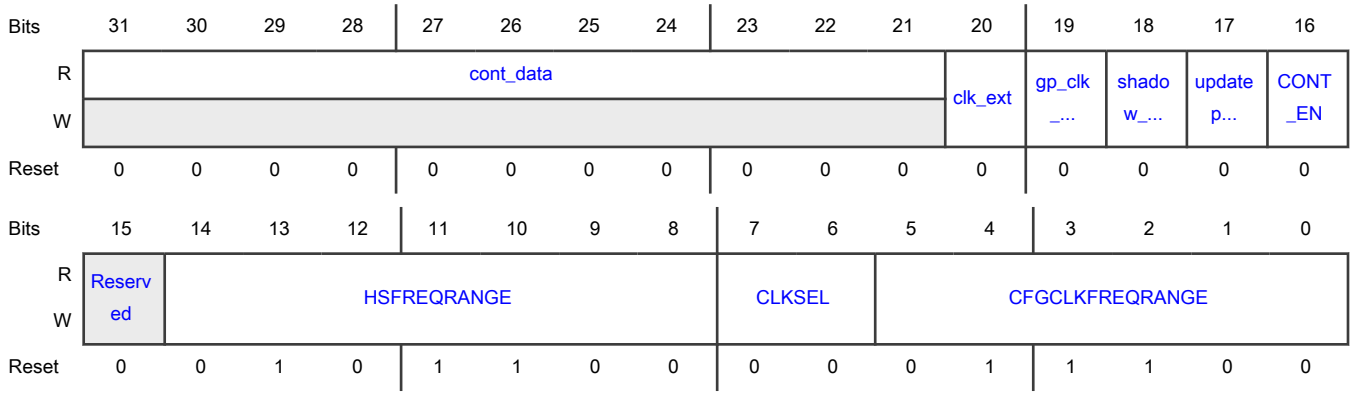
82.4.1.17 DSI Register (DSI)

MIPI D-PHY DSI control and status information.

Offset

Register	Offset
DSI	4Ch

Diagram



Fields

Field	Description
31-21 cont_data	cont_data Read-only status register Each pin of this output bus is used to verify the continuity from the chips pin all the way to the PHYs analog ports cont_data[0]: rext cont_data[1]: clkn cont_data[2]: clkp cont_data[3]: datan0 cont_data[4]: datap0 cont_data[5]: datan1 cont_data[6]: datap1

Table continues on the next page...

Table continued from the previous page...

Field	Description
	cont_data[7]: datan2 cont_data[8]: datap2 cont_data[9]: datan3 cont_data[10]: datap3
20 clk_ext	clk_ext
19 gp_clk_en	Enable for clkout_gp clock 0 - Disable clkout_gp clock 1 - Enable clkout_gp clock
18 shadow_clear	Shadow registers clear A pulse on this signal clears the shadow registers to their reset default value.
17 updatepll	Control for PLL operation frequency udpated A pulse on this signal triggers the PLL control signals to latch to the PLL shadow registers to set the PLL operating fequency.
16 CONT_EN	Continuity test mode enable This signal places the macro in IO continuity test mode default vault : 1'b0 0 - Disable I/O continuity test mode. 1 - Enable I/O continuity test mode.
15 —	Reserved
14-8 HSFREQRANGE	High-speed frequency range selection default: 7'b0101100
7-6 CLKSEL	Control of PLL clock output selection 00 - Clocks stopped 01 - Clock generation 10 - buffered clkext 11 - Forbidden
5-0 CFGCLKFREQRANGE	System clock frequency cfgclkfreqrange[5:0] = round[(Fcfg_clk(MHz)-17)*4]; Fcfg_clk is 24 MHz, so cfgclkfreqrange[5:0] = 6'b011100

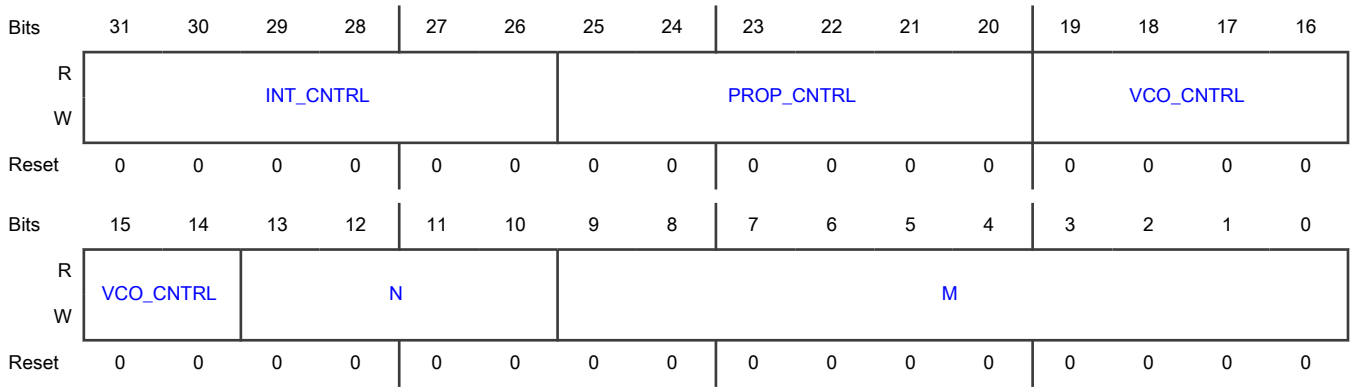
82.4.1.18 DSI Write Register 0 (DSI_W0)

DSI controls for M, N, charge pump, VCO operation.

Offset

Register	Offset
DSI_W0	50h

Diagram



Fields

Field	Description
31-26 INT_CNTRL	Integral charge pump control for SoC direct PLL control This signal can control the PLL if pll_shadow_control = 1'b0 Power down value=000000
25-20 PROP_CNTRL	Proportional charge pump control for SoC direct PLL control This signal can control the PLL if pll_shadow_control = 1'b0 Power down value=000000
19-14 VCO_CNTRL	VCO operating range for SoC direct PLL control Needs to be set appropriately to frequency of operation. This signal can control the PLL if pll_shadow_control = 1'b0
13-10 N	Control of the input frequency division ratio N (1 to 16) for SoC direct PLL control This signal can control the PLL if pll_shadow_control = 1'b0.
9-0 M	Control of the feedback multiplication ratio M (40 to 625) for SoC direct PLL control This signal can control the PLL if pll_shadow_control = 1'b0.

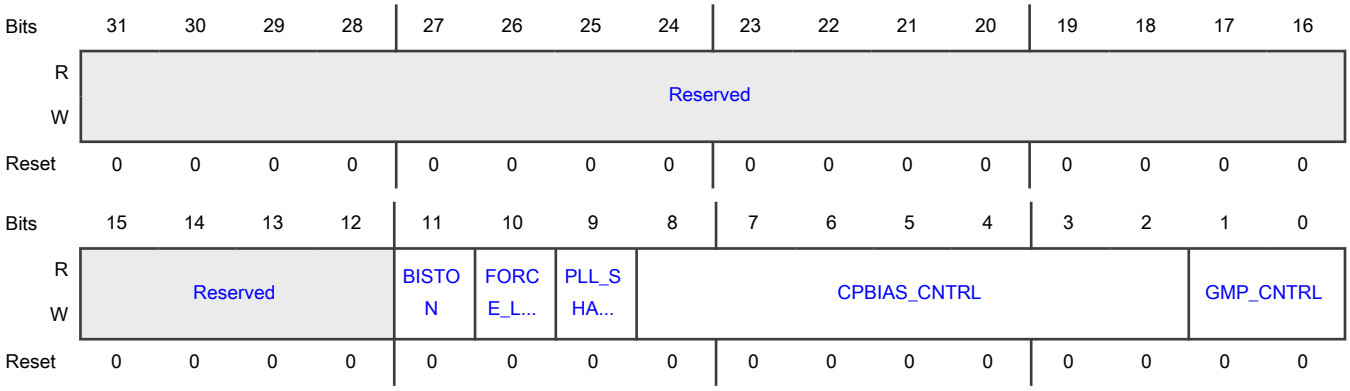
82.4.1.19 DSI Write Register 1 (DSI_W1)

DSI controls for force lock,charge pump bias control,shadow control, and BIST start

Offset

Register	Offset
DSI_W1	54h

Diagram



Fields

Field	Description
31-12 —	Reserved
11 BISTON	BIST start 0 - Do not start BIST 1 - Start BIST
10 FORCE_LOCK	PLL force lock control 0 - According to lock detector 1 - Force PLL internal lock indication
9 PLL_SHADOW_CONTROL	PLL configuration select PLL can be configured through D-PHY or through SoC control and shadow registers. 0 - SoC control and shadow registers 1 - D-PHY control
8-2 CPBIAS_CNTRL	Charge pump bias control
1-0 GMP_CNTRL	Effective loop-filter resistance (=1/gmp) control

82.4.1.20 DSI Read Register 0 (DSI_R0)

DSI status for M, N, VCO operating range, and charge pump.

Offset

Register	Offset
DSI_R0	58h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INT_CNTRL						PROP_CNTRL						VCO_CNTRL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VCO_CNTRL		N				M									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-26 INT_CNTRL	Integral charge pump status
25-20 PROP_CNTRL	Proportional charge pump status
19-14 VCO_CNTRL	VCO operating range
13-10 N	Input frequency division ratio N (1 to 16)
9-0 M	Feedback multiplication ratio M (40 to 625)

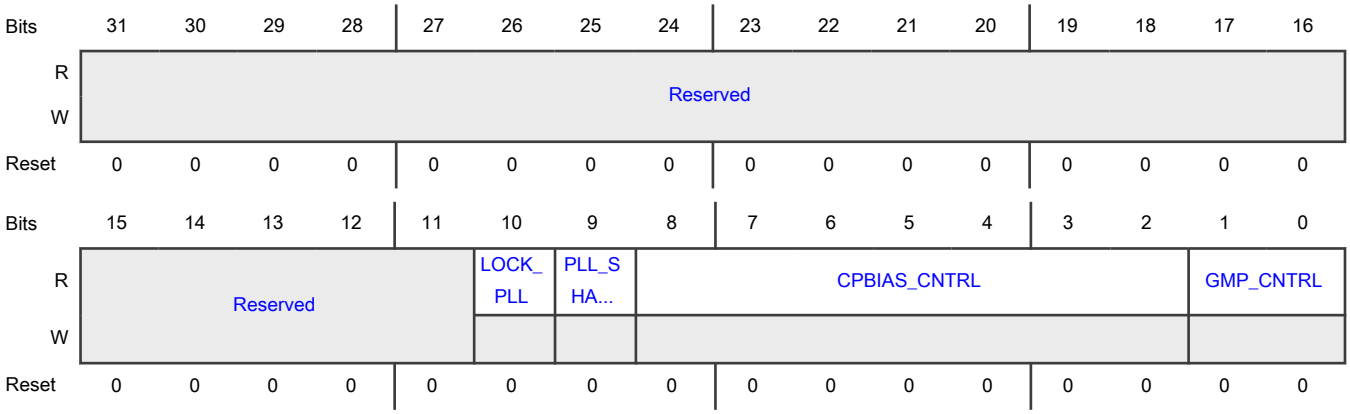
82.4.1.21 DSI Read Register 1 (DSI_R1)

DSI status for force lock,charge pump bias control,shadow control, and PLL lock.

Offset

Register	Offset
DSI_R1	5Ch

Diagram



Fields

Field	Description
31-11 —	Reserved
10 LOCK_PLL	PLL lock state 0 - PLL is not locked 1 - PLL is locked
9 PLL_SHADOW_CONTROL	PLL configuration select
8-2 CPBIAS_CNTRL	Charge pump bias status
1-0 GMP_CNTRL	Effective loop-filter resistance (=1/gmp) status

82.4.1.22 Display Mux Control Register (DISPLAY_MUX)

Display parameters

Offset

Register	Offset
DISPLAY_MUX	60h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				PARALLEL_DISP_FORMAT				MIPI_DSI_RGB666...		MIPI_DSI_RGB565...		LCDIF_CROSS_LINE_PATTERN			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31-11 —	Reserved
10-8 PARALLEL_DISP_FORMAT	Parallel display format configuration 000 - RGB888 -> RGB888 001 - RGB888 -> RGB666. Truncate the two least significant bits of each color component (pass through the 6 most significant bits of each color component.) 010 - RGB565 -> RGB565 011 - RGB555 -> RGB555 100 - YUV -> YCbCr 24 bits 101 - YUV -> YUV444
7-6 MIPI_DSI_RGB666_MAP_CFG	MIPI DSI RGB666 bit map configuration, aligned with MIPI DSI pixel component bit map configuration. 00 - Config 1 01 - Config 2
5-4 MIPI_DSI_RGB565_MAP_CFG	MIPI DSI RGB565 bit map configuration, aligned with MIPI DSI pixel component bit map configuration Refer to MIPI DSI Figure 2-2 16-bit configuration. 00 - Config 1 01 - Config 2 10 - Config 3

Table continues on the next page...

Table continued from the previous page...

Field	Description
3-0 LCDIF_CROSS_LINE_PATTERN	LCDIF pixel component bit map configuration, aligned with the cross line pattern configuration on LCDIF. 0000 - RGB888 to RGB888 0001 - RBG888 to RGB888 0010 - GBR888 to RGB888 0011 - GRB888 to RGB888 0100 - BRG888 to RGB888 0101 - BGR888 to RGB888 0110 - RGB888 to RGB666 0111 - RGB565 to RGB565 1000 - YUYV to YUYV from LCDIF[16:0] 1001 - UYVY to YUYV from LCDIF[16:0] 1100 - YUYV to YUYV from LCDIF[23:8] 1101 - UYVY to YUYV from LCDIF[23:8]

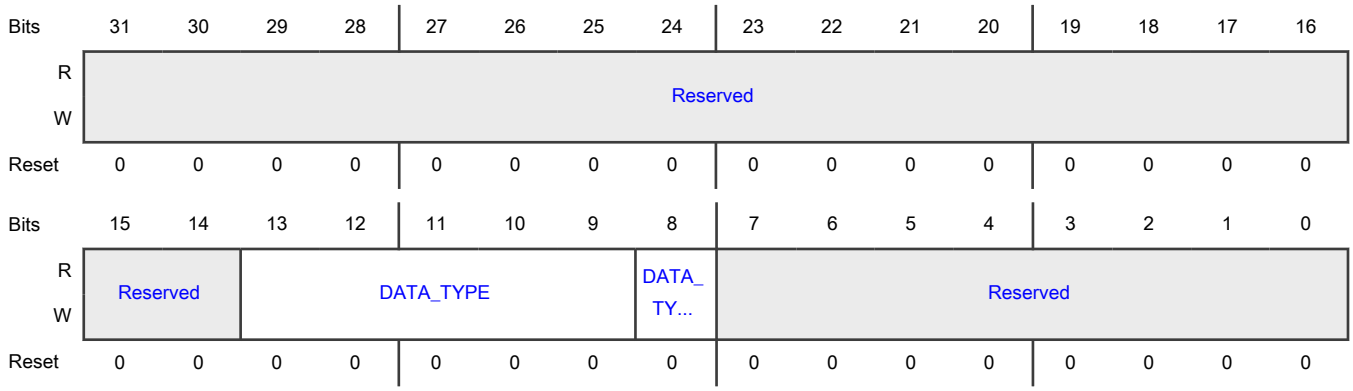
82.4.1.23 Parallel Camera Interface Register (IF_CTRL_REG)

Control parameters for the parallel camera interface.

Offset

Register	Offset
IF_CTRL_REG	70h

Diagram



Fields

Field	Description
31-14 —	Reserved
13-9 DATA_TYPE	Date type to ISI control bus [11:7] 0_0000 - Null data 0_0100 - RGB format 0_1000 - YUV444 format 1_0000 - Reserved 1_0010 - Reserved 1_1000 - Reserved 1_1010 - Reserved 1_1100 - RAW
8 DATA_TYPE_SEL	Pixel link data type select 0 - Pixel Link data type comes from the CSI interface 1 - Pixel Link data type comes from IF_CTRL_REG DATA_TYPE[4:0]
7-0 —	Reserved

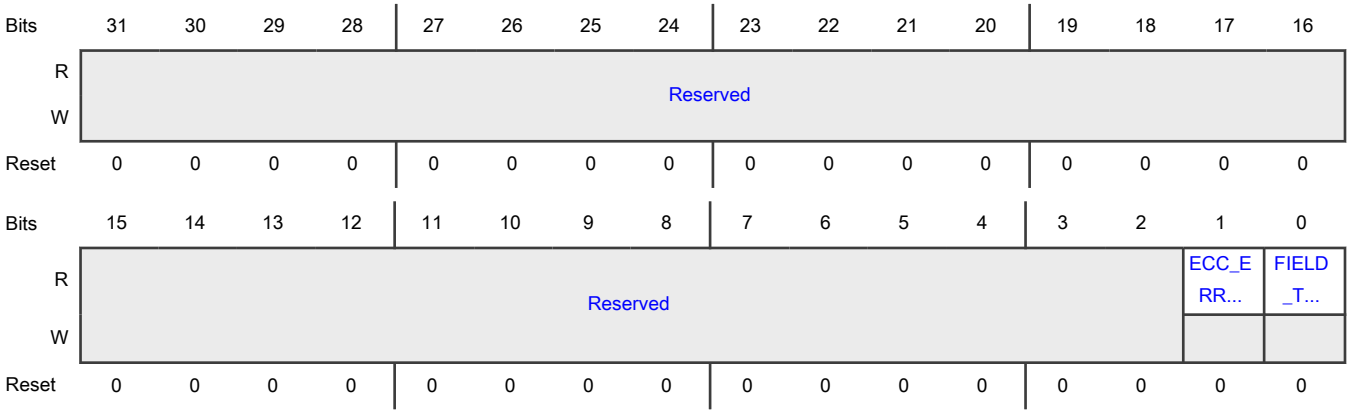
82.4.1.24 Interface Status Register (INTERFACE_STATUS)

Provides ECC error status and field toggle status information.

Offset

Register	Offset
INTERFACE_STATUS	74h

Diagram



Fields

Field	Description
31-2 —	Reserved
1 ECC_ERROR_STATUS	ECC error status 0 - No ECC error detected 1 - ECC error detected
0 FIELD_TOGGLE_STATUS	Field toggle status 0 - VSYNC field toggle mode disabled 1 - VSYNC field toggle mode enabled

82.4.1.25 Interface Control Register (INTERFACE_CTRL_REG)

Controls for HSNC, VSYNC, DE, and CCIR error correction.

Offset

Register	Offset
INTERFACE_CTRL_REG	78h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SOFTRST	Reserved		MASK_VSYNC_CNTR		DATA_TYPE_IN				UV_SWAP...	HSYNC_PULSE			Reserved	HSYNC_O...	RAW_OUT...
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VALID_S...	GCLK_MO...	VSYNC_F...	HSYNC_F...	CCIR_EC...	CCIR_VS...	CCIR_NT...	CCIR_VI...	CCIR_EN	CCIR_EX...	PIXEL_D...	DE_POL	VSYNC_P...	HSYNC_P...	PIXEL_C...	CSI_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Description
31 SOFTRST	CSI interface software reset 0 - Assert software reset of CSI interface. 1 - Do not perform software reset of CSI interface.

Table continues on the next page...

Table continued from the previous page...

Field	Description
30-29 —	Reserved
28-27 MASK_VSYNC_CNTR	CSI mask VSYNC counter 00 - No mask 01 - Mask 1 frame 10 - Mask 2 frames 11 - Mask 3 frames
26-23 DATA_TYPE_IN	CSI input data type 0000 - UYVY bt656 8-bit 0001 - UYVY bt656 10-bit 0010 - RGB 8-bit 0011 - BGR 8-bit 0100 - Reserved 0101 - YUV422 YVYU 8-bit 0110 - YUV444 YUV 8-bit 0111 - Reserved 1000 - Reserved 1001 - Bayer 8-bit 1010 - Bayer 10-bit 1011 - Reserved 1100 - Reserved
22 UV_SWAP_EN	UV swap enable 0 - UV swap disable 1 - UV swap enable
21-19 HSYNC_PULSE	HSYNC_PULSE HSYNC Pulse width
18 —	Reserved
17 HSYNC_OUT_SEL	HSYNC output select 0 - HYSNC output level 1 - HYSNC output pulse

Table continues on the next page...

Table continued from the previous page...

Field	Description
16 RAW_OUT_SEL	RAW output select 0 - Right justified output 1 - Left justified to 14-bit output
15 VALID_SEL	Data valid select 0 - hsync data valid 1 - data en data valid.don't support in i.MX93 0 - HSYNC data valid. 1 - Data enable valid. Not supported.
14 GCLK_MODE_EN	Gate clock mode enable 0 - Disable 1 - Enable
13 VSYNC_FORCE_EN	VSYNC force enable 0 - Do not override VSYNC. 1 - Override VSYNC.
12 HSYNC_FORCE_EN	HSYNC force enable 0 - Do not override HSYNC. 1 - Override HSYNC.
11 CCIR_ECC_CORR_EN	CCIR error correction enable This bit enables and disables the ECC error correction. If single-bit ECC error occurs and this bit is disabled, only ECC_INT will be generated without error correction. If single-bit ECC error occurs and this bit is enabled, the ECC error will be corrected and ECC_INT will not be generated. 0 - ECC error correction is disabled. 1 - ECC error correction is enabled.
10 CCIR_VSYNC_RST_EN	CCIR_VSYNC_RESET_EN If this bit is set, the CCIR Decoder FSM will be reset by VSYNC signal in every frame starting.
9 CCIR_NTSC_EN	CCIR NTSC enable 0 - PAL 1 - NTSC
8 CCIR_VIDEO_MODE	CCIR video mode 0 - Progressive mode 1 - Interlace

Table continues on the next page...

Table continued from the previous page...

Field	Description
7 CCIR_EN	CCIR mode enable 0 - CCIR mode disable 1 - CCIR mode enable
6 CCIR_EXT_VSYNC_EN	External VSYNC enable When CCIR_EN bit is enabled and this bit is set to 1, the external VSYNC signal will be used
5 PIXEL_DATA_POL	PIXEL_DATA polarity control 0 - PIXEL_DATA output to Pixel Link is not inverted. 1 - PIXEL_DATA output to Pixel Link is inverted.
4 DE_POL	DE polarity control 0 - DE output to Pixel Link is not inverted. 1 - DE output to Pixcel Link is inverted.
3 VSYNC_POL	VSYNC polarity control 0 - VSYNC output to Pixel Link is not inverted. 1 - VSYNC output to Pixel Link is inverted.
2 HSYNC_POL	HSYNC polarity control 0 - HSYNC output to Pixel Link is not inverted. 1 - HSYNC output to Pixel Link is inverted.
1 PIXEL_CLK_POL	Pixel clock polarity control 0 - Pixel clcok is not inverted. 1 - Pixel clock input is inverted.
0 CSI_EN	CSI interface enable 0 - Disable CSI interface 1 - Enable CSI interface

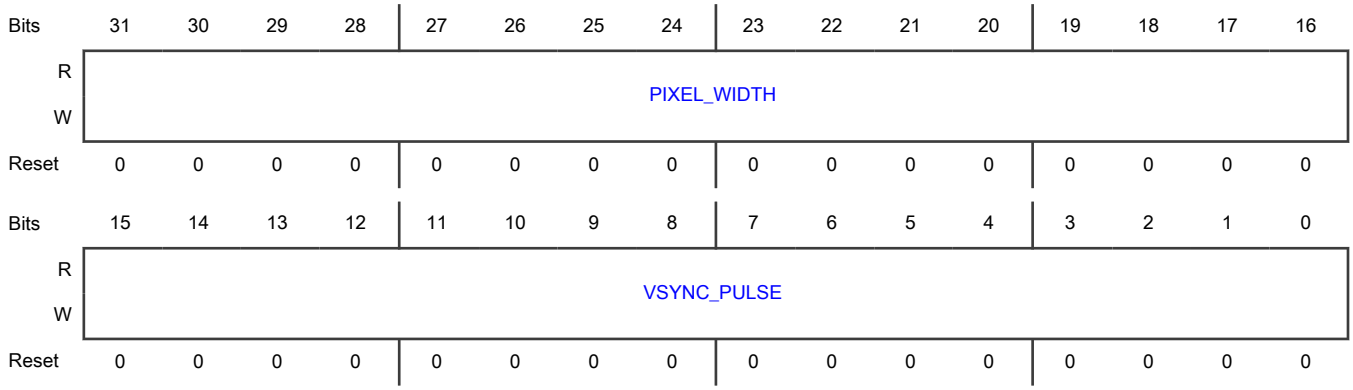
82.4.1.26 Interface Control Register 1 (INTERFACE_CTRL_REG1)

Controls for VSYNC pulse and pixel widths.

Offset

Register	Offset
INTERFACE_CTRL_REG1	7Ch

Diagram



Fields

Field	Description
31-16 PIXEL_WIDTH	CSI interface pixel width
15-0 VSYNC_PULSE	VSYNC pulse width

Appendix A
Release notes

A.1 About_this_manual changes

No substantial content changes.

A.2 Introduction changes

No substantial content changes.

A.3 Memory Map changes

No substantial content changes.

A.4 IRQ and DMA assignment changes

No substantial content changes.

A.5 Enhanced Direct Memory Access (eDMA3)

A.5.1 Chip-specific eDMA3 changes

- Added section.

A.5.2 eDMA3 module changes

No substantial content changes.

A.6 Enhanced Direct Memory Access (eDMA4)

A.6.1 Chip-specific eDMA4 changes

- Added section.

A.6.2 eDMA4 module changes

No substantial content changes.

A.7 Security Overview changes

- Updated Secure debug.

A.8 Fusemap changes

• Added SPEED_GRADING, ECID, SI_REV and MARKET_SEGMENT fuses to Fusemap attachment.
• Updated watchdog timeout select options in Boot Fusemap.

A.9 FSB module changes

No substantial content changes.

A.10 System Boot changes

• In Boot Types, added note "In low power boot mode, only Cortex-M33 core runs after POR. Cortex-A55 core cannot be triggered by Cortex-M33 firmware".
• Updated watchdog timeout select options in Watchdog.

A.11 IOMUXC GPR module changes

No substantial content changes.

A.12 Arm Modules Overview changes

No substantial content changes.

A.13 A55 Platform Overview changes

No substantial content changes.

A.14 ARM Cortex CM33 TCM MCM and ECC MCM (CM33_CACHE_TCM_ECC_MCM)

A.14.1 Chip-specific CM33 changes

No substantial content changes.

A.14.2 CM33 Platform Overview changes

No substantial content changes.

A.15 MCM module changes

No substantial content changes.

A.16 XCACHE module changes

No substantial content changes.

A.17 CM33 Performance Monitor (CMX-PERFMON)

A.17.1 Chip-specific CMX changes

No substantial content changes.

A.17.2 CMX_PERFMON changes

No substantial content changes.

A.18 NPU module changes

No substantial content changes.

A.19 OCRAM module changes

No substantial content changes.

A.20 MECC256 changes

No substantial content changes.

A.21 MU module changes

- Updated receive and transmit interrupts related points in Interrupts.

A.22 SEMA42 module changes

No substantial content changes.

A.23 Trusted Resource Domain Controller (TRDC)

A.23.1 Chip-specific TRDC changes

No substantial content changes.

A.23.2 TRDC module changes

- Reserved NSE bit in register MBC0_DOM0_MEMm_BLK_CFG_Wn.
- Updated the chapter title from "Master Domain Assignment Controller XRDC MDAC" to "Domain Assignment Controller (TRDC_DAC)".

A.24 System Counter (SYS_CTR)

A.24.1 Chip-specific SYS_CTR changes

No substantial content changes.

A.24.2 SYS_CTR module changes

No substantial content changes.

A.25 System Debug changes

No substantial content changes.

A.26 JTAG Controller (JTAGC)

A.26.1 Chip-specific JTAG changes

No substantial content changes.

A.26.2 JTAGC module changes

No substantial content changes.

A.27 External Signals changes

No substantial content changes.

A.28 IOMUXC module changes

- Updated description of all FSEL1 bits.
- Updated description of all DSE bits.

A.29 General-Purpose Input/Output (GPIO)

A.29.1 Chip-specific GPIO changes

- In Chip-specific GPIO information

A.29.2 GPIO module changes

No substantial content changes.

A.30 Clocks and Power Overview changes

No substantial content changes.

A.31 CCM module changes

- In Clock Root, updated media_apb_clk_root frequency to 125 MHz.

A.32 GPC changes

No substantial content changes.

A.33 SRC module changes

- In SRC General Purpose Register 20 (GPR20), fixed typo.

A.34 Battery-Backed Non-Secure Module (BBNSM)

A.34.1 Chip-specific BBNSM changes

No substantial content changes.

A.34.2 BBNSM module changes

No substantial content changes.

A.35 External_memory controllers

No substantial content changes.

A.36 DDR Memory Controller (DDRC)

A.36.1 Chip-specific DDR changes

No substantial content changes.

A.36.2 DDRC module changes

- In Features, updated LPDDR4 note.

- In MD_EN, added clarification/procedure to disable the fields.
- In DDR SDRAM Debug Status 2 (DDRDSR_2), updated register description.

- In Memory Error Detect (ERR_DETECT), changed bit 13 to Reserved.
- In Memory Error Disable (ERR_DISABLE), changed bit 8 to Reserved.

- In Event signals table, updated PMC10_66 to PMC10_65 and PMC9_66 to PMC9_65.

- Added DDR SDRAM Control Configuration 4 (DDR_SDRAM_CFG_4)
- In LPDDR4 LPDDR4X, 16-bit data bus, no interleaving, DDRC register descriptions and Block diagram

Table continues on the next page...

• Added DDR SDRAM Control Configuration 4 (DDR_SDRAM_CFG_4), changed name of bits 12-16 from FRQCH_RET to PHY_RET.
• Updated description of TIMING_CFG_10[TSTAB].
• In Overview removed "Detect most errors within a nibble".

A.37 Flexible Serial Peripheral Interface (FlexSPI)

A.37.1 Chip-specific FLEXSPI changes

No substantial content changes.

A.37.2 FlexSPI module changes

No substantial content changes.

A.38 AHB Address Remap module changes

No substantial content changes.

A.39 Ultra Secured Digital Host Controller (uSDHC)

A.39.1 Chip-specific USDHC changes

No substantial content changes.

A.39.2 uSDHC module changes

No substantial content changes.

A.40 CAN (FlexCAN)

A.40.1 Chip-specific FLEXCAN changes

No substantial content changes.

A.40.2 FlexCAN module changes

• In the "Maximum CAN bit rate in data phase on CAN FD frames with Enhanced RX FIFO disabled" table, changed the column heading "Payload size" to "Minimum payload size".
• Changed all instances of ISO 11898-1:2015 to 11898-1_2024.

A.41 Flexible I/O (FlexIO)

A.41.1 Chip-specific FLEXIO changes

No substantial content changes.

A.41.2 FLEXIO module changes

No substantial content changes.

A.42 Universal Serial Bus Controller (USB)

A.42.1 Chip-specific USB changes

No substantial content changes.

A.42.2 USB CTRL module changes

No substantial content changes.

A.43 fusemap changes

No substantial content changes.

A.44 Ethernet Controller with TSN (ENET_QoS)

A.44.1 Chip-specific ENET changes

No substantial content changes.

A.44.2 ENET_QOS module changes

- Added RDES2[13:11] description in Table 421
- Added description for 1588_EVENT_AUX_IN, 1588_EVENT_IN, and 1588_EVENT_OUT in Table 428

A.45 ENET module changes

- Updated the field description of RCR[RGMII_EN]

A.46 Audio Overview changes

- In Audio Interfaces, Added "SAI1 supports 2 lanes TX and 1 lane RX."

A.47 PDM Microphone Interface (PDM)

A.47.1 Chip-specific PDM changes

No substantial content changes.

A.47.2 MICFIL module changes

No substantial content changes.

A.48 Synchronous Audio Interface (SAI)

A.48.1 Chip-specific SAI changes

- In Chip-specific SAI information, Added "SAI1 supports 2 lanes TX and 1 lane RX."

A.48.2 SAI module changes

No substantial content changes.

A.49 AUDIO_XCVR module changes

- Minor editorial changes in Block diagram.
- Updated External signals section.
- Removed "If eARC mode,(1.8 GHz) then AV_PLL_DIV_ADDR (0x0060) = 0x00962002" from PLL Programming Sequence section.

A.50 Medium Quality Sound (MQS)

A.50.1 Chip-specific MQS changes

No substantial content changes.

A.50.2 MQS module changes

- Implemented inclusive language; changed "Master clock" with "Initiator clock".

A.51 Audio Overview changes

No substantial content changes.

A.52 LCD Interface (LCDIF)

A.52.1 Chip-specific LPIT changes

No substantial content changes.

A.52.2 LCDIF changes

No substantial content changes.

A.53 MIPI DSI changes

No substantial content changes.

A.54 MIPI Tx PHY changes

No substantial content changes.

A.55 MIPI_CSI2 changes

No substantial content changes.

A.56 MIPI Rx PHY changes

No substantial content changes.

A.57 Pixel Pipeline (PXP)

A.57.1 Chip-specific PXP changes

No substantial content changes.

A.57.2 PXP changes

- Removed note "For downscaling, the bilinear filter provides the factor of 2 and does not support the factor of 1.####_####_####" from Scaling Operation.

A.58 Image Sensing Interface (ISI)

A.58.1 Chip-specific ISI changes

No substantial content changes.

A.58.2 ISI module changes

- Deleted inapplicable information related to the second display controller throughout the block guide
- Added additional register sets for Channel 1 in ISI register descriptions

- Added note for bypassing channel when input format is RAW in the field description of CHNL_CTRL*n*[CHNL_BYPASS]

- Updated Features

Table continues on the next page...

- Updated Figure 353 in the function description of CHNL_IMG_CTRL*n*
- Updated Table 549 in the function description of CHNL_MEM_RD_CTRL*n*

A.59 Parallel Camera Interface changes

No substantial content changes.

A.60 LVDS Display Bridge and Pixel Mapper

A.60.1 Chip-specific LVDS changes

No substantial content changes.

A.60.2 LDB Module changes

- Removed "Memory Map".

A.61 Low Power Inter-Integrated Circuit (LPI2C)

A.61.1 Chip-specific LPI2C changes

No substantial content changes.

A.61.2 LPI2C module changes

- Updated Features.
- In Controller FIFO Control (MFCR), updated cross-references to point to MSR bit fields instead of SSR bit fields.

A.62 Low Power Serial Peripheral Interface (LPSPI)

A.62.1 Chip-specific LPSPI changes

No substantial content changes.

A.62.2 LPSPI module changes

- Updated CFGR1[AUTOPCS].

A.63 Low Power Universal Asynchronous Receiver/Transmitter (LPUART)

A.63.1 Chip-specific LPUART changes

No substantial content changes.

A.63.2 LPUART module changes

- In RXINV, updated bit field description and bit field value description.
- In MSBF, updated bit field description.
- In Parameter (PARAM), updated bit field description.
- In IDLECFG, updated bit field description.
- In IDLE, updated bit field description.

A.64 Improved Inter-Integrated Circuit (I3C)

A.64.1 Chip-specific I3C changes

No substantial content changes.

A.64.2 I3C module changes

- Updated MDMACTRL[DMATB].
- Removed Target Reset and RSTACT CCC section.
- Added Reset detector module.

- Updated Block diagram.
- Updated Controller requirements.

A.65 Timers Overview changes

No substantial content changes.

A.66 Low-Power Timer (LPTMR)

A.66.1 Chip-specific LPTMR changes

No substantial content changes.

A.66.2 LPTMR module changes

- Switched step5 and step6 in both Application 1: Generate an interrupt every 100 ms using 32.768 kHz clock source and Application 2: Generate an interrupt once a minute using 32.768 kHz clock source

A.67 Low Power Periodic Interrupt Timer (LPIT)

A.67.1 Chip-specific LPIT changes

No substantial content changes.

A.67.2 LPIT module changes

No substantial content changes.

A.68 Timer/PWM Module (TPM)

A.68.1 Chip-specific TPM changes

No substantial content changes.

A.68.2 TPM module changes

• Added CONF[GTBSYNC]
• Updated Figure 393

A.69 Timer (TSTMR)

A.69.1 Chip-specific TSTMR changes

No substantial content changes.

A.69.2 TSTMR module changes

No substantial content changes.

A.70 Watchdog timer (WDOG)

A.70.1 Chip-specific WDOG changes

No substantial content changes.

A.70.2 WDOG module changes

• Some editorial updates.

A.71 Analog Overview changes

No substantial content changes.

A.72 Trigger Mux (TRGMUX)

A.72.1 Chip-specific TRGMUX changes

No substantial content changes.

A.72.2 TRGMUX module changes

No substantial content changes.

A.73 Analog-to-Digital Converter (SAR_ADC)

A.73.1 Chip-specific ADC changes

No substantial content changes.

A.73.2 SAR_ADC module changes

No substantial content changes.

A.74 Thermal Monitoring Unit (TMU)

A.74.1 Chip-specific TMU changes

No substantial content changes.

A.74.2 TMU changes

No substantial content changes.

A.75 PLL module changes

No substantial content changes.

A.76 OSC changes

No substantial content changes.

A.77 BLK_CTRL_NS_AONMIX module changes

No substantial content changes.

A.78 BLK_CTRL_S_AONMIX module changes

No substantial content changes.

A.79 BLK_CTRL_WAKEUPMIX changes

No substantial content changes.

A.80 BLK_CTRL_NICMIX module changes

No substantial content changes.

A.81 BLK_CTRL_MLMIX module changes

No substantial content changes.

A.82 DDRMIX module changes

No substantial content changes.

A.83 BLK_CTRL_MEDIAMIX module changes

No substantial content changes.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

EdgeLock — is a trademark of NXP B.V.

Synopsys & Designware — are registered trademarks of Synopsys, Inc.

Synopsys — Portions Copyright © 2018-2022 Synopsys, Inc. Used with permission. All rights reserved.

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© 2026 NXP B.V. All rights reserved.

For more information, please visit: https://www.nxp.com

Date of release: 10 February 2026
Document identifier: IMX93RM